National Technical University of Athens
School of Rural, Surveying and Geoinformatics Engineering
Remote Sensing Laboratory

Learning Visual and Multimodal Representations

Bill PSOMAS

PhD Dissertation

Supervisor: Prof. Konstantinos KARANTZALOS

Athens, September 2024

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Αγρονόμων Τοπογράφων Μηχανικών - Μηχανικών Γεωπληροφορικής
Εργαστήριο Τηλεπισκόπησης

Εκμάθηση Οπτικών και Πολυτροπικών Αναπαραστάσεων

Βασίλειος ΨΩΜΑΣ

Διδακτορική Διατριβή

Επιβλέπων: Καθ. Κωνσταντίνος ΚΑΡΑΝΤΖΑΛΟΣ

Αθήνα, Σεπτέμβριος 2024

Committee

Advisory Committee

Konstantinos Karantzalos
Professor, National Technical University of Athens, Supervisor

Giorgos Tolias
Assosiate Professor, Czech Technical University in Prague, Member

Demetre Argialas
Emeritus Professor, National Technical University of Athens, Member

Examination Committee

Konstantinos Karantzalos
Professor, National Technical University of Athens, Supervisor

Giorgos Tolias
Assosiate Professor, Czech Technical University in Prague, Member

Demetre Argialas
Professor, National Technical University of Athens, Member

Vasilia Karathanassi
Professor, National Technical University of Athens

Nikos Komodakis
Assistant Proffesor, University of Crete

Ioannis Papoutsis
Assistant Professor, National Technical University of Athens

Maria Vakalopoulou
Assistant Professor, CentraleSupélec, University Paris-Saclay

Abstract

Representations lie at the heart of artificial intelligence, enabling machines to perceive, interpret and interact with the world. Visual representations, extracted from images or videos, enable tasks such as image classification, image retrieval, and object detection. Visual-textual representations, bridging the gap between the visual and linguistic domains, enable tasks like image captioning, visual question answering, and cross-modal retrieval. The ability to learn and manipulate these representations is paramount for advancing the state-of-the-art in computer vision and beyond. In this dissertation, we investigate novel methods for learning both visual (unimodal) and visual-textual (multimodal) representations, focusing mainly on applications in deep metric learning, image classification, and composed image retrieval. We address the challenges of learning representations from both data-centric and model-centric perspectives, aiming to unlock new capabilities for visual understanding and interaction.

In visual representation learning, we first focus on *data* and introduce *Metrix*, a deep metric learning method utilizing mixup for data augmentation. Metrix addresses the challenge of interpolating both examples and target labels, overcoming the non-additive nature of traditional metric learning loss functions. By generalizing existing loss functions to incorporate mixup, Metrix enhances learning and explores new embedding space regions. We introduce a novel metric, *utilization*, to measure this exploration. Experiments on four benchmark datasets, including various mixup settings, show that Metrix significantly outperforms state-of-the-art methods, improving robustness and generalization. This work exemplifies our aim to advance visual representation learning through innovative data augmentation.

Next, we shift our focus to the *model* architecture, introducing *SimPool*, a simple attention-based pooling method at the end of network designed to replace the default one in both convolutional neural networks (CNNs) and vision transformers (ViTs). We develop a generic pooling framework and formulate existing pooling methods as its instantiations, allowing us to analyze, compare and discuss their properties. Through this, we finally derive SimPool, which improves performance in supervised and self-supervised settings on standard benchmarks and downstream tasks. SimPool generates high-quality attention maps that accurately delineate object boundaries, significantly enhancing object localization and robustness to background changes. It improves object discovery metrics and performs efficiently, even when removing ViT blocks, thus optimizing the balance between performance and model complexity. This work exemplifies our aim to advance visual representation learning through innovative model architecture component.

Transitioning to visual-textual representations, we introduce *FreeDom*, a training-

free method for zero-shot composed image retrieval in open-world domain conversion. FreeDom leverages the descriptive power of a frozen vision-language model (VLM) and employs textual inversion, enabling flexible image and text query composition. Unlike traditional methods that invert query images to the continuous latent space of tokens, FreeDom's inversion into the discrete input space of text is pivotal for its success. Experiments on four benchmark domain conversion datasets, including three newly introduced by us, demonstrate its superior performance. Additionally, FreeDom performs on par with the best methods in generic composed image retrieval. This work exemplifies our aim to advance multimodal representation learning through innovative discrete-space textual inversion.

Expanding on visual-textual representations, we now focus on their applications in remote sensing to introduce a novel task: remote sensing composed image retrieval (RSCIR). This task aims to provide a more expressive and flexible search capability within the remote sensing domain. We explore and qualitatively evaluate the unique challenges and capabilities this task introduces. Users can now pair a query image with a query text specifying modifications related to color, shape, size, texture, density, context, quantity, or the presence of certain classes. To quantitatively assess this, we establish a benchmark, PatternCom, and an evaluation protocol focusing on shape, color, density, and quantity modifications. Our method, *WeiCom*, operates training-free by utilizing a frozen vision-language model and incorporates a modality control parameter for generating more image- or text-oriented results based on the specific search needs. This work exemplifies our aim to advance multimodal representation learning by introducing a flexible method that showcases the potential of this novel task in a new domain.

Acknowledgements

I would like to express my deepest gratitude to the people who have guided, supported, and inspired me throughout my PhD journey. First and foremost, I am immensely thankful to Konstantinos Karantzalos, who was always there when I needed him, offering solutions to every challenge that arose. His consistent advice, reliability, and unwavering support made our collaboration exceptional, and for that, I am deeply grateful. A heartfelt thank you to Yannis Avrithis, the person who shaped me into the researcher and scientist I am today. He generously shared his knowledge, principles, and passion without expecting anything in return. His dedication, countless hours of guidance, and moral example have left an indelible mark on me. Yannis, your influence is immeasurable, and I hold the utmost respect and appreciation for you. I would also like to thank Demetre Argialas, who introduced me to the world of artificial intelligence many years ago, and Giorgos Tolias, with whom I collaborated later in my PhD, and despite the shorter time together, our smooth and productive partnership assures me it will continue in the future. To the examination committee, Vasilia Karathanassi, Nikos Komodakis, Ioannis Papoutsis, and Maria Vakalopoulou, thank you for your willingness to participate in the examination of this dissertation. Your presence and insights were invaluable.

My heartfelt thanks go to all the members of the RSLab. We shared many intense, rewarding moments, working from morning till night during these years. In particular, I want to express my gratitude to Ioannis Kakogeorgiou, with whom I had the closest collaboration. Our friendship and professional partnership enriched this journey, and I hope we meet again in the future.

A special thank you to Shashanka Venkataramanan for our fantastic collaboration during a crucial phase of my PhD. The lessons I learned in that period were pivotal to my progress. Likewise, my appreciation extends to Piera Riccio for her trust and the fruitful collaboration.

I owe profound gratitude to my parents, Dimitra and Manos, for their unwavering support throughout my life. Their encouragement, perseverance, and belief in the pursuit of knowledge have been my guiding light. I am eternally grateful for their love and the values they instilled in me. I also wish to thank all my friends who stood by my side throughout this journey.

Finally, my deepest thanks go to Evita, the most beloved person in my life. Your endless love, understanding, and patience have been my strongest pillars of support. You believed in me when I doubted myself, and you stood by me with grace and strength. For everything, I thank you from the bottom of my heart.

Εκτεταμένη Περίληψη

Η αλληλεπίδρασή μας με τον κόσμο ξεκινά από τη στιγμή της γέννησής μας μέσω της αισθητηριακής αντίληψης. Αρχικά, η όρασή μας είναι θολή, διακρίνοντας μόνο φως και σκιές. Μέσα σε λίγες ημέρες, μπορούμε να διακρίνουμε χρώματα και να αναγνωρίζουμε πρόσωπα. Η αντίληψη του βάθους αναπτύσσεται λίγες εβδομάδες αργότερα, ενώ η ικανότητα εστίασης σε αντικείμενα έρχεται στους δύο μήνες. Στους έξι μήνες, μπορούμε να βλέπουμε καθαρές εικόνες. Καθώς το οπτικό σύστημα ωριμάζει, κωδικοποιούμε, αποθηκεύουμε και ανακτούμε λεπτομερείς νοητικές εικόνες αντικειμένων, τόπων και ανθρώπων. Αυτή η διαδικασία επιτρέπει την αναγνώριση αντικειμένων, την κατανόηση των χωρικών σχέσεων και την πλοήγηση στο περιβάλλον μας.

Παράλληλα με την ανάπτυξη της οπτικής αντίληψης, εξελίσσονται και οι ικανότητες ομιλίας και γλώσσας. Λίγο μετά τη γέννηση μας αναγνωρίζουμε τους ήχους της ομιλίας και στους έξι μήνες αρχίζουμε να φλυαρούμε. Γύρω στο πρώτο έτος, λέμε τις πρώτες μας λέξεις, οδηγώντας σταδιακά στην αύξηση του λεξιλογίου και στη διαμόρφωση προτάσεων. Μέχρι τα τρία χρόνια, συμμετέχουμε σε σύνθετες συνομιλίες, κατανοώντας τη σύνταξη και τη γραμματική. Η ανάπτυξη της γλώσσας περιλαμβάνει την κωδικοποίηση, αποθήκευση και ανάκτηση γλωσσικών και ακουστικών εμπειριών, που διευκολύνονται από τις κοινωνικές αλληλεπιδράσεις και το γλωσσικό περιβάλλον. Αυτή η διαδικασία επιτρέπει την κατανόηση της σημασίας των λέξεων, τη χρήση της γλώσσας για την έκφραση σκέψεων και συναισθημάτων, καθώς και την ανάπτυξη της ικανότητας για πολύπλοκη επικοινωνία και κοινωνική αλληλεπίδραση.

Η κωδικοποίηση, η αποθήκευση και η ανάκτηση πληροφοριών είναι το κοινό στοιχείο, λοιπόν, είτε πρόκειται για οπτικές είτε γλωσσικές είτε ακουστικές εμπειρίες. Οι αισθητηριακές και γνωστικές μας εμπειρίες κωδικοποιούνται και αποθηκεύονται ως σύνθετες αναπαραστάσεις, που είναι πιθανώς μοναδικές για κάθε άτομο. Για παράδειγμα, η λέξη "σπίτι" προκαλεί διαφορετικές νοητικές εικόνες σε κάθε άτομο, βάσει των μοναδικών εμπειριών του. Αυτό υποδηλώνει ότι στον χώρο των αναπαραστάσεών μας, η γλωσσική αναπαράσταση της λέξης "σπίτι" συνδέεται στενά με την οπτική αναπαράσταση που έχει δημιουργηθεί από τις εμπειρίες μας με τα σπίτια. Αυτές οι προσωπικές και μοναδικές αναπαραστάσεις επιτρέπουν την αναγνώριση αντικειμένων και την κατανόηση των εννοιών με έναν τρόπο που ενσωματώνει τόσο τα αισθητηριακά όσο και τα γλωσσικά στοιχεία της εμπειρίας μας. Μεταβαίνοντας από τον άνθρωπο στη μηχανή, από τη φυσική στην τεχνητή νοημοσύνη, οι αναπαραστάσεις εξακολουθούν να έχουν πρωτεύοντα ρόλο, καθώς επιτρέπουν στις μηχανές να αντιλαμβάνονται, να ερμηνεύουν και να αλληλεπιδρούν με τον κόσμο. Οι (τεχνητές) οπτικές αναπαραστάσεις, που εξάγονται από εικόνες ή βίντεο, επιτρέπουν εργασίες όπως η ταξινόμηση εικόνων, η ανάκτηση εικόνων και η ανίχνευση αντικειμένων. Οι (τεχνητές) οπτικο-γλωσσικές αναπαραστάσεις, γεφυρώνοντας το χάσμα μεταξύ των οπτικών και γλωσσικών τομέων,

επιτρέπουν εργασίες όπως η λεκτική περιγραφή εικόνων, η απάντηση οπτικών ερωτήσεων και η σύνθετη ανάκτηση εικόνων. Η ικανότητα εκμάθησης και χειρισμού αυτών των αναπαραστάσεων είναι κρίσιμη για την προώθηση της τεχνολογίας αιχμής στην υπολογιστική όραση, αλλά και πέρα από αυτήν.

Στην παρούσα διδακτορική διατριβή, διερευνώνται καινοτόμες μέθοδοι για την εκμάθηση τόσο οπτικών (μονοτροπικών), όσο και οπτικο-γλωσσικών (πολυτροπικών) αναπαραστάσεων, εστιάζοντας κυρίως σε εφαρμογές στην βαθιά εκμάθηση μετρικής, την ταξινόμηση εικόνων και τη σύνθετη ανάκτηση εικόνων. Αντιμετωπίζονται οι προκλήσεις της εκμάθησης αναπαραστάσεων αναπτύσοντας μεθόδους εστιασμένες τόσο στα δεδομένα, όσο και στα μοντέλα, με τελικό στόχο την επίτευξη νέων δυνατοτήτων για οπτική και οπτικο-γλωσσική κατανόηση και αλληλεπίδραση.

Πιο συγκεκριμένα, η πρώτη ενότητα εστιάζει στα δεδομένα για την εκμάθηση οπτικών αναπαραστάσεων και εισάγει το *Metrix*, μια μέθοδο βαθιάς εκμάθησης μετρικής που χρησιμοποιεί την ανάμειξη για επαύξηση των δεδομένων. Στη βαθιά εκμάθηση μετρικής, σε αντίθεση με την ταξινόμηση εικόνων, οι κατηγορίες (και οι κατανομές) στην εκπαίδευση και τον έλεγχο είναι διαφορετικές. Επομένως, είναι αναμενόμενο μια μέθοδος επαύξησης δεδομένων που χρησιμοποιεί την παρεμβολή, όπως η ανάμειξη, να είναι ακόμα πιο χρήσιμη και αποτελεσματική εδώ από ότι στην ταξινόμηση. Ωστόσο, οι πρόσφατες προσπάθειες περιορίζονται κυρίως σε ειδικές περιπτώσεις παρεμβολής ενσωματώσεων και αντιμετωπίζουν προβλήματα με την παρεμβολή ετικετών. Αυτό ίσως οφείλεται και στο γεγονός ότι οι συναρτήσεις απώλειας της βαθιάς εκμάθησης μετρικής, σε αντίθεση με τη διασταυρούμενη εντροπία της ταξινόμηση εικόνων, είναι κατά βάση μη προσθετικές. Αυτό θέτει το ερώτημα: ποιος είναι ο κατάλληλος τρόπος για να οριστούν και να παρεμβληθούν ετικέτες στη βαθιά εκμάθησης μετρικής;

Η πρώτη ενότητα απαντά σε αυτό το ερώτημα, αντιμετωπίζοντας την πρόκληση της παρεμβολής τόσο των εικόνων όσο και των ετικετών, υπερβαίνοντας τη μη προσθετική φύση των παραδοσιακών συναρτήσεων απώλειας της βαθιάς εκμάθησης μετρικής. Συγκρίνοντας την ταξινόμηση εικόνων με τη βαθιά εκμάθηση μετρικής, παρατηρείται ότι η δεύτερη δε διαφέρει και τόσο από την πρώτη, αν οι εικόνες αντικατασταθούν από ζεύγη εικόνων και οι ετικέτες κατηγορίας από "θετικές" ή "άρνητικές" ετικέτες, ανάλογα με το αν οι ετικέτες κατηγορίας των μεμονομένων εικόνων είναι ίδιες ή όχι. Υπό αυτή την έννοια, ένας απλός τρόπος να γίνει η ανάμειξη ετικετών είναι η χρήση μιας ετικέτας δύο κατηγοριών ανά ζεύγος εικόνων και η γραμμική της παρεμβολή όπως στην τυπική ανάμειξη. Γενικεύοντας τις υπάρχουσες συναρτήσεις απώλειας, η πρώτη ενότητα καταφέρνει να ενσωματώσει αυτού του είδους την ανάμειξη σε αυτές και έτσι να ορίσει το Metrix, το οποίο βελτιώνει αισθητά τη μάθηση και εξερευνά νέες περιοχές του ενσωματωμένου χώρου. Για να μετρηθεί αυτή η εξερεύνηση, εισάγεται ένας νέος μετρικός δείκτης, η "άξιοποίηση". Ακόμα, ορίζεται η "θετικότητα" μιας μικτής ετικέτας δύο κατηγοριών και μελετάται πως ακριβώς αυξάνεται ως συνάρτηση του συντελεστή παρεμ-

βολής, τόσο θεωρητικά όσο και εμπειρικά. Εκτεταμένα πειράματα σε τέσσερα πρότυπα σύνολα δεδομένων, συμπεριλαμβανομένων πειραμάτων διαφόρων ρυθμίσεων ανάμειξης, δείχνουν ότι το Metrix ξεπερνά σημαντικά τις υπάρχουσες μεθόδους, βελτιώνοντας την ανθεκτικότητα και τη γενίκευση. Η πρώτη ενότητα ακολουθεί και συνδράμει ενεργά στο στόχο προαγωγής της εκμάθησης οπτικών αναπαραστάσεων μέσω καινοτόμων επαυξήσεων δεδομένων και συγκεκριμένα μέσω της εισαγωγής της ανάμειξης Metrix στη βαθιά εκμάθηση μετρικής.

Παραμένοντας στην εκμάθηση οπτικών αναπαραστάσεων, η δεύτερη ενότητα μετατοπίζει την εστίαση στην αρχιτεκτονική του μοντέλου, εισάγοντας το *SimPool*, μια απλή μέθοδο συγκέντρωσης στο τέλος του δικτύου, σχεδιασμένη να αντικαθιστά την προεπιλεγμένη τόσο σε συνελικτικά νευρωνικά δίκτυα όσο και σε οπτικούς μετασχηματιστές. Τα συνελικτικά δίκτυα είναι ιεραρχικά, αποτελούμενα από εναλλασσόμενα επίπεδα συνέλιξης και συγκέντρωσης. Δημιουργούν την τελική ολική αναπαράσταση εικόνας συνήθως μέσω της ολικής συγκέντρωσης μέσης τιμής, η οποία συμπιέζει ολόκληρο τον χάρτη χαρακτηριστικών σε ένα μοναδικό διάνυσμα στο τέλος του δικτύου. Από την άλλη, οι οπτικοί μετασχηματιστές ξεκινούν διαχωρίζοντας την εικόνα σε τμήματα, κωδικοποιώντας το καθένα ως μια μονάδα, συμπεριλαμβανομένης και μιας ειδικής μονάδας, της μονάδας ταξινόμησης. Η συγκέντρωση βασίζεται σε αυτή την μαθαινόμενη μονάδα ταξινόμησης, η οποία, ξεκινώντας από τον χώρο εισόδου, υποβάλλεται στον ίδιο μηχανισμό αυτο-προσοχής με τις υπόλοιπες μονάδες σε όλα τα επίπεδα του δικτύου και έτσι παρέχει μια ολική αναπαράσταση εικόνας. Δηλαδή, το δίκτυο καταλήγει σε μια ολική συγκέντρωση σταθμισμένης μέσης τιμής, χρησιμοποιώντας ως βάρη την προσοχή της μονάδας ταξινόμησης πάνω στις υπόλοιπες μονάδες των τμημάτων της εικόνας. Η μονάδα ταξινόμησης παρέχει χάρτες προσοχής δωρεάν, ωστόσο, όπως έχει σημειωθεί, αυτοί είναι χαμηλής ποιότητας, εκτός αν είναι στο πλαίσιο αυτο-επιβλεπόμενης μάθησης. Στη δεύτερη ενότητα, υποστηρίζεται ότι ο οπτικός μετασχηματιστής μπορεί να επαναδιατυπωθεί αρχιτεκτονικά σε δύο ρεύματα, όπου το ένα εξάγει μια οπτική αναπαράσταση στις μονάδες των τμημάτων της εικόνας, ενώ το άλλο εκτελεί χωρική συγκέντρωση στη μονάδα ταξινόμησης. Υπό αυτή την έννοια, η διαδικασία της συγκέντρωσης μπορεί να απομονωθεί και από τους δύο τύπους δικτύων και να αντικατασταθεί με μία νέα. Αυτό εγείρει τα ακόλουθα ερωτήματα:

Μπορεί να δημιουργηθεί μια απλή διαδικασία συγκέντρωσης στο τελευταίο στάδιο είτε των συνελικτικών δικτύων είτε των οπτικών μετασχηματιστών, η οποία βελτιώνει την προεπιλεγμένη αντιστοίχως; Μπορεί αυτή η διαδικασία να παρέχει υψηλής ποιότητας χάρτες προσοχής που διαχωρίζουν τα όρια των αντικειμένων και για τους δυο τύπους δικτύων; Ισχύουν αυτές οι ιδιότητες τόσο υπό επιβλεπόμενη όσο και υπό αυτο-επιβλεπόμενη μάθηση; Για να απαντηθούν αυτά να ερωτήματα, αναπτύσσεται ένα γενικό πλαίσιο συγκέντρωσης, παραμετροποιημένο από: (α) την αρχικοποίηση της συγκέντρωσης, (β) τον αριθμό των διανυσμάτων συγκέντρωσης, (γ) την επαναληψιμότητα

της συγκέντρωσης, (δ) πιθανούς μετασχηματισμούς σε κάθε στάδιο της διαδικασίας, (ε) ζεύγη ομοιότητας, (στ) μηχανισμό προσοχής και κανονικοποίηση και (ζ) τη συνάρτηση που ορίζει την πράξη της συγκέντρωσης. Στη συνέχεια, μια σειρά από υπάρχουσες μεθόδους συγκέντρωσης επαναδιατυπώνονται ως υλοποιήσεις αυτού του πλαισίου. Συζητώντας τις ιδιότητες κάθε μεθόδου, μέσω του πλαισίου, προτείνεται ένας νέος, απλός, μηχανισμός συγκέντρωσης βασισμένος στη διασταυρούμενη προσοχή, που ονομάζεται SimPool. Το SimPool βελτιώνει την απόδοση τόσο σε επιβλεπόμενη, όσο και αυτο-επιβλεπόμενη μάθηση σε πρότυπα σύνολα δεδομένων και κατάντη εργασίες. Δημιουργεί υψηλής ποιότητας χάρτες προσοχής που οριοθετούν με ακρίβεια τα όρια των αντικειμένων, βελτιώνοντας έτσι σημαντικά τον εντοπισμό αντικειμένων και την ανθεκτικότητα στις αλλαγές στο υπόβαθρο της εικόνας. Βελτιώνει τις μετρικές ανακάλυψης αντικειμένων και αποδίδει αποτελεσματικά, ακόμη και έπειτα από αφαίρεση επιπέδων του οπτικού μετασχηματιστή, βελτιστοποιώντας έτσι την ισορροπία μεταξύ απόδοσης και πολυπλοκότητας μοντέλου. Η δεύτερη ενότητα ακολουθεί και συνδράμει ενεργά στο στόχο προαγωγής της εκμάθησης οπτικών αναπαραστάσεων μέσω καινοτόμων αρχιτεκτονικών συστατικών μοντέλου και συγκεκριμένα μέσω της εισαγωγής της συγκέντρωσής SimPool.

Μεταβαίνοντας στις οπτικο-γλωσσικές αναπαραστάσεις, στην τρίτη ενότητα, εισάγεται το *FreeDom*, μια μέθοδο χωρίς εκπαίδευση για σύνθετη ανάκτηση εικόνων μηδενικής λήψης σε σενάρια μετατροπής τομέων ανοιχτού κόσμου. Η ανάκτηση εικόνας-προς-εικόνα είναι μια εργασία στην υπολογιστική όραση με εφαρμογές σε ορόσημα, προϊόντα μόδας, πρόσωπα και ιατρικές εικόνες, μεταξύ άλλων. Η ανάκτηση πραγματοποιείται αποκλειστικά με βάση το οπτικό περιεχόμενο του ερωτήματος. Από την άλλη πλευρά, αν το αντικείμενο μπορεί να περιγραφεί με κείμενο, τότε εφαρμόζεται η ανάκτηση κειμένου-προς-εικόνα. Ο πιο ευέλικτος τρόπος για να εκφραστεί η πρόθεση του χρήστη είναι ένα ερώτημα που περιλαμβάνει τόσο μια εικόνα όσο και μια περιγραφή κειμένου. Αυτό διερευνάται στην σύνθετη ανάκτηση εικόνων, η οποία στοχεύει στην ανάκτηση εικόνων που δεν είναι μόνο οπτικά παρόμοιες με την ερώτηση εικόνας, αλλά και τροποποιημένες σύμφωνα με τις συγκεκριμένες λεπτομέρειες του ερωτήματος κειμένου. Στην τρίτη ενότητα, η εστίαση βρίσκεται κυρίως σε μια παραλλαγή της σύνθετης ανάκτησης εικόνων, στην μετατροπή τομέα, όπου το ερώτημα κειμένου λειτουργεί ως περιγραφή του τομέα στόχου. Σε αντίθεση με την παραδοσιακή διατομεακή ανάκτηση εικόνων, στην οποία τα μοντέλα εκπαιδεύονται να χρησιμοποιούν ερωτήματα από έναν πηγαίο τομέα και να ανακτούν εικόνες από έναν άλλο τομέα στόχο, σε αυτή την ενότητα αντιμετωπίζεται μια πιο πρακτική, ανοιχτή ρύθμιση τομέα, όπου το ερώτημα και η βάση δεδομένων μπορεί να προέρχονται από οποιονδήποτε άγνωστο τομέα. Εξετάζονται διαφορετικές παραλλαγές σεναρίων μετατροπής τομέων, στις οποίες το ερώτημα εικόνας ορίζεται σε επίπεδο κατηγορίας ή σε επίπεδο παραδείγματος, ενώ ο τομέας δύναται να αντιστοιχεί σε περιγραφές σχετικές με το στυλ ή την γενικότερη οπτική περίσταση.

Το FreeDom αξιοποιώντας την περιγραφική δύναμη ενός προ-εκπαιδευμένου και παγωμένου οπτικο-γλωσσικού μοντέλου παράλληλα με την αντιστροφή κειμένου, επιτρέπει την ευέλικτη σύνθεση ερωτημάτων εικόνας και ερωτημάτων κειμένου. Σε αντίθεση με τις παραδοσιακές μεθόδους που αντιστρέφουν τα ερωτήματα εικόνων στο συνεχή χώρο χαρακτηριστικών, η αντιστροφή του FreeDom στον διακριτό εισαγωγικό χώρο του κειμένου είναι καθοριστική για την επιτυχία του. Αυτό μπορεί πιθανώς να αποδίδεται στο γεγονός ότι η αντιστροφή με βάση τη βελτιστοποίηση, δηλαδή η αντιστροφή στο συνεχή χώρο χαρακτηριστικών, μπορεί να παρέχει λύσεις σε περιοχές αυτού που δεν είχαν χρησιμοποιηθεί ποτέ ως είσοδοι στο οπτικο-γλωσσικό μοντέλο· κάτι που δεν ισχύει για την προτεινόμενη αντιστροφή. Πειράματα σε τέσσερα πρότυπα σύνολα δεδομένων μετατροπής τομέων, συμπεριλαμβανομένων τριών νεοεισαχθέντων, δείχνουν την ανώτερη απόδοσή του FreeDom σε σύγκριση με τις υπάρχουσες μεθόδους. Επιπλέον, αποδίδει εξίσου καλά με τις αποδοτικότερες μεθόδους σύνθετης ανάκτησης εικόνων ακόμα και σε γενικότερα σενάρια. Η τρίτη ενότητα ακολουθεί και συνδράμει ενεργά στο στόχο προαγωγής της εκμάθησης οπτικο-γλωσσικών (πολυτροπικών) αναπαραστάσεων μέσω καινοτόμων αντιστροφών και συγκεκριμένα μέσω της εισαγωγής της αντιστροφής σε διακριτό χώρο της μεθόδου FreeDom.

Επεκτείνοντας τις οπτικο-γλωσσικές αναπαραστάσεις, η τέταρτη ενότητα επικεντρώνεται στις εφαρμογές τους στην τηλεπισκόπηση και εισάγει μια νέα εργασία: την σύνθετη ανάκτηση εικόνων τηλεπισκόπησης. Τα τελευταία χρόνια, η παρατήρηση της Γης μέσω της τηλεπισκόπησης έχει σημειώσει τεράστια αύξηση στον όγκο των δεδομένων, δημιουργώντας μια πρόκληση στη διαχείριση και εξαγωγή σχετικών πληροφοριών. Η ικανότητα οργάνωσης εκτεταμένων αρχείων και γρήγορης ανάκτησης συγκεκριμένων εικόνων είναι κρίσιμη. Η ανάκτηση εικόνων τηλεπισκόπησης, που στοχεύει στην αναζήτηση και ανάκτηση εικόνων από βάσεις δεδομένων τηλεπισκόπησης, έχει αναδειχθεί ως βασική λύση. Ωστόσο, αυτές οι μέθοδοι αντιμετωπίζουν ένα σημαντικό περιορισμό: την εξάρτηση από ένα ερώτημα μιας μόνο τροπικότητας. Αυτός ο περιορισμός συχνά περιορίζει τους χρήστες από το να εκφράσουν πλήρως τις συγκεκριμένες απαιτήσεις τους, ειδικά δεδομένης της σύνθετης και δυναμικής φύσης της επιφάνειας της Γης όπως απεικονίζεται στις τηλεπισκοπικές εικόνες. Η τέταρτη ενότητα στοχεύει να προσφέρει μια πιο εκφραστική και ευέλικτη δυνατότητα αναζήτησης στον τομέα της τηλεπισκόπησης. Στην σύνθετη ανάκτηση εικόνων τηλεπισκόπησης, οι χρήστες μπορούν να συνδυάζουν ένα ερώτημα εικόνας με ένα ερώτημα κειμένου που καθορίζει τροποποιήσεις που σχετίζονται με το χρώμα, το σχήμα, το μέγεθος, την υφή, την πυκνότητα, τη γενικότερη οπτική περίσταση, την ποσότητα ή την παρουσία συγκεκριμένων κατηγοριών. Για την ποσοτική αξιολόγηση αυτών, δημιουργείται ένα σετ δεδομένων, το PatternCom, και ένα πρωτόκολλο αξιολόγησης που εστιάζει σε τροποποιήσεις σχετικές με το χρώμα, το σχήμα, την πυκνότητα, τη γενικότερη οπτική περίσταση, την ποσότητα ή την παρουσία. Προτείνεται η μέθοδος σύνθετης ανάκτηση εικόνων τηλεπισκόπησης WeiCom, η οποία λειτουργεί χωρίς εκπαίδευση χρησιμοποιώντας ένα προ-εκπαιδευμένο και παγωμένο

οπτικο-γλωσσικό μοντέλο. Η μέθοδος ενσωματώνει μια παράμετρο ελέγχου τροπικότητας για την παραγωγή αποτελεσμάτων που προσανατολίζονται περισσότερο στην εικόνα ή στο κείμενο, ανάλογα με τις συγκεκριμένες ανάγκες αναζήτησης. Η τέταρτη ενότητα ακολουθεί και συνδράμει ενεργά στο στόχο προαγωγής της εκμάθησης οπτικο-γλωσσικών (πολυτροπικών) αναπαραστάσεων, εισάγοντας μια ευέλικτη μέθοδο που παρουσιάζει τις δυνατότητες της νέας, προτεινόμενης εργασίας, συνοδευόμενη απο ένα νέο σετ δεδομένων.

Το αντικείμενο της παρούσας διδακτορικής διατριβής αναπτύσσεται μέσω παράθεσης, ανάλυσης, συζήτησης και σύγκρισης των πρόσφατων εξελίξεων της σχετικής βιβλιογραφίας, σε συνδυασμό με τις παρατηρήσεις και τα συμπεράσματα που προκύπτουν από τα πειραματικά αποτελέσματα της εφαρμογής των νέων μεθόδων. Οι νέες μέθοδοι κάθε ενότητας χαρακτηρίζονται από την ανώτερη απόδοσή τους ή/και τη μειωμένη πολυπλοκότητά τους σε σχέση με τις υπάρχουσες μεθόδους. Τα περιεχόμενα της διατριβής μπορούν να χωριστούν σε τρία μέρη. Το πρώτο μέρος επικεντρώνεται στις οπτικές αναπαραστάσεις, αναλύοντας την ανάπτυξη και αξιολόγηση καινοτόμων μεθόδων που βασίζονται στα δεδομένα και στην αρχιτεκτονική του μοντέλου. Το δεύτερο μέρος εμβαθύνει στις οπτικο-γλωσσικές (πολυτροπικές) αναπαραστάσεις, παρουσιάζοντας καινοτόμες προσεγγίσεις για σύνθετη ανάκτηση εικόνων και τις εφαρμογές τους σε διάφορους τομείς. Τέλος, το τρίτο μέρος, παρέχει μια περίληψη των ευρημάτων, συμπεράσματα και προτάσεις για μελλοντική εργασία. Τα τρία μέρη αναπτύσσονται σε 6 κεφάλαια:

Στο Κεφάλαιο 1, εισάγεται η σημασία των αναπαραστάσεων στην ανθρώπινη και τεχνητή νοημοσύνη, θέτοντας το πλαίσιο για την έρευνα στην εκμάθηση οπτικών και οπτικο-γλωσσικών αναπαραστάσεων. Παρουσιάζονται οι κύριους στόχοι και οι συνεισφορές αυτής της διδακτορικής διατριβής.

Στο Κεφάλαιο 2, παρουσιάζεται η νέα μέθοδος επαύξησης δεδομένων, Metrix, η οποία σχεδιάστηκε να αντιμετωπίζει την πρόκληση της παρεμβολής τόσο των παραδειγμάτων όσο και των ετικετών στόχου στη βαθιά εκμάθηση μετρικής. Συζητούνται τα θεωρητικά θεμέλια, η ανάπτυξη της μεθόδου και η αξιολόγηση σε διάφορα πρότυπα σύνολα δεδομένων, επιδεικνύοντας σημαντικές βελτιώσεις στην ανθεκτικότητα και τη γενίκευση.

Στο Κεφάλαιο 3, εισάγεται το SimPool, μια μέθοδος συγκέντρωσης βασισμένη στο μηχανισμό προσοχής, η οποία σχεδιάστηκε για να αντικαθιστά την προεπιλεγμένη τόσο σε συνελικτικά νευρωνικά δίκτυα όσο και σε οπτικούς μετασχηματιστές. Αναλύεται η ανάπτυξη ενός γενικού πλαισίου συγκέντρωσης, η διατύπωση των υπαρχουσών μεθόδων εντός αυτού του πλαισίου και η εξαγωγή του SimPool. Οι βελτιώσεις στην απόδοση, τόσο ποιοτικές όσο και ποσοτικές, επιβεβαιώνονται σε πρότυπα σύνολα δεδομένων και κατάντη εργασίες.

Στο Κεφάλαιο 4, εισάγεται το FreeDom, μια μέθοδος χωρίς εκπαίδευση για σύνθετη ανάκτηση εικόνων μηδενικής λήψης σε σενάρια μετατροπής τομέων ανοιχτού κόσμου. Η καινοτόμος προσέγγιση αξιοποιεί ένα προ-εκπαιδευμένο και παγωμένο οπτικο-γλωσσικό μοντέλο και χρησιμοποιεί αντιστροφή σε διακριτό χώρο. Τα πειραματικά αποτελέσματα σε πολλαπλά πρότυπα σύνολα δεδομένων δείχνουν την ανώτερη απόδοσή του και τις δυνατότητες για περαιτέρω εφαρμογές σε γενική σύνθετη ανάκτηση εικόνων.

Στο Κεφάλαιο 5, εισάγεται μια νέα εργασία: η σύνθετη ανάκτηση εικόνων τηλεπισκόπησης. Συζητούνται οι μοναδικές προκλήσεις και δυνατότητες αυτής της εργασίας, παρουσιάζοντας ένα νέο πρότυπο σύνολο δεδομένων, το PatternCom, και μια μέθοδο χωρίς εκπαίδευση, τη WeiCom, που χρησιμοποιεί παράμετρο ελέγχου τροπικότητας. Η αποτελεσματικότητα της μεθόδου αξιολογείται μέσω διαφόρων τροποποιήσεων χαρακτηριστικών, αναδεικνύοντας τις δυνατότητες για βελτίωση των δυνατοτήτων αναζήτησης στον τομέα της τηλεπισκόπησης.

Στο Κεφάλαιο 6, συνοψίζονται τα κύρια ευρήματα και οι συνεισφορές της έρευνας. Τονίζονται οι πρόοδοι που έγιναν τόσο στην εκμάθηση οπτικών όσο και οπτικογλωσσικών αναπαραστάσεων και συζητούνται οι πιθανές μελλοντικές κατευθύνσεις για την επέκταση της εργασίας.

Οι βασικές ιδέες, μέθοδοι και αποτελέσματα της διδακτορικής διατριβής έχουν δημοσιευτεί ή βρίσκονται σε διαδικασία δημοσίευσης σε διεθνή επιστημονικά συνέδρια υψηλού κύρους με πολύ σημαντικό αριθμό αναφορών. Εκτός από τις δημοσιεύσεις που συμπεριλήφθηκαν στη διατριβή, κατά τη διάρκεια της έρευνας εκπονήθηκαν και δημοσιεύτηκαν περαιτέρω δημοσιεύσεις σε αντίστοιχα συνέδρια και περιοδικά, οι οποίες όμως δεν περιλαμβάνονται σε αυτή τη διατριβή. Επιπλέον, στο πλαίσιο της διάδοσης της έρευνας, πραγματοποιήθηκαν συμμετοχές και παρουσιάσεις σε συμπόσια, καθώς και προφορικές παρουσιάσεις σε προσκεκλημένες ομιλίες.

Κάθε δημοσίευση υποστηρίζεται από ένα υψηλής ποιότητας και καλά τεκμηριωμένο αποθετήριο. Αυτά τα αποθετήρια τηρούν τα πρότυπα της κοινότητας και στοχεύουν να κάνουν την έρευνα προσιτή, προωθήσιμη και αναπαραγώγιμη. Κάθε αποθετήριο περιλαμβάνει κώδικα, προ-εκπαιδευμένα μοντέλα, καθώς και οποιαδήποτε νέα σύνολα δεδομένων εισάγονται. Επιπλέον, για ορισμένες δημοσιεύσεις, έχουν δημιουργηθεί διαδραστικά επιδείγματα για την ενίσχυση της εμπλοκής των χρηστών.

Contents

1 Introduction

Contents

1.1 Current Challenges and Motivation

From the moment we are born, our interaction with the world begins through sensory perception. Initially, our vision is blurry, and we can only distinguish between light and dark shades. Within the first few days, we can discriminate colors from white and start to recognize faces, particularly our mother's. Depth perception begins to develop a few weeks later, a crucial skill that precedes crawling. By the age of two months, we can focus on particular objects, but it is not until around six months

that we can perceive sharp images. This progressive refinement of visual acuity is driven by both biological maturation and the accumulation of sensory experiences.As our visual system matures, we begin to encode, store, and retrieve detailed mental images of objects, places, and people. This encoding process occurs over varying time scales, from rapid eye movements to long-term memory storage. During our early childhood, the brain is highly plastic, enabling the formation of complex visual memories. However, as we grow older, the ability to recall memories from our early years diminishes, a phenomenon known as childhood amnesia. Despite this, the capacity to form and retain new visual memories persists throughout life, allowing us to maintain an updated mental image of familiar environments and individuals.

Parallel to the development of visual perception is the evolution of speech and language capabilities. We begin to recognize and respond to speech sounds soon after birth. By six months, we can differentiate between phonemes of our native language and start babbling, producing sounds that mimic the rhythm and intonation of speech. Around our first birthday, we begin to utter our first words. This marks the beginning of a rapid expansion in vocabulary and the ability to form simple sentences. By the age of three, we can engage in complex conversations, demonstrating an understanding of syntax and grammar. The development of language involves both auditory and cognitive processes. We encode speech sounds and patterns, store these auditory experiences, and retrieve them to produce language. This development is facilitated by social interactions and the linguistic environment, highlighting the importance of experience in shaping language abilities. In the process of learning a language, we gradually grasp the semantics and syntax of our native tongue, enabling us to convey complex ideas and emotions. By integrating auditory inputs with contextual cues and visual stimuli, we develop a sophisticated understanding of our environment and how to communicate within it.

The *encoding*, *storage*, and *retrieval* of information are thus common processes, whether they involve visual, linguistic, or auditory experiences. Our sensory and cognitive experiences are encoded and stored as complex representations, which are likely unique to each individual. For instance, the word "home" evokes different mental images for each person, based on their unique experiences. This suggests that in our representational space, the linguistic representation of the word "home" is closely linked to the visual representation formed from our experiences with homes. These personal and unique representations enable visual recognition and concept understanding in a way that integrates both sensory and linguistic elements of our experience.

Transitioning from humans to machines, from human to artificial intelligence, representations continue to play a crucial role as they enable machines to perceive, interpret, and interact with the world. Visual representations, extracted from images

or videos, facilitate tasks such as image classification, image retrieval, and object detection. Visual-textual representations, bridging the gap between the visual and linguistic domains, enable tasks like image captioning, visual question answering, and composed image retrieval. The ability to learn and manipulate these representations is essential for advancing cutting-edge technology in computer vision and beyond.

This doctoral dissertation focuses on the development and application of novel methods for learning visual and visual-textual (multimodal) representations, with a specific emphasis on deep metric learning, image classification, and composed image retrieval. The research explores both data-centric and model-centric approaches to enhance the capabilities of machine learning models in visual understanding and interaction. It addresses significant challenges in representation learning and introduces new methods and benchmarks to advance the state-of-the-art in this field. This chapter provides a brief introduction to the research conducted, highlighting the motivation, current challenges, objectives, and scientific contributions of this work.

1.1.1 Visual Representations

Visual representation learning lies at the core of computer vision, enabling machines to perform tasks that require understanding and interpreting visual information [1]. Among these tasks, image classification, where the goal is to categorize an image into one of several predefined classes, is one of the most studied in machine and deep learning. It is a common source of pre-trained models for transfer learning to other tasks [2], [3]. It has been studied under different supervision settings [4], [5], knowledge transfer [6] and data augmentation [7], including the recent research on *mixup* [8], [9], where embeddings and labels are interpolated.

Deep metric learning aims to learn an explicit non-linear mapping from input space to low-dimensional embedding space, such that positive pairs of examples are close in the embedding space, while negative pairs are far apart [10]. That is, deep metric learning is about learning from pairwise interactions such that inference relies on instance embeddings, *e.g.* for nearest neighbor classification [10], instance-level retrieval [11], few-shot learning [12], face recognition [13] and semantic textual similarity [14]. Following [15], it is most often fully supervised by one class label per example, like classification. However, unlike classification, classes (and distributions) at training and inference are different in deep metric learning. Data augmentation techniques, particularly mixup, have proven to be highly effective in improving model generalization. Mixup involves interpolating between pairs of examples and their corresponding labels, thus generating new training examples [8], [9]. Thus, one might expect interpolation-based data augmentation like mixup to be even more

important in deep metric learning than in classification. Yet, recent attempts are mostly limited to special cases of embedding interpolation and have trouble with label interpolation [16]. This might be attributed to the fact that deep metric learning loss functions, unlike cross-entropy of image classification, are non-additive, thus adapting mixup presents unique challenges [17]. This raises the questions:

- *What is a proper way to define and interpolate images and labels for deep metric learning?*
- *Can this interpolation be effective when applied to input, feature and embedding spaces?*
- *Does this interpolation lead to better exploration and improved visual representations?*

Visual representation learning and spatial pooling are two interconnected processes since the study of Gabor filters [18] and early convolutional networks [19]. On the one hand, convolutional networks [20]–[22] are hierarchical, consisting of alternating convolutional and local pooling layers. They build a final global image representation typically via global average pooling [21], which compresses the entire feature map into a single vector at the network's end. On the other hand, vision transformers [23] start by splitting the image into patches, encoding each as a token, including a special classification token called CLS, inherited from language models [24]. Pooling is based on this learnable CLS token, which, beginning at the input space, undergoes the same self-attention operation with patch tokens across all layers and provides a global image representation. That is, the network ends in global weighted (average) pooling, using as weights the attention of CLS over the patch tokens.

Few works that have studied anything other than CLS for pooling in transformers are mostly limited to global average pooling [25]–[28]. CLS offers attention maps for free, but those are typically of low quality unless in a self-supervised setting [29], which is not well studied. Few works that attempt to rectify this in the supervised setting include a spatial entropy loss [30], shape distillation from convolutional networks [31] and skipping computation of self-attention, observing that the quality of self-attention is still good at intermediate layers [32]. It has also been found beneficial to inject the CLS token only at the last few layers [33].

Vision transformers can be reformulated in two streams, where one is extracting a visual representation on patch tokens and the other is performing spatial pooling on the CLS token; whereas, convolutional networks undergo global spatial pooling at the very last step, before the classifier. In this sense, one can isolate the pooling process from both kinds of networks and replace it by a new one. This raises the question:s:

- *Can a pooling process be derived at the very last step of either convolutional or transformer encoders that improves over their default?*
- *Can this process provide high-quality attention maps that delineate object boundaries, for both networks?*
- *Can these properties hold under both supervised and self-supervised settings?*

To this end, the motivation for the visual representation learning part of this dissertation is driven by the need to enhance the quality and robustness of visual representations through both data-centric and model-centric methods.

From a data-centric perspective, the challenge of effectively integrating interpolation-based data augmentation in deep metric learning sparked our interest in exploring new ways to define and apply such techniques to improve model generalization and exploration. From a model-centric perspective, the limitations of existing pooling methods in convolutional and transformer encoders highlighted the need for a more effective approach. This inspired us to investigate pooling methods that could enhance attention map quality and maintain performance across different training settings. These motivations underpin our quest to address the fundamental challenges in visual representation learning, as reflected in the critical questions posed above.

1.1.2 Multimodal Representations

Visual-textual (multimodal) representation learning, bridging the gap between the visual and linguistic domains, enable machines to perform tasks that require understanding and integrating information across both (various) modalities, such as image captioning, visual question answering and composed image retrieval. Among these tasks, composed image retrieval [34]–[39] offers a flexible way to express the user intent in a query comprising both an image and a text description. In contrast to image-to-image retrieval [40], [41], where the retrieval is based solely on the visual content of the query, and text-to-image-retrieval [42]–[44], which is only successful if the object can be accurately described in words, composed image retrieval combines both modalities, aiming to retrieve images that are not only visually similar to the query image but also modified in accordance with the specifics of the query text.

Traditionally, composed image retrieval methods are supervised by triplets [34], [45], [46], which limited their application to specific domains like fashion [47]–[49] and physical states [50] due to the labor-intensive process of labeling. The emergence of zero-shot composed image retrieval [39], [51], which refers to the ability of a model to perform retrieval in a domain it was not explicitly trained on, expanded the range of possible applications by leveraging vision-language models [52]–[54]. However, exist-

ing zero-shot composed image retrieval methods are either trained using unlabeled images [39], [51] or not trained [55], but rely heavily on large language models [56]. Most zero-shot composed image retrieval methods perform textual inversion by mapping the query image to the continuous latent space of word tokens [39], [51], which involves an optimization process.

Meanwhile, cross-domain image retrieval [57] addresses the challenge of retrieving images across different visual domains, such as style [58], texture [58] or lighting conditions [59]. Early cross-domain image retrieval methods are supervised and often struggle with generalization due to the domain gap between the query image and the target database, making it difficult to retrieve relevant images when the visual characteristics of the domains differ significantly. More recent methods dispense with the need for labeled cross-modal pairs and are unsupervised [60]–[63]. However, generalization to an unseen domain is rarely demonstrated [64], while no method includes the domain of the query image in the database, which would make the task more challenging. Unlike cross-domain retrieval, domain conversion [39] addresses a more practical, open-domain setting. In this setting, the query and database may be from any unseen domain, while the text query serves as a description of the target domain. This task is suitable for mitigating some of the limitations of convnetional cross-domain image retrieval by allowing for more flexible and dynamic retrieval across a broader range of domains. This raises the questions:

- *Can a pre-trained, frozen vision-language model be leveraged for the task of composed image retrieval without the need for additional models or fine-tuning?*
- *Given its zero-shot capabilities, can the specific task of domain conversion be focused on and expanded, thereby addressing some limitations of the closely related task of cross-domain image retrieval?*
- *Can discrete space inversion, as opposed to continuous space, be utilized along with the pre-trained, frozen vision-language model?*
- *Can an approach performing well on domain conversion, perform on par with or better than other methods in more generic composed image retrieval tasks?*

In recent years, earth observation through remote sensing has witnessed an enormous growth in data volume, creating a challenge in managing and extracting relevant information. This surge is largely attributed to the proliferation of open satellite data programs, which have democratized access to earth observation data and broadened the scope of research and applications in various fields. The capacity to efficiently organize extensive archives and quickly *retrieve* specific images is crucial. Remote sensing image retrieval [65], which aims to search and retrieve images from RS image archives, has emerged as a key solution. However, remote sensing image retrieval

methods face a limitation: reliance on a query of a single modality. This constraint often restricts users from fully expressing their specific requirements, especially given the complex and dynamic nature of Earth's surface as depicted in remote sensing imagery. Ideally, users would benefit from a system that allows them to articulate nuanced modifications or specifications in conjunction with an image-based query. This is where composed image retrieval comes into play. Composed image retrieval, integrating both image and text in the search query, is designed to retrieve images that are not only visually similar to the query image but also relevant to the details of the query text. This raises the following questions:

- *Can composed image retrieval be effectively introduced into the domain of remote sensing to enhance the expressiveness and flexibility of search capabilities?*
- *Can a training-free approach utilizing a vision-language model provide sufficient descriptive power for various attribute modifications in remote sensing imagery?*
- *Can a modality control parameter be used to balance the influence of image- and text-based components in the retrieval process to meet specific user needs?*

To this end, the motivation for the visual-textual (multimodal) representation learning part of this dissertation is driven by the need to enhance the quality and generalization of multimodal representations in the task of composed image retrieval.

From a zero-shot learning perspective, the challenge of leveraging pre-trained, frozen vision-language models sparked our interest in exploring new methods to expand their capabilities. This includes addressing the limitations of current approaches that depend on large language models, extensive training data or continuous space inversion. In remote sensing, the complexity and diversity of Earth observation data underscored the limitations of single-modality retrieval methods. This inspired us to explore more expressive retrieval approaches that integrate both image and text modalities, aiming to enhance search expressiveness and flexibility. These motivations underpin our quest to address challenges in multimodal representation learning, as reflected in the critical questions posed above.

1.2 Goal

The general aim of this dissertation is summarized in the following:

"*Address the challenges of learning visual and visual-textual (multimodal) representations from either data-centric or model-centric perspectives by developing innovative*

methods, aiming to enhance the quality, robustness and generalization of models."

1.2.1 Objectives

The main objectives of this dissertation are summarized in the following:

Objective 1 *To investigate and develop a data-augmentation method addressing the challenge of interpolating both examples and target labels in deep metric learning.*

Objective 2 *To systematically evaluate this data-augmentation method, ensuring it enhances exploration, thus improving robustness and generalization.*

Objective 3 *To investigate and develop a pooling method at the very last step of both convolutional and transformer encoders that improves over their corresponding default, under both supervised and self-supervised settings.*

Objective 4 *To design this pooling method as attention-based, ensuring that it provides high-quality attention maps, solving the attention deficit of transformers, and improving robustness, localization and interpretability.*

Objective 5 *To leverage a pre-trained, frozen vision-language model for composed image retrieval without the need for additional models, additional data or fine-tuning.*

Objective 6 *To expand the specific task of domain conversion in composed image retrieval, addressing limitations of the closely related task of cross-domain image retrieval.*

Objective 7 *To investigate and develop a method utilizing discrete-space inversion, in contrast to the continuous one, enhancing the retrieval performance.*

Objective 8 *To introduce composed image retrieval into remote sensing, enhancing the expressiveness and flexibility of search capabilities, accompanied by a benchmark dataset.*

Objective 9 *To investigate and develop a training-free method leveraging a pre-trained, frozen a vision-language model, suitable for attribute modification in remote sensing imagery.*

1.2.2 Contributions

The main contributions of this dissertation are summarized in the following:

Contribution 1 *The development of a generic way of representing and interpolating labels, allowing the straightforward extension of any kind of mixup*

to deep metric learning for a large class of loss functions. A novel mixup method is proposed within this generic formulation . This is related to Objective 1.

Contribution 2 *The systematic evaluation of the novel mixup method under different settings, including mixup at different representation levels, mixup of different pairs of examples, loss functions and hard example mining. The introduction of a new evaluation metric, utilization, validating that a representation more appropriate for test classes is implicitly learned during the exploration of the embedding space in the presence of the novel mixup method. The definition of "positivity",* i.e. *the event that a mixed example behaves as "positive" for an anchor, and the study of how it increases as a function of the interpolation factor, both theoretically and empirically. This is related to Objective 2.*

Contribution 3 *The formulation of a generic pooling framework that allows for easy inspection and qualitative comparison of a wide range of methods. Utilizing this framework, a simple, attention-based, non-iterative, universal pooling mechanism is derived, providing a single vector global representation. This is related to Objectives 3 and 4.*

Contribution 4 *The provision of high-quality attention maps for free by the novel method. The "high quality" property of these maps is evaluated not only qualitatively, but also through experiments on object localization, object discovery, and robustness to background changes that explicitly use the attention maps. This is related to Objective 4.*

Contribution 5 *The development of a training-free, discrete-space inversion method, leveraging a pre-trained and frozen vision-language model for composed image retrieval, without the need for additional models, additional data or fine-tuning. The novel method outperforms state-of-the-art methods in the domain conversion task, which was expanded by introducing four new benchmark datasets, and demonstrates competitive performance in generic composed image retrieval. This is related to Objectives 5, 6, and 7.*

Contribution 6 *The introduction of composed image retrieval into remote sensing, enhancing the expressiveness and flexibility of search. This includes the creation of a benchmark dataset to facilitate evaluation. Furthermore, the development of a simple, training-free method leveraging a pre-trained and frozen vision-language model suitable for attribute modification, utilizing a control parameter for more image- or text-oriented search results. This is related to Objectives 8 and 9.*

1.2.3 Dissemination of Research

The research conducted during this PhD has been disseminated through various scientific publications, invited talks, and online materials, as detailed below. The publications forming this dissertation, in the order of appearance within it, are:

Conference Papers

S. Venkataramanan*, **B. Psomas***, E. Kijak, L. Amsaleg, K. Karantzalos, Y. Avrithis, «It takes two to tango: Mixup for deep metric learning», in International Conference on Learning Representations (ICLR), 2022

B. Psomas, I. Kakogeorgiou, K. Karantzalos, Y. Avrithis, «Keep it simpool: Who said supervised transformers suffer from attention deficit?», in International Conference on Computer Vision (ICCV), 2023

N. Efthymiadis, **B. Psomas**, Z. Laskar, K. Karantzalos, Y. Avrithis, O. Chum, G. Tolias, «Composed image retrieval for training-free domain conversion», under review in Winter Conference on Applications of Computer Vision (WACV), 2024.

B. Psomas, I. Kakogeorgiou, N. Efthymiadis, G. Tolias, O. Chum, Y.Avrithis, K. Karantzalos, «Composed image retrieval for remote sensing», in IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2024

The following publications were conducted during the PhD but are not included in this dissertation:

Journal Articles

M. Sdraka, I. Papoutsis, **B. Psomas**, K. Vlachos, K. Ioannidis, K. Karantzalos, I. Gialampoukidis, S. Vrochidis, «Deep learning for downscaling remote sensing images: Fusion and super-resolution», IEEE Geoscience and Remote Sensing Magazine (GRSM), vol. 10, no. 3, pp. 202–255, 2022

Conference Papers

I. Kakogeorgiou, S. Gidaris, **B. Psomas**, Y. Avrithis, A. Bursuc, K. Karantzalos, N. Komodakis, «What to hide from your students: Attention-guided masked image modeling», in European Conference on Computer Vision (ECCV), 2022

P. Riccio, **B. Psomas**, F. Galati, F. Escolano, T. Hofmann, N. Oliver, «Openfilter: A framework to democratize research access to social media AR filters», Advances in Neural Information Processing Systems (NeurIPS), 2022

S. Vellas, **B. Psomas**, K. Karadima, D. Danopoulos, A. Paterakis, G. Lentaris, D. Soudris, K. Karantzalos, «Evaluation of resource-efficient crater detectors on embedded systems», in IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2024

As part of the research dissemination, attendance and poster presentations were made at symposiums:

Symposiums

«It takes two to tango: Mixup for deep metric learning», in 1^{st} ELLIS Doctoral Symposium 2023, Tübingen, Germany, 27 September - 1 October, 2021

«Openfilter: A framework to democratize research access to social media ar filters», in 2^{nd} ELLIS Doctoral Symposium 2023, Alicante, Spain, 19 - 23 September, 2022

In addition, oral presentations were made for invited talks:

Invited Talks

«Leveraging Attention in Masked Image Modeling and Pooling», in 49th Pattern Recognition and Computer Vision Colloquium, Czech Technical University in Prague, Prague, 4 April 2024

Finally, each publication included in this dissertation is supported by a high-quality, well-documented repository. These repositories adhere to community standards and aim to make the research accessible, promotable, and reproducible. Each repository contains the code, pre-trained models, and any new datasets introduced. Additionally, for some publications, interactive demos have been created to enhance user engagement. The details are listed below:

Online Material

Repository for «It takes two to tango: Mixup for deep metric learning». The repository includes code, pre-trained models, as also links to presentation slides and poster.

Repository for «Keep it simpool: Who said supervised transformers suffer from attention deficit?». The repository includes code, pre-trained models, as also links to presentation slides, poster and interactive demo.

Repository for «Composed image retrieval for remote sensing». The repository includes code, pre-trained models, as also links to presentation slides.

Additionally, the following repositories support publications conducted during the PhD but not included in this dissertation:

Online Material

Repository for «What to Hide from Your Students: Attention-Guided Masked Image Modeling». The repository includes code, pre-trained models, as also links to presentation slides and poster.

Repository for «Openfilter: A framework to democratize research access to social media AR filters». The repository includes code, datasets, as also links to presentation slides and poster.

Repository for «Evaluation of resource-efficient crater detectors on embedded system». The repository includes code, pre-trained models, as also a link to presentation slides.

1.3 Outline

The contents of this dissertation can be divided into three parts. The first part focuses on visual representations, detailing the development and evaluation of novel data-centric (chapter 2) and model-centric (chapter 3) methods. The second part delves into visual-textual (multimodal) representations, presenting innovative approaches for composed image retrieval and their applications in various domains (chapter 4 and chapter 5). Finally, chapter 6 provides a summary of the findings, conclusions, and suggestions for future work. A brief description of each chapter is presented below:

Chapter 2: Learning Visual Representations via Data Augmentation. This chapter presents the novel data augmentation method, *Metrix*, designed to address the challenge of interpolating both examples and target labels in deep metric learning. The theoretical foundation, method development, and evaluation across various benchmark datasets are discussed, demonstrating significant improvements in robustness and generalization.

Chapter 3: Learning Visual Representations via Model Architecture Component. Focusing on model architecture, this chapter introduces *SimPool*, an attention-based pooling method for convolutional neural networks and vision transformers. The

chapter details the development of a generic pooling framework, the formulation of existing methods within this framework, and the derivation of SimPool. The performance improvements, both qualitative and quantitative, are validated on standard benchmarks and downstream tasks.

Chapter 4: Extracting Multimodal Representations via Discrete-Space Inversion. Transitioning to visual-textual representations, this chapter introduces *FreeDom*, a training-free method for zero-shot composed image retrieval in open-world domain conversion. The innovative approach leverages a frozen vision-language model and employs discrete-space textual inversion. Experimental results across multiple benchmark datasets demonstrate its superior performance and the potential for further applications in generic composed image retrieval.

Chapter 5: Extracting Multimodal Representations for Remote Sensing Composed Image Retrieval. Expanding the scope of visual-textual representations, this chapter introduces a novel task: remote sensing composed image retrieval. The chapter discusses the unique challenges and capabilities of this task, presenting a new benchmark dataset, *PatternCom*, and a training-free method, *WeiCom* utilizing a modality control parameter. The method's effectiveness is evaluated through various attribute modifications, showcasing its potential for enhancing search capabilities in the remote sensing domain.

Chapter 6: Conclusion. This chapter concludes the dissertation by summarizing the key findings and contributions of the research. It highlights the advancements made in both visual and multimodal representation learning and discusses potential future directions for extending the work.

2

Learning Visual Representations via Data Augmentation

Contents

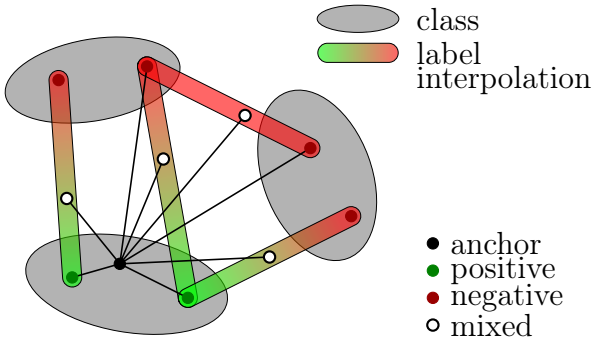


Figure 2.1: *Metrix* (= *Metric Mix*) allows an anchor to interact with **positive** (same class), **negative** (different class) and interpolated examples, which also have interpolated labels.

2.1 Revisiting Deep Metric Learning: The Role of Data Interpolation in Visual Representation Learning

Classification is one of the most studied tasks in machine learning and deep learning. It is a common source of pre-trained models for *transfer learning* to other tasks [2], [3]. It has been studied under different *supervision settings* [4], [5], *knowledge transfer* [6] and *data augmentation* [7], including the recent research on *mixup* [8], [9], where embeddings and labels are interpolated.

Deep metric learning is about learning from pairwise interactions such that inference relies on instance embeddings, *e.g.* for *nearest neighbor classification* [10], *instance-level retrieval* [11], *few-shot learning* [12], *face recognition* [13] and *semantic textual similarity* [14].

Following [15], it is most often fully supervised by one class label per example, like classification. The two most studied problems are *loss functions* [66] and *hard example mining* [67], [68]. Tuple-based losses with example weighting [69] can play the role of both.

Unlike classification, classes (and distributions) at training and inference are different in metric learning. Thus, one might expect interpolation-based data augmentation like mixup to be even more important in metric learning than in classification. Yet, recent attempts are mostly limited to special cases of embedding interpolation and have trouble with label interpolation [16]. This raises the question: *what is a proper way to define and interpolate labels for metric learning?*

In this work, we observe that metric learning is not different from classification, where examples are replaced by pairs of examples and class labels by "positive" or "negative", according to whether class labels of individual examples are the same or not. The positive or negative label of an example, or a pair, is determined in relation to a given example which is called an *anchor*. Then, as shown in Figure 2.1, a straightforward way is to use a *binary* (two class) label per pair and interpolate it linearly as in standard mixup. We call our method *Metric Mix*, or *Metrix* for short.

To show that mixing examples improves representation learning, we quantitatively measure the properties of the test distributions using *alignment* and *uniformity* [70]. *Alignment* measures the clustering quality and *uniformity* measures its distribution over the embedding space; a well clustered and uniformly spread distribution indicates higher representation quality. We also introduce a new metric, *utilization*, to measure the extent to which a test example, seen as a query, lies near any of the training examples, clean or mixed. By quantitatively measuring these three metrics, we show that interpolation-based data augmentation like mixup is very important in metric learning, given the difference between distributions at training and inference.

In summary, we make the following contributions:

1. We define a generic way of representing and interpolating labels, which allows straightforward extension of any kind of mixup to deep metric learning for a large class of loss functions. We develop our method on a generic formulation that encapsulates these functions (section 2.3).
2. We define the "positivity" of a mixed example and we study precisely how it increases as a function of the interpolation factor, both in theory and empirically (subsection 2.3.6).
3. We systematically evaluate mixup for deep metric learning under different settings, including mixup at different representation levels (input/manifold), mixup of different pairs of examples (anchors/positives/negatives), loss functions and hard example mining (subsection 2.4.3).
4. We introduce a new evaluation metric, *utilization*, validating that a representation more appropriate for test classes is implicitly learned during exploration of the embedding space in the presence of mixup (subsection 2.4.5).

5. We improve the state of the art on four common metric learning benchmarks (subsection 2.4.3).

2.2 Contextualizing Deep Metric Learning and Data Interpolation

Metric learning Metric learning aims to learn a metric such that *positive* pairs of examples are nearby and *negative* ones are far away. Conventionally, methods have been linear using the Mahalanobis distance or non-linear using kernels [71]. In *deep metric learning*, we learn an explicit non-linear mapping from raw input to a low-dimensional *embedding space* [10], where the Euclidean distance has the desired properties. Although learning can be unsupervised [72], deep metric learning has mostly followed the supervised approach, where positive and negative pairs are defined as having the same or different class label, respectively [15].

Loss functions can be distinguished into pair-based and proxy-based [66]. *Pair-based* losses use pairs of examples [67], [72], which can be defined over triplets [13], [73]–[75], quadruples [76] or tuples [10], [69], [77]. *Proxy-based* losses use one or more proxies per class, which are learnable parameters in the embedding space [78]–[82]. Pair-based losses capture data-to-data relations, but they are sensitive to noisy labels and outliers. They often involve terms where given constraints are satisfied, which produce zero gradients and do not contribute to training. This necessitates *mining* of hard examples that violate the constraints, like semi-hard [13] and distance weighted [67]. By contrast, proxy-based losses use data-to-proxy relations, assuming proxies can capture the global structure of the embedding space. They involve less computations that are more likely to produce nonzero gradient, hence have less or no dependence on mining and converge faster.

Mixup *Input mixup* [8] linearly interpolates between two or more examples in the input space for data augmentation. Numerous variants take advantage of the structure of the input space to interpolate non-linearly, *e.g.* for images [83]–[89]. *Manifold mixup* [9] interpolates intermediate representations instead, where the structure is learned. This can be applied to or assisted by decoding back to the input space [90]–[94]. In both cases, corresponding labels are linearly interpolated too. Most studies are limited to cross-entropy loss for classification. Pairwise loss functions have been under-studied, as discussed below.

Interpolation for pairwise loss functions As discussed in subsection 2.3.3, interpolating target labels is not straightforward in pairwise loss functions. In *deep*

metric learning, *embedding expansion* [16], HDML [95] and *symmetrical synthesis* [96] interpolate pairs of embeddings in a deterministic way within the same class, applying to pair-based losses, while *proxy synthesis* [17] interpolates between classes, applying to proxy-based losses. None performs label interpolation, which means that [17] risks synthesizing false negatives when the interpolation factor λ is close to 0 or 1.

In *contrastive representation learning*, MoCHi [97] interpolates anchor with negative embeddings but not labels and chooses $\lambda \in [0, 0.5]$ to avoid false negatives. This resembles thresholding of λ at 0.5 in OptTransMix [93]. Finally, *i-mix* [98] and MixCo [99] interpolate pairs of anchor embeddings as well as their (virtual) class labels linearly. There is only one positive, while all negatives are clean, so it cannot take advantage of interpolation for relative weighting of positives/negatives per anchor [69].

By contrast, Metrix is developed for deep metric learning and applies to a large class of both pair-based and proxy-based losses. It can interpolate inputs, intermediate features or embeddings of anchors, (multiple) positives or negatives *and* the corresponding two-class (positive/negative) labels per anchor, such that relative weighting of positives/negatives depends on interpolation.

2.3 Integrating Mixup into Deep Metric Learning

2.3.1 Preliminaries

Problem formulation We are given a training set $X \subset \mathcal{X}$, where $\mathcal{X}$ is the input space. For each *anchor* $a \in X$, we are also given a set $P(a) \subset X$ of *positives* and a set $N(a) \subset X$ of *negatives*. The positives are typically examples that belong to the same class as the anchor, while negatives belong to a different class. The objective is to train the parameters θ of a model $f : X \to \mathbb{R}^d$ that maps input examples to a d-dimensional *embedding*, such that positives are close to the anchor and negatives are far away in the embedding space. Given two examples $x, x' \in \mathcal{X}$, we denote by $s(x, x')$ the *similarity* between x, x' in the embedding space, typically a decreasing function of Euclidean distance. It is common to ℓ_2-normalize embeddings and define $s(x, x') := \langle f(x), f(x') \rangle$, which is the *cosine similarity*. To simplify notation, we drop the dependence of f, s on θ.

Pair-based losses [10], [69], [72], [73] use both anchors and positives/negatives in X, as discussed above. *Proxy-based* losses define one or more learnable *proxies* $\in \mathbb{R}^d$ per class, and only use proxies as anchors [80] or as positives/negatives [78], [79], [81]. To accommodate for uniform exposition, we extend the definition of similarity as $s(v, x) := \langle v, f(x) \rangle$ for $v \in \mathbb{R}^d, x \in \mathcal{X}$ (proxy anchors) and $s(x, v) := \langle f(x), v \rangle$

for $x \in \mathcal{X}, v \in \mathbb{R}^d$ (proxy positives/negatives). Finally, to accommodate for mixed embeddings in subsection 2.3.5, we define $s(v, v') := \langle v, v' \rangle$ for $v, v' \in \mathbb{R}^d$.

Thus, we define $s : (\mathcal{X} \cup \mathbb{R}^d)^2 \to \mathbb{R}$ over pairs of either inputs in $\mathcal{X}$ or embeddings in $\mathbb{R}^d$. We discuss a few representative loss functions below, before deriving a generic form.

Contrastive The contrastive loss [72] encourages positive examples to be pulled towards the anchor and negative examples to be pushed away by a margin $m \in \mathbb{R}$. This loss is *additive* over positives and negatives, defined as:

$$\ell_{\text{cont}}(a; \theta) := \sum_{p \in P(a)} -s(a, p) + \sum_{n \in N(a)} [s(a, n) - m]_+ \tag{2.1}$$

Multi-Similarity The multi-similarity loss [69] introduces *relative weighting* to encourage positives (negatives) that are farthest from (closest to) the anchor to be pulled towards (pushed away from) the anchor by a higher weight. This loss is *not* additive over positives and negatives:

$$\begin{aligned} \ell_{\text{MS}}(a; \theta) :=& \frac{1}{\beta} \log \left(1 + \sum_{p \in P(a)} e^{-\beta(s(a,p)-m)} \right) + \\ & \frac{1}{\gamma} \log \left(1 + \sum_{n \in N(a)} e^{\gamma(s(a,n)-m)} \right) \end{aligned} \tag{2.2}$$

Here, $\beta, \gamma \in \mathbb{R}$ are scaling factors for positives, negatives respectively.

Proxy Anchor The proxy anchor loss [80] defines a learnable *proxy* in $\mathbb{R}^d$ for each class and only uses proxies as anchors. For a given anchor (proxy) $a \in \mathbb{R}^d$, the loss has the same form as (2.2), although similarity s is evaluated on $\mathbb{R}^d \times \mathcal{X}$.

2.3.2 Generic Loss Formulation

We observe that both additive (2.1) and non-additive (2.2) loss functions involve a sum over positives $P(a)$ and a sum over negatives $N(a)$. They also involve a decreasing function of similarity $s(a, p)$ for each positive $p \in P(a)$ and an increasing function of similarity $s(a, n)$ for each negative $n \in N(a)$. Let us denote by ρ^+, ρ^- this function for positives, negatives respectively. Then, non-additive functions differ

Loss	Anchor	P/N	$\tau(x)$	$\sigma^+(x)$	$\sigma^-(x)$	$\rho^+(x)$	$\rho^-(x)$
Contrastive [72]	X	X	x	x	x	$-x$	$[x-m]_+$
Lifted structure [75]	X	X	$[x]_+$	$\log(x)$	$\log(x)$	e^{-x}	e^{x-m}
Binomial dev. [100]	X	X	x	$\log(1+x)$	$\log(1+x)$	$e^{-\beta(x-m)}$	$e^{\gamma(x-m)}$
Multi-similarity [69]	X	X	x	$\frac{1}{\beta}\log(1+x)$	$\frac{1}{\gamma}\log(1+x)$	$e^{-\beta(x-m)}$	$e^{\gamma(x-m)}$
Proxy Anchor [80]	proxy	X	x	$\frac{1}{\beta}\log(1+x)$	$\frac{1}{\gamma}\log(1+x)$	$e^{-\beta(x-m)}$	$e^{\gamma(x-m)}$
NCA [101]	X	X	x	$-\log(x)$	$\log(x)$	e^{x}	e^{x}
ProxyNCA [78]	X	proxy	x	$-\log(x)$	$\log(x)$	e^{x}	e^{x}
SoftTriple [79]	X	proxy	x	$-\log(x)$	$\log(x)$	$e^{\beta(x-m)}$	$e^{\beta(x-m)}+\sum e^{\beta x}$
EPSHN [102]	X	X	x	$-\log(x)$	$\log(x)$	e^{x}	$e^{x_+}+e^{x}$
ProxyNCA++ [81]	X	proxy	x	$-\log(x)$	$\log(x)$	$e^{x/T}$	$e^{x/T}$

Table 2.1: *Loss functions under the generic loss formulation.* Anchor/positive/negative: X: embedding of input example from training set X by f; proxy: learnable parameter in $\mathbb{R}^d$; T: temperature. All loss functions are encompassed by (2.3) using the appropriate definition of functions $\tau, \sigma^+, \sigma^-, \rho^+, \rho^-$ as given here.

from additive by the use of a nonlinear function σ^+, σ^- on positive and negative terms respectively, as well as possibly another nonlinear function τ on their sum:

$$\ell(a;\theta) := \tau\left(\sigma^+\left(\sum_{p\in P(a)}\rho^+(s(a,p))\right)+\sigma^-\left(\sum_{n\in N(a)}\rho^-(s(a,n))\right)\right) \tag{2.3}$$

With the appropriate choice for $\tau, \sigma^+, \sigma^-, \rho^+, \rho^-$, this definition encompasses contrastive (2.1), multi-similarity (2.2) or proxy anchor as well as many pair-based or proxy-based loss functions, as shown in Table 2.1. It does not encompass the *triplet loss* [73], which operates on pairs of positives and negatives, forming triplets with the anchor. The triplet loss is the most challenging in terms of mining because there is a very large number of pairs and only few contribute to the loss. We only use function τ to accommodate for *lifted structure* [10], [75], where $\tau(x) := [x]_+$ is reminiscent of the triplet loss. We observe that multi-similarity [69] differs from *binomial deviance* [100] only in the weights of the positive and negative terms. Proxy anchor [80] is a proxy version of multi-similarity [69] on anchors and ProxyNCA [78] is a proxy version of NCA [101] on positives/negatives.

This generic formulation highlights the components of the loss functions that are additive over positives/negatives and paves the way towards incorporating mixup.

2.3.3 Improving Representations Using Mixup

To improve the learned representations, we follow [8], [9] in mixing inputs and features from intermediate network layers, respectively. Both are developed for classification.

Input mixup [8] augments data by linear interpolation between a pair of input examples. Given two examples $x, x' \in \mathcal{X}$ we draw $\lambda \sim \text{Beta}(\alpha, \alpha)$ as *interpolation factor* and mix x with x' using the standard mixup operation $\text{mix}_\lambda(x, x') := \lambda x + (1-\lambda)x'$.

Manifold mixup [9] linearly interpolates between intermediate representations (features) of the network instead. Referring to 2D images, we define $g_m : \mathcal{X} \to \mathbb{R}^{c \times w \times h}$ as the mapping from the input to intermediate layer m of the network and $f_m : \mathbb{R}^{c \times w \times h} \to \mathbb{R}^d$ as the mapping from intermediate layer m to the embedding, where c is the number of channels (feature dimensions) and $w \times h$ is the spatial resolution. Thus, our model f can be expressed as the composition $f = f_m \circ g_m$.

For manifold mixup, we follow [94] and mix either features of intermediate layer m or the final embeddings. Thus, we define three *mixup types* in total:

$$f_\lambda(x, x') := \begin{cases} f(\text{mix}_\lambda(x, x')), & \text{input mixup} \\ f_m(\text{mix}_\lambda(g_m(x), g_m(x'))), & \text{feature mixup} \\ \text{mix}_\lambda(f(x), f(x')), & \text{embedding mixup} \end{cases} \tag{2.4}$$

Function $f_\lambda : \mathcal{X}^2 \to \mathbb{R}^d$ performs both input and manifold mixup. We explore different mixup types in subsection 2.4.4.

2.3.4 Label Representation

Classification In supervised classification, each example $x \in X$ is assigned an one-hot encoded label $y \in \{0, 1\}^C$, where C is the number of classes. Label vectors are also linearly interpolated: given two labeled examples $(x, y), (x', y')$, the interpolated label is $\text{mix}_\lambda(y, y')$. The loss (cross-entropy) is a continuous function of the label vector. We extend this idea to metric learning.

Metric learning Positives $P(a)$ and negatives $N(a)$ of anchor a are defined as having the same or different class label as the anchor, respectively. To every example in $P(a) \cup N(a)$, we assign a binary (two-class) label $y \in \{0, 1\}$, such that $y = 1$ for positives and $y = 0$ for negatives:

$$U^+(a) := \{(p, 1) : p \in P(a)\} \tag{2.5}$$
$$U^-(a) := \{(n, 0) : n \in N(a)\} \tag{2.6}$$

Thus, we represent both positives and negatives by $U(a) := U^+(a) \cup U^-(a)$. We now rewrite the generic loss function (2.3) as:

$$\ell(a;\theta) := \tau\left(\sigma^+\left(\sum_{(x,y)\in U(a)} y\rho^+(s(a,x))\right) + \sigma^-\left(\sum_{(x,y)\in U(a)} (1-y)\rho^-(s(a,x))\right)\right) \tag{2.7}$$

Here, every labeled example (x, y) in $U(a)$ appears in both positive and negative terms. However, because label y is binary, only one of the two contributions is nonzero. Now, in the presence of mixup, we can linearly interpolate labels exactly as in classification.

2.3.5 Mixed Loss Function

Mixup For every anchor a, we are given a set $M(a)$ of pairs of examples to mix. This is a subset of $(S(a)\cup U(a))\times U(a)$ where $S(a) := (a, 1)$. That is, we allow mixing between positive-negative, positive-positive and negative-negative pairs, where the anchor itself is also seen as positive. We define the possible choices of *mixing pairs* $M(a)$ in subsection 2.4.1 and we assess them in subsection 2.4.4. Let $V(a)$ be the set of corresponding *labeled mixed embeddings*:

$$V(a) := \{(f_\lambda(x, x'), \mathrm{mix}_\lambda(y, y')) : ((x, y), (x', y')) \in M(a), \lambda \sim \mathrm{Beta}(\alpha, \alpha)\}, \tag{2.8}$$

where f_λ is defined by (2.4). With these definitions in place, the generic loss function $\widetilde{\ell}$ over mixed examples takes exactly the same form as (2.7), with only $U(a)$ replaced by $V(a)$:

$$\widetilde{\ell}(a;\theta) := \tau\left(\sigma^+\left(\sum_{(v,y)\in V(a)} y\rho^+(s(a,v))\right) + \sigma^-\left(\sum_{(v,y)\in V(a)} (1-y)\rho^-(s(a,v))\right)\right), \tag{2.9}$$

where similarity s is evaluated on $\mathcal{X} \times \mathbb{R}^d$ for pair-based losses and on $\mathbb{R}^d \times \mathbb{R}^d$ for proxy anchor. Now, every labeled embedding (v, y) in $V(a)$ appears in both positive and negative terms and *both* contributions are nonzero for positive-negative pairs, because after interpolation, $y \in [0, 1]$.

Error function Parameters θ are learned by minimizing the error function, which is a linear combination of the *clean loss* (2.3) and the *mixed loss* (2.9), averaged over all anchors

$$E(X;\theta) := \frac{1}{|X|}\sum_{a\in X} \ell(a;\theta) + w\widetilde{\ell}(a;\theta), \tag{2.10}$$

where $w \geq 0$ is the *mixing strength*. At least for manifold mixup, this combination comes at little additional cost, since clean embeddings are readily available.

Interpretation To better understand the two contributions of a labeled embedding (v, y) in $V(a)$ to the positive and negative terms of (2.9), consider the case of positive-negative mixing pairs, $M(a) \subset U^+(a) \times U^-(a)$. Then, for $((x, y), (x', y')) \in M(a)$, the mixed label is $\mathrm{mix}_\lambda(y, y') = \mathrm{mix}_\lambda(1, 0) = \lambda$ and (2.9) becomes

$$\begin{aligned}\widetilde{\ell}(a;\theta) = \tau\Bigg(&\sigma^+\left(\sum_{(v,\lambda)\in V(a)} \lambda\rho^+(s(a,v))\right)\\ &+ \sigma^-\left(\sum_{(v,\lambda)\in V(a)} (1-\lambda)\rho^-(s(a,v))\right)\Bigg)\end{aligned} \tag{2.11}$$

Thus, the mixed embedding v is both positive (with weight λ) and negative (with weight $1 - \lambda$). Whereas for positive-positive mixing, that is, for $M(a) \subset U^+(a)^2$, the mixed label is 1 and the negative term vanishes. Similarly, for negative-negative mixing, that is, for $M(a) \subset U^-(a)^2$, the mixed label is 0 and the positive term vanishes.

In the particular case of contrastive (2.1) loss, positive-negative mixing (2.11) becomes:

$$\widetilde{\ell}_{\mathrm{cont}}(a;\theta) := \sum_{(v,\lambda)\in V(a)} -\lambda s(a,v) + \sum_{(v,\lambda)\in V(a)} (1-\lambda)[s(a,v) - m]_+ \tag{2.12}$$

Similarly, for multi-similarity (2.2):

$$\begin{aligned}\widetilde{\ell}_{\mathrm{MS}}(a;\theta) :=& \frac{1}{\beta}\log\left(1 + \sum_{(v,\lambda)\in V(a)} \lambda e^{-\beta(s(a,v)-m)}\right) +\\ &\frac{1}{\gamma}\log\left(1 + \sum_{(v,\lambda)\in V(a)} (1-\lambda) e^{\gamma(s(a,v)-m)}\right)\end{aligned} \tag{2.13}$$

2.3.6 Analysis: Mixed Embeddings and Positivity

Positivity Under positive-negative mixing, (2.11) shows that a mixed embedding v with interpolation factor λ behaves as both positive and negative to different extents, depending on λ: mostly positive for λ close to 1, mostly negative for λ close to 0. The net effect depends on the derivative of the loss with respect to the similarity $\partial\widetilde{\ell}(a;\theta)/\partial s(a,v)$: if the derivative is negative, then v behaves as positive and vice versa. This is clear from the chain rule

$$\frac{\partial\widetilde{\ell}(a;\theta)}{\partial v} = \frac{\partial\widetilde{\ell}(a;\theta)}{\partial s(a,v)} \cdot \frac{\partial s(a,v)}{\partial v}, \tag{2.14}$$

because $\partial s(a,v)/\partial v$ is a vector pointing in a direction that makes a, v more similar and the loss is being minimized. Let $\mathrm{Pos}(a,v)$ be the event that v behaves as "positive", *i.e.*, $\partial\widetilde{\ell}(a;\theta)/\partial s(a,v) \leq 0$ and minimizing the loss will increase the similarity $s(a,v)$.

Multi-similarity We estimate the probability of $\mathrm{Pos}(a,v)$ as a function of λ in the case of multi-similarity with a single embedding v obtained by mixing a positive with a negative:

$$\widetilde{\ell}_{\mathrm{MS}}(a;\theta) = \frac{1}{\beta}\log\left(1+\lambda e^{-\beta(s(a,v)-m)}\right) + \frac{1}{\gamma}\log\left(1+(1-\lambda)e^{\gamma(s(a,v)-m)}\right) \tag{2.15}$$

In this case, $\mathrm{Pos}(a,v)$ occurs if and only if

$$\frac{\partial\widetilde{\ell}_{\mathrm{MS}}(a;\theta)}{\partial s(a,v)} = \frac{-\lambda e^{-\beta(s(a,v)-m)}}{(1+\lambda e^{-\beta(s(a,v)-m)})} + \frac{(1-\lambda)e^{\gamma(s(a,v)-m)}}{(1+(1-\lambda)e^{\gamma(s(a,v)-m)})} \leq 0 \tag{2.16}$$

By letting $t := s(a,v) - m$, this condition is equivalent to

$$\frac{(1-\lambda)e^{\gamma t}}{(1+(1-\lambda)e^{\gamma t})} \leq \frac{\lambda e^{-\beta t}}{(1+\lambda e^{-\beta t})} \tag{2.17}$$

$$(1-\lambda)e^{\gamma t}(1+\lambda e^{-\beta t}) \leq \lambda e^{-\beta t}(1+(1-\lambda)e^{\gamma t}) \tag{2.18}$$

$$(1-\lambda)e^{\gamma t} + \lambda(1-\lambda)e^{(\gamma-\beta)t} \leq \lambda e^{-\beta t} + \lambda(1-\lambda)e^{(\gamma-\beta)t} \tag{2.19}$$

$$e^{(\beta+\gamma)t} \leq \frac{\lambda}{1-\lambda} \tag{2.20}$$

$$(\beta+\gamma)(s(a,v)-m) \leq \ln\left(\frac{\lambda}{1-\lambda}\right) \tag{2.21}$$

$$s(a,v) \leq \frac{1}{\beta+\gamma}\ln\left(\frac{\lambda}{1-\lambda}\right) + m. \tag{2.22}$$

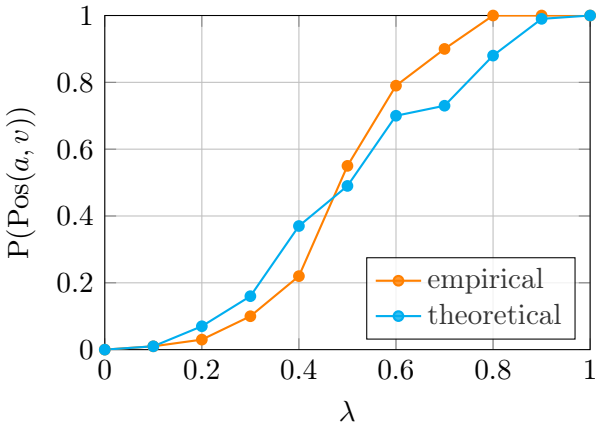


Figure 2.2: *"Positivity" of mixed embeddings* vs. λ. We measure $\mathrm{P}(\mathrm{Pos}(a,v))$ **empirically** as $\mathrm{P}(\partial\widetilde{\ell}_{\mathrm{MS}}(a;\theta)/\partial s(a,v) \leq 0)$ and **theoretically** by (2.23), where F_λ is again measured from data. We use embedding mixup on MS (2.2) on CUB200 at epoch 0, based on the setup of subsection 2.4.1.

Finally, the probability of $\mathrm{Pos}(a,v)$ as a function of λ is

$$\mathrm{P}(\mathrm{Pos}(a,v)) = F_\lambda\left(\frac{1}{\beta+\gamma}\ln\left(\frac{\lambda}{1-\lambda}\right)+m\right), \tag{2.23}$$

where F_λ is the CDF of similarities $s(a,v)$ between anchors a and mixed embeddings v with interpolation factor λ.

In Figure 2.2, we measure the probability of $\mathrm{Pos}(a,v)$ as a function of λ in two ways. First, we measure the derivative $\partial\widetilde{\ell}_{\mathrm{MS}}(a;\theta)/\partial s(a,v)$ for anchors a and mixed embeddings v over the entire dataset and we report the empirical probability of this derivative being non-positive versus λ. Second, we measure $\mathrm{P}(\mathrm{Pos}(a,v))$ theoretically using (2.23), where the CDF of similarities $s(a,v)$ is again measured empirically for a and v over the dataset, as a function of λ. Despite the simplifying assumption of a single positive and a single negative in deriving (2.23), we observe that the two measurements agree in general. They are both increasing functions of λ of sigmoidal shape, they roughly yield $\mathrm{P}(\mathrm{Pos}(a,v)) \geq 0.5$ for $\lambda \geq 0.5$ and they confirm that a mixed embedding is mostly positive for λ close to 1 and mostly negative for λ close to 0.

Dataset	CUB200 [103]	Cars196 [104]	SOP [10]	In-Shop [105]
Objects	birds	cars	household furniture	clothes
# classes	200	196	22,634	7,982
# training images	5,894	8,092	60,026	26,356
# testing images	5,894	8,093	60,027	26,356
# training classes	100	98	11,318	3991
# testing classes	100	98	11,318	3991
sampling	random	random	balanced	balanced
samples per class	–	–	5	5
classes per batch	65†	70†	20	20
learning rate	1×10^{-4}	1×10^{-4}	3×10^{-5}	1×10^{-4}

Table 2.2: *Statistics and settings* for the four datasets we use in our experiments. †: average.

2.4 Evaluating the Impact of Mixup on Deep Metric Learning: Performance and Insights

2.4.1 Setup

Datasets and sampling We experiment on Caltech Birds (CUB200) [103], Stanford Cars (Cars196) [104], Stanford Online Products (SOP) [10] and In-Shop Clothing retrieval (In-Shop) [105] image datasets. Dataset statistics are summarized in Table 2.2. Since the number of classes is large compared to the batch size in SOP and In-Shop, batches would rarely contain a positive pair when sampled uniformly at random. Hence, we use *balanced sampling* [106], *i.e.*, a fixed number of classes and examples per class, as shown in Table 2.2. For fair comparison with baseline methods, images are randomly flipped and cropped to 224×224 at training. At inference, we resize to 256×256 and then center-crop to 224×224.

Network, features and embeddings We use ResNet-50 [21] (R-50) pretrained on ImageNet-1k [107] as a backbone network. We obtain the intermediate representation (*feature*), a $7 \times 7 \times 2048$ tensor, from the last convolutional layer. Following [80], we combine adaptive average pooling with max pooling, followed by a fully-connected layer to obtain the *embedding* of $d = 512$ dimensions.

Loss functions We reproduce *contrastive* (Cont) [72], *multi-similarity* (MS) [69], *proxy anchor* (PA) [80] and *ProxyNCA++* [81] and we evaluate them under different mixup types. For MS (2.2), following [66], we use $\beta = 18$, $\gamma = 75$ and $m = 0.77$. For PA, we use $\beta = \gamma = 32$ and $m = 0.1$, as reported by the authors. As baselines, we reproduce and compare with *triplet* [74], *lifted structure* [10], ProxyNCA [78], *margin* [67] and SoftTriple [79] losses, without mixup. By reporting published results,

we also compare with D&C [108] and EPSHN [102].

Methods We compare our method, *Metrix*, with *proxy synthesis* (PS) [17], *i-mix* [98] and MoCHi [97]. For PS, we adapt the official code[1] to PA on all datasets, and use it with PA only, because it is designed for proxy-based losses. PS has been shown superior to [16], [96], although in different networks. MoCHi and *i-mix* are meant for contrastive representation learning.

Evaluation protocol We follow the standard evaluation protocol of [10], where half of the classes are used for training and the other half for testing. For each test example used as a query, we find its K-nearest neighbors within the test set's embedding space, excluding the query itself. Each query is assigned a score of 1 if it has at least one neighbor from the same class, and 0 otherwise. We measure Recall@K, which is the mean of these scores across all test examples.

Implementation details We train R-50 using AdamW [109] optimizer for 100 epochs with a batch size 100. The initial learning rate per dataset is shown in Table 2.2. The learning rate is decayed by 0.1 for Cont and by 0.5 for MS and PA on CUB200 and Cars196. For SOP and In-Shop, we decay the learning rate by 0.25 for all losses. The weight decay is set to 0.0001.

2.4.2 Mixup Settings

In mixup for classification, given a batch of n examples, it is standard to form n pairs of examples by pairing the batch with a *random permutation* of itself, resulting in n mixed examples, either for input or manifold mixup. In metric learning, it is common to obtain n embeddings and then use all $\frac{1}{2}n(n-1)$ pairs of embeddings in computing the loss. We thus treat mixup types differently.

Input mixup Mixing all pairs would be computationally expensive in this case, because we would compute $\frac{1}{2}n(n-1)$ embeddings. A random permutation would not produce as many hard examples as can be found in all pairs. Thus, for each anchor (each example in the batch), we use the k *hardest negative* examples and mix them with positives or with the anchor. We use $k = 3$ by default.

Manifold mixup Originally, manifold mixup [9] focuses on the *first* few layers of the network. Mixing all pairs would then be even more expensive than input mixup, because intermediate features (tensors) are even larger than input examples. Hence,

[1] https://github.com/navervision/proxy-synthesis

we focus on the *last* few layers instead, where features and embeddings are compact, and we mix all pairs. We use feature mixup by default and call it *Metrix/feature* or just *Metrix*, while input and embedding mixup are called *Metrix/input* and *Metrix/embed*, respectively. All options are studied in subsection 2.4.4.

Mixing pairs Whatever the mixup type, we use clean examples as anchors and we define a set $M(a)$ of pairs of examples to mix for each anchor a, with their labels (positive or negative). By default, we mix positive-negative or anchor-negative pairs, according to $M(a) := U^+(a) \times U^-(a)$ and $M(a) := S(a) \times U^-(a)$, respectively, where $U^-(a)$ is replaced by hard negatives only for input mixup. The two options are combined by choosing uniformly at random in each iteration. More options are studied in subsection 2.4.4.

Implementation details For any given mixup type or set of mixup pairs, the interpolation factor λ is drawn from $\text{Beta}(\alpha, \alpha)$ with $\alpha = 2$. We empirically set the mixup strength (2.10) to $w = 0.4$ for positive-negative pairs and anchor-negative pairs.

2.4.3 Results

Improving the state of the art As shown in Table 2.3, *all* three mixup types (input, feature, embedding) consistently improve the performance of all baseline losses (Cont, MS, PA, ProxyNCA++) across all datasets. Metrix (feature mixup) works best, followed by Metrix/embed (embedding mixup) and Metrix/input (input mixup). Surprisingly, MS outperforms PA and ProxyNCA++ under mixup on all datasets but SOP, where the three losses are on par. This is despite the fact that baseline PA outperforms MS on CUB200 and Cars-196, while ProxyNCA++ outperforms MS on SOP and In-Shop. Both contrastive and MS are significantly improved by mixup. By contrast, improvements on PA and ProxyNCA++ are marginal, which may be due to the already strong performance of PA, or further improvement is possible by employing different mixup methods that take advantage of the image structure.

In terms of Recall@1, our MS+Metrix is best overall, improving by 3.6% (67.8 → 71.4) on CUB200, 1.8% (87.8 → 89.6) on Cars196, 4.1% (76.9 → 81.0) on SOP and 2.1% (90.1 → 92.2) on In-Shop. The same solution sets new state of the art, outperforming the previously best PA by 1.7% (69.7 → 71.4) on CUB200, MS by 1.8% (87.8 → 89.6) on Cars196, ProxyNCA++ by 0.3% (80.7 → 81.0) on SOP and SoftTriple by 1.2% (91.0 → 92.2) on In-Shop. Importantly, while the previous state of the art comes from a different loss per dataset, MS+Metrix is almost consistently best across all datasets.

Method	CUB200			Cars196			SOP			In-Shop		
	R@1	R@2	R@4	R@1	R@2	R@4	R@1	R@10	R@100	R@1	R@10	R@20
Triplet [74]	63.5	75.6	84.4	77.3	85.4	90.8	70.5	85.6	94.3	85.3	96.6	97.8
LiftedStructure [10]	65.9	75.8	84.5	81.4	88.3	92.4	76.1	88.6	95.2	88.6	97.6	98.4
ProxyNCA [78]	65.2	75.6	83.8	81.2	87.9	92.6	73.2	87.0	94.4	86.2	95.9	97.0
Margin [67]	65.0	76.2	84.6	82.1	88.7	92.7	74.8	87.8	94.8	88.6	97.0	97.8
SoftTriple [79]	67.3	77.7	86.2	86.5	91.9	95.3	79.8	91.2	96.3	91.0	97.6	98.3
D&C [108]*	65.9	76.6	84.4	84.6	90.7	94.1	75.9	88.4	94.9	85.7	95.5	96.9
EPSHN [102]*	64.9	75.3	83.5	82.7	89.3	93.0	78.3	90.7	96.3	87.8	95.7	96.8
Contrastive [72]	64.7	75.9	84.6	81.6	88.2	92.7	74.9	87.0	93.9	86.4	94.7	96.2
+Metrix/input	66.3	77.1	85.2	82.9	89.3	93.7	75.8	87.8	94.6	87.7	95.9	96.5
+Metrix	67.4	77.9	85.7	85.1	91.1	94.6	77.5	89.1	95.5	89.1	95.7	97.1
+Metrix/embed	66.4	77.6	85.4	83.9	90.3	94.1	76.7	88.6	95.2	88.4	95.4	96.8
Multi-Similarity [69]	67.8	77.8	85.6	87.8	92.7	95.3	76.9	89.8	95.9	90.1	97.6	98.4
+Metrix/input	69.0	79.1	86.0	89.0	93.4	96.0	77.9	90.6	95.9	91.8	98.0	98.9
+Metrix	**71.4**	80.6	86.8	**89.6**	**94.2**	96.0	81.0	92.0	**97.2**	**92.2**	**98.5**	98.6
+Metrix/embed	70.2	80.4	86.7	88.8	92.9	95.6	78.5	91.3	96.7	91.9	98.3	98.7
Proxy Anchor [80]*	69.7	80.0	87.0	87.7	92.9	95.8	–	–	–	–	–	–
Proxy Anchor [80]	69.5	79.3	87.0	87.6	92.3	95.5	79.1	90.8	96.2	90.0	97.4	98.2
+Metrix/input	70.5	81.2	87.8	88.2	93.2	96.2	79.8	91.4	96.5	90.9	98.1	98.4
+Metrix	71.0	**81.8**	**88.2**	89.1	93.6	**96.7**	**81.3**	91.7	96.9	91.9	98.2	**98.8**
+Metrix/embed	70.4	81.1	87.9	88.9	93.3	96.4	80.6	91.7	96.6	91.6	98.3	98.3
ProxyNCA++ [81]*	69.0	79.8	87.3	86.5	92.5	95.7	80.7	92.0	96.7	90.4	98.1	98.8
ProxyNCA++ [81]	69.1	79.5	87.7	86.6	92.1	95.4	80.4	91.7	96.7	90.2	97.6	98.4
+Metrix/input	69.7	79.9	88.3	87.5	92.9	96.0	80.9	92.2	96.9	91.4	98.1	98.8
+Metrix	70.4	80.6	88.7	88.5	93.4	96.5	**81.3**	**92.7**	97.1	91.9	98.1	98.8
+Metrix/ embed	70.2	80.2	88.2	88.1	93.0	96.2	81.1	92.4	97.0	91.6	98.1	98.8

Table 2.3: *Improving the SOTA with Metrix* using ResNet-50 with embedding size $d = 512$ on four datasets. R@K (%): Recall@K; higher is better. *: reported by authors.

Method		CUB200			Cars196			SOP			In-Shop		
	Mixing Pairs	R@1	R@2	R@4	R@1	R@2	R@4	R@1	R@10	R@100	R@1	R@10	R@20
Contrastive [72]	–	64.7	75.9	84.6	81.6	88.2	92.7	74.9	87.0	93.9	86.4	94.7	96.3
+ *i*-Mix [98]	anc-neg	65.8	76.2	84.9	82.0	88.5	93.2	75.2	87.3	94.2	87.1	95.4	96.1
+ Metrix/input	pos-neg/anc-neg	**66.3**	**77.1**	**85.2**	**82.9**	**89.3**	**93.7**	**75.8**	**87.8**	**94.6**	**87.7**	**95.9**	**96.5**
+MoCHi [97]	neg-neg	63.1	74.3	83.8	76.3	84.0	89.3	68.9	83.1	91.8	81.8	91.9	93.9
+MoCHi [97]	anc-neg	65.2	75.8	84.2	82.5	88.0	92.9	75.8	87.1	94.8	87.2	92.8	94.9
+Metrix/embed	pos-neg/anc-neg	**66.4**	**77.6**	**85.4**	**83.9**	**90.3**	**94.1**	**76.7**	**88.6**	**95.2**	**88.4**	**95.4**	**96.9**
Proxy Anchor [80]	–	69.7	80.0	87.0	87.6	92.3	95.5	79.1	90.8	96.2	90.0	97.4	98.2
+PS [17]	pos-neg/neg-neg	70.0	79.8	87.2	87.9	92.8	95.6	79.6	90.9	96.4	90.3	97.4	98.0
+Metrix/embed	pos-neg/anc-neg	**70.4**	**81.1**	**87.9**	**88.9**	**93.3**	**96.4**	**80.6**	**91.7**	**96.6**	**91.6**	**98.3**	**98.3**

Table 2.4: *Comparison of Metrix/input and Metrix/embed with other mixing methods* using R-50 with embedding size $d = 512$ on four datasets. R@K (%): Recall@K; higher is better. PS: Proxy Synthesis.

Alternative mixing methods In Table 2.4, we compare Metrix/input with i-Mix [98] and Metrix/embed with MoCHi [97] using contrastive loss, and with PS [17] using PA. MoCHi and PS mix embeddings only, while labels are always negative. For i-Mix, we mix anchor-negative pairs ($S(a) \times U^-(a)$). For MoCHi, the anchor is clean and we mix negative-negative ($U^-(a)^2$) and anchor-negative ($S(a) \times U^-(a)$) pairs, where $U^-(a)$ is replaced by $k = 100$ hardest negatives and $\lambda \in (0, 0.5)$ for anchor-negative. PS mixes embeddings of different classes and treats them as new classes. For clean anchors, this corresponds to positive-negative ($U^+(a) \times U^-(a)$) and negative-negative ($U^-(a)^2$) pairs, but PS also supports mixed anchors.

In terms of Recall@1, Metrix/input outperforms i-Mix with anchor-negative pairs by 0.5% (65.8 → 66.3) on CUB200, 0.9% (82.0 → 82.9) on Cars196, 0.6% (75.2 → 75.8) and 0.6% (87.1 → 87.7) on In-Shop. Metrix/embed outperforms MoCHI with anchor-negative pairs by 1.2% (65.2 → 66.4) on CUB200, 1.4% (82.5 → 83.9) on Cars196, 0.9% (75.8 → 76.7) and 1.2% (87.2 → 88.4) on In-Shop. The gain over MoCHi with negative-negative pairs is significantly higher. Metrix/embed also outperforms PS by 0.4% (70.0 → 70.4) on CUB200, 1% (87.9 → 88.9) on Cars196, 1% (79.6 → 80.6) on SOP and 1.3% (90.3 → 91.6) on In-Shop.

Computational complexity On CUB200 dataset, using a batch size of 100 on an NVIDIA RTX 2080 Ti GPU, the average training time in ms/batch is 586 for MS and 817 for MS+Metrix. The 39% increase in complexity is reasonable for 3.6% increase in R@1. Furthermore, the average training time in ms/batch is 483 for baseline PA, 965 for PA+Metrix and 1563 for PS [17]. While the computation cost of PS is higher than Metrix by 62%, Metrix outperform PS by 0.4% and 1.3% in terms of R@1 and R@2 respectively (Table 2.4). At inference, the computational cost is equal for all methods.

Qualitative results of retrieval Figure 2.3 shows qualitative results of retrieval on CUB200 using contrastive loss, with and without Metrix. This dataset has large intra-class variations such as pose variation and background clutter. Baseline contrastive loss may fail to retrieve the correct images due to these challenges. The ranking is improved in the presence of Metrix.

2.4.4 Ablations

We perform ablations on Cars196 using R-50 with $d = 512$, applying mixup on contrastive loss.

Hard negatives We study the effect of the number k of hard negatives using different mixup types. The set of mixing pairs is chosen from (positive-negative,

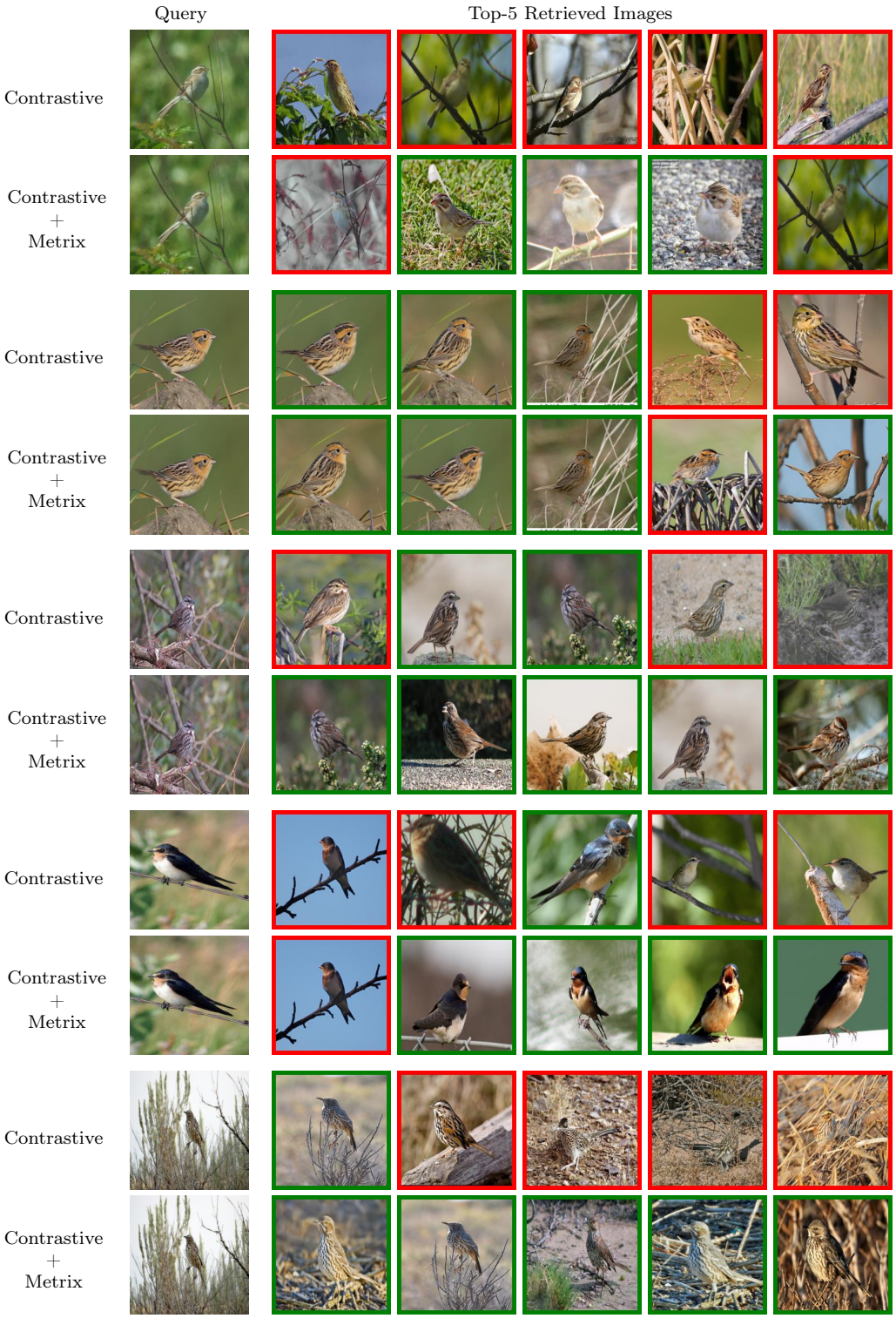


Figure 2.3: *Retrieval results* on CUB200 using contrastive loss, with and without Metrix. Retrieved images: **correct**, **incorrect**.

Study	Hard Negatives k	Mixing Pairs	Mixup Type	R@1	R@2	R@4	R@8
baseline				81.6	88.2	92.7	95.8
hard negatives	1	pos-neg/anc-neg	input	82.0	89.1	93.1	96.1
	2	pos-neg/anc-neg	input	82.5	89.2	93.4	96.2
	3	pos-neg/anc-neg	input	82.9	89.3	93.7	95.5
	20	pos-neg/anc-neg	feature	83.5	90.1	94.0	96.5
	40	pos-neg/anc-neg	feature	84.0	90.4	94.2	96.8
	all	pos-neg/anc-neg	feature	**85.1**	**91.1**	**94.6**	**97.0**
	20	pos-neg/anc-neg	embed	82.7	89.2	93.4	96.1
	40	pos-neg/anc-neg	embed	83.0	90.0	93.8	96.4
	all	pos-neg/anc-neg	embed	83.4	89.9	94.1	96.4
mixing pairs	–	pos-pos	input	81.0	88.2	92.6	95.6
	3	pos-neg	input	82.4	89.1	93.3	95.6
	3	anc-neg	input	81.8	89.0	93.6	95.4
	–	pos-pos	feature	81.1	88.3	92.9	95.8
	all	pos-neg	feature	**84.0**	**90.2**	94.2	96.6
	all	anc-neg	feature	83.7	90.1	**94.4**	**96.7**
	–	pos-pos	embed	78.3	85.7	90.8	94.4
	all	pos-neg	embed	83.1	90.0	93.9	96.6
	all	anc-neg	embed	82.7	89.5	93.5	96.3
mixup type combinations	{1, all}	pos-neg/anc-neg	{input, feature}	83.7	94.2	95.9	96.7
	{3, all}	pos-neg/anc-neg	{input, embed}	83.0	90.9	94.1	96.4
	{all, all}	pos-neg/anc-neg	{feature, embed}	84.7	90.6	94.4	96.9
	{1, all, all}	pos-neg/anc-neg	{input, feature, embed}	**85.3**	**94.9**	**96.2**	**97.1**

Table 2.5: *Ablation study of Metrix* using contrastive loss and R-50 with embedding size $d = 512$ on Cars196. R@K (%): Recall@K; higher is better.

anchor-negative) uniformly at random per iteration. We choose $k = 3$ for input mixup. For feature/embedding mixup, we mix all pairs in a batch by default, but also study $k \in \{20, 40\}$. As shown in Table 2.5, $k = 3$ for input and all pairs for feature/embedding mixup works best. Still, using few hard negatives for feature/embedding mixup is on par or outperforms input mixup. All choices significantly outperform the baseline.

Mixing pairs We study the effect of mixing pairs $M(a)$, in particular, $U^+(a)^2$ (positive-positive), $U^+(a) \times U^-(a)$ (positive-negative) and $S(a) \times U^-(a)$ (anchor-negative), again using different mixup types. As shown in Table 2.5, when using a single set of mixing pairs during training, positive-negative and anchor-negative consistently outperform the baseline, while positive-positive is actually outperformed by the baseline. This may be due to the lack of negatives in the mixed loss (2.9), despite the presence of negatives in the clean loss (2.3). Hence, we only use positive-negative and anchor-negative by default, combined by choosing uniformly at random in each iteration.

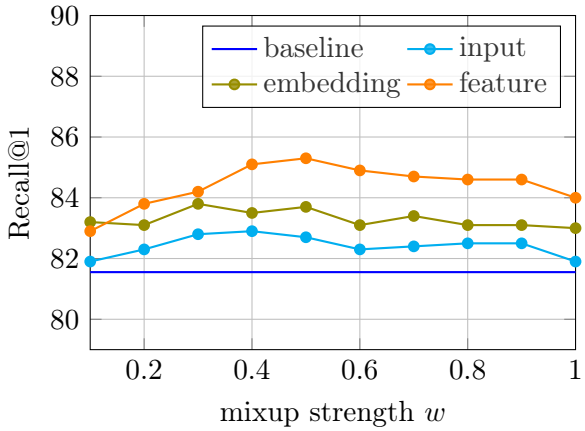


Figure 2.4: *Effect of mixup strength* for different mixup types using contrastive loss and R-50 with embedding size $d = 512$ on Cars196. Recall@K (%): higher is better.

Mixup types We study the effect of mixup type (input, feature, embedding), when used alone. The set of mixing pairs is chosen from (positive-negative, anchor-negative) uniformly at random per iteration. As shown in both "hard negatives" and "mixing pairs" parts of Table 2.5, our default feature mixup works best, followed by embedding and input mixup.

Mixup type combinations We study the effect of using more than one mixup type (input, feature, embedding), chosen uniformly at random per iteration. The set of mixing pairs is also chosen from (positive-negative, anchor-negative) uniformly at random per iteration. As shown in Table 2.5, mixing inputs, features and embeddings works best. Although this solution outperforms feature mixup alone by 0.2% Recall@1 (85.1 $\rightarrow$ 85.3), it is computationally expensive because of using input mixup. The next best efficient choice is mixing features and embeddings, which however is worse than mixing features alone (84.7 *vs*. 85.1). This is why we chose feature mixup by default.

Mixup strength w We study the effect of the mixup strength w in the combination of the clean and mixed loss (2.10) for different mixup types. As shown in Figure 2.4, mixup consistently improves the baseline and the effect of w is small, especially for input and embedding mixup. Feature mixup works best and is slightly more sensitive.

Ablations on CUB200 We perform additional ablations on CUB200 using R-50 with $d = 128$ by applying contrastive loss. All results are shown in Table 2.6. One may draw the same conclusions as from Table 2.5 on Cars196 with $d = 512$, which

Study	Hard Negatives k	Mixing Pairs	Mixup Type	R@1	R@2	R@4	R@8
baseline				64.7	75.9	84.6	87.6
hard negatives	1	pos-neg/anc-neg	input	62.4	73.9	83.0	89.7
	2	pos-neg/anc-neg	input	62.7	74.2	**83.6**	90.0
	3	pos-neg/anc-neg	input	63.1	74.5	83.5	90.3
	20	pos-neg/anc-neg	feature	63.9	75.0	83.9	89.9
	40	pos-neg/anc-neg	feature	63.5	75.2	83.5	89.8
	all	pos-neg/anc-neg	feature	**64.5**	**75.4**	84.3	**90.6**
	20	pos-neg/anc-neg	embed	63.1	74.3	83.1	90.0
	40	pos-neg/anc-neg	embed	63.5	74.7	83.6	90.1
	all	pos-neg/anc-neg	embed	64.0	75.1	84.8	90.9
mixing pairs	–	pos-pos	input	58.7	70.7	80.1	87.1
	3	pos-neg	input	62.9	75.1	83.4	90.6
	3	anc-neg	input	62.8	74.7	83.6	90.1
	–	pos-pos	feature	61.0	73.1	82.5	89.7
	all	pos-neg	feature	**63.9**	**75.0**	**83.9**	89.9
	all	anc-neg	feature	63.8	74.8	83.6	**90.2**
	–	pos-pos	embed	59.7	72.2	82.7	89.5
	all	pos-neg	embed	63.8	75.1	83.3	90.5
	all	anc-neg	embed	63.5	75.0	83.9	90.5
mixup type combinations	{1, all}	pos-neg/anc-neg	{input, feature}	63.9	75.1	**84.9**	90.5
	{3, all}	pos-neg/anc-neg	{input, embed}	63.4	74.9	84.5	90.1
	{all, all}	pos-neg/anc-neg	{feature, embed}	64.2	75.2	84.1	90.7
	{1, all, all}	pos-neg/anc-neg	{input, feature, embed}	**65.3**	**76.2**	84.4	**91.2**

Table 2.6: *Ablation study of Metrix* using contrastive loss and R-50 with embedding size $d = 128$ on CUB200. R@K (%): Recall@K; higher is better.

confirms that our choice of hard negatives and mixup pairs is generalizable across different datasets and embedding sizes.

In particular, following the settings of subsection 2.4.4, we observe in Table 2.6 that using $k = 3$ hard negatives for input mixup and all pairs for feature/embedding mixup achieves the best performance in terms of Recall@1. Similarly, using a single set of mixing pairs, positive-negative and anchor-negative consistently outperform the baseline, whereas positive-positive is inferior than the baseline. Furthermore, combining positive-negative and anchor-negative pairs by choosing uniformly at random in each iteration achieves the best overall performance.

We also study the effect of using more than one mixup type (input, feature,embedding), chosen uniformly at random per iteration. The set of mixing pairs is also chosen from (positive-negative, anchor-negative) uniformly at random per iteration in this study. From Table 2.6, we observe that although mixing input, features and embedding works best with an improvement of 0.8% over feature mixup alone (64.5 $\rightarrow$ 65.3), it is computationally expensive due to using input mixup. The

next best choice is mixing features and embeddings, which is worse than using feature mixup alone (64.2 *vs*. 64.5). This confirms our choice of using feature mixup as default.

2.4.5 How Mixup Improves Representations

We analyze how Metrix improves representation learning, given the difference between distributions at training and inference. As discussed in section 2.1, since the classes at inference are unseen at training, one might expect interpolation-based data augmentation like mixup to be even more important than in classification. This is so because, by mixing examples during training, we are exploring areas of the embedding space beyond the training classes. We hope that this exploration would possibly lead the model to implicitly learn a representation more appropriate for the test classes, if the distribution of the test classes lies near these areas.

Alignment and Uniformity In terms of quantitative measures of properties of the training and test distributions, we follow [70]. This work introduces two measures – *alignment* and *uniformity* (the lower the better) to be used both as loss functions (on the training set) and as evaluation metrics (on the test set). *Alignment* measures the expected pairwise distance between positive examples in the embedding space. A small value of alignment indicates that the positive examples are clustered together. *Uniformity* measures the (log of the) expected pairwise similarity between all examples regardless of class, using a Gaussian kernel as similarity. A small value of uniformity indicates that the distribution is more uniform over the embedding space, which is particularly relevant to our problem. Meant for contrastive learning, [70] use the same training and test classes, while in our case they are different.

By training with contrastive loss on CUB200 and then measuring on the test set, we achieve an alignment (lower the better) of 0.28 for contrastive loss, 0.28 for i-Mix [98] and 0.19 for Metrix/input. MoCHi [97] and Metrix/embed achieve an alignment of 0.19 and 0.17, respectively. We also obtain a uniformity (lower the better) of -2.71 for contrastive loss, -2.13 for i-Mix and -3.13 for Metrix/input. The uniformity of MoCHi and Metrix/embed is -3.18 and -3.25, respectively. This indicates that Metrix helps obtain a test distribution that is more uniform over the embedding space, where classes are better clustered and better separated.

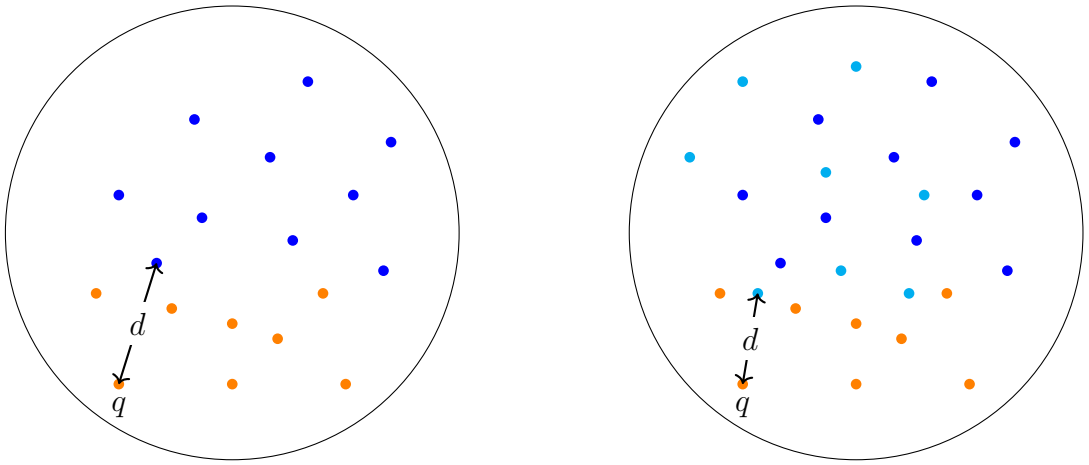
Utilization The measures proposed by [70] are limited to a single distribution or dataset, either the training set (as loss functions) or the test set (as evaluation metrics). It is more interesting to measure the extent to which a test example, seen as a query, lies near any of the training examples, clean or mixed. For this, we

introduce the measure of *utilization* $u(Q, X)$ of the training set X by the test set Q as:

$$u(Q, X) = \frac{1}{|Q|} \sum_{q \in Q} \min_{x \in X} \|f(q) - f(x)\|^2 \tag{2.24}$$

Utilization measures the average, over the test set Q, of the minimum distance of a query q to a training example $x \in X$ in the embedding space of the trained model f (lower is better). A low value of utilization indicates that there are examples in the training set that are similar to test examples. When using mixup, we measure utilization as $u(Q, \hat{X})$, where $\hat{X}$ is the augmented training set including clean and mixed examples over a number of epochs and f remains fixed. Because $X \subset \hat{X}$, we expect $u(Q, \hat{X}) < u(Q, X)$, that is, the embedding space is better explored in the presence of mixup (Figure 2.5).

By using contrastive loss on CUB200, utilization drops from 0.41 to 0.32 when using Metrix. This indicates that test samples are indeed closer to mixed examples than clean in the embedding space. This validates our hypothesis that a representation more appropriate for test classes is implicitly learned during exploration of the embedding space in the presence of mixup.



(a) Only clean examples (b) Clean and mixed examples

Figure 2.5: *Exploring the embedding space* when using (a) only clean examples (b) clean and mixed examples. Given a query q, the distance d to its nearest training embedding (clean or mixed) is smaller with mixup (b) than without (a). Examples: **clean train**; **(clean) test**; **mixed train**.

2.5 Conclusion

Based on the argument that metric learning is binary classification of pairs of examples into "positive" and "negative", we have introduced a direct extension of mixup from classification to metric learning. Our formulation is generic, applying to a large class of loss functions that separate positives from negatives per anchor and involve component functions that are additive over examples. Those are exactly loss functions that require less mining. We contribute a principled way of interpolating labels, such that the interpolation factor affects the relative weighting of positives and negatives. Other than that, our approach is completely agnostic with respect to the mixup method, opening the way to using more advanced mixup methods for metric learning.

We consistently outperform baselines using a number of loss functions on a number of benchmarks and we improve the state of the art using a single loss function on all benchmarks, while previous state of the art was not consistent in this respect. Surprisingly, this loss function, multi-similarity [69], is not the state of the art without mixup. Because metric learning is about generalizing to unseen classes and distributions, our work may have applications to other such problems, including transfer learning, few-shot learning and continual learning.

Acknowledgement This work was partially supported by the EU RAMONES project grant No. 101017808 and was performed using the HPC resources from GRNET S.A. project pr011028.

3

Learning Visual Representations via Model Architecture Component

Contents

3.1 Revisiting Pooling Mechanisms in Visual Representation Learning: From Convolutional Networks to Vision Transformers

Extracting visual representations and spatial pooling have been two interconnected processes since the study of 2D Gabor filters [18] and early convolutional networks [19]. Modern *convolutional networks* [22], [110] gradually perform local pooling and downsampling throughout the architecture to extract a low-resolution feature tensor, followed by a last step of global spatial pooling. *Vision transformers* [23] only downsample at input tokenization and then preserve resolution, but pooling takes place again throughout the architecture via the interaction of patch tokens with a CLS token, inherited from language models [111].

The pooling operation has been studied extensively in instance-level tasks on convolutional networks [112], [113], but less so in category-level tasks or transformers. Pooling in transformers is based on weighted averaging, using as weights the 2D *attention map* of the CLS token at the last layer. However, this attention map is typically of low quality, unless under self-supervision [29].

In this work, we argue that vision transformers can be reformulated in two streams, where one is extracting a visual representation on patch tokens and the other is performing spatial pooling on the CLS token; whereas, convolutional networks undergo global spatial pooling at the very last step, before the classifier. In this sense, one can isolate the pooling process from both kinds of networks and replace it by a new one. This raises the following questions:

1. *Can we derive a simple pooling process at the very last step of either convolutional or transformer encoders that improves over their default?*
2. *Can this process provide high-quality attention maps that delineate object boundaries, for both networks?*
3. *Do these properties hold under both supervised and self-supervised settings?*

To answer these questions, we develop a *generic pooling framework*, parametrized by: (a) the number of vectors in the pooled representation; (b) whether pooling is iterative or not; (c) mappings at every stage of the process; (d) pairwise similarities, attention function and normalization; and (e) a function determining the pooling

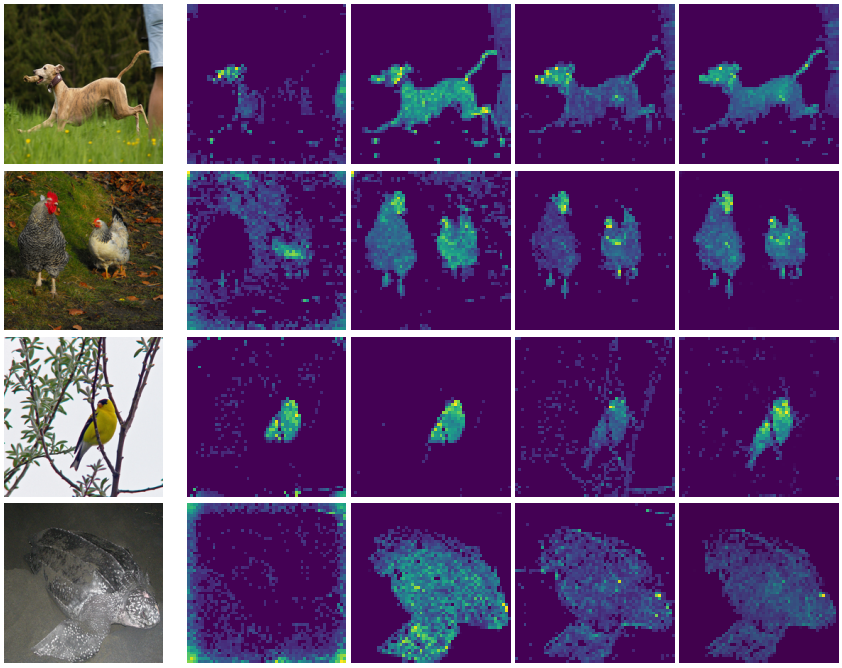


Figure 3.1: We introduce SimPool, a simple attention-based pooling method at the end of network, obtaining clean attention maps under supervision or self-supervision, improving pre-training and downstream task performance. Attention maps of ViT-S [23] trained on ImageNet-1k [114]. For baseline, we use the mean attention map of the CLS token. For SimPool, we use the attention map $\mathbf{a}$ (3.65). Input image: 896×896; patches: 16×16; attention map: 56×56.

operation.

We then formulate a number of existing pooling methods as instantiations of this framework, including (a) simple pooling mechanisms in convolutional networks [110], [113], [115]–[117], (b) iterative methods on more than one vectors like k-means [118], [119], (c) feature re-weighting mechanisms originally designed as network components rather than pooling [120], [121], and (d) vision transformers [23], [33]. Finally, by discussing the properties of each group of methods, we derive a new, simple, attention-based pooling mechanism as a replacement of the default one for both convolutional and transformer encoders. SimPool provides high-quality attention maps that delineate object boundaries, under both supervised and self-supervised settings, as shown for ViT-S [23] in Figure 3.1.

In summary, we make the following contributions:

1. We formulate a generic pooling framework that allows easy inspection and qualitative comparison of a wide range of methods.
2. We introduce a simple, attention-based, non-iterative, universal pooling mechanism that provides a single vector representation and answers all the above questions in the affirmative.
3. We conduct an extensive empirical study that validates the superior qualitative properties and quantitative performance of the proposed mechanism on standard benchmarks and downstream tasks.

3.2 Contextualizing Pooling Mechanisms in Convolutional Networks and Vision Transformers

Spatial pooling of visual input is the process by which spatial resolution is reduced to 1×1, such that the input is mapped to a single vector. This process can be gradual and interleaved with mapping to a feature space, because any feature space is amenable to smoothing or downsampling. The objective is robustness to deformation while preserving important visual information.

Via a similarity function, *e.g.* dot product, the vector representation of an image can be used for efficient matching to class representations for category-level tasks or to the representation of another image for instance-level tasks. One may obtain more than one vectors per image as a representation, but this requires a particular kernel for matching.

Background The study of receptive fields in neuroscience [122] lead to the development of 2D *Gabor filters* [18] as a model of the first processing layer in the visual cortex. Visual descriptors based on filter banks in the frequency domain [123] and orientation histograms [124], [125] can be seen as efficient implementations of the same idea. Apart from mapping to a new space—that of filter responses or orientation bins—they involve a form of smoothing, at least in some orientation, and weighted local spatial pooling.

Textons [126] can be seen as a second layer, originally studied in the context of texture discrimination [127] and segmentation [126], [128] and taking the form of multidimensional histograms on Gabor filter responses. The *bag of words* model [129], [130] is based on the same idea, as a histogram on other visual descriptors. Again, apart from mapping to a new space—that of textons or visual words—they involve local or global spatial pooling.

Histograms and every step of building visual features can be seen as a form of non-

linear coding followed by pooling [131]. *Coding* is maybe the most important factor. For example, a high-dimensional mapping before pooling, optionally followed by dimension reduction after pooling, can reduce interference between elements [132]–[134]. *Weighting* of individual elements is also important in attending important regions [135]–[137] and in preventing certain elements from dominating others [138]–[140].

The *pooling operation* itself is any symmetric (permutation-invariant) set function, which can be expressed in the form $F(X) = g\left(\sum_{x \in X} f(x)\right)$ [141]. The most common is average and maximum [131], [142], [143].

Common ways to obtain a representation of *multiple vectors* are using a spatial partition [135] or a partition in the feature space [144], [145].

Convolutional networks Following findings of neuroscience, early convolutional networks [19], [146] are based on learnable *convolutional layers* interleaved with fixed *spatial pooling layers* that downsample, which is an instance of the coding-pooling framework. The same design remains until today [22], [110], [147], [148]. Again, apart from mapping to a new space, convolutional layers involve a form of weighted local pooling. Again, the operation in pooling layers is commonly average [146] or maximum [142], [147].

Early networks end in a fully-connected layer over a feature tensor of low resolution [146]–[148]. This evolved into spatial pooling, *e.g. global average pooling* (GAP) for classification [110], [149], regional pooling for detection [150], or global maximum followed by a pairwise loss [115] for instance-level tasks. This is beneficial for downstream tasks and interpretability [151].

The spatial pooling operation at the end of the network is widely studied in instance level-tasks [112], [113], [115], giving rise to forms of *spatial attention* [117], [152]–[155], In category-level tasks, it is more common to study *feature re-weighting* as components of the architecture [120], [121], [156]. The two are closely related because *e.g.* the weighted average is element-wise weighting followed by sum. Most modern pooling operations are learnable.

Pooling can be *spatial* [117], [153]–[156], *over channels* [120], or both [121], [152]. CBAM [121] is particularly related to our work in the sense that it includes global average pooling followed by a form of spatial attention, although the latter is not evident in its original formulation and although CBAM is designed as a feature re-weighting rather than pooling mechanism.

One may obtain a representation of *multiple vectors e.g.* by some form of clustering [157] or optimal transport [118].

Vision transformers Pairwise interactions between features are forms of *self-attention* that can be seen as alternatives to convolution or forms of pooling. They have commonly been designed as architectural components of convolutional networks, again over the spatial [158]–[161] or the channel dimensions [162], [163]. Originating in language models [24], *vision transformers* [23] streamlined these approaches and became the dominant competitors of convolutional networks.

Transformers commonly downsample only at the input, forming spatial *patch tokens*. Pooling is based on a learnable CLS ("classification") token, which, beginning at the input space, undergoes the same self-attention operation with patch tokens and eventually provides a global image representation. That is, the network ends in global weighted average pooling, using as weights the attention of CLS over the patch tokens. Pooling is still gradual, since CLS interacts with patch tokens throughout the network depth.

Several variants of transformers often bring back ideas from convolutional networks, including spatial hierarchy [25], relative position encoding [164], [165], re-introducing convolution [166], [167], re-introducing pooling layers [25], [168]–[170], or simple pooling instead of attention [171]. In this sense, downsampling may occur inside the transformer, *e.g.* for classification [25], [168] or detection [169], [170].

Few works that have studied anything other than CLS for pooling in transformers are mostly limited to GAP [25]–[28]. CLS offers attention maps for free, but those are typically of low quality unless in a self-supervised setting [29], which is not well studied. Few works that attempt to rectify this in the supervised setting include a spatial entropy loss [30], shape distillation from convolutional networks [31] and skipping computation of self-attention, observing that the quality of self-attention is still good at intermediate layers [32]. It has also been found beneficial to inject the CLS token only at the last few layers [33].

We are thus motivated to question why the pooling operation at the end of the network needs to be different in convolutional networks and vision transformers and why pooling with a CLS token needs to be performed across the network depth. We study pooling in both kinds of networks, in supervised and self-supervised settings alike. We derive a simple, attention-based, universal pooling mechanism that applies equally to all cases, improving both performance and the quality of attention maps.

3.3 Formulating a Unified Pooling Framework and Deriving SimPool

We develop a generic pooling framework that encompasses many simple or more complex pooling methods, iterative or not, attention-based or not. We then examine a number of methods as instantiations of this framework. Finally, we discuss their properties and make particular choices in designing our solution.

3.3.1 A Generic Pooling Framework

Preliminaries Let $\mathbf{X} \in \mathbb{R}^{d \times W \times H}$ be the 3-dimensional *feature tensor* obtained from the last layer of a network for a given input image, where d is the number of feature channels and W, H are the width and height. We represent the image by the *feature* matrix $X \in \mathbb{R}^{d \times p}$ by flattening the spatial dimensions of $\mathbf{X}$, where $p := W \times H$ is the number of spatial locations. Let $\mathbf{x}_i \in \mathbb{R}^p$ denote the i-th row of X, that is, corresponding to the 2-dimensional feature map in channel i, and $\mathbf{x}_{\bullet j} \in \mathbb{R}^d$ denote the j-th column of X, that is, the feature vector of spatial location j.

By $\mathbf{1}_n \in \mathbb{R}^n$, we denote the all-ones vector. Given an $m \times n$ matrix $A \geq 0$, by $\eta_1(A) := \operatorname{diag}(A\mathbf{1}_n)^{-1}A$ we denote row-wise ℓ_1-normalization; similarly, $\eta_2(A) := A \operatorname{diag}(\mathbf{1}_m^\top A)^{-1}$ for column-wise.

Pooling process The objective of pooling is to represent the image by one or more vectors, obtained by interaction with X, either in a single step or by an iterative process. We denote the pooling process by function $\pi : \mathbb{R}^{d \times p} \to \mathbb{R}^{d' \times k}$ and the output vectors by matrix $U = \pi(X) \in \mathbb{R}^{d' \times k}$, where d' is the number of dimensions, possibly $d' = d$, and k is the number of vectors. In the most common case of a single vector, $k = 1$, we denote U by $\mathbf{u} \in \mathbb{R}^{d'}$. We discuss here the general iterative process; single-step pooling is the special case where the number of iterations is 1.

Initialization We define $X^0 := X$ and make a particular choice for $U^0 \in \mathbb{R}^{d^0 \times k}$, where $d^0 := d$. The latter may depend on the input X, in which case it is itself a simple form of pooling or not; for example, it may be random or a learnable parameter over the entire training set.

Pairwise interaction Given U^t and X^t at iteration t, we define the *query* and *key* matrices

$$Q = \phi_Q^t(U^t) \in \mathbb{R}^{n^t \times k} \tag{3.1}$$

$$K = \phi_K^t(X^t) \in \mathbb{R}^{n^t \times p}. \tag{3.2}$$

Here, functions $\phi_Q^t : \mathbb{R}^{d^t \times k} \to \mathbb{R}^{n^t \times k}$ and $\phi_K^t : \mathbb{R}^{d^t \times p} \to \mathbb{R}^{n^t \times p}$ may be the identity, linear or non-linear mappings to a space of the same ($n^t = d^t$) or different dimensions. We let K, Q interact pairwise by defining the $p \times k$ matrix $S(K,Q) := ((s(\mathbf{k}_{\bullet i}, \mathbf{q}_{\bullet j}))_{i=1}^p)_{j=1}^k$, where $s : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ for any n is a similarity function. For example, s can be dot product, cosine similarity, or a decreasing function of some distance. In the case of dot product, $s(\mathbf{x}, \mathbf{y}) := \mathbf{x}^\top \mathbf{y}$ for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, it follows that $S(K,Q) = K^\top Q \in \mathbb{R}^{p \times k}$.

Attention We then define the *attention* matrix

$$A = h(S(K,Q)) \in \mathbb{R}^{p \times k}. \tag{3.3}$$

Here, $h : \mathbb{R}^{p \times k} \to [0,1]^{p \times k}$ is a nonlinear function that may be elementwise, for instance relu or exp, normalization over rows or columns of $S(K,Q)$, or it may yield a form of correspondence or assignment between the columns of K and Q, possibly optimizing a cost function.

Attention-weighted pooling We define the *value* matrix

$$V = \phi_V^t(X^t) \in \mathbb{R}^{n^t \times p}. \tag{3.4}$$

Here, function $\phi_V^t : \mathbb{R}^{d^t \times p} \to \mathbb{R}^{n^t \times p}$ plays a similar role with ϕ_Q^t, ϕ_K^t. *Attention-weighted pooling* is defined by

$$Z = f^{-1}(f(V)A) \in \mathbb{R}^{n^t \times k}. \tag{3.5}$$

Here, $f : \mathbb{R} \to \mathbb{R}$ is a nonlinear elementwise function that determines the pooling operation, for instance, average or max-pooling. The product $f(V)A$ defines k linear combinations over the columns of $f(V)$, that is, the features at different spatial locations. If the columns of A are ℓ_1-normalized, then those are convex combinations. Thus, matrix A defines the weights of an averaging operation.

Output Finally, we define the output matrices corresponding to image features and pooling,

$$X^{t+1} = \phi_X^t(X^t) \in \mathbb{R}^{d^{t+1} \times p} \tag{3.6}$$

$$U^{t+1} = \phi_U^t(Z) \in \mathbb{R}^{d^{t+1} \times k}. \tag{3.7}$$

Functions $\phi_X^t : \mathbb{R}^{n^t \times p} \to \mathbb{R}^{d^{t+1} \times p}$ and $\phi_U^t : \mathbb{R}^{n^t \times k} \to \mathbb{R}^{d^{t+1} \times k}$ play a similar role with $\phi_Q^t, \phi_K^t, \phi_V^t$ but also determine the dimensionality d^{t+1} for the next iteration.

At this point, we may iterate by returning to the "pairwise interaction" step, or terminate, yielding U^{t+1} as U with $d' = d^{t+1}$. Non-iterative methods do not use ϕ_X^t.

Algorithm 1: Our generalized pooling framework.

input : p: #patches, d: dimension
input : $X \in \mathbb{R}^{d \times p}$: features
option : k: #pooled vectors
option : INIT: pooling initialization
option : T: #iterations
option : $\{\phi_Q^t\}, \{\phi_K^t\}$: query, key mappings
option : s: pairwise similarity function
option : h: attention function
option : $\{\phi_V^t\}$: value mapping
option : f: pooling function
option : $\{\phi_X^t\}, \{\phi_U^t\}$: output mappings
output: d': output dimension
output: $U \in \mathbb{R}^{d' \times k}$: pooled vectors

$d^0 \leftarrow d$ ▷ input dimension
$X^0 \leftarrow X \in \mathbb{R}^{d^0 \times k}$ ▷ initialize features
$U^0 \leftarrow \text{INIT}(X) \in \mathbb{R}^{d^0 \times k}$ ▷ initialize pooling
for $t = 0, \ldots, T-1$ **do**
  $Q \leftarrow \phi_Q^t(U^t) \in \mathbb{R}^{n^t \times k}$ ▷ query (3.1)
  $K \leftarrow \phi_K^t(X^t) \in \mathbb{R}^{n^t \times p}$ ▷ key (3.2)
  $S \leftarrow \mathbf{0}_{p \times k}$ ▷ pairwise similarity
  for $i \in [p]$, $j \in [k]$ **do**
    $s_{ij} \leftarrow s(\mathbf{k}_{\bullet i}, \mathbf{q}_{\bullet j})$
  $A \leftarrow h(S) \in \mathbb{R}^{p \times k}$ ▷ attention (3.3)
  $V \leftarrow \phi_V^t(X^t) \in \mathbb{R}^{n^t \times p}$ ▷ value (3.4)
  $Z \leftarrow f^{-1}(f(V)A) \in \mathbb{R}^{n^t \times k}$ ▷ pooling (3.5)
  $X^{t+1} \leftarrow \phi_X^t(X^t) \in \mathbb{R}^{d^{t+1} \times p}$ ▷ update feat. (3.6)
  $U^{t+1} \leftarrow \phi_U^t(Z) \in \mathbb{R}^{d^{t+1} \times k}$ ▷ update pool. (3.7)
$d' \leftarrow d^T$ ▷ output dimension
$U \leftarrow U^T$ ▷ pooled vectors

Algorithm Our generalized pooling framework is summarized in algorithm 1. As *input*, it takes the features $X \in \mathbb{R}^{d \times p}$, representing p patch embeddings of dimension d. As *output*, it returns the pooled vectors $U \in \mathbb{R}^{d' \times k}$, that is, k vectors of dimension d'. As *options*, it takes the number k of vectors to pool; the pooling initialization function INIT; the number T of iterations; the query and key mappings $\{\phi_Q^t\}, \{\phi_K^t\}$; the pairwise similarity function s; the attention function h; the value mapping $\{\phi_V^t\}$; the pooling function f; and the output mappings $\{\phi_X^t\}, \{\phi_U^t\}$.

The mappings and dimensions within iterations may be different at each iteration, and all optional functions may be learnable. As such, the algorithm is general enough to incorporate any deep neural network. However, the focus is on pooling, as is evident by the pairwise similarity between queries (pooled vectors) and keys (features) in line 9, which is a form of *cross-attention*.

Notation By id we denote the identity mapping. Given $n \in \mathbb{N}$, we define $[n] := \{1, \dots, n\}$. By $\mathbb{1}_A$ we denote the indicator function of set A, by δ_{ij} the Kronecker delta and by $[P]$ the Iverson bracket of statement P. By $A \circ B$ we denote the Hadamard product of matrices A, B and by $A^{\circ n}$ the Hadamard n-th power of A. We recall that by η_1, η_2 we denote the row-wise and column-wise ℓ_1-normalization of a matrix, respectively, while $\boldsymbol{\sigma}_2$ is column-wise softmax.

3.3.2 A Pooling Landscape

Table 3.1 examines, in *groups*, a number of pooling methods as instantiations of our framework. The objective is to get insight into their basic properties. How this table is obtained is detailed below.

Group 1: Simple methods with $k = 1$

We examine methods with $k = 1$ that are non-iterative, not attention-based, there are no query Q, key K, similarity matrix S or function h, and the attention is a vector $\mathbf{a} \in \mathbb{R}^p$ that is either fixed or a function directly of X. These methods have been studied in category-level tasks [110], [116] or mostly in instance-level tasks [113], [115], [117]. With the exception of HOW [117], the value matrix is $V = X$, that is, $\phi_V = \text{id}$, and we are pooling into vector $\mathbf{u} = \mathbf{z} \in \mathbb{R}^d$, that is, $\phi_U = \text{id}$. Then, (3.5) takes the form

$$\mathbf{u} = f^{-1}(f(X)\mathbf{a}) \in \mathbb{R}^d, \tag{3.8}$$

and we focus on instantiating it to identify function f and attention vector $\mathbf{a}$. With the exception of LSE [116], where $f(x) = e^{rx}$ with learnable scale r, function f is

#	Method	Cat	Iter	k	U^0	$\phi_Q(U)$	$\phi_K(X)$	$s(\mathbf{x},\mathbf{y})$	A	$\phi_V(X)$	$f(x)$	$\phi_X(X)$	$\phi_U(Z)$
1	GAP [110]	✓		1					$\mathbf{1}_p/p$	X	$f_{-1}(x)$		Z
	max [115]			1					$\mathbf{1}_p$	X	$f_{-\infty}(x)$		Z
	GeM [113]			1					$\mathbf{1}_p/p$	X	$f_\alpha(x)$		Z
	LSE [116]	✓		1					$\mathbf{1}_p/p$	X	e^{rx}		Z
	HOW [117]			1					$\mathrm{diag}(X^\top X)$	$\textsc{fc}(\mathrm{avg}_3(X))$	$f_{-1}(x)$		Z
2	OTK [118]	✓		k	U	U	X	$-\lVert\mathbf{x}-\mathbf{y}\rVert^2$	$\textsc{Sinkhorn}(e^{S/\epsilon})$	$\psi(X)$	$f_{-1}(x)$		Z
	k-means		✓	k	random	U	X	$-\lVert\mathbf{x}-\mathbf{y}\rVert^2$	$\eta_2(\arg\max_1(S))$	X	$f_{-1}(x)$	X	Z
	Slot [119]*	✓	✓	k	U	W_QU	W_KX	$\mathbf{x}^\top\mathbf{y}$	$\boldsymbol{\sigma}_2(S/\sqrt{d})$	W_VX	$f_{-1}(x)$	X	$\textsc{mlp}(\textsc{gru}(Z))$
3	SE [120]	✓		1	$\pi_A(X)$	$\sigma(\textsc{mlp}(U))$				$\mathrm{diag}(\mathbf{q})X$		V	
	CBAM [121]*	✓		1	$\pi_A(X)$	$\sigma(\textsc{mlp}(U))$	X	$\mathbf{x}^\top\mathbf{y}$	$\sigma(\mathrm{conv}_7(S))$	$\mathrm{diag}(\mathbf{q})X$		$V\,\mathrm{diag}(\mathbf{a})$	
4	ViT [23]*	✓	✓	1	U	$g_m(W_QU)$	$g_m(W_KX)$	$\mathbf{x}^\top\mathbf{y}$	$\boldsymbol{\sigma}_2(S_i/\sqrt{d})_{i=1}^m$	$g_m(W_VX)$	$f_{-1}(x)$	$\textsc{mlp}(\textsc{msa}(X))$	$\textsc{mlp}(g_m^{-1}(Z))$
	CaiT [33]*	✓	✓	1	U	$g_m(W_QU)$	$g_m(W_KX)$	$\mathbf{x}^\top\mathbf{y}$	$\boldsymbol{\sigma}_2(S_i/\sqrt{d})_{i=1}^m$	$g_m(W_VX)$	$f_{-1}(x)$	X	$\textsc{mlp}(g_m^{-1}(Z))$
5	SimPool	✓		1	$\pi_A(X)$	W_QU	W_KX	$\mathbf{x}^\top\mathbf{y}$	$\boldsymbol{\sigma}_2(S/\sqrt{d})$	$X-\min X$	$f_\alpha(x)$		Z

Table 3.1: *A landscape of pooling methods.* Cat: used in category-level tasks; Iter: iterative; k: #pooled vectors; U^0: initialization; $\phi_Q(U)$: query mapping; $\phi_K(X)$: key mapping; $s(\mathbf{x},\mathbf{y})$: similarity function; A: spatial attention; $\phi_V(X)$: value mapping; $f(x)$: pooling function; $\phi_X(X)$, $\phi_U(Z)$: output mappings; *: simplified. π_A: GAP; σ: sigmoid; $\boldsymbol{\sigma}_2$: softmax over columns; η_2: column normalization; g_m: partitioning in m groups. steel blue: ours; gray: common choices with ours; **green**: learnable; **red**: hyperparameter.

f_α (3.9) and we seek to identify α.

$$f_\alpha(x) := \begin{cases} x^{\frac{1-\alpha}{2}}, & \text{if } \alpha \neq 1, \\ \ln x, & \text{if } \alpha = 1. \end{cases} \tag{3.9}$$

As studied by Amari [172], function f_α is defined for $x \geq 0$ ($\alpha \neq 1$) or $x > 0$ ($\alpha = 1$). It reduces to the maximum, quadratic mean (RMS), arithmetic mean, geometric mean, harmonic mean, and minimum for $\alpha = -\infty, -3, -1, 1, 3, +\infty$, respectively. It has been proposed as a transition from average to max-pooling [143] and is known as GeM [113], with $\gamma = (1-\alpha)/2 > 1$ being a *learnable* parameter.

Global average pooling (GAP) [110], [149] That is:

$$\pi_A(X) := \frac{1}{p}\sum_{j=1}^{p} \mathbf{x}_{\bullet j} = X\mathbf{1}_p/p = f_{-1}^{-1}(f_{-1}(X)\mathbf{a}), \tag{3.10}$$

where $f_{-1}(x) = x^{\frac{1-(-1)}{2}} = x$, thus $f_{-1} = \text{id}$, and $\mathbf{a} = \mathbf{1}_p/p$.

Max pooling [115] Assuming $X \geq 0$,

$$\pi_{\max}(X) := \max_{j\in[p]} \mathbf{x}_{\bullet j} = \lim_{\gamma\to\infty} \left(\sum_{j=1}^{p} \mathbf{x}_{\bullet j}^{\gamma}\right)^{\frac{1}{\gamma}} \tag{3.11}$$

$$= \lim_{\gamma\to\infty} (X^\gamma \mathbf{1}_p)^{\frac{1}{\gamma}} = f_{-\infty}^{-1}(f_{-\infty}(X)\mathbf{a}), \tag{3.12}$$

where all operations are taken element-wise and $\mathbf{a} = \mathbf{1}_p$.

Generalized mean (GeM) [113] Assuming $X \geq 0$,

$$\pi_{\text{GeM}}(X) := \left(\frac{1}{p}\sum_{j=1}^{p} \mathbf{x}_{\bullet j}^{\gamma}\right)^{\frac{1}{\gamma}} \tag{3.13}$$

$$= (X^\gamma \mathbf{1}_p/p)^{\frac{1}{\gamma}} = f_\alpha^{-1}(f_\alpha(X)\mathbf{a}), \tag{3.14}$$

where all operations are taken element-wise, $\gamma = (1-\alpha)/2$ is a learnable parameter and $\mathbf{a} = \mathbf{1}_p/p$.

SimPool has the same pooling function but is based on an attention mechanism.

Log-sum-exp (LSE) [116]

$$\pi_{\text{LSE}}(X) := \frac{1}{r} \log \left(\frac{1}{p} \sum_{j=1}^{p} \exp(r\mathbf{x}_{\bullet j}) \right) \tag{3.15}$$

$$= f^{-1}(f(X)\mathbf{a}), \tag{3.16}$$

where all operations are taken element-wise, r is a learnable scale parameter, $f(x) = e^{rx}$ and $\mathbf{a} = \mathbf{1}_p/p$.

HOW [117] The attention value of each feature $\mathbf{x}_{\bullet j}$ is its norm $\|\mathbf{x}_{\bullet j}\|$. That is,

$$\mathbf{a} = (\|\mathbf{x}_{\bullet 1}\|, \ldots, \|\mathbf{x}_{\bullet p}\|)^\top = (X^{\circ 2})^\top \mathbf{1}_d \tag{3.17}$$

$$= \text{diag}(X^\top X) \in \mathbb{R}^p, \tag{3.18}$$

obtained by pooling over channels. The value matrix is

$$V = \phi_V(X) = \text{FC}(\text{avg}_3(X)) \in \mathbb{R}^{d' \times p}, \tag{3.19}$$

where avg_3 is 3×3 local average pooling, FC is a fixed fully-connected (1×1 convolutional) layer incorporating centering, PCA dimension reduction and whitening according to the statistics of the local features of the training set and $d' < d$ is the output dimension. Then,

$$\mathbf{z} = \sum_{j=1}^{p} a_j \mathbf{v}_{\bullet j} = V\mathbf{a} = f_{-1}^{-1}(f_{-1}(V)\mathbf{a}) \in \mathbb{R}^{d'}, \tag{3.20}$$

where $f_{-1} = \text{id}$ as in GAP. Finally, the output is $\mathbf{u} = \eta^2(\mathbf{z})$, where the mapping $\phi_U = \eta^2$ is ℓ_2-normalization.

Group 2: Iterative methods with $k > 1$

We examine methods, which, given $X \in \mathbb{R}^{d \times p}$ and $k < p$, seek $U \in \mathbb{R}^{d \times k}$ by iteratively optimizing a kind of assignment between columns of X and U. The latter are called references [118], centroids [173], or slots [119]. Assignment can be soft [118], [119] or hard [173]. It can be an assignment of columns of X to columns of U [119], [173] or both ways [118]. The algorithm may contain learnable components [118], [119] or not [173].

Optimal transport kernel embedding (OTK) [118] Pooling is based on a learnable parameter $U \in \mathbb{R}^{d \times k}$. We define the $p \times k$ *cost* matrix $C = (c_{ij})$ consisting of the pairwise squared Euclidean distances between columns of X and U, *i.e.*,

$c_{ij} = \|\mathbf{x}_{\bullet i} - \mathbf{u}_{\bullet j}\|^2$. We seek a $p \times k$ non-negative *transportation plan* matrix $P \in \mathcal{P}$ representing a joint probability distribution over features of X and U with uniform marginals:

$$\mathcal{P} := \{P \in \mathbb{R}_+^{p \times k} : P\mathbf{1}_k = \mathbf{1}_p/p, P^\top \mathbf{1}_p = \mathbf{1}_k/k\}. \tag{3.21}$$

The objective is to minimize the expected, under P, pairwise cost with entropic regularization

$$P^* := \arg\min_{P \in \mathcal{P}} \langle P, C\rangle - \epsilon H(P), \tag{3.22}$$

where $H(P) = -\mathbf{1}_p^\top (P \circ \log P)\mathbf{1}_k$ is the entropy of P, $\langle \cdot, \cdot \rangle$ is the Frobenius inner product and $\epsilon > 0$ controls the sparsity of P. The optimal solution is $P^* = \textsc{Sinkhorn}(e^{-C/\epsilon})$, where exponentiation is element-wise and SINKHORN is the Sinkhorn-Knopp algorithm [174], which iteratively ℓ_1-normalizes rows and columns of a matrix until convergence [175]. Finally, pooling is defined as

$$U = \psi(X)P^* \in \mathbb{R}^{d' \times k}, \tag{3.23}$$

where $\psi(X) \in \mathbb{R}^{d' \times p}$ and $\psi : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a Nyström approximation of a kernel embedding in $\mathbb{R}^d$, *e.g.* a Gaussian kernel [118], which applies column-wise to $X \in \mathbb{R}^{d \times p}$.

We conclude that OTK [118] is a instance of our pooling framework with learnable $U_0 = U \in \mathbb{R}^{d \times k}$, query/key mappings $\phi_Q = \phi_K = \text{id}$, pairwise similarity function $s(\mathbf{x}, \mathbf{y}) = -\|\mathbf{x} - \mathbf{y}\|^2$, attention matrix $A = h(S) = \textsc{Sinkhorn}(e^{S/\epsilon}) \in \mathbb{R}^{p \times k}$, value mapping $\phi_V = \psi$, average pooling function $f = f_{-1}$ and output mapping $\phi_U = \text{id}$.

Although OTK is not formally iterative in our framework, SINKHORN internally iterates indeed to find a soft-assignment between the features of X and U.

k-means [173] k-means aims to find a $d \times k$ matrix U minimizing the sum of squared Euclidean distances of each column $\mathbf{x}_{\bullet i}$ of X to its nearest column $\mathbf{u}_{\bullet j}$ of U:

$$J(U) := \sum_{i=1}^{p} \min_{j \in [k]} \|\mathbf{x}_{\bullet i} - \mathbf{u}_{\bullet j}\|^2. \tag{3.24}$$

Observe that (3.60) is the special case $k = 1$, where the unique minimum $\mathbf{u}^* = \pi_A(X)$ is found in closed form (3.61). For $k > 1$, the distortion measure J is non-convex and we are only looking for a local minimum.

The standard k-means algorithm is initialized by a $d \times k$ matrix U^0 whose columns are k of the columns of X sampled at random and represent a set of k *centroids* in $\mathbb{R}^d$. Given U^t at iteration t, we define the $p \times k$ *distance* matrix $D = (d_{ij})$ consisting of the pairwise squared Euclidean distances between columns of X and U^t, *i.e.*, $d_{ij} = \left\|\mathbf{x}_{\bullet i} - \mathbf{u}^t_{\bullet j}\right\|^2$. For $i \in [p]$, feature $\mathbf{x}_{\bullet i}$ is *assigned* to the nearest centroid $\mathbf{u}^t_{\bullet j}$ with index

$$c_i = \arg\min_{j \in [k]} d_{ij}, \tag{3.25}$$

where ties are resolved to the lowest index. Then, at iteration $t+1$, centroid $\mathbf{u}^t_{\bullet j}$ is *updated* as the mean of features $\mathbf{x}_{\bullet i}$ assigned to it, *i.e.*, for which $c_i = j$:

$$\mathbf{u}^{t+1}_{\bullet j} = \frac{1}{\sum_{i=1}^p \delta_{c_i j}} \sum_{i=1}^p \delta_{c_i j} \mathbf{x}_{\bullet i}. \tag{3.26}$$

Let $\arg\min_1(D)$ be the $p \times k$ matrix $M = (m_{ij})$ with

$$m_{ij} = \delta_{c_i j} = \left[j = \arg\min_{j' \in [k]} d_{ij'} \right]. \tag{3.27}$$

That is, each row $\mathbf{d}_i \in \mathbb{R}^k$ of D yields a row $\mathbf{m}_i \in \{0,1\}^k$ of M that is an one-hot vector indicating the minimal element over $\mathbf{d}_i$. Define operator $\arg\max_1$ accordingly. Then, (3.26) can be written in matrix form as

$$U^{t+1} = X \eta_2(\arg\max_1(-D)) \in \mathbb{R}^{d \times k}. \tag{3.28}$$

We conclude that k-means is an iterative instance of our pooling framework with the columns of $U^0 \in \mathbb{R}^{d \times k}$ sampled at random from the columns of X, query/key mappings $\phi_Q = \phi_K = \mathrm{id}$, pairwise similarity function $s(\mathbf{x}, \mathbf{y}) = -\left\|\mathbf{x} - \mathbf{y}\right\|^2$, attention matrix $A = h(S) = \eta_2(\arg\max_1(S)) \in \mathbb{R}^{p \times k}$, value mapping $\phi_V = \mathrm{id}$, average pooling function $f = f_{-1}$ and output mappings $\phi_X = \phi_U = \mathrm{id}$.

Slot attention [119] Pooling is initialized by a random $d' \times k$ matrix U^0 sampled from a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with shared, learnable mean $\mu \in \mathbb{R}^{d'}$ and standard deviation $\sigma \in \mathbb{R}^{d'}$. Given U^t at iteration t, define the query $Q = W_Q \mathrm{LN}(U^t) \in \mathbb{R}^{n \times k}$ and key $K = W_K \mathrm{LN}(X) \in \mathbb{R}^{n \times p}$, where LN is LayerNorm [176] and n is a common dimension. An attention matrix is defined as

$$A = \eta_1(\boldsymbol{\sigma}_2(K^\top Q / \sqrt{n})) \in \mathbb{R}^{p \times k}. \tag{3.29}$$

Then, with value $V = W_V \text{LN}(X) \in \mathbb{R}^{n \times p}$, pooling is defined as the weighted average

$$Z = VA \in \mathbb{R}^{n \times k}. \tag{3.30}$$

Finally, U^t is updated according to

$$G = \text{GRU}(Z) \in \mathbb{R}^{d' \times k} \tag{3.31}$$

$$U^{t+1} = G + \text{MLP}(\text{LN}(G)) \in \mathbb{R}^{d' \times k}, \tag{3.32}$$

where GRU is a *gated recurrent unit* [177] and MLP a multi-layer perceptron with ReLU activation and a residual connection [119].

We now simplify the above formulation by removing LayerNorm and residual connections.

We conclude that slot attention [119] is an iterative instance of our pooling framework with U^0 a random $d' \times k$ matrix sampled from $\mathcal{N}(\mu, \sigma^2)$ with learnable parameters $\mu, \sigma \in \mathbb{R}^{d'}$, query mapping $\phi_Q(U) = W_Q U \in \mathbb{R}^{n \times k}$, key mapping $\phi_K(X) = W_K X \in \mathbb{R}^{n \times p}$, pairwise similarity function $s(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$, attention matrix $A = h(S) = \eta_1(\boldsymbol{\sigma}_2(S/\sqrt{n})) \in \mathbb{R}^{p \times k}$, value mapping $\phi_V(X) = W_V X \in \mathbb{R}^{n \times p}$, average pooling function $f = f_{-1}$, output mapping $\phi_U(Z) = \text{MLP}(\text{GRU}(Z)) \in \mathbb{R}^{d' \times k}$ and output dimension d'.

SimPool is similar in its attention mechanism, but is non-iterative with $k = 1$ and initialized by GAP.

Group 3: Feature re-weighting, $k = 1$

We examine two methods, originally proposed as components of the architecture, which use attention mechanisms to re-weight features in the channel or the spatial dimension. We modify them by placing at the end of the network, followed by GAP. We thus reveal that they serve as attention-based pooling. This includes pairwise interaction, although this was not evident in their original formulation.

Squeeze-and-excitation block (SE) [120] The *squeeze* operation aims to mitigate the limited receptive field of convolutional networks, especially in the lower layers. It uses global average pooling over the spatial dimension,

$$\mathbf{u}^0 = \pi_A(X) \in \mathbb{R}^d. \tag{3.33}$$

Then, the *excitation* operation aims at capturing channel-wise dependencies and involves two steps. In the first step, a learnable gating mechanism forms a vector

$$\mathbf{q} = \sigma(\text{MLP}(\mathbf{u}^0)) \in \mathbb{R}^d, \tag{3.34}$$

where σ is the sigmoid function and MLP concists of two linear layers with ReLU activation in-between and forming a bottlenect of hidden dimension d/r. This vector expresses an importance of each channel that is not mutually exclusive. The second step re-scales each channel (row) of X by the corresponding element of $\mathbf{q}$,

$$V = \text{diag}(\mathbf{q})X \in \mathbb{R}^{d \times p}. \tag{3.35}$$

The output $X' = V \in \mathbb{R}^{d \times p}$ is a new tensor of the same shape as X, which can be used in the next layer. In this sense, the entire process is considered a block to be used within the architecture of convolutional networks at several layers. This yields a new family of networks, called *squeeze-and-excitation networks* (SENet).

However, we can also see it as a pooling process if we perform it at the end of a network, followed by GAP:

$$\mathbf{z} = \pi_A(V) = \text{diag}(\mathbf{q})X\mathbf{1}_p/p \in \mathbb{R}^d, \tag{3.36}$$

We conclude that this modified SE block is a non-iterative instance of our pooling framework with $\mathbf{u}^0 = \pi_A(X) \in \mathbb{R}^d$, query mapping $\phi_Q(\mathbf{u}) = \sigma(\text{MLP}(\mathbf{u})) \in \mathbb{R}^d$, no key K, similarity matrix S of function h, uniform spatial attention $\mathbf{a} = \mathbf{1}_p/p$, value mapping $\phi_V(X) = \text{diag}(\mathbf{q})X \in \mathbb{R}^{d \times p}$ and average pooling function $f = f_{-1}$.

The original design does not use $\mathbf{a}$ or $\mathbf{z}$; instead, it has an output mapping $\phi_X(X) = V = \text{diag}(\mathbf{q})X \in \mathbb{R}^{d \times p}$. Thus, it can be used iteratively along with other mappings of X to form a modified network architecture.

Convolutional block attention module (CBAM) [121] This is an extension of SE [120] that acts on both the channel and spatial dimension in similar ways. *Channel attention* is similar to SE: It involves (a) global average and maximum pooling of X over the spatial dimension,

$$U^0 = (\pi_A(X) \;\; \pi_{\max}(X)) \in \mathbb{R}^{d \times 2}; \tag{3.37}$$

(b) a learnable gating mechanism forming vector

$$\mathbf{q} = \sigma(\text{MLP}(U^0)\mathbf{1}_2/2) \in \mathbb{R}^d, \tag{3.38}$$

which is defined as in SE [120] but includes averaging over the two columns before σ; and (c) re-scaling channels (rows) of X by $\mathbf{q}$,

$$V = \text{diag}(\mathbf{q})X \in \mathbb{R}^{d \times p}. \tag{3.39}$$

Spatial attention performs a similar operation in the spatial dimension: (a) global average and maximum pooling of V over the channel dimension,

$$S = (\pi_A(V^\top) \;\; \pi_{\max}(V^\top)) \in \mathbb{R}^{p\times 2}; \tag{3.40}$$

(b) a learnable gating mechanism forming vector

$$\mathbf{a} = \sigma(\text{conv}_7(S)) \in \mathbb{R}^p, \tag{3.41}$$

where conv_7 is a a convolutional layer with kernel size 7×7; and (c) re-scaling features (columns) of V by $\mathbf{a}$,

$$X' = V\,\text{diag}(\mathbf{a}) \in \mathbb{R}^{d\times p}. \tag{3.42}$$

The output X' is a new tensor of the same shape as X, which can be used in the next layer. In this sense, CBAM is a block to be used within the architecture, like SE [120]. However, we can also see it as a *pooling process* if we perform it at the end of a network, followed by GAP:

$$\mathbf{z} = \pi_A(X') = V\,\text{diag}(\mathbf{a})\mathbf{1}_p/p = V\mathbf{a}/p \in \mathbb{R}^d. \tag{3.43}$$

We also *simplify* CBAM by removing max-pooling from both attention mechanisms and keeping average pooling only. Then, (3.40) takes the form

$$\mathbf{s} = \pi_A(V^\top) = V^\top\mathbf{1}_d/d = (\text{diag}(\mathbf{q})X)^\top\mathbf{1}_d/d \tag{3.44}$$

$$= X^\top\mathbf{q}/d \in \mathbb{R}^p. \tag{3.45}$$

This reveals *pairwise interaction* by dot-product similarity between $\mathbf{q}$ as query and X as key. It was not evident in the original formulation, because dot product was split into element-wise product followed by sum.

We conclude that this modified CBAM module is a non-iterative instance of our pooling framework with $\mathbf{u}^0 = \pi_A(X) \in \mathbb{R}^d$, query mapping $\phi_Q(\mathbf{u}) = \sigma(\text{MLP}(\mathbf{u}))/d \in \mathbb{R}^d$, key mapping $\phi_K = \text{id}$, pairwise similarity function $s(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top\mathbf{y}$, spatial attention $\mathbf{a} = h(\mathbf{s}) = \sigma(\text{conv}_7(\mathbf{s}))/p \in \mathbb{R}^p$, value mapping $\phi_V(X) = \text{diag}(\mathbf{q})X \in \mathbb{R}^{d\times p}$, average pooling function $f = f_{-1}$ and output mapping $\phi_U = \text{id}$.

The original design does not use $\mathbf{z}$; instead, it has an output mapping $\phi_X(X) = V\,\text{diag}(\mathbf{a}) = \text{diag}(\mathbf{q})X\,\text{diag}(\mathbf{a}) \in \mathbb{R}^{d\times p}$. Thus, it can be used iteratively along with other mappings of X to form a modified network architecture.

SimPool is similar in that $\mathbf{u}^0 = \pi_A(X)$ but otherwise its attention mechanism is different: there is no channel attention while in spatial attention there are learnable query/key mappings and competition between spatial locations.

Group 4: Transformers

We re-formulate the standard ViT [23] in two streams, where one performs pooling and the other feature mapping. We thus show that the pooling stream is an iterative instance of our framework, where iterations are blocks. We then examine the variant CaiT [33], which is closer to SimPool in that pooling takes place in the upper few layers with the features being fixed.

Vision transformer (ViT) [23] The transformer encoder *tokenizes* the input image, *i.e.*, it splits the image into p non-overlapping *patches* and maps them to patch token embeddings of dimension d through a linear mapping. It then concatenates a learnable CLS token embedding, also of dimension d, and adds a learnable *position embedding* of dimension d to all tokens. It is thus initialized as

$$F^0 = (\mathbf{u}^0 \;\; X^0) \in \mathbb{R}^{d\times(p+1)}, \tag{3.46}$$

where $\mathbf{u}^0 \in \mathbb{R}^d$ is the initial CLS token embedding and $X^0 \in \mathbb{R}^{d\times p}$ contains the initial patch embeddings.

The encoder contains a sequence of *blocks*. Given token embeddings $F^t = (\mathbf{u}^t \;\; X^t) \in \mathbb{R}^{d\times(p+1)}$ as input, a block performs the following operations:

$$G^t = F^t + \text{MSA}(\text{LN}(F^t)) \in \mathbb{R}^{d\times(p+1)} \tag{3.47}$$

$$F^{t+1} = G^t + \text{MLP}(\text{LN}(G^t)) \in \mathbb{R}^{d\times(p+1)}, \tag{3.48}$$

where LN is LayerNorm [176] and MLP is a network of two affine layers with a ReLU activation in-between, applied to all tokens independently. Finally, at the end of block $T-1$, the image is pooled into vector $\mathbf{u} = \text{LN}(\mathbf{u}^T)$.

Given $F^t \in \mathbb{R}^{d\times(p+1)}$, the *multi-head self-attention* (MSA) operation uses three linear mappings to form the query $Q = W_Q F^t$, key $K = W_K F^t$ and value $V = W_V F^t$, all in $\mathbb{R}^{d\times(p+1)}$. It then splits each of the three into m submatrices, each of size $d/m \times (p+1)$, where m is the number of *heads*.

Given a stacked matrix $A = (A_1; \ldots; A_m) \in \mathbb{R}^{d\times n}$, where $A_i \in \mathbb{R}^{d/m\times n}$ for $i \in [m]$, we denote splitting as

$$\mathcal{A} = g_m(A) = \{A_1, \ldots, A_m\} \subset \mathbb{R}^{d/m\times n}. \tag{3.49}$$

Thus, with $\mathcal{Q} = g_m(Q) = \{Q_i\}$, $\mathcal{K} = g_m(K) = \{K_i\}$, $\mathcal{V} = g_m(V) = \{V_i\}$, self-attention is defined as

$$A_i = \boldsymbol{\sigma}_2\left(K_i^\top Q_i/\sqrt{d'}\right) \in \mathbb{R}^{(p+1)\times(p+1)} \tag{3.50}$$

$$Z_i = V_i A_i \in \mathbb{R}^{d'\times(p+1)}, \tag{3.51}$$

for $i \in [m]$, where $d' = d/m$. Finally, given $\mathcal{Z} = \{Z_i\}$, submatrices are grouped back and an output linear mapping yields the output of MSA:

$$U = W_U g_m^{-1}(\mathcal{Z}) \in \mathbb{R}^{d\times(p+1)}. \tag{3.52}$$

Here, we decompose the above formulation into two parallel streams. The first operates on the CLS token embedding $\mathbf{u}^t \in \mathbb{R}^d$, initialized by learnable parameter $\mathbf{u}^0 \in \mathbb{R}^d$ and iteratively performing pooling. The second operates on the patch embeddings $X^t \in \mathbb{R}^{d\times p}$, initialized by $X^0 \in \mathbb{R}^{d\times p}$ as obtained by tokenization and iteratively performing feature extraction. We focus on the first one.

Given $\mathbf{u}^t \in \mathbb{R}^d$, $X^t \in \mathbb{R}^{d\times p}$ at iteration t, we form the query $\mathcal{Q} = g_m(W_Q\text{LN}(\mathbf{u}^t))$, key $\mathcal{K} = g_m(W_K\text{LN}(X^t))$ and value $\mathcal{V} = g_m(W_V\text{LN}(X^t))$. *Cross-attention* between $\mathcal{Q}$ and $\mathcal{K}, \mathcal{V}$ follows for $i \in [m]$:

$$\mathbf{a}_i = \boldsymbol{\sigma}_2\left(K_i^\top \mathbf{q}_i/\sqrt{d'}\right) \in \mathbb{R}^p \tag{3.53}$$

$$\mathbf{z}_i = V_i\mathbf{a}_i \in \mathbb{R}^{d'}. \tag{3.54}$$

Finally, denoting $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$, the CLS token embedding at iteration $t+1$ is given by

$$\mathbf{g}^t = \mathbf{u}^t + W_U g_m^{-1}(\mathcal{Z}) \in \mathbb{R}^d \tag{3.55}$$

$$\mathbf{u}^{t+1} = \mathbf{g}^t + \text{MLP}(\text{LN}(\mathbf{g}^t)) \in \mathbb{R}^d. \tag{3.56}$$

We now simplify the above formulation by removing LayerNorm and residual connections. We also remove the dependence of self-attention of patch embeddings on the CLS token.

We conclude that ViT [23] is an iterative instance of our pooling framework with learnable $\mathbf{u}^0 \in \mathbb{R}^d$, query mapping $\phi_Q(\mathbf{u}) = g_m(W_Q\mathbf{u}) \subset \mathbb{R}^{d'}$ with $d' = d/m$, key mapping $\phi_K(X) = g_m(W_K X) \subset \mathbb{R}^{d'\times p}$, pairwise similarity function $s(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top\mathbf{y}$, spatial attention $\mathcal{A} = h(\mathcal{S}) = \{\boldsymbol{\sigma}_2(\mathbf{s}_i/\sqrt{d'})\}_{i=1}^m \subset \mathbb{R}^p$, value mapping $\phi_V(X) = g_m(W_V X) \subset \mathbb{R}^{d'\times p}$, average pooling function $f = f_{-1}$ and output mappings $\phi_X(X) = \text{MLP}(\text{MSA}(X)) \in \mathbb{R}^{d\times p}$ and $\phi_U(\mathcal{Z}) = \text{MLP}(W_U g_m^{-1}(\mathcal{Z})) \in \mathbb{R}^d$.

Although $k = 1$, splitting into m submatrices and operating on them independently is the same as defining m query vectors in $\mathbb{R}^d$ via the block-diagonal matrix

$$Q = \begin{pmatrix} \mathbf{q}_1 & \ldots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{q}_m \end{pmatrix} \in \mathbb{R}^{d\times m}. \tag{3.57}$$

Q interacts with K by dot product, essentially operating in m orthogonal subspaces. This gives rise to an attention matrix $A \in \mathbb{R}^{p \times m}$ containing $\mathbf{a}_i$ (3.53) as columns and a pooled matrix $Z \in \mathbb{R}^{d \times m}$ containing $\mathbf{z}_i$ (3.54) as columns.

Thus, the m heads in multi-head attention bear similarities to the k pooled vectors in our formulation. The fact that transformer blocks act as iterations strengthens our observation that methods with $k > 1$ are iterative. However, because of linear maps at every stage, there is no correspondence between heads across iterations.

Class-attention in image transformers (CaiT) [33] This work proposes two modifications in the architecture of ViT [23]. The first is that the encoder consists of two stages. In stage one, patch embeddings are processed alone, without a CLS token. In stage two, a learnable CLS token is introduced that interacts with patch embeddings with cross-attention, while the patch embeddings remain fixed. The second modification is that it introduces two learnable diagonal matrices $\Lambda_G^t, \Lambda_X^t \in \mathbb{R}^{d \times d}$ at each iteration (block) t and uses them to re-weight features along the channel dimension.

Thus, stage one is specified by a modification of (3.47), (3.48) as follows:

$$G^t = X^t + \Lambda_G^t \mathrm{MSA}(\mathrm{LN}(X^t)) \in \mathbb{R}^{d \times p} \tag{3.58}$$

$$X^{t+1} = G^t + \Lambda_X^t \mathrm{MLP}(\mathrm{LN}(G^t)) \in \mathbb{R}^{d \times p}. \tag{3.59}$$

This is similar to [120], [121], only here the parameters are learnable rather than obtained by GAP. Similarly, stage two is specified by a modification of (3.53)-(3.56). Typically, stage two consists only of a few (1-3) iterations.

> We conclude that a simplified version of stage two of CaiT [33] is an iterative instance of our pooling framework with the same options as ViT [23] except for the output mapping $\phi_X = \mathrm{id}$.

SimPool is similar in that there are again two stages, but stage one is the entire encoder, while stage two is a single non-iterative cross-attention operation between features and their GAP, using function f_α for pooling.

Slot attention [119] is also similar to stage two of CaiT, performing few iterations of cross-attention between features and slots with $\phi_X = \mathrm{id}$, but with a single head, $k > 1$ and different mapping functions.

3.3.3 SimPool

Group 5 of Table 3.1 is our method, SimPool. A schematic overview is given in Figure 3.2.

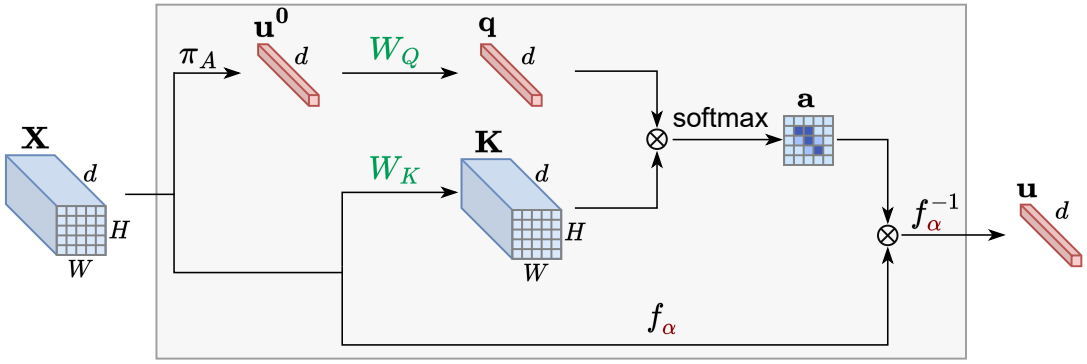


Figure 3.2: *Overview of SimPool.* Given an input tensor $\mathbf{X} \in \mathbb{R}^{d \times W \times H}$ flattened into $X \in \mathbb{R}^{d \times p}$ with $p := W \times H$ patches, one stream forms the initial representation $\mathbf{u}^0 = \pi_A(X) \in \mathbb{R}^d$ (3.62) by *global average pooling* (GAP), mapped by $W_Q \in \mathbb{R}^{d \times d}$ (3.63) to form the *query* vector $\mathbf{q} \in \mathbb{R}^d$. Another stream maps X by $W_K \in \mathbb{R}^{d \times d}$ (3.64) to form the *key* $K \in \mathbb{R}^{d \times p}$, shown as tensor $\mathbf{K}$. Then, $\mathbf{q}$ and K interact to generate the attention map $\mathbf{a} \in \mathbb{R}^p$ (3.65). Finally, the pooled representation $\mathbf{u} \in \mathbb{R}^d$ is a generalized weighted average of X with $\mathbf{a}$ determining the weights and scalar function f_α determining the pooling operation (3.67).

Pooling process We are striving for a simple design. While pooling into $k > 1$ vectors would yield a more discriminative representation, either these would have to be concatenated, as is the case of multi-head attention, or a particular similarity kernel would be needed beyond dot product, which we consider to be beyond the scope of this work. We rather argue that it is the task of the encoder to learn a single vector representation of objects, even if those are composed of different parts. This argument is stronger when pre-training is performed on images mostly depicting one object, like ImageNet-1k.

We observe in Table 3.1 that only methods explicitly pooling into $k > 1$ vectors or implicitly using $m > 1$ heads are iterative. We explain why in the next paragraph. Following this insight, we perform pooling in a single step.

In summary, our solution is limited to a single vector $\mathbf{u} \in \mathbb{R}^d$ for pooling, that is, $k = 1$, and is non-iterative.

Initialization We observe in Table 3.1 that single-step attention-based methods in Group 3 initialize $\mathbf{u}^0$ by GAP. We hypothesize that, since attention is based on pairwise similarities, it is essential that $\mathbf{u}^0$ is chosen such that its similarities with X are maximized on average, which would help to better discriminate between foreground (high similarity) and background (low similarity). Indeed, for $s(\mathbf{x}, \mathbf{y}) = -\|\mathbf{x} - \mathbf{y}\|^2$, the sum of squared Euclidean distances of each column $\mathbf{x}_{\cdot i}$ of X to

$\mathbf{u} \in \mathbb{R}^d$

$$J(\mathbf{u}) = \frac{1}{2} \sum_{i=1}^{p} \|\mathbf{x}_{\bullet i} - \mathbf{u}\|^2 \tag{3.60}$$

is a convex distortion measure with unique minimum the average of vectors $\{\mathbf{x}_{\bullet i}\}$

$$\mathbf{u}^* := \arg\min_{\mathbf{u} \in \mathbb{R}^d} J(\mathbf{u}) = \frac{1}{p} \sum_{i=1}^{p} \mathbf{x}_{\bullet i} = \pi_A(X), \tag{3.61}$$

which can be found in closed form. By contrast, for $k > 1$ vectors, distortion can only be minimized iteratively, *e.g.* by k-means. We therefore choose:

$$\mathbf{u}^0 = \pi_A(X) = X\mathbf{1}_p/p. \tag{3.62}$$

Pairwise interaction, attention We follow the attention mechanism of transformers, in its simplest possible form. In particular, we use a single head, $m = 1$, like Slot Attention [119] (which however uses k vectors). We find that the query and key mappings are essential in learning where to attend as a separate task from learning the representation for the given task at hand. In particular, we use linear mappings ϕ_Q, ϕ_K with learnable parameters $W_Q, W_K \in \mathbb{R}^{d \times d}$ respectively:

$$\mathbf{q} = \phi_Q(\mathbf{u}^0) = W_Q \mathbf{u}^0 \in \mathbb{R}^d \tag{3.63}$$

$$K = \phi_K(X) = W_K X \in \mathbb{R}^{d \times p}. \tag{3.64}$$

As in transformers, we define pairwise similarities as dot product, that is, $S(K, \mathbf{q}) = K^\top \mathbf{q} \in \mathbb{R}^{p \times k}$, and attention as scaled softmax over columns (spatial locations), that is, $h(S) := \boldsymbol{\sigma}_2(S/\sqrt{d})$:

$$\mathbf{a} = \boldsymbol{\sigma}_2\left(K^\top \mathbf{q}/\sqrt{d}\right) \in \mathbb{R}^p, \tag{3.65}$$

where $\boldsymbol{\sigma}_2(S) := \eta_2(\exp(S))$ and exp is taken elementwise.

Attention-weighted pooling As shown in Table 3.1, the average pooling operation ($f = f_{-1}$) is by far the most common. However, the more general function f_α (3.9) has shown improved performance in instance-level tasks [113]. For $\alpha < -1$ ($\gamma > 1$) in particular, it yields an intermediate operation between average and max-pooling. The latter is clearly beneficial when feature maps are sparse, because it better preserves the non-zero elements.

We adopt $f = f_\alpha$ for its genericity: the only operation that is not included as a special case in Table 3.1 is log-sum-exp [116]. This choice assumes $X \geq 0$. This

is common in networks ending in relu, like ResNet [110], which is also what makes feature maps sparse. However, vision transformers and modern convolutional networks like ConvNeXt [22] do not end in relu; hence X has negative elements and is not necessarily sparse. We therefore define

$$V = \phi_V(X) = X - \min X \in \mathbb{R}^{d \times p}, \tag{3.66}$$

where the minimum is taken over all elements of X, such that f_α operates only on non-negative numbers.

We also define $\mathbf{u} = \phi_U(\mathbf{z}) = \mathbf{z}$ and the output dimension is $d' = d$. Thus, the mappings ϕ_V, ϕ_U are parameter-free. The argument is that, for average pooling for example ($f = f_{-1}$ in (3.5)), any linear layers before or after pooling would commute with pooling, thus they would form part of the encoder rather than the pooling process. Moreover, Table 3.1 shows that ϕ_U is non-identity only for iterative methods.

In summary, we define SimPool (SP) as

$$\mathbf{u} = \pi_{\text{SP}}(X) := f_\alpha^{-1}(f_\alpha(V)\mathbf{a}) \in \mathbb{R}^d, \tag{3.67}$$

where $V \in \mathbb{R}^{d \times p}$ is the value (3.66) and $\mathbf{a} \in \mathbb{R}^p$ is the attention map (3.65). Parameter α is learned in GeM [113], but we find that treating it as a hyperparameter better controls the quality of the attention maps.

Algorithm 2: SimPool. Green: learnable.

input : d: dimension, p: patches
input : features $X \in \mathbb{R}^{d \times p}$
output: pooled vector $\mathbf{u} \in \mathbb{R}^d$

$\mathbf{u}^0 \leftarrow X\mathbf{1}_p/p \in \mathbb{R}^d$ ▷ initialization (3.62)
$X \leftarrow \text{LN}(X) \in \mathbb{R}^{d \times p}$ ▷ LayerNorm [176]
$\mathbf{q} \leftarrow W_Q\mathbf{u}^0 \in \mathbb{R}^d$ ▷ query (3.63)
$K \leftarrow W_K X \in \mathbb{R}^{d \times p}$ ▷ key (3.64)
$\mathbf{a} \leftarrow \boldsymbol{\sigma}_2(K^\top \mathbf{q}/\sqrt{d}) \in \mathbb{R}^p$ ▷ attention (3.65)
$V \leftarrow X - \min X \in \mathbb{R}^{d \times p}$ ▷ value (3.66)
$\mathbf{u} \leftarrow f_\alpha^{-1}(f_\alpha(V)\mathbf{a}) \in \mathbb{R}^d$ ▷ pooling (3.9),(3.67)

Algorithm SimPool is summarized in algorithm 2. The addition to what presented above is LayerNorm after obtaining $\mathbf{u}^0$ and before K, V. That is, (3.64) and (3.66) are modified as

$$K = \phi_K(X) = W_K \text{LN}(X) \in \mathbb{R}^{d \times p}. \tag{3.68}$$

$$V = \phi_V(X) = \text{LN}(X) - \min \text{LN}(X) \in \mathbb{R}^{d \times p}. \tag{3.69}$$

As shown in Table 3.12, it is our choice in terms of simplicity, performance, and attention map quality to apply LayerNorm to key and value and linear layers to query and key. The learnable parameters are W_Q and W_K.

In summary, SimPool is a non-iterative instance of our pooling framework with $k = 1$, $\mathbf{u}^0 = \pi_A(X) \in \mathbb{R}^d$, query mapping $\phi_Q(\mathbf{u}) = W_Q\mathbf{u} \in \mathbb{R}^d$, key mapping $\phi_K(X) = W_K\text{LN}(X) \in \mathbb{R}^{d\times p}$, pairwise similarity function $s(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top\mathbf{y}$, spatial attention $\mathbf{a} = h(\mathbf{s}) = \boldsymbol{\sigma}_2(\mathbf{s}/\sqrt{d}) \in \mathbb{R}^p$, value mapping $\phi_V(X) = \text{LN}(X) - \min \text{LN}(X) \in \mathbb{R}^{d\times p}$, average pooling function $f = f_\alpha$ and output mapping $\phi_U = \text{id}$.

3.4 Assessing SimPool: Performance, Properties, and Insights

3.4.1 Datasets, Networks and Evaluation Protocols

Supervised pre-training We train ResNet-18, ResNet-50 [110], ConvNeXt-S [22], ViT-S and ViT-B [23] for *image classification* on ImageNet-1k. For the analysis subsection 3.4.2 and ablation subsection 3.4.4, we train ResNet-18 on the first 20% of training examples per class of ImageNet-1k [114] (called ImageNet-20%) for 100 epochs. For the benchmark of subsection 3.4.3, we train ResNet-50 for 100 and 200 epochs, ConvNeXt-S and ViT-S for 100 and 300 epochs and ViT-B for 100 epochs, all on the 100% of ImageNet-1k. We evaluate on the full validation set in all cases and measure top-1 classification accuracy. The baseline is the default per network, *i.e.* GAP for convolutional networks and CLS token for transformers.

Self-supervised pre-training On the 100% of ImageNet-1k, we train DINO [29] with ResNet-50, ConvNeXt-S and ViT-S for 100 epochs. We evaluate on the validation set by k-NN and *linear probing* on the training set. For *linear probing*, we train a linear classifier on top of features as in DINO [29]. For k-NN [178], we freeze the model and extract features, then use a k-nearest neighbor classifier with $k = 10$.

Downstream tasks We fine-tune supervised and self-supervised ViT-S on CIFAR-10 [179], CIFAR-100 [179] and Oxford Flowers [180] for *image classification*, measuring top-1 classification accuracy. CIFAR-100 is just like CIFAR-10, except it has 100 classes containing 600 images each. Oxford Flowers consists of 102 flower categories containing between 40 and 258 images each. We perform *object localization* without fine-tuning using supervised and self-supervised ViT-S on CUB [103] and ImageNet-1k, measuring MaxBoxAccV2 [181]. We perform *unsupervised object*

discovery without fine-tuning using self-supervised ViT-S with DINO-SEG [29] and LOST [182] on VOC07 [183] trainval, VOC12 [183] trainval and COCO 20K [184], measuring CorLoc [185]. The latter is a subset of COCO2014 trainval dataset [184], comprising 19,817 randomly selected images. VOC07 comprises 9,963 images depicting 24,640 annotated objects. VOC12 comprises 11,530 images depicting 27,450 annotated objects. We perform *semantic segmentation* with fine-tuning using self-supervised ViT-S on ADE20K [186], measuring mIoU, mAcc, and aAcc. We fine-tune a linear layer. The training set of ADE20K consists of 20k images and the validation set of 2k images in 150 classes. We validate *robustness* against background changes using ViT-S on ImageNet-9 [187] (IN-9) and its variations. We use the linear head and linear probe for supervised and self-supervised ViT-S, respectively, measuring top-1 classification accuracy. IN-9 contains nine coarse-grained classes with seven variations of both background and foreground. We perform *image retrieval*, extracting features from a self-supervised ResNet-50 and ViT-S and evaluating them on $\mathcal{R}$Oxford and $\mathcal{R}$Paris [188], measuring mAP. $\mathcal{R}$Oxford and $\mathcal{R}$Paris are the revisited Oxford [189] and Paris [190] datasets, comprising 5,062 and 6,412 images collected from Flickr [191] by searching for Oxford and Paris landmarks respectively. We perform *fine-grained classification*, extracting features from a supervised and self-supervised ResNet-50 and ViT-S and evaluating them on Caltech-UCSD Birds (CUB200) [103], Stanford Cars (CARS196) [104], In-Shop Clothing Retrieval (In-Shop) [192] and Stanford Online Products (SOP) [193], measuring Revall@K. Dataset statistics are summarized in Table 2.2.

Ablations For the ablations of subsection 3.4.4, we train supervised ResNet-18 and ViT-T for *image classification* on ImageNet-20% and ImageNet-1k respectively.

3.4.2 Experimental Analysis

Figure 3.3 evaluates different methods in groups following Table 3.1, regardless of their original design for (a) pooling or not, (b) different tasks, *e.g.* instance-level or category-level, (c) different networks, *e.g.* convolutional or transformers.

Group 1 consists of simple pooling methods with: (a) no parameters: GAP [149], max [115], GAP+max [194]; and (b) scalar parameter: GeM [113] and LSE [116]. HOW [117] is the only method to use (parameter-free) attention. GeM is performing the best, with LSE following second. These methods are inferior to those in other groups.

Group 2 incorporates methods with $k > 1$ vectors. We set $k = 3$ and take the maximum of the 3 logits per class. OTK and Slot use attention. Slot attention [119] works best, outperforming k-means by 1.3%.

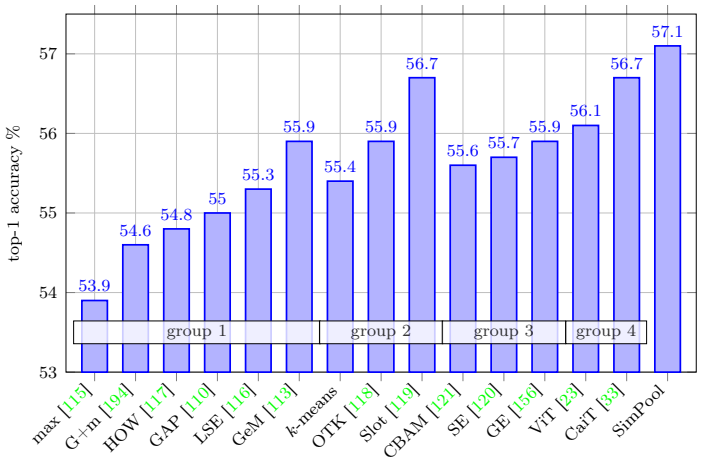


Figure 3.3: *Image classification* on ImageNet-20. Supervised training of ResNet-18 for 100 epochs.

Group 3 refers to parametric attention-based methods, weighting features based on their importance for the task: CBAM [121], Squeeze-Excitation [120] and Gather-Excite [156]. While originally designed as components within the architecture, we adapt them to pooling by GAP at the end. Gather-Excite [156] performs best.

Group 4 refers to parametric attention-based methods found in vision transformers. ViT [23] refers to multi-head self-attention learnable CLS and four heads, which we incorporate as a single layer at the end of the model. CaiT [33] is the same but using only cross-attention between CLS and patch embeddings. CaiT performs the best.

SimPool outperforms all other methods. Seeing this experiment as a tournament, we select the best performing method of each group and qualify it for the benchmark of subsection 3.4.3.

3.4.3 Benchmark

Image Classification Table 3.2 compares SimPool with baseline and tournament winners per group of subsection 3.4.2 on supervised pre-training for classification. For 100 epochs, SimPool outperforms all methods, consistently improving the baseline by 0.6% using convolutional networks, 1.6% using ViT-S and 1.0% using ViT-B. Gather-Excite [156] improves over the baseline only on convolutional networks, while Slot [119] only on ViT-S. CaiT improves over the baseline only for ConvNeXt-S. By contrast, SimPool improves everywhere. For more than 100 epochs, SimPool improves the baseline by 0.5% using ResNet-50, 0.4% using ConvNeXt-S and 0.8%

Method	Epochs	ResNet-50	ConvNext-S	ViT-S	ViT-B
Baseline	100	77.4	81.1	72.7	74.1
CaiT [33]	100	77.3	81.2	72.6	-
Slot [119]	100	77.3	80.9	72.9	-
GE [156]	100	77.6	81.3	72.6	-
SimPool	100	**78.0**	**81.7**	**74.3**	**75.1**
Baseline	300	78.1†	83.1	77.9	-
SimPool	300	**78.7**†	**83.5**	**78.7**	-

Table 3.2: *Image classification* top-1 accuracy (%) on ImageNet-1k. Supervised pre-training for 100 and 300 epochs. Best competitors selected per group from Figure 3.3. Baseline: GAP for convolutional, CLS for transformers; †: 200 epochs.

Method	Epochs	ResNet-50		ConvNeXt-S		ViT-S	
		k-NN	Prob	k-NN	Prob	k-NN	Prob
Baseline	100	61.8	63.0	65.1	68.2	68.9	71.5
SimPool	100	**63.8**	**64.4**	**68.8**	**72.2**	**69.8**	**72.8**

Table 3.3: *Image classification* top-1 accuracy (%) on ImageNet-1k. Self-supervised pre-training with DINO [29] for 100 epochs. Prob: linear probing; Baseline: GAP for convolutional, CLS for transformers.

using ViT-S.

Table 3.3 evaluates self-supervised pre-training for 100 epochs. SimPool improves over the baseline by 2.0% k-NN and 1.4% linear probing on ResNet-50; 3.7% k-NN and 4.0% linear probing on ConvNeXt-S; and 0.9% k-NN and 1.3% linear probing on ViT-S. We also train ViT-S with DINO on ImageNet-1k for 300 epochs. SimPool improves over the baseline by 0.4% k-NN (72.2 $\rightarrow$ 72.6) and 0.7% linear probing (74.3 $\rightarrow$ 75.0).

Fine-tuning for classification Table 3.4 evaluates fine-tuning for classification on different datasets [179], [180] of a supervised and a self-supervised ViT-S. SimPool brings small improvement over the baseline in all cases.

Method	Supervised			Self-Supervised		
	CIFAR-10	CIFAR-100	Flowers	CIFAR-10	CIFAR-100	Flowers
Baseline	98.1	86.0	97.1	98.7	89.8	98.3
SimPool	**98.4**	**86.2**	**97.4**	**98.9**	**89.9**	**98.4**

Table 3.4: *Image classification* accuracy (%), fine-tuning for classification for 1000 epochs. ViT-S pre-trained on ImageNet-1k for 100 epochs. Self-supervision with DINO [29].

Method	Supervised		Self-Supervised	
	CUB	ImageNet	CUB	ImageNet
Baseline	63.1	53.6	82.7	62.0
SimPool	**77.9**	**64.4**	**86.1**	**66.1**
Baseline@20	62.4	50.5	65.5	52.5
SimPool@20	**74.0**	**62.6**	**72.5**	**58.7**

Table 3.5: *Localization accuracy* MaxBoxAccV2 on CUB test and ImageNet-1k validation set. ViT-S pre-trained on ImageNet-1k for 100 epochs. Self-supervision with DINO [29]. @20: at epoch 20.

Method	DINO-seg [29], [182]			LOST [182]		
	VOC07	VOC12	COCO	VOC07	VOC12	COCO
Baseline	30.8	31.0	36.7	55.5	59.4	46.6
SimPool	**53.2**	**56.2**	**43.4**	**59.8**	**65.0**	**49.4**
Baseline@20	14.9	14.8	19.9	50.7	56.6	40.9
SimPool@20	**49.2**	**54.8**	**37.9**	**53.9**	**58.8**	**46.1**

Table 3.6: *Object discovery* CorLoc. ViT-S pre-trained on ImageNet-1k for 100 epochs. Self-supervision with DINO [29]. @20: at epoch 20.

Object localization Accurate localization can have a significant impact on classification accuracy, particularly under multiple objects, complex scenes and background clutter. Table 3.5 evaluates localization accuracy under both supervision settings. SimPool significantly improves the baseline by up to 7% MaxBoxAccV2 when self-supervised and up to 14% when supervised. In the latter case, the gain is already up to 12% at epoch 20.

Unsupervised object discovery Table 3.6 studies LOST [182], which uses the raw features of a vision transformer pre-trained using DINO [29] for unsupervised single-object discovery, as well as the baseline DINO-seg [29], [182], which uses the attention maps instead. SimPool significantly outperforms the baseline on all datasets by up to 25.2% CorLoc for DINO-seg and 5.6% for LOST on VOC12. Again, the gain is significant already at the first 20 epochs.

Semantic segmentation We evaluate semantic segmentation on ADE20K [186] under self-supervised pre-training. To evaluate the quality of the learned representation, we only fine-tune a linear layer on top of the fixed patch features, as in iBOT [195]. SimPool improves the baseline by 1,5% mIoU (26.4 $\rightarrow$ 27.9), 1.7% mAcc (34.0 $\rightarrow$ 35.7) and 1.0% aAcc (71.6 $\rightarrow$ 72.6). These results testify the improved quality of the learned representations when pre-training with SimPool.

Method	OF	MS	MR	MN	NF	OBB	OBT	IN-9
	Supervised							
Baseline	66.4	79.1	67.4	65.5	37.2	12.9	15.2	92.0
SimPool	**71.8**	**80.2**	**69.3**	**67.3**	**42.8**	**15.2**	**15.6**	**92.9**
	Self-supervised + Linear probing							
Baseline	**87.3**	87.9	78.5	76.7	47.9	**20.0**	**16.9**	95.3
SimPool	**87.3**	**88.1**	**80.6**	**78.7**	**48.2**	17.8	16.7	**95.6**

Table 3.7: *Background robustness* on IN-9 [187] and its variations; more details in the appendix. ViT-S pre-trained on ImageNet-1k for 100 epochs. Self-supervision with DINO [29].

Network	Method	$\mathcal{R}$Oxford		$\mathcal{R}$Paris	
		Medium	Hard	Medium	Hard
ResNet-50	Baseline	27.2	7.9	47.3	19.0
	SimPool	**29.7**	**8.7**	**51.6**	**23.0**
ViT-S	Baseline	29.4	10.0	54.6	26.2
	SimPool	**32.1**	**10.6**	**56.5**	**27.3**

Table 3.8: *Image retrieval* mAP (%) without fine-tuning on $\mathcal{R}$Oxford and $\mathcal{R}$Paris [188]. Self-supervised pre-training with DINO [29] on ImageNet-1k for 100 epochs.

Background changes Deep neural networks often rely on the image background, which can limit their ability to generalize well. To achieve better performance, these models must be able to cope with changes in the background and prioritize the foreground. To evaluate SimPool robustness to the background changes, we use the ImageNet-1k-9 [187] (IN-9) dataset. In four of these datasets, *i.e.*, Only-FG (OF), Mixed-Same (MS), Mixed-Rand (MR), and Mixed-Next (MN), the background is modified. The rest, *i.e.*, No-FG (NF), Only-BG-B (OBB), and Only-BG-T (OBT), feature masked foregrounds. Table 3.7 shows that SimPool improves over the baseline under both supervision settings with only 2 out of 8 exceptions under DINO [29] pre-training. The latter is justified, given that none of the foreground objects or masks are present in these settings.

Image retrieval While classification accuracy indicates ability of a model to recognize objects of the same classes as those it was trained for, it does not necessarily reflect its ability to capture the visual similarity between images, when tested on a dataset from a different distribution. Here, we evaluate this property of visual features using ResNet-50 and ViT-S; for particular object retrieval without fine-tuning on $\mathcal{R}$Oxford and $\mathcal{R}$Paris [188]. In Table 3.8, we observe that SimPool is very

network	Method	CUB200			CARS196			SOP			In-Shop		
		R@1	R@2	R@4	R@1	R@2	R@4	R@1	R@10	R@100	R@1	R@10	R@20
		Supervised											
ResNet-50	Baseline	42.7	55.2	67.7	42.3	54.2	65.7	48.3	63.2	71.8	**27.6**	**49.9**	**56.5**
	SimPool	**43.0**	**55.2**	**67.9**	**43.8**	**56.2**	**67.4**	**48.7**	**64.1**	**72.9**	27.0	49.9	56.5
ViT-S	Baseline	55.8	68.3	78.3	38.2	50.3	61.8	54.1	69.2	81.6	30.9	56.5	63.2
	SimPool	**56.8**	**69.6**	**79.2**	**38.9**	**50.7**	**63.3**	**54.2**	**69.4**	**81.9**	**32.8**	**57.6**	**64.3**
		Self-supervised											
ResNet-50	Baseline	26.0	36.2	46.9	**34.1**	**44.2**	**55.0**	51.2	65.3	76.5	37.1	58.4	64.1
	SimPool	**30.7**	**40.9**	**53.3**	33.6	43.6	54.3	**52.1**	**66.5**	**77.2**	**38.1**	**60.0**	**65.6**
ViT-S	Baseline	56.7	69.4	80.5	37.5	47.5	58.4	**59.8**	**74.4**	**85.4**	40.4	63.9	70.3
	SimPool	**61.8**	**74.4**	**83.6**	**37.6**	**48.0**	**58.4**	59.5	73.9	85.0	**41.1**	**64.3**	**70.8**

Table 3.9: *Fine-grained classification* Recall@K (R@K, %) without fine-tuning on four datasets, following the same protocol as [196], [197]. Models pre-trained on ImageNet-1k for 100 epochs. Self-supervision with DINO [29].

effective, improving the retrieval performance of both models on all datasets and evaluation protocols over the baseline.

Fine-grained classification We evaluate fine-grained classification using ResNet-50 and ViT-S, both supervised and self-supervised, following [197]. We extract features from test set images and directly apply nearest neighbor search, measuring Recall@K. Table 3.9 shows that SimPool is superior to the baseline in most of the datasets, models and supervision settings, with the exception of ResNet-50 supervised on In-Shop, ResNet-50 self-supervised on Cars196 and ViT-S self-supervised on SOP (3 out of 16 cases). The improvement is roughly 1-2% Recall@1 in most cases, and is most pronounced on self-supervised on CUB200, roughly 5%.

Method	ResNet-18		ResNet-50		ConvNeXt-S		ViT-S	
	#par	flo	#par	flo	#par	flo	#par	flo
Baseline	**11.7**	**1.82**	**25.6**	**4.13**	**50.2**	**8.68**	**22.1**	**4.24**
CaiT	18.0	1.85	75.9	4.60	57.3	8.75	23.8	4.29
Slot	14.6	1.87	71.7	4.89	56.7	8.79	23.7	4.30
GE	11.7	1.83	26.1	4.15	50.3	8.69	22.1	4.25
SimPool	12.2	1.84	33.9	4.34	51.4	8.71	22.3	4.26

Table 3.10: *Computation resources* on Imagenet-1k, with $d = 512$ (ResNet-18), 2048 (ResNet-50), 768 (ConvNeXt), 384 (ViT-S). #par: # parameters, in millions; flo: GFLOPS.

Network	Pooling	Depth	Init	Accuracy	#params
Base	GAP	12	12	73.3	22.1M
Base	cls	12	0	72.7	22.1M
Base + 1		13	0	73.2	23.8M
Base + 2		14	0	73.7	25.6M
Base + 3		15	0	73.8	27.4M
Base + 4		16	0	73.9	29.2M
Base + 5		17	0	**74.6**	30.9M
Base	SimPool	12	12	**74.3**	22.3M
Base − 1		11	11	73.9	20.6M
Base − 2		10	10	73.6	18.7M
Base − 3		9	9	72.5	17.0M

Table 3.11: *Trade-off between performance and parameters.* Supervised pre-training of ViT-S on ImageNet-1k for 100 epochs. Init: Initial layer of pooling token. Base: original network. Base+b (Base−b): b blocks added to (removed from) the network.

Computation resources Table 3.10 shows the number of parameters and floating point operations per second for the best competitors of Figure 3.3. Resources depend on the embedding dimension d. SimPool is higher than the baseline but not the highest.

Performance *vs*. parameters Table 3.11 aims to answer the question of how much the performance improvement of SimPool is due to parameters of the query and key mappings. Interestingly, ViT-S works better with GAP than the default cls. SimPool adds 0.2M parameters to the network. For fair comparison, we remove blocks from the network (Base) when using SimPool and add blocks when using cls. We find that, to exceed the accuracy of Base SimPool, Base cls needs 5 extra blocks, *i.e.*, 9M more parameters. Equally interestingly, removing 3 blocks from Base SimPool is only slightly worse than Base cls, having 5M fewer parameters.

3.4.4 Ablations

We ablate the design and components of SimPool. More ablations are found in the appendix. In particular, for function f_α (3.9), we set $\gamma = 2$ for convolutional networks and $\gamma = 1.25$ for transformers by default, where $\gamma = (1 - \alpha)/2$ is a hyperparameter.

Design In Table 3.12 (left), we ablate (a) the attention function h (3.3); (b) the number of iterations with shared parameters at every iteration (Layers) or not (Iter); (c) the initialization U^0; (d) the pairwise similarity function s; (e) the number k of pooled vectors, obtained by k-means instead of GAP. We also

Ablation	Option	Acc	Linear			LN			Acc
$h(S)$	$\boldsymbol{\sigma}_2(S_i/\sqrt{d})_{i=1}^m$	56.6	Q	K	V	Q	K	V	
	$\eta_2(\boldsymbol{\sigma}_1(S/\sqrt{d}))$	55.6	✓	✓	✓	✓	✓	✓	**57.0**
Layers	3	56.8	✓	✓		✓	✓	✓	**56.6**
	5	55.9	✓			✓	✓	✓	56.5
Iter	3	56.5		✓		✓	✓	✓	56.4
	5	56.4				✓	✓	✓	55.6
U^0	U	56.3	✓	✓		✓	✓		56.3
	$\text{diag}(X^\top X)$	56.6	✓	✓		✓			56.0
$s(\mathbf{x},\mathbf{y})$	$-\|\mathbf{x}-\mathbf{y}\|^2$	56.5	✓	✓			✓		56.2
	cosine	56.3	✓	✓			✓	✓	**56.6**
k (max)	2	56.5	✓	✓				✓	56.4
	5	56.4	✓	✓		✓		✓	56.2
k (concat)	2	56.5					✓	✓	56.2
	5	55.9	✓	✓					54.4
ϕ_Q, ϕ_K	$W_Q = W_K$	56.4							54.5
SimPool		**57.1**	GAP						55.0

Table 3.12: SimPool ablation on ImageNet-20% using ResNet-18 trained for 100 epochs. Ablation of (left) design; (right) linear and LayerNorm (LN) [176] layers. q, k, v: query, key, value. $\boldsymbol{\sigma}_2(S_i/\sqrt{d})_{i=1}^m$: same as our default, but with multi-head attenion, $m = 4$ heads; k (max): maximum taken over output logits; k (concat): concatenation and projection to the same output dimensions d'. Green: learnable parameter; blue: winning choice per group of experiments; steel blue: Our chosen default. Using pooling operation $f = f_\alpha$ (3.9) (left); $f = f_{-1}$ (right).

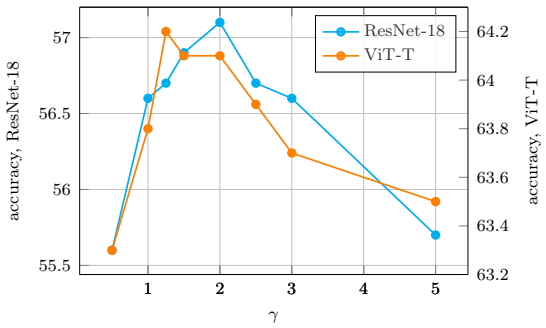


Figure 3.4: *Image classification* top-1 accuracy (%) *vs. exponent* $\gamma = (1-\alpha)/2$ (3.67) for ResNet-18 supervised on ImageNet-20% and ViT-T supervised on ImageNet-1k, both for 100 epochs.

consider queries and keys sharing the same mapping, $W_Q = W_K$. We observe that multi-head, few iterations and initialization by $\text{diag}(X^\top X)$ perform slightly worse, without adding any extra parameters, while setting $W_Q = W_K$ performs slightly worse, having 50% less parameters.

Linear and LayerNorm layers In Table 3.12 (right), we systematically ablate linear and LayerNorm (LN) [176] layers on query q, key k and value v. We strive for performance and quality while at the same time having a small number of components and parameters. In this sense, we choose the setup that includes linear layers on q, k and LN on k, v, yielding 56.6 accuracy. We observe that having linear and LN layers everywhere performs best under classification accuracy. However, this setup has attention maps of lower quality and more parameters.

Pooling parameter α (3.67) We ablate the effect of parameter α of the pooling function f_α (3.67) on the classification performance of SimPool using ResNet-18 on ImageNet-20% and ViT-T on ImageNet-1k for 100 epochs. We find learnable α (or $\gamma = (1-\alpha)/2$) to be inferior both in terms of performance and attention map quality. For ResNet-18 on ImageNet-20%, it gives top-1 accuracy 56.0%. Clamping to $\gamma = 5$ gives 56.3% and using a $10\times$ smaller learning rate gives 56.5%.

In Figure 3.4, we set exponent γ to be a hyperparameter and observe that for both networks, values between 1 and 3 are relatively stable. Specifically, the best choice is 2 for ResNet-18 and 1.25 for ViT-T. Thus, we choose exponent 2 for convolutional networks (ResNet-18, ResNet-50 and ConvNeXt-S) and 1.25 for vision transformers (ViT-T, ViT-S and ViT-B).

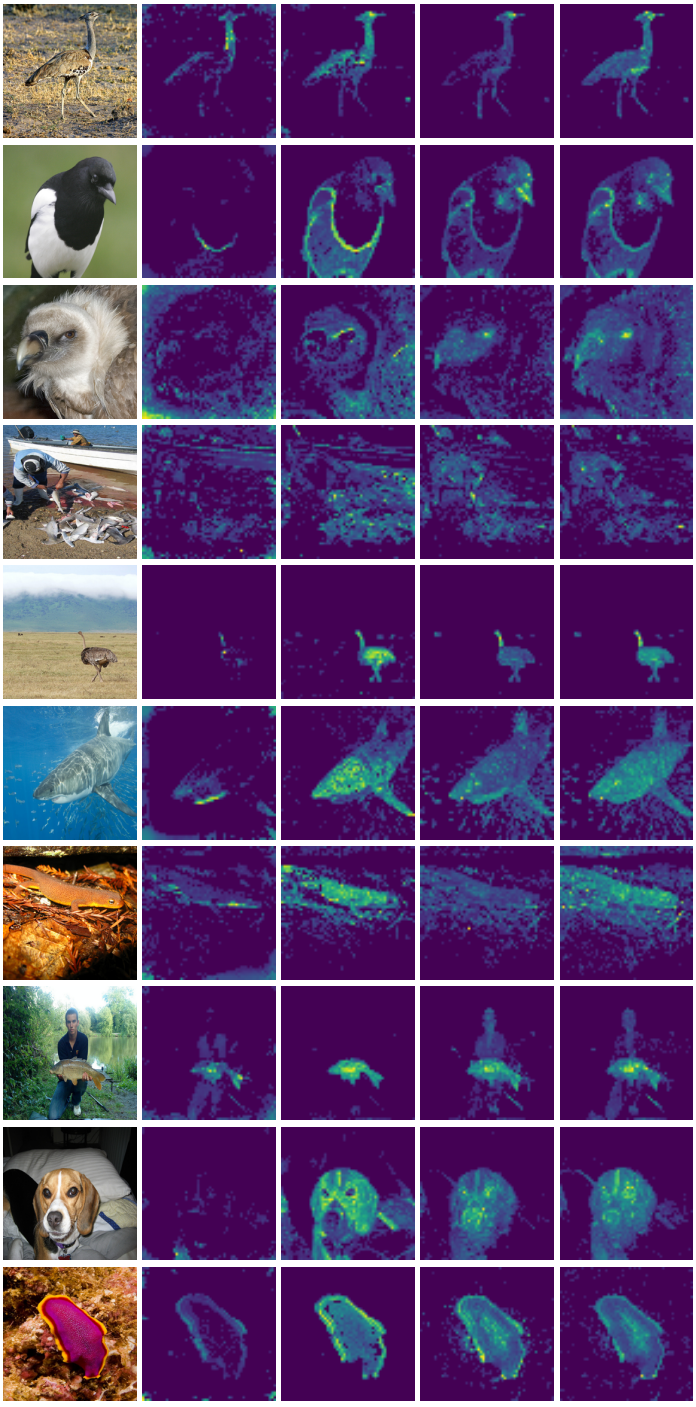


Figure 3.5: *Attention maps* of ViT-S [23] trained on ImageNet-1k for 100 epochs under supervision and self-supervision [29]. For ViT-S baseline, we use the mean attention map of the CLS token. For SimPool, we use the attention map $\mathbf{a}$ (3.65). Input image resolution: 896×896; patches: 16×16; output attention map: 56×56.

3.4.5 Visualizations

Attention maps: ViT Figure 3.5 shows attention maps of supervised and self-supervised ViT-S trained on ImageNet-1k. The ViT-S baseline uses the CLS token for pooling by default. For SimPool, we remove the CLS stream entirely from the encoder and use the attention map $\mathbf{a}$ (3.65).

We observe that under *self-supervision*, the attention map quality of SimPool is on par with the baseline and in some cases the object of interest is slightly more pronounced, *e.g.*, rows 1, 3, 6 and 7.

What is more impressive is *supervised* training. In this case, the baseline has very low quality of attention maps, focusing only on part of the object of interest (*e.g.*, rows 1, 2, 5, 6, 10), focusing on background more than self-supervised (*e.g.*, rows 1, 4, 6, 7, 8), even missing the object of interest entirely (*e.g.*, rows 3, 9). By contrast, the quality of attention maps of SimPool is superior even to self-supervised, attending more to the object surface and less background.

Segmentation masks Figure 3.6 shows the same images for the same setting as in Figure 3.5, but this time overlays segmentation masks on top input images, corresponding to more than 60% mass of the attention map. Again, SimPool is on par with baseline when self-supervised, supervised baseline has poor quality and supervised SimPool is a lot better, although its superiority is not as evident as with the raw attention maps.

Object localization Figure 3.7 visualizes object localization results, comparing bounding boxes of SimPool with the baseline. The results are obtained from the experiments of Table 3.5, using ViT-S with supervised pre-training. We observe that the baseline systematically fails to localize the objects accurately. On the other hand, SimPool allows reasonable localization of the object of interest just from the attention map, without any supervision other than the image-level label.

Attention maps: The effect of γ Figure 3.8 and Figure 3.9 visualize the effect of exponent $\gamma = (1 - \alpha)/2$ of pooling operation f_α (3.9) on the quality of the attention maps of ResNet-18 and ViT-T, respectively. The use of the average pooling operation f_{-1} as opposed to f_α (3.9) is referred to as no γ. For ResNet-18, we observe that for $\gamma < 1.25$ or $\gamma > 3.0$, the attention maps are of low quality, failing to delineate the object of interest (*e.g.*, row 1), missing the object of interest partially (*e.g.*, rows 2, 4) or even entirely (*e.g.*, row 3). For ViT-T, it is impressive that for γ around or equal to 1.25, the attention map quality is high, attending more (*e.g.*, row 4) or even exclusively (*e.g.*, rows 1, 2) the object instead of background.

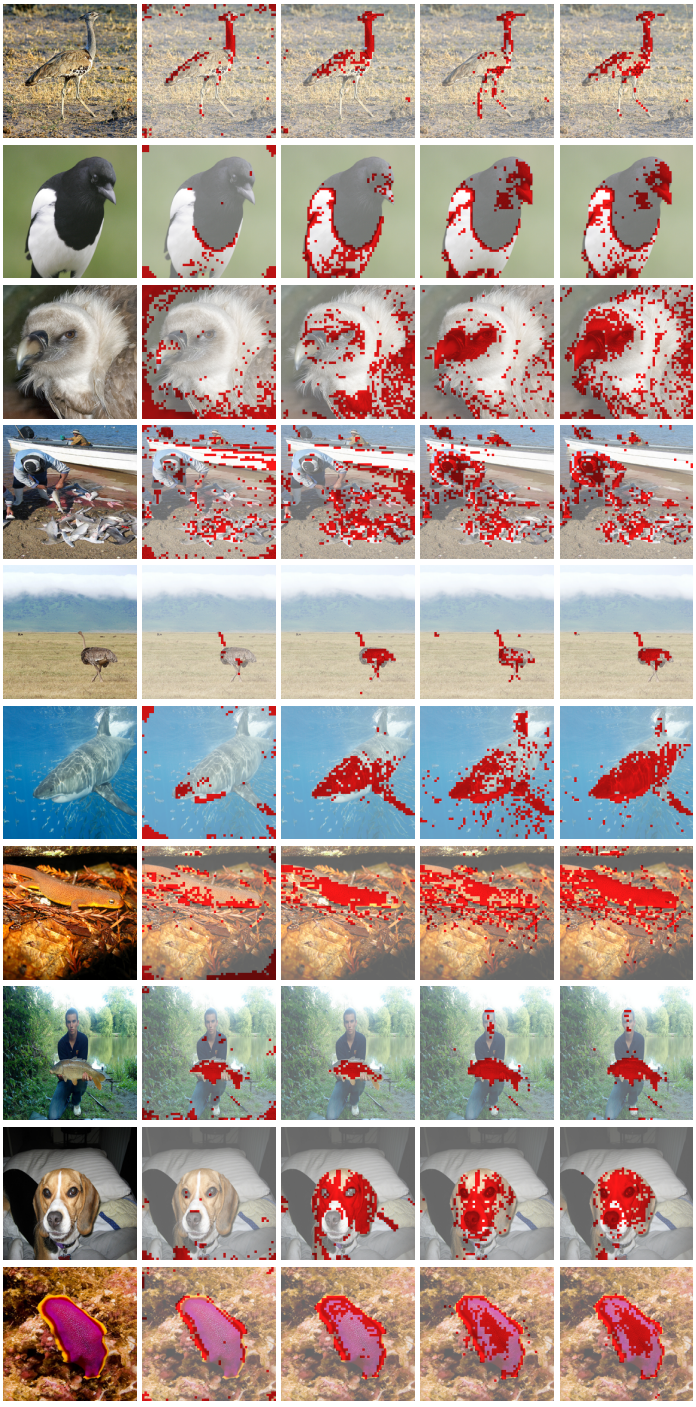


Figure 3.6: *Segmentation masks* of ViT-S [23] trained on ImageNet-1k for 100 epochs under supervision and self-supervision [29]. For ViT-S baseline, we use the attention map of the CLS token. For SimPool, we use the attention map $\mathbf{a}$ (3.65). Same as Figure 3.5, with attention map value thresholded at 60% of mass and mask overlaid on input image.

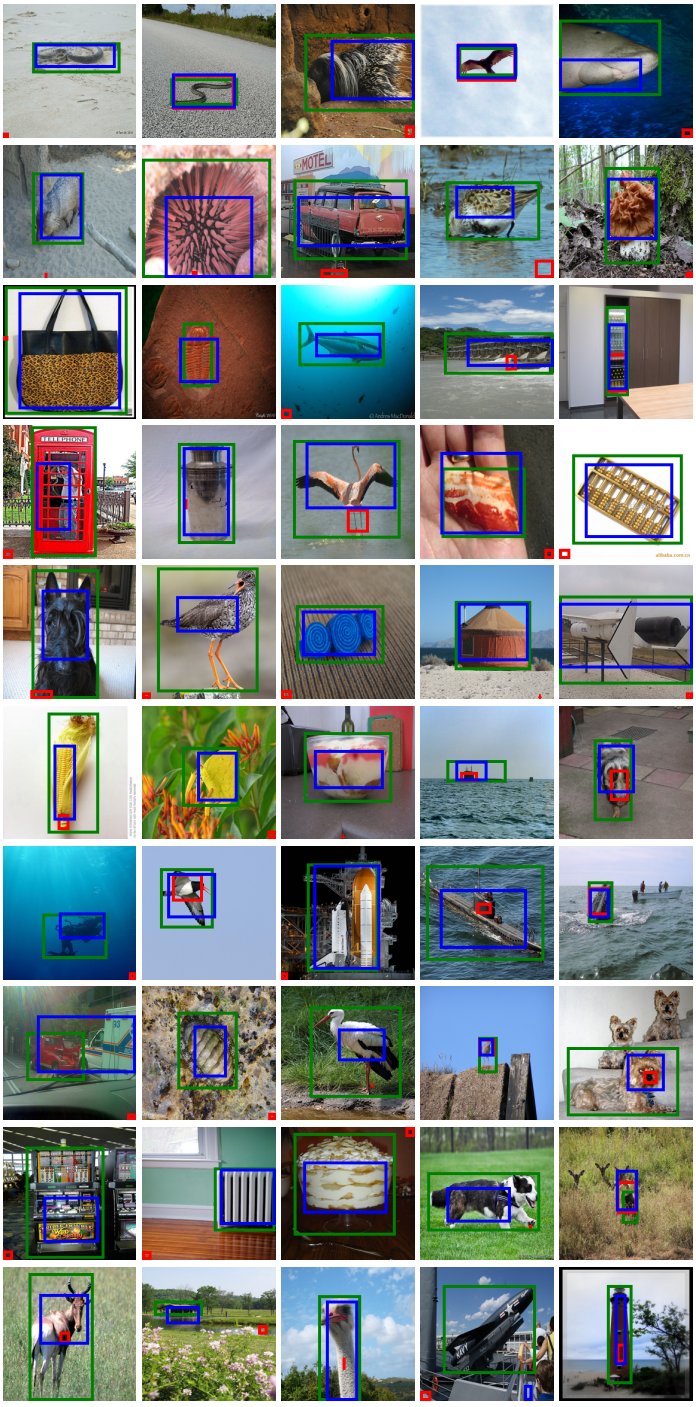


Figure 3.7: *Object localization* on ImageNet-1k with ViT-S [23] supervised pre-training on ImageNet-1k-1k for 100 epochs. Bounding boxes obtained from experiment of Table 3.5, following [181]. **Green**: ground-truth bounding boxes; **red**: baseline, predicted by the attention map of the CLS token; **blue**: predicted by SimPool, using the attention map $\mathbf{a}$ (3.65).

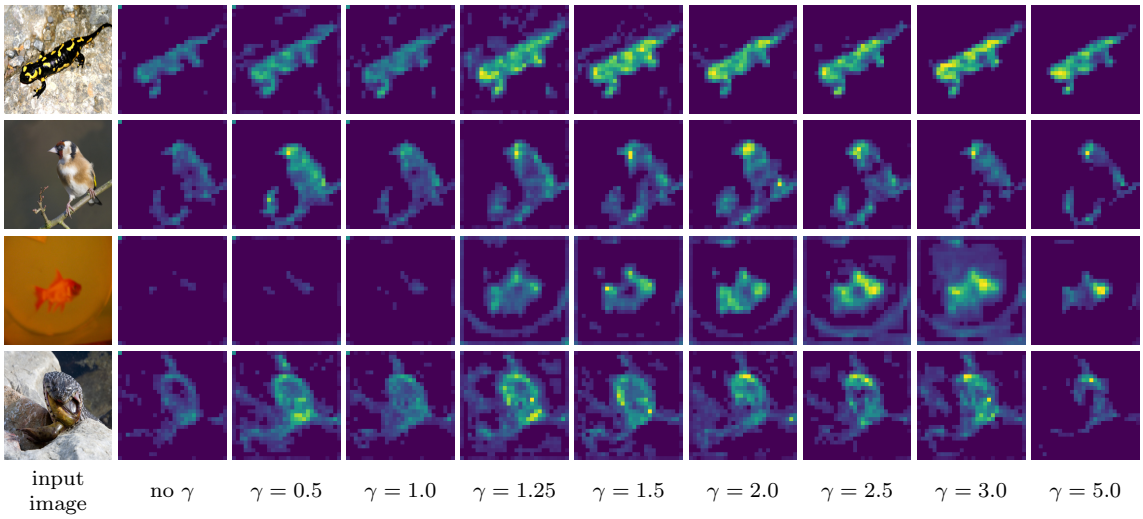


Figure 3.8: *The effect of* γ. Attention maps of ResNet-18 [110] with SimPool using different values of γ trained on ImageNet-20% for 100 epochs under supervision. We use the attention map $\mathbf{a}$ (3.65). Input image resolution: 896×896; output attention map: 28×28; no γ: using the average pooling operation f_{-1} instead of f_α (3.9). We set $\gamma = 2$ by default for convolutional networks.

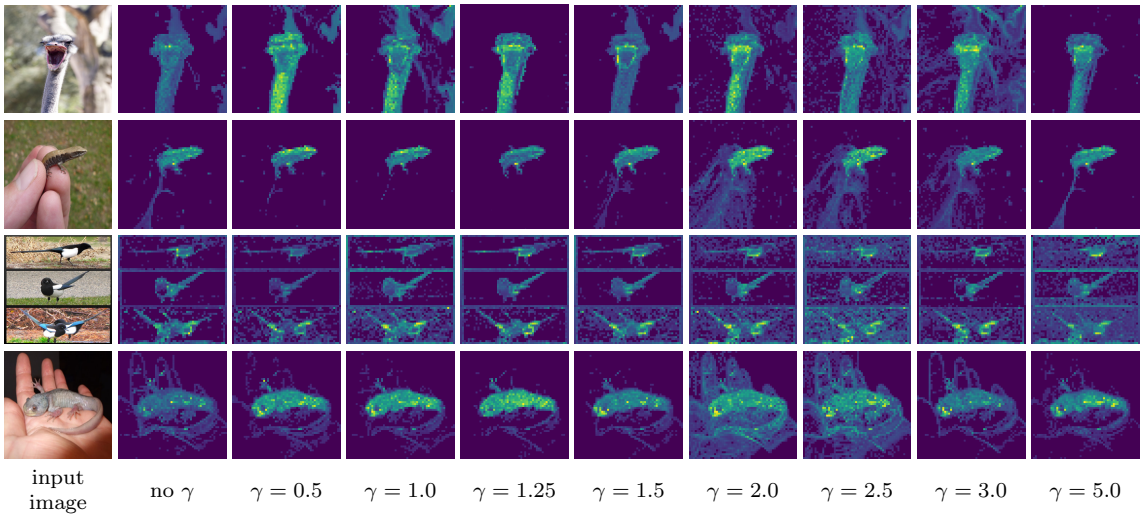


Figure 3.9: *The effect of* γ. Attention maps of ViT-T [23] with SimPool using different values of γ trained on ImageNet-1k for 100 epochs under supervision. We use the attention map $\mathbf{a}$ (3.65). Input image resolution: 896×896; patches: 16×16; output attention map: 56×56; no γ: using the average pooling operation f_{-1} instead of f_α (3.9). We set $\gamma = 1.25$ by default for transformers.

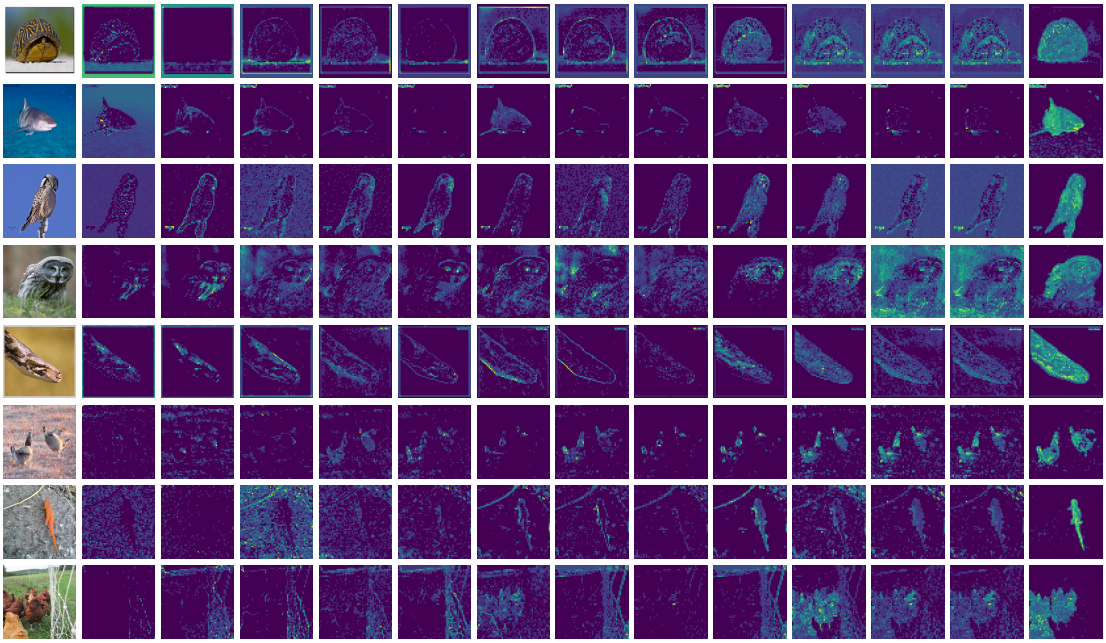


Figure 3.10: *CLS vs. SimPool.* Attention maps of ViT-T [23] trained on ImageNet-1k for 100 epochs under supervision. For CLS, we use the mean attention map of the CLS token of each block. For SimPool, we use the attention map $\mathbf{a}$ (3.65). Input image resolution: 896×896; patches: 16×16; output attention map: 56×56.

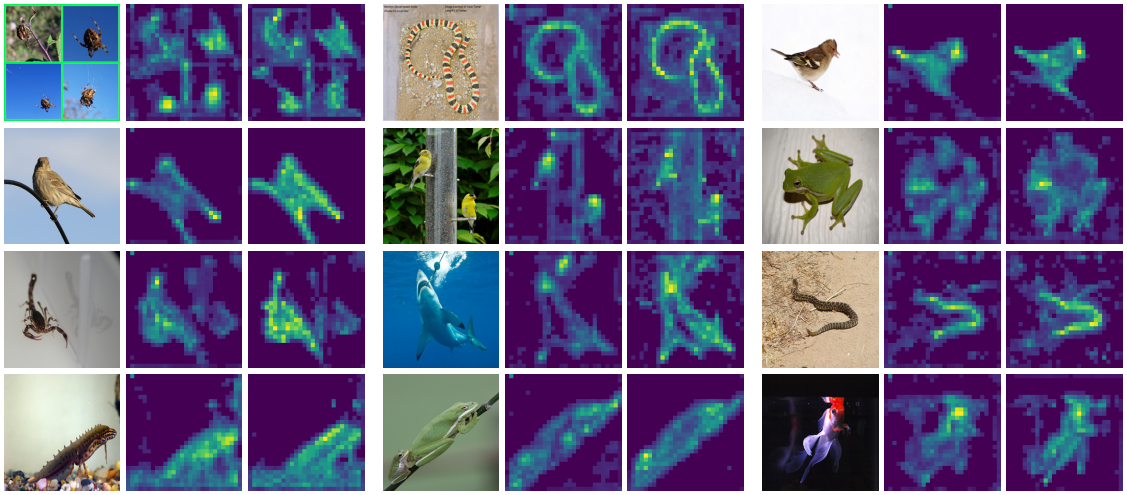


Figure 3.11: *Attention maps* of ResNet-50 [110] trained on ImageNet-1k for 100 epochs under supervision and self-supervision [29]. We use the attention map $\mathbf{a}$ (3.65). Input image resolution: 896×896; output attention map: 28×28.

Attention maps: CLS vs. SimPool Figure 3.10 compares the quality of the attention maps of supervised ViT-T trained with CLS to that of SimPool. For CLS, we visualize the mean attention map of the heads of the CLS token for each of the 12 blocks. For SimPool, we visualize the attention map $\mathbf{a}$ (3.65). SimPool has attention maps of consistently higher quality, delineating and exclusively focusing on the object of interest (*e.g.*, rows 3, 5, 7). It is impressive that while CLS interacts with patch tokens in 12 different blocks, it is inferior to SimPool, which interacts only once at the end.

Attention maps: ResNet, ConvNeXt Figure 3.11 and Figure 3.12 show attention maps of supervised and self-supervised ResNet-50 and ConvNeXt-S, respectively. Both networks are pre-trained on ImageNet-1k for 100 epochs. We use the attention map $\mathbf{a}$ (3.65). We observe that SimPool enables the default ResNet-50 and ConvNeXt-S to obtain raw attention maps of high quality, focusing on the object of interest and not on background or other objects. This is not possible with the default global average pooling and is a property commonly thought of vision transformers when self-supervised [29]. Between supervised and self-supervised SimPool, the quality differences are small, with self-supervised being slightly superior.

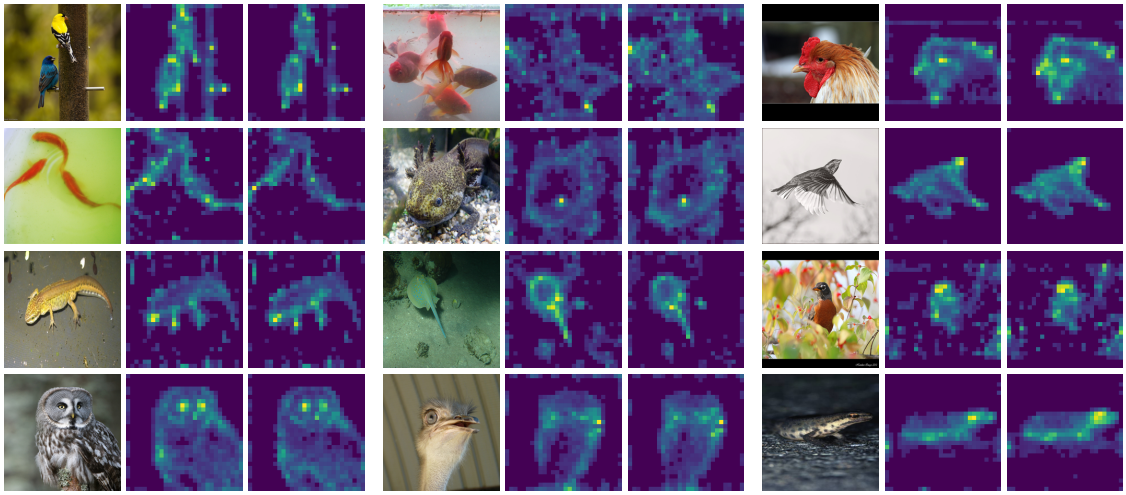


Figure 3.12: *Attention maps* of ConvNeXt-S [22] trained on ImageNet-1k for 100 epochs under supervision and self-supervision [29]. We use the attention map $\mathbf{a}$ (3.65). Input image resolution: 896×896; output attention map: 28×28.

3.5 Conclusion

We have introduced SimPool, a simple, attention-based pooling mechanism that acts at the very last step of either convolutional or transformer encoders, delivering highly superior quantitative results on several benchmarks and downstream tasks. In addition, SimPool delivers decent attention maps in both convolutional and transformer networks under both supervision and self-supervision with remarkable improvement in delineating object boundaries for supervised transformers. Despite this progress, we believe that investigating why the standard CLS-based attention fails under supervision deserves further study.

Acknowledgements This work was supported by the Hellenic Foundation for Research and Innovation (HFRI) under the BiCUBES project (grant: 03943). It was also supported by the RAMONES and iToBos EU Horizon 2020 projects, under grants 101017808 and 965221, respectively. NTUA thanks NVIDIA for the support with the donation of GPU hardware.

4

Extracting Multimodal Representations via Discrete-Space Inversion

Contents

4.1 Revisiting Composed Image Retrieval: A Training-Free Approach for Multimodal Representations

Image-to-image retrieval is a computer vision task with applications to landmarks [198], fashion products [199], face recognition [200], remote sensing [201] and medical images [202], among others. The retrieval is performed purely according to the visual content of the query [40], [41]. On the other hand, if the object can be described with text, then, text-to-image retrieval [42]–[44] applies. The most flexible way to express the user intent is a query comprising both an image and a text description. This is explored in *composed image retrieval* (CIR) [34]–[39], which aims to retrieve target images not only visually similar to the query image, but also modified in accordance with the specifics of the text query.

Traditionally, CIR methods are supervised by *triplets* [34], [45], [46]. However, the labor-intensive process of labeling confined early research to specific applications in fashion [47]–[49], physical states [50], object attributes and object composition [34], [203], [204]. The emergence of vision-language models (VLM) [52]–[54] led to their integration into CIR. Initially, this has been achieved by fine-tuning using triplets [37]. More recently, *zero-shot composed image retrieval* (ZS-CIR) [39], [51] significantly increases the spectrum of possible applications. Most existing methods employ *textual inversion*, *i.e.*, mapping the query image to text, thus allowing query composition purely by means of text. Also, most methods are trained using unlabeled images [39], [51], or are not trained at all [55], but they do require the use of large language models [56] (LLM).

In this paper, we focus on a specific variant of composed image retrieval, namely *domain conversion*, where the text query serves as a description of the target domain [39]. Unlike conventional cross-domain retrieval [59], where models are trained to use queries of a source domain and retrieve items from another target domain, we address a more practical, *open-domain* setting, where the query and database may be from any unseen domain. We target different variants of this task, where

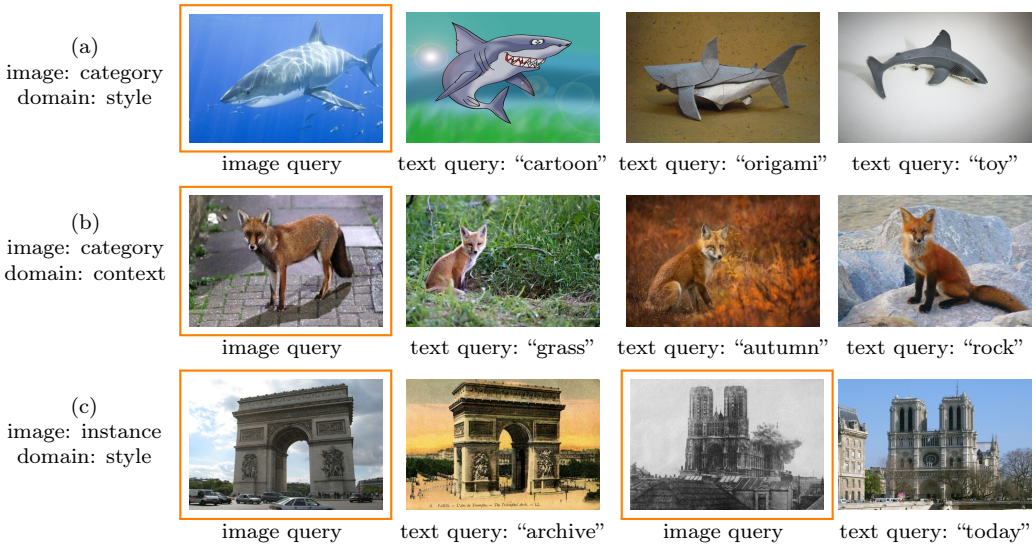


Figure 4.1: We introduce FreeDom, a training-free, composed image retrieval method for domain conversion based on CLIP [52]. Given an *image query* (**framed**) and a *text query* that names a *domain*, we retrieve images having the class of the image query and the domain in the text query. We target a range of applications where classes can be defined at *category level* (a,b) [205], [206] or *instance level* (c) [207], and domains can be defined as *styles* (a,c), or *context/environment* (b). For each image query, retrieved images shown for different text queries.

the class of the query object is defined at *category-level* or at *instance-level*, while the domain corresponds to descriptions of *style* or *context*, as shown in Figure 4.1. Even though domain conversion is a subset of the tasks handled by existing CIR methods, the variants considered in our work reflect a wider set of applications than what was encountered in prior art [39].

Large pre-trained VLMs provide powerful representations of objects, domains and their combinations. Our approach is *training-free* by using a frozen VLM and performing textual inversion in a *non-parametric* way, assuming access to an external large memory of words. Inversion maps images to the discrete input space of text, instead of the continuous latent space of word tokens as in prior work [51]. Compared to such alternative, our memory-based inversion is not only more efficient and intuitive, but also comes with great performance benefits. While our emphasis lies in domain conversion, the proposed approach is versatile and applicable to various composed image retrieval tasks, where its performance is competitive to the state-of-the-art approaches.

In summary, we make the following contributions:

1. We are the first to focus on composed image retrieval in the context of domain conversion, and introduce three new benchmarks additional to the one explored in existing work.
2. We introduce FreeDom, a training-Free CIR method for DOMain conversion that operates in an open world by inheriting the capabilities of a frozen CLIP model.
3. We demonstrate that textual inversion performs better in the discrete input space of known words than in the continuous latent space of pseudo-words.
4. We outperform all existing methods by a large margin on four different benchmarks. Our experimental results form a testbed for future comparisons in this task.

4.2 Contextualizing Composed Image Retrieval: Advances and Methods

Composed image retrieval (CIR) Image-to-image [40], [41] and text-to-image [42]–[44] retrieval provide useful ways to explore large image collections. Nevertheless, composed image retrieval provides more flexible ways to express the query and enables novel applications. TIRG [34] is the first to *compose* image and text as a search query, where text serves as a modification of the image to refine the retrieval results. Training is supervised with cross-entropy loss, using triplets of

the form *reference image, query text, target image*. Following the same setting, JVSM [45] learns image-text compositional embeddings in a unified space using multiple matching losses.

Other methods exploit attention in the form of a multi-modal disentangled non-local blocks [46] to correlate text with image regions [36] extracted by an RPN [208], to perform the composition at multiple depths [35], to modulate content [38], and to discover the relation between composed query and target image [209]. DRA [210] learns a dual relation alignment network, while MPC [204] introduces a variant of the task with multiple queries. All these methods perform training from scratch and rely on triplets related to fashion [47]–[49], physical states [50], object attributes and object composition [34], [203], [204]. Labeling such triplets is expensive and limits the widespread use of CIR.

Inspired by vision-language foundation models [52], [53], [211], recent work builds upon them in different ways. CIRPLANT [212] and FashionVLP [213] extract features from a reference image as well as text features using a tokenizer [111], [214] and fine-tune the VLM [52], [214], [215] using triplets. CLIP4CIR [37] fine-tunes CLIP [52] and trains a small network to combine image and text features using triplets. BLIP4CIR [216] builds upon CLIP4CIR with BLIP [54] and trains using reversed triplets, along with the original ones.

Due to the need for richer and more data, datasets are collected by crowd-sourcing with human-generated text [212], by exploiting LAION-5B [217], [218] or VQA v2.0 [219], [220], or by automatically synthesizing millions of high-quality triplets [221] using generative models [222]. All these methods benefit by the compositional ability of vision-language models [52], [54], [214], [215], but still rely on triplet datasets.

Pic2Word [39] relies on a VLM and is the first to avoid triplets and to perform domain conversion as one of the tasks. It follows self-supervised training to invert the query image to a text token. Thus, query composition takes place in the text domain by combining this token with the query text. SEARLE [51] performs textual inversion with test-time optimization per query image. Then, a network is trained to imitate the result of such optimization, so that inference is performed more efficiently. Our memory-based textual inversion avoids pre-training or test-time optimization and is shown to outperform all prior work by a large margin.

Concurrent to our work, notable efforts in the field include CIReVL [55], SPRC [223], and ISA [224], each contributing unique to the composition. CIReVL [55] composes image and text solely in the language domain. It uses CLIP [52] as image and text encoder, BLIP-2 [225] to caption the reference image and GPT-3.5 turbo [56] to re-compose the generated caption based on the query text. All models are pre-trained,

making the method training-free, modular, but also computationally intensive.

Training-free use of VLMs The emergence of vision-language models (VLMs) [52]–[54], [225] revolutionized the field of multimodal learning. Trained on massive datasets [217], these models have instrumental abilities to map images and text into a shared embedding space and are successful in training-free scenarios. MaskCLIP [226] and CLIP-DIY [227] demonstrate the intrinsic potential of CLIP for semantic segmentation, while FLDM [228] highlights its effectiveness in text-guided video editing. The training-free paradigm extends to text-guided image editing [229] and layout control [230], both using cross-attention. VLMs are promising in specialized applications too, such as deepfake detection [231], cross-domain image composition [232] and phrase localization [233]. Related to our training-free approach, CIReVL [55] uses VLMs and LLMs to compose image and text queries in the language domain.

Cross-domain image retrieval (CDIR) This is the task where the query image and database images come from different domains and the challenge is to bridge domain gap [57]. As visual domain, one might consider style [58], color [234], texture [58], context [59], lighting conditions [59] or images captured using different sensors [235]. One main line of research is in sketch-based image retrieval [58], [236]–[241], and another on consumer scenarios such as street-to-shop [59], [242], [243].

Early methods do not generalize to new object classes or domains. This is the goal of zero-shot sketch-based retrieval [244]–[246]. More recent methods dispense with the need for labeled cross-modal pairs and are unsupervised [60]–[63]. Generalization to an unseen domain is only demonstrated by UCDR [64]. Nevertheless, no CDIR method includes the domain of the query image in the database, which becomes meaningful in our task, *i.e.* domain conversion with image-text queries.

4.3 Revisiting Textual Inversion: A Discrete-Space Retrieval-Based Approach

4.3.1 Preliminaries

Composed image retrieval is the task where the goal is to retrieve images based on an *composed image-text query*, that is, a query that consists of a *visual* part, the query image, and a *textual* part, the query text. In this work, we focus on a specific variant of composed image retrieval that targets *domain conversion* [39].

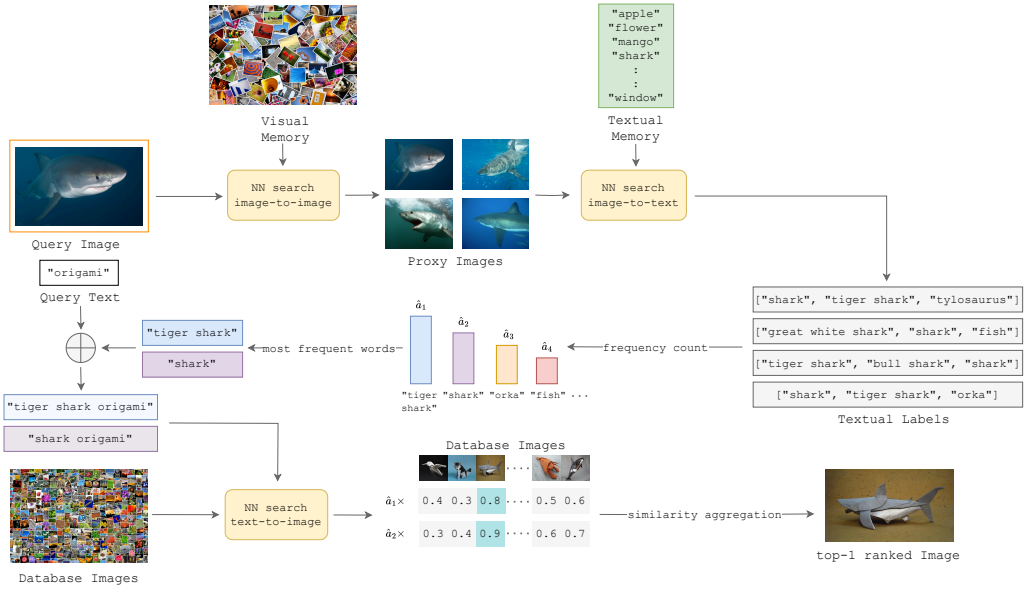


Figure 4.2: *Overview of* FREEDOM. Given a query image and a query text indicating the target domain, we first retrieve proxy images to the query, by an image-to-image search over a visual memory. Then, we associate a set of text labels to each proxy image, by an image-to-text search over a textual memory. Each of the most frequent text labels is composed with the query text in the text space and images are retrieved from the database by text-to-image search. The resulting lists of similarities are linearly combined with the frequencies of occurrence as weights. Here: $k = 4$, $n = 3$, $m = 2$.

In particular, the query image y depicts an object of *class* $C(y)$ in *source domain* $D(y)$, while the query text t represents the *target domain*, $D(t)$. The two elements are jointly referred to as the *composed query*, $q = (y, t)$. Given an image dataset X, the goal is to retrieve images from X whose class is same as that of the query image, $C(y)$, and whose domain is same as that of the query text, $D(t)$. Retrieval amounts to ranking images $x \in X$ according to their *composed similarity* $s(q, x) \in \mathbb{R}$ to the query q.

We rely on a pre-trained vision-language model that consists of a *visual encoder* $f : \mathcal{I} \to \mathbb{R}^d$ and a *text encoder* $g : \mathcal{T} \to \mathbb{R}^d$, which map respectively input images from image space $\mathcal{I}$ and words[1] from text space $\mathcal{T}$ to the same embedding space of dimension d. Using those encoders, we extract a visual embedding $\mathbf{y} = f(y) \in \mathbb{R}^d$ and text embedding $\mathbf{t} = g(t) \in \mathbb{R}^d$ for the query. Similarly, the embedding of an image $x \in X$ or a word w are denoted by $\mathbf{x} = f(x) \in \mathbb{R}^d$ and $\mathbf{w} = g(w) \in \mathbb{R}^d$, respectively. All embeddings are ℓ_2-normalized.

Given the visual-language model, the goal in this work is to represent the composed query in the same embedding space as images and text. That is, define a *composed encoder* $h : \mathcal{I} \times \mathcal{T} \to \mathbb{R}^d$ such that the composed query q is mapped to $\mathbf{q} = h(q) = h(y, t) \in \mathbb{R}^d$, again ℓ_2-normalized. For any image $x \in X$, this allows us to express the composed similarity as the cosine similarity

$$s(q, x) := h(q)^\top f(x) = h(y, t)^\top f(x). \tag{4.1}$$

Thus, given the encoders f, g, the goal is to define h.

4.3.2 Expanded Textual Inversion

Textual inversion The ability of the text encoder, by design, to combine different concepts at its input and map them jointly to the embedding space, motivates us to represent the composed query $q = (y, t)$ entirely in the text space $\mathcal{T}$ and then map it to the embedding space, using g. The query text t is already in $\mathcal{T}$; but to map the query image y to $w^* \in \mathcal{T}$, we need to first embed it to $\mathbf{y} = f(y)$ and then map it from the embedding space back to $\mathcal{T}$:

$$w^* = g^{-1}(f(y)). \tag{4.2}$$

The process of mapping the query image y to text in $\mathcal{T}$ is called *textual inversion* and the challenge is that the inverse mapping g^{-1} is unknown.

Common approaches are *pre-training* [39] and *test-time optimization* [51], both representing w^* in the latent space of vector tokens. The former defines a decoder

[1] With the term *words*, we shall refer to both words and sentences.

and trains it on a dataset to *learn* the inverse mapping g^{-1} of the text encoder. Its challenge is the sheer scale of training required to reach anywhere close to the quality of the pre-trained text encoder g. The latter defines a variable at the input of g and finds the optimal solution w^* such that $g(w^*) = f(y)$. Its challenge is that there is a multitude of local optimal solutions, thus the approach overly relies on the initialization of the variable, which remains unknown.

Memory-based inversion Contrary to existing approaches, we achieve the inversion by nearest neighbor search over an external vocabulary $V \subset \mathcal{T}$ of words [247], without training or optimization, and we find w^* in the discrete text space $\mathcal{T}$ rather than in the continuous latent space of vector tokens.

In particular, if $V = \{v_1, \dots, v_N\}$, we map vocabulary words v_i to text embeddings $\mathbf{v}_i = g(v_i)$ for $i = 1, \dots, N$. The *text memory* of words v_i and associated embeddings $\mathbf{v}_i$ is the restriction $g|_V : V \to \mathcal{T}$ of g to V:

$$g|_V : \{v_1, \dots, v_N\} \to \{\mathbf{v}_1, \dots, \mathbf{v}_N\}. \tag{4.3}$$

Given an embedding $\mathbf{v}_i \in \mathcal{V} = g(V) = \{\mathbf{v}_1, \dots, \mathbf{v}_N\}$, we can instantly determine the associated word v_i. This process essentially defines the restriction $g^{-1}|_{\mathcal{V}} : \mathcal{V} \to V$ of the otherwise unknown inverse g^{-1} to $\mathcal{V}$. For brevity, we refer to $g^{-1}|_{\mathcal{V}}$ as g^{-1} in the following.

What remains is, given the query embedding $\mathbf{y} \in \mathbb{R}^d$, to approximate it by one or more vectors in $\mathcal{V}$. We do this by finding the m nearest neighbors of $\mathbf{y}$ in $\mathcal{V}$,

$$\mathcal{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_m\} = \mathrm{NN}_m(\mathbf{y}; \mathcal{V}), \tag{4.4}$$

given by descending order of (cosine) similarity. Since $\mathcal{W} \subset \mathcal{V}$, we can map an embedding $\mathbf{w}_i \in \mathcal{W}$ by g^{-1} back to the associated word $w_i = g^{-1}(\mathbf{w}_i)$ in V. Thus, all neighbors are mapped to words

$$W = \{w_1, \dots, w_m\} = g^{-1}(\mathcal{W}). \tag{4.5}$$

Putting everything together, this set of words is given by $W = \phi_V(y)$, where *NN-inversion*

$$\phi_V(y) := g^{-1}(\mathrm{NN}_m(f(y); g(V))) \tag{4.6}$$

is an approximation of (4.2) by vocabulary V. The larger the vocabulary, the better the quality of approximation—but the more expensive the process. We use function ϕ_V (4.6) to define different versions of composed encoder h below.

Memory-based inversion is similar to zero-shot recognition, where the query image y is represented by a set of words W from vocabulary V. We call the words found by $\phi_V(y)$ the *text labels* or *labels* of y.

Single-word inversion The closest word w_1 to the query image is merged with the query text t to form a composed query $w_1 \oplus t$ in the text space alone, where $\oplus$ denotes space-delimited string concatenation. Thus, h becomes

$$h_1(y, t) := \beta_1(\phi_V(y), t), \tag{4.7}$$

where ϕ_V is given by (4.6) and

$$\beta_1(W, t) := g(w_1 \oplus t). \tag{4.8}$$

A single word may often work well. Nevertheless, using more words may help when the correct class is not top-ranked or when a collection of words may better represent a particular image query.

Multi-word inversion: early fusion We take advantage of the ability of the text encoder to combine a number of words in its input. Now, we form a composed query $w_1 \oplus \ldots \oplus w_m \oplus t$ in the text space and h becomes

$$h_E(y, t) := \beta_E(\phi_V(y), t), \tag{4.9}$$

where ϕ_V is given by (4.6) and

$$\beta_E(W, t) := g(w_1 \oplus \cdots \oplus w_m \oplus t). \tag{4.10}$$

We refer to this approach as *early fusion*, since the words w_i are combined at the earliest possible stage.

Multi-word inversion: late fusion Early fusion may be sensitive to words assigned incorrectly by NN-inversion ϕ_V (4.6). The other extreme is *late fusion*, whereby we compose words at the latest possible stage. In particular, we form one composed query $w_i \oplus t$ in the text space for each word w_i, embed it separately and form a linear combination of these embeddings. Thus, h becomes

$$h_L(y, t) := \beta_L(\phi_V(y), t), \tag{4.11}$$

where ϕ_V is given by (4.6),

$$\beta_L(W, t, \mathbf{a}) := \sum_{i=1}^{m} a_i g(w_i \oplus t) \tag{4.12}$$

and $a_i \in \mathbb{R}$ is a weight associated with word w_i, by default uniform $\mathbf{a} = \mathbf{1} \in \mathbb{R}^m$. Because of the linearity of (4.1), this is equivalent to m independent queries followed by a linear combination of the resulting similarities.

Memory-based expansion Even if we use multiple nearest neighbors, the underlying cross-modal (image-to-text) similarity of (4.4) remains challenging. To achieve more reliable zero-shot recognition, we employ a retrieval-based augmentation mechanism. We use a *visual memory* of images z_i and associated embeddings $\mathbf{z}_i = f(z_i)$ from an external image set Z. We first expand the query image through a set of k *proxy images*, found as nearest neighbors of $\mathbf{y}$ in the embeddings $\mathcal{Z} = f(Z)$ based on unimodal (image-to-image) similarity:

$$\mathcal{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_k\} = \mathrm{NN}_k(\mathbf{y}; \mathcal{Z}). \tag{4.13}$$

Then, following (4.4), for each proxy image $\mathbf{y}_j$ in the embedding space, we find its n nearest neighbors in $\mathcal{V}$ and the associated words in the text space

$$W_j = \{w_{j1}, \dots, w_{jn}\} = g^{-1}(\mathrm{NN}_n(\mathbf{y}_j; \mathcal{V})). \tag{4.14}$$

From the union $W^+ = \cup_{j=1}^k W_j$ with $|W^+| \leq nk$ because of repeating words w_{ji}, we select the m most frequent words $\hat{W} = \{\hat{w}_1, \dots, \hat{w}_m\}$ and we define $\hat{a}_i$ as the frequency associated with word $\hat{w}_i$. We write this filtered set of words as $\hat{W} = \phi^+_{X,V}(y)$ as a function of the image query y, where $\phi^+_{X,V}$ is called *expanded NN-inversion*. Finally, h can be defined via either early fusion

$$h_{E^+}(y, t) := \beta_E(\phi^+_{X,V}(y), t), \tag{4.15}$$

where β_E is given by (4.10), or by late fusion

$$h_{L^+}(y, t) := \beta_L(\phi^+_{X,V}(y), t, \mathbf{1}) \quad \text{and} \tag{4.16}$$

$$h_{L^+_\alpha}(y, t) := \beta_L(\phi^+_{X,V}(y), t, \hat{\mathbf{a}}), \tag{4.17}$$

with uniform and frequency weights $\mathbf{1}, \hat{\mathbf{a}} \in \mathbb{R}^m$, respectively, where β_L is given by (4.12). This last version expressed by (4.17) is the complete FreeDom method, summarized in Figure 4.2.

4.4 Benchmarking Composed Image Retrieval: Performance Analysis and Insights

4.4.1 Datasets, Networks and Evaluation Protocol

We target a range domain conversion applications where classes can be defined at *category level* [205], [206], [248] or *instance level* [207], and domains can be defined as *styles* [205], [207], [248], or *context/environment* [206], and additionally evaluate on generic composed image retrieval benchmarks.

Datasets for domain conversion ImageNet-R [205] has renditions of 200 ImageNet-1k [249] classes comprising 30,000 images. Following Pic2Word, experiments are preformed with four domains: *cartoon*, *origami*, *sculpture* and *toy*. MiniDomainNet [248] is a subset of DomainNet [250] with about 140,000 images of 126 classes and four domains: *clipart*, *painting*, *real* and *sketch*. Although this is a classification dataset, we adapt it for retrieval by using the official test set as our query set and the rest as the database. Nico++ [206] is an Out-Of-Distribution classification benchmark of $88,866$ real photographs from six domains: *autumn*, *dimlight*, *grass*, *outdoor*, *rock* and *water*. It contains 60 categories. The query set is composed of 10% randomly selected images. The rest is used as the database. Large time lags location (LTLL) [251] contains images of 25 locations captured over a range of more than 150 years; 225 historical and 275 modern ones. Experiments are perfomed on two domains: *today* and *archive*.

Datasets for generic composed image retrieval are also used in our performance evaluation, namely FashionIQ [49], CIRR [212] and CIRCO [51].

Network We use the OpenAI pre-trained CLIP with a ViT-L/14 image encoder [252].

Evaluation Protocol Unlike Precision@k and Recall@k, which are commonly used in literature and focus on specific points in a ranked list, mean Average Precision (mAP) provides a more comprehensive assessment by considering precision across the entire ranking. Thus, we choose mAP as our evaluation metric for both domain conversion and generic composed image retrieval benchmarks. On CIRR and CIRCO, we evaluate on the validation set, which is a valid choice since no method performs training/validation on this set; the test set is behind an evaluation server.

Text and visual memory We use the $20k$ words of the Open Images V7 dataset [247] as the text memory. It is sufficiently large and it is used in zero-shot recognition [253] as well as by SEARLE [51], which also allows direct comparison between the methods. Unless otherwise stated, we use the image database as visual memory.

4.4.2 Simple Baselines

Unimodal Unimodal-query baselines rely only on similarity using one of the query modalities: *text-only* by $h_T(y,t) := g(t)$ and *visual-only* by $h_V(y,t) := f(y)$, referred to as "Text" and "Image" respectively in Table 4.1. They are both expected to fail since the final similarity does not capture both aspects of the composed query.

Product This baseline is a combination of the two unimodal approaches by using the product of the corresponding similarities. It is referred to as "Text × Image" in Table 4.1.

Sum A common baseline in the literature combines these two unimodal approaches by summation, *i.e.* $h_S(y,t) := g(t)+f(y)$, referred to as "Text + Image" in Table 4.1. The problem is that the text and image embeddings follow very different distributions.

4.4.3 Advanced Baselines

InstructPix2Pix In this baseline, InstructPix2Pix [254] is used to generate an image from our visual and textual queries. Then retrieval is done by image-to-image similarities. The performance of this baseline is low, which is an indicator that the combination of the two modalities through the visual encoder is sub-optimal. We observe that even though several of the generated images are quite successful, at the same time a large amount of them are completely unsuccessful.

FreeDom w/ img-cap In this baseline we assume access to a dataset of image-caption pairs; we use the first 40M images and captions of LAION 400M [255]. This set forms a joint visual-textual memory. Images are retrieved from this memory, and their captions are treated as the text labels of textual inversion. Those are combined with the query text and late fusion follows with weights equal to the similarities between query and memory images. The hyperparameters are chosen to be the same as our standard FreeDom. Interestingly, this baseline surpasses FreeDom on LTLL for the case of "today" as source.

FreeDom w/ captioners In this baseline we use two captioners, namely BLIP [54], and BLIP2 [225]. Every query image is captioned by each captioner and the results are used as two text labels of the image. Subsequently, our standard processing pipeline is followed, while the weights are the similarities of each caption with the query image. This baseline is 15 times slower than the standard FreeDom and is consistently beneath, even though it uses extra architectures.

4.4.4 Competitors

We compare FreeDom to recently proposed zero-shot composed image retrieval methods on four datasets: Pic2Word [39], CompoDiff [221], SEARLE [51], and WeiCom [256]. For all competitors, the same ViT-L/14 [252] is used as image

encoder and also as text encoder for Pic2Word and SEARLE, while CompoDiff uses ViT-G/14 [252] as text encoder. All these methods are run and evaluated by us.

Pic2Word [39] achieves textual inversion in the latent space of text tokens through a three-layered MLP. In all experiments with Pic2Word, we use the officially pre-trained mapping network released by the authors. For ImageNet-R and MiniDN, the composed query has the same format as in the original paper: "a [*target domain*] of *", *e.g.* "a cartoon of *". For NICO++, the composed query is "a * in [*target domain*]", *e.g.* "a * in autumn". Finally, for the LTLL dataset, we use the composed query "a [*target domain*] photo of *", *e.g.* "an archive photo of *".

CompoDiff [221] is built on top of frozen CLIP. We follow the publicly released official implementation for our experiments. We use the officially pre-trained denoising Transformer released by the authors. We do not use any mask, nor any mixed text condition (negative query text). The query text includes only the target domain word, i.e. "[*target domain*]".

SEARLE [51] performs textual inversion by test-time optimization to represent query images in the latent space of vector tokens. We opt for the optimization variant, instead of their feed-forward network, since it is shown to perform better. We use the publicly released official implementation for our experiments and refer to the version with default optimization hyper-parameters as "SEARLE (default)" and to our improved hyper-parameters by "SEARLE (tuned)" in Table 4.1. Each query image is associated to different concepts retrieved from a vocabulary, which is similar to the text labels of our method. We refer to the number of those concepts by m in Table 4.4. The final composed queries are adapted for each dataset in the same way as for Pic2Word. For SEARLE (tuned), we perform hyper-parameter search for learning rate in {0.2, 0.02, 0.002, 0.0002}, optimization iterations in {5,10,50,200,350,500} and number of textual labels m in {1,3,7,10,15}. The best hyper-parameters across datasets are: $lr = 0.0002$, $iters = 350$ and $m = 1$.

WeiCom [256] is a composed image retrieval method specialized for remote sensing. It fits a normal distribution to the similarities between the text query $g(t)$ and all the database images $f(x)$ for $x \in X$, and similarly for the image query $f(y)$. It uses the corresponding cumulative distribution function for each distribution in order to transform the similarities. It transforms the similarities of the two distributions closer to the uniform distribution, and it combines the similarities by summation.

4.4.5 Experimental Results

Comparison with SOTA As shown in Table 4.1, FreeDom outperforms all baselines and competitors by a large margin and for every individual source domain. In particular, it outperforms the second best method on ImageNet-R by 15.83% mAP, on MiniDomainNet by 14.32%, on NICO++ by 10.96% and on LTLL by 6.49%. On ImageNet-R, the second best method is SEARLE (tuned). On MiniDomainNet, CompoDiff and SEARLE (tuned) are the second and third best, respectively, while WeiCom performs lower than the simple baselines "Text × Image" and "Text + Image". On NICO++, SEARLE (tuned) is the only competitor that beats the simple baselines for more than 1%, while Pic2Word performs lower than the simple baseline "Text × Image". On LTLL, WeiCom is the second best method, while interestingly, the baseline "Text × Image" performs higher than Pic2Word, CompoDiff, and SEARLE (default).

Additionally, in Table 4.2, we compare FreeDom with all previous methods that evaluate Recall@k on ImageNet-R. Following the literature [39], [51], [221], we evaluate using only Photo as the source domain and compare with baselines and competitors. Baselines and SEARLE are performed by us, Pic2Word performance is reported from the original paper, the rest of the Pic2Word experiments, ARTEMIS [209], CLIP4CIR [37], and CompoDiff are reported from the CompoDiff paper. FreeDom outperforms all baselines and competitors by a large margin. CIReVL is the second best in most of the cases.

Qualitative analysis In Figure 4.3 we show the histogram of similarities between the query and the positives and negatives of different kinds of images. The unimodal baselines fail as expected. The sum baseline gives high importance to the image query. The normalized variant of WeiCom improves that to a small extent but performs poorly as well. Early fusion improves further but gives too much importance to the object, since several text labels are merged with a single word for the domain. Late fusion significantly improves this imbalance. The use of proxy images and weighting further boost the performance, which is visualized by the blue histogram moving to the right relative to the orange.

4.4.6 Ablations

Impact of hyper-parameters In Table 4.3 we show the impact of the number k of proxy images and number of nearest words from the vocabulary. The experiment is performed for a fixed number of text labels m. Observe that none of the combinations with $k = 1$ or $n = 1$ is the best. Therefore, the two steps of nearest neighbor search are meaningful. Note that the query image is part of the database,

meaning that $k = 1$ corresponds to no expansion. In the rest of our experiments, we set $k = 20$ and $n = 7$, which appear to perform well across all datasets. In Table 4.4, we show the impact of varying m, where benefits are demonstrated on all datasets by going beyond one text label and performance is stable for a large range of values.

Method components In Table 4.4 we show an experiment where each of our components is added one by one leading to the final method. Additionally, we directly compare to SEARLE. SEARLE performs best if a single concept is used. Our simplest variant with a single text label performs better than SEARLE on three out of four datasets, showcasing the benefit of our textual inversion in discrete words. FreeDom benefits by using additional text labels and outperforms SEARLE on all datasets. Late fusion gives a large boost in performance compared to early fusion and so does expansion to proxy images. The use of weights is beneficial on average and only harms slightly on LTLL.

(a) ImageNet-R [205]

Method	Car	Ori	Pho	Scu	Toy	Avg
Text	0.82	0.63	0.68	0.78	0.77	0.74
Image	4.27	3.12	0.84	5.86	5.09	3.84
Text × Image	8.19	5.62	6.98	8.95	9.43	7.83
Text + Image	6.61	4.45	2.18	9.18	8.62	6.21
InstructPix2Pix	3.90	5.70	1.97	5.70	5.62	4.58
FreeDom w/ i.-c.	15.11	6.70	19.77	18.08	16.58	15.24
FreeDom w/ c.	16.68	11.74	17.44	15.68	16.94	15.70
Pic2Word	7.60	5.53	7.64	9.39	9.27	7.88
CompoDiff	13.71	10.61	8.76	15.17	16.17	12.88
SEARLE (default)	10.16	4.48	3.18	10.11	8.88	7.37
SEARLE (tuned)	18.11	9.02	9.94	17.26	15.83	14.04
WeiCom	10.07	7.61	10.06	11.26	13.38	10.47
FreeDom	**35.93**	**11.66**	**27.95**	**36.56**	**37.24**	**29.87**

(b) MiniDomainNet [248]

Method	Clip	Paint	Pho	Ske	Avg
Text	0.63	0.52	0.63	0.51	0.57
Image	7.15	7.31	4.37	7.78	6.65
Text × Image	8.99	8.65	15.85	5.88	9.85
Text + Image	9.58	9.98	9.22	8.52	9.32
InstructPix2Pix	8.57	8.86	7.08	7.20	7.93
FreeDom w/ i.-c.	21.88	17.54	31.78	15.35	21.64
FreeDom w/ c.	27.65	17.42	33.42	17.24	23.91
Pic2Word	13.39	8.63	17.96	8.03	12.00
CompoDiff	19.06	24.27	23.41	25.05	22.95
SEARLE (default)	15.14	10.49	9.89	12.50	12.00
SEARLE (tuned)	25.04	18.72	23.77	19.61	21.78
WeiCom	7.52	7.04	15.13	4.40	8.52
FreeDom	**41.90**	**31.67**	**41.14**	**34.35**	**37.27**

(c) NICO++ [206]

Method	Aut	Dim	Gra	Out	Roc	Wat	Avg
Text	1.00	0.99	1.15	1.23	1.10	1.05	1.09
Image	6.45	4.85	5.67	7.67	7.65	5.65	6.32
Text × Image	8.24	6.36	12.11	12.71	10.46	8.84	9.79
Text + Image	8.47	6.58	9.22	11.90	11.20	8.41	9.30
InstructPix2Pix	4.18	2.66	4.60	4.78	5.19	3.56	4.16
FreeDom w/ i.-c.	15.56	11.64	19.34	19.18	17.56	13.81	16.18
FreeDom w/ c.	14.07	9.54	18.67	20.86	17.34	12.37	15.48
Pic2Word	9.79	8.09	11.24	11.27	11.01	7.16	9.76
CompoDiff	10.07	7.83	10.53	11.41	11.93	10.15	10.32
SEARLE (default)	9.32	8.81	10.95	12.64	11.37	8.79	10.32
SEARLE (tuned)	13.49	13.73	17.91	17.99	15.79	11.84	15.13
WeiCom	8.58	7.39	13.04	13.17	11.32	9.73	10.54
FreeDom	**24.36**	**24.42**	**30.05**	**30.49**	**26.87**	**20.35**	**26.09**

(d) LTLL [207]

Method	Today	Arch	Avg
Text	5.32	6.12	5.72
Image	8.45	24.53	16.49
Text × Image	16.44	29.92	23.18
Text + Image	9.60	26.13	17.87
InstructPix2Pix	9.83	20.02	14.92
FreeDom w/ i.-c.	**42.58**	19.16	30.87
FreeDom w/ c.	26.52	18.76	22.19
Pic2Word	17.86	24.67	21.27
CompoDiff	15.45	27.76	21.61
SEARLE (default)	13.48	24.33	18.90
SEARLE (tuned)	20.82	30.10	25.46
WeiCom	24.56	28.63	26.60
FreeDom	30.68	**35.50**	**33.09**

Table 4.1: *Domain conversion mAP (%)* on four datasets; comparison of FreeDom with baselines and competitors. For each source domain (e.g. Toy) average mAP over all target domains. AVG: average mAP over all source-target domain combinations; FreeDom w/ i.-c.: FreeDom with img-cap; FreeDom w/ c.: FreeDom with captioners.

Method	Cartoon		Origami		Toy		Sculpture		Avg	
	R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50
Text	0.15	0.95	0.87	3.73	0.71	1.77	0.36	1.89	0.52	2.09
Image	0.31	4.51	0.21	1.73	0.54	5.65	0.33	4.04	0.35	3.98
Text + Image	1.96	12.91	2.18	10.68	1.34	9.89	1.82	12.15	1.83	11.41
Pic2Word	8.00	21.90	13.50	25.60	8.70	21.60	10.00	23.80	10.05	23.23
Pic2Word (CC-3M)	7.35	18.53	12.79	25.54	10.39	22.96	10.24	23.76	10.19	22.70
Pic2Word (LAION 2B-en)	8.17	20.86	14.08	25.06	8.73	22.07	10.43	23.63	10.35	22.91
ARTEMIS w/ CompoDiff	11.42	23.81	15.49	25.44	11.21	24.01	10.84	21.07	12.24	23.58
CLIP4Cir w/ CompoDiff	10.90	24.12	16.08	25.60	11.01	23.57	10.45	21.86	12.11	23.79
CompoDiff (T5-XL)	8.43	20.40	15.73	25.69	11.19	22.48	9.19	18.45	11.14	21.76
CompoDiff (CLIP+T5-XL)	12.91	24.40	17.22	26.40	11.57	26.11	11.53	22.54	13.31	24.86
CompoDiff (CLIP)	13.21	24.06	17.03	26.17	11.22	26.25	11.24	22.96	13.18	24.86
KEDs	14.80	34.20	23.50	34.80	16.50	36.30	17.40	36.40	18.00	35.40
SEARLE (default)	1.49	12.38	3.78	13.88	1.99	15.34	2.18	15.34	2.36	14.24
SEARLE (tuned)	10.17	30.32	17.02	32.00	8.23	9.10	11.60	32.41	11.76	30.96
WeiCom	11.61	24.36	15.24	23.72	8.00	17.89	13.81	26.18	12.17	23.04
CIReVL	19.20	42.80	22.2	**43.10**	**30.20**	41.30	23.40	45.00	23.75	43.05
FreeDom	**23.77**	**48.80**	**32.86**	42.82	25.71	**47.47**	**27.87**	**48.96**	**27.55**	**47.01**

Table 4.2: *Domain conversion Recall@k (%)* on ImageNet-R. Comparison of FreeDom with baselines and competitors. Source domain: PHOTO; target domains: CARTOON, ORIGAMI, TOY, and SCULPTURE. AVG: average performance over all target domains. Top: baseline methods; middle: methods that require training. bottom: training-free methods. †: run by us.

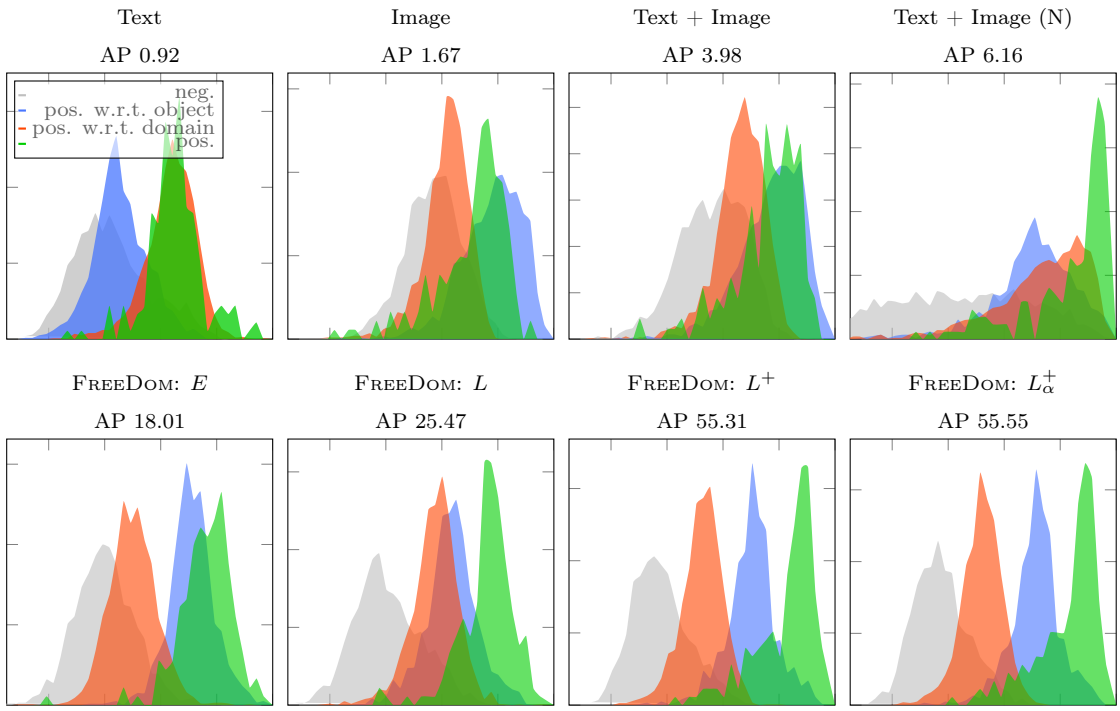


Figure 4.3: *Histogram of similarities* between a query and database images: **negative** (wrong object and domain); **positive** only w.r.t. the **object** (correct object, wrong domain); **positive** only w.r.t. the **domain** (wrong object, correct domain); **positive** (correct object and domain). E: early fusion; L: late fusion; L^+: late fusion with memory-based expansion; L^+_α: late fusion with memory-based expansion and histogram frequencies as weights; AP: average precision of the query. For better visualization, we sample an equal number of negatives, positives w.r.t. object and positives w.r.t. domain, while the values in the histogram of positives are multiplied by 10. Image query from MiniDomainNet; text query: "clipart".

	Avg					ImageNet-R					MiniDN					NICO++					LTLL				
k \ n	1	7	15	30	45	1	7	15	30	45	1	7	15	30	45	1	7	15	30	45	1	7	15	30	45
1	25.0	28.0	28.0	28.0	28.0	26.2	25.8	25.8	25.8	25.8	30.2	32.1	32.1	32.1	32.1	19.3	23.2	23.2	23.2	23.2	24.4	30.8	30.8	30.8	30.8
10	29.6	31.4	31.2	30.8	30.1	29.1	30.1	29.7	28.1	26.9	35.4	36.7	36.1	35.4	34.7	23.6	25.7	25.8	25.6	25.3	30.1	33.3	33.3	34.1	33.5
20	29.6	31.6	31.4	30.4	29.3	29.2	29.9	29.3	27.5	25.9	36.2	37.3	36.8	35.9	35.2	24.2	26.1	26.2	26.0	25.7	28.8	33.1	33.2	32.3	30.4
30	29.5	30.5	30.6	29.4	28.1	29.0	29.7	29.1	27.1	25.5	36.5	37.5	37.1	36.1	35.3	24.5	26.3	26.4	26.3	25.9	28.1	28.7	29.8	28.0	25.6
40	28.8	29.4	29.3	27.8	26.7	28.6	29.5	28.9	26.7	24.8	36.7	37.6	37.2	36.2	35.3	24.6	26.4	26.5	26.4	26.0	25.4	24.2	24.6	22.0	20.8
50	27.8	28.6	28.5	26.8	25.9	28.1	29.2	28.6	26.2	24.2	36.8	37.7	37.3	36.3	35.3	24.8	26.4	26.6	26.5	26.1	21.7	21.1	21.7	18.4	17.9

Table 4.3: *The effect of number k of proxy images vs. number n of labels for each proxy in* FreeDom, measured in domain conversion mAP. Values in the range $k \in \{10,20\}$ and $n \in \{7,15\}$ are better on average performance and competitive across datasets.

	Avg					ImageNet-R					MiniDn					NICO++					LTLL				
m	1	3	7	10	15	1	3	7	10	15	1	3	7	10	15	1	3	7	10	15	1	3	7	10	15
SRL	19.5	19.3	18.7	18.2	17.7	9.3	8.9	8.5	8.4	10.2	24.3	24.2	22.7	21.9	20.8	15.9	15.9	16.0	16.0	13.7	28.4	28.2	27.5	26.5	26.2
E	25.1	25.5	21.3	19.4	-	26.2	24.2	17.5	15.1	-	30.2	31.5	26.2	23.0	-	19.3	16.3	12.2	11.1	-	24.5	29.8	29.3	28.3	-
E^+	26.9	28.0	23.2	20.6	-	28.5	27.2	19.2	16.4	-	34.8	35.4	28.6	24.8	-	22.3	18.1	13.3	12.0	-	22.0	31.3	31.6	29.2	-
L	25.1	28.1	28.0	27.4	26.3	26.2	27.3	25.8	24.7	23.0	30.2	33.0	32.1	30.9	28.9	19.3	22.7	23.2	22.8	21.9	24.5	29.3	30.8	31.2	31.6
L^+	26.9	30.7	31.3	30.5	28.4	28.5	29.8	29.1	27.8	25.3	34.8	37.6	36.3	34.9	32.0	22.3	25.5	25.7	25.2	23.9	22.0	29.9	34.2	34.2	32.5
L^+_α	26.9	30.6	31.6	31.5	31.0	28.5	30.1	29.9	29.3	28.4	34.8	37.7	37.3	36.8	36.2	22.3	25.5	26.1	26.1	25.9	22.0	29.1	33.1	33.9	33.7

Table 4.4: *The impact of the number m of selected labels on each* FreeDom *component*, measured in domain conversion mAP. SRL: comparison to SEARLE (tuned) that uses m words from the corpus to guide the textual inversion optimization. Components: E: early fusion; L: late fusion; E^+: early fusion with memory-based expansion; L^+ late fusion with memory-based expansion; L^+_α: late fusion with memory-based expansion and frequency-based weighting.

Memory	Avg	ImageNet-R	MiniDn	NICO++	LTLL
No Memory	27.96	25.75	32.05	23.19	30.83
LAION 40M	28.57	25.00	33.85	24.31	31.11
Database + LAION 40M	29.52	26.07	34.92	24.91	32.17
Database	31.58	29.87	37.27	26.09	33.09

Table 4.5: *Impact of the visual memory:* performance comparison between no visual memory and a visual memory comprising 40 million LAION [255] images, or the database, or their union.

Impact of visual memory In Table 4.5 we show the performance for using different datasets as a visual memory. Compared to no visual memory at all, all options improve the performance on average, with an exception on ImageNet-R/LAION due to the low availability of images in specific domains such as *origami*. Therefore, the efficacy of the memory remains robust even when dealing with unstructured datasets such as LAION. Additionally, the inclusion of task-relevant images, even in small proportions, proves advantageous. The best improvements are achieved using the database as memory, which is our default choice.

4.4.7 Oracle Experiments

Information injection In the inversion problem, a common challenge arises from the source domain appearing within the text labels, which introduces conflicting domains into the query composition. Conversely, the correct query class or the source domain may not be found in the m text labels. To study the impact of each, we conduct an oracle experiment and summarize the results in Table 4.6. In the first two columns we compare early and late fusion with memory-based inversion after adding the query class to the m text labels. Late fusion achieves almost twice the gain compared to early fusion, showing that it can benefit more from the correct information. In the last two columns we compare the performance after including the source domain as a distractor. Late fusion suffers almost half the loss compared to early fusion, showing more robust behavior to incorrect information.

Sensitivity to the vocabulary We question whether FreeDom strongly depends on having the most appropriate word for describing the query object class in the vocabulary. To reflect that, we perform an oracle experiment where the name of the ground truth class of the query image is used to remove its ℓ nearest words from the vocabulary. After removal with $\ell = 5$, FreeDom performs 23.48, 31.95, 23.60 and 30.67 mAP on ImageNet-R, MiniDomainNet, NICO++, and LTLL, respectively. Even with the lack of the most appropriate words, FreeDom is still the best performing method.

Dataset	Object Gain		Domain Loss	
	E^+	L^+	E^+	L^+
ImageNet-R	+0.72	+1.16	-4.30	-1.64
NICO++	+0.21	+0.40	-0.66	-1.29
MiniDN	+0.72	+0.59	-7.43	-2.79
LTLL	+1.56	+4.00	-2.86	-1.58
Avg	+0.80	+1.54	-3.81	-1.83

Table 4.6: *Oracle experiment* to study the impact of inliers and outliers in the m text labels. Inlier represents the query object name and outlier is the source domain name. Gain or loss by adding inliers or outliers in text labels, respectively, is reported. Under memory-based expansion (E^+ and L^+), late fusion benefits from inliers and is more robust to outliers.

Performance upper-bound We use a single text label that is the name of the ground-truth object class for the image query. In this case, FreeDom achieves 46.58, 34.00, 46.06 and 31.18 mAP on ImageNet-R, MiniDomainNet, NICO++ and LTLL, respectively. This reference performance indicates there is still more space for improvement in the category-level benchmarks. However, this oracle experiment underperforms FreeDom on LTLL. This is no surprise as class names are not representative of the depicted object for this instance-level benchmark.

4.4.8 Beyond Domain Conversion Benchmarks

In this work we focus on the task of domain conversion, motivated by the significance of its applications. Addressing the challenges of this task, particularly the utilization of bi-modal queries, and the open-world recognition across domains and objects, prove to be non-trivial. Given that our method handles these challenges well and taking into account that these challenges extend universally to the generic composed image retrieval, we evaluate FreeDom on benchmarks of the generic task and present the results in Table 4.7.

Even though FreeDom is training-free, the results indicate that it is either the single best approach (CIRCO) or on par with another method (with SEARLE on CIRR) or below the best by a small margin (Fashion-IQ). Simple baselines perform comparably to all methods (Fashion-IQ), a little lower than other methods (CIRR) or a little higher than other methods (CIRCO). In conclusion, the improvements of all methods over the baselines on these tasks appear comparatively modest when compared to improvements of FreeDom on domain conversion. On CIRR and CIRCO, we evaluate on the validation set, which is a valid choice since no method performs training/validation on this set; the test set is behind an evaluation server.

Method	Fashion-IQ	CIRR	CIRCO
Text + Image (N)	6.77	18.84	10.61
Text × Image	10.58	24.42	10.92
Pic2Word	10.52	29.75	8.16
SEARLE	**10.70**	31.50	14.48
FreeDom (default)	9.67	28.96	13.06
FreeDom ($k = 1$)	10.04	30.60	**15.43**
FreeDom ($k = 1, n = 15, m = 15$)	10.30	**31.51**	13.41

Table 4.7: *Composed image retrieval beyond domain conversion:* performance (mAP) evaluation performed by us for all methods on generic composed retrieval benchmarks. FreeDom is evaluated for the default parameters and for additional setups found to be beneficial.

4.4.9 Visualizations

Figure 4.4 shows visualizations of the top-k ranked database images of FreeDom on ImageNet-R. We use Photo as source domain and convert to any target domain. FreeDom is able to retrieve correct images in all cases. Figure 4.5 shows visualizations of the top-k ranked database images of FreeDom on MiniDomainNet. We perform Sketch → Photo conversion, i.e. sketch-based image retrieval [244]–[246]. Interestingly, FreeDom is performing well in this task, in contrast to Pic2Word [39].

Furthermore, we present challenging cases where state-of-the-art methods underperform and we demonstrate the performance of FreeDom. Figure 4.6 shows visualizations of the top-k ranked database images of FreeDom *vs*. competitors on instance-level LTLL. We perform Archive → Today and Today → Archive domain conversions. We observe that competitors confuse both domains and instances. Figure 4.7 shows visualizations of the top-k ranked database images of FreeDom *vs*. competitors on NICO++. We perform Autumn → Dimlight and Grass → Autumn domain conversions. FreeDom has the best retrieval results, while competitors fail almost everywhere.

In our visual examples we excluded exact duplicates and we performed aspect ratio change for better presentation.

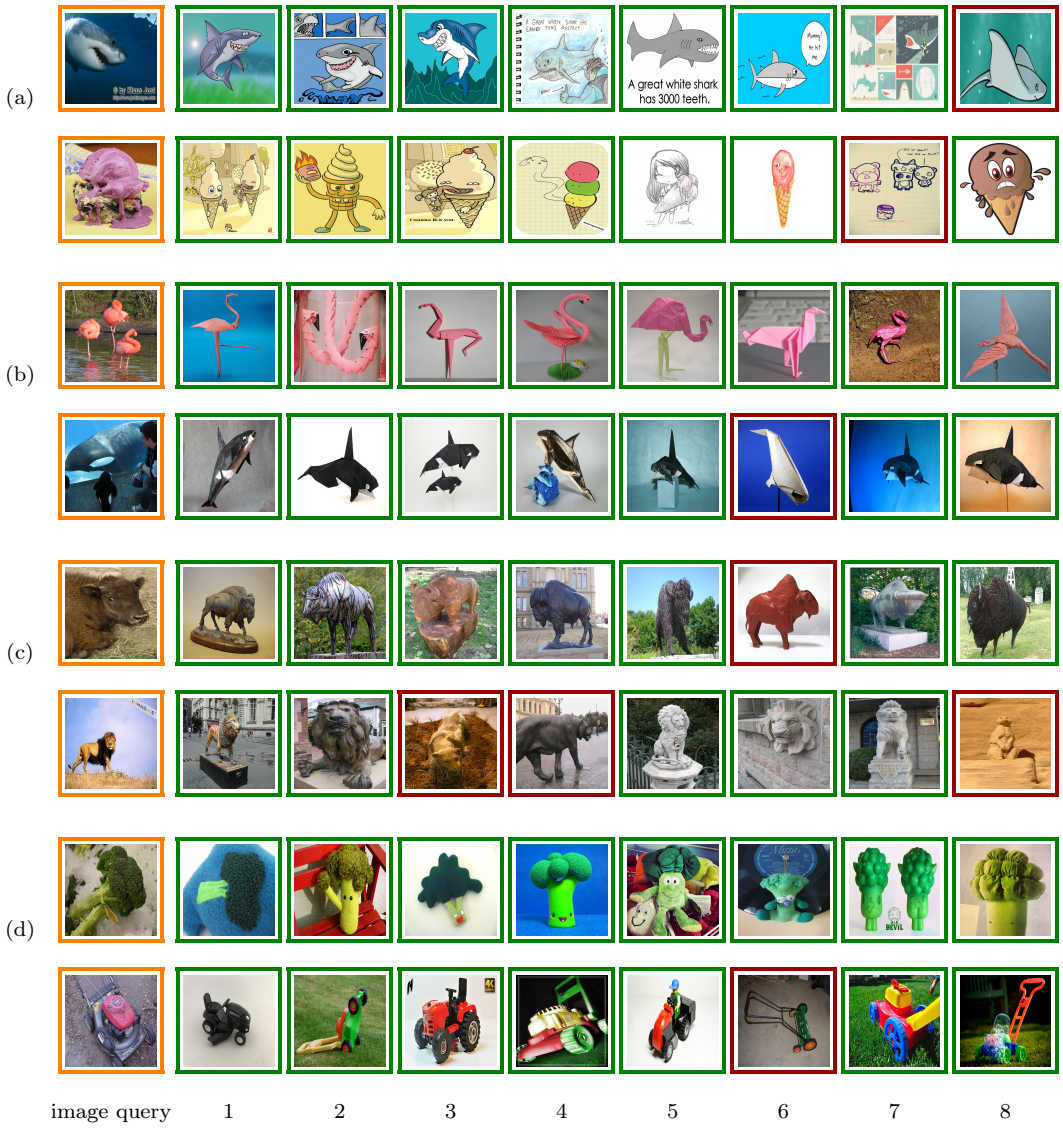


Figure 4.4: *top-k retrieval results of* FREEDOM. Domain conversion on ImageNet-R: (a) PHOTO → CARTOON; (b) PHOTO → ORIGAMI; (c) PHOTO → SCULPTURE; (d) PHOTO → TOY. **Orange**: image query, **green**: correctly retrieved; **red**: incorrectly retrieved; k=8.

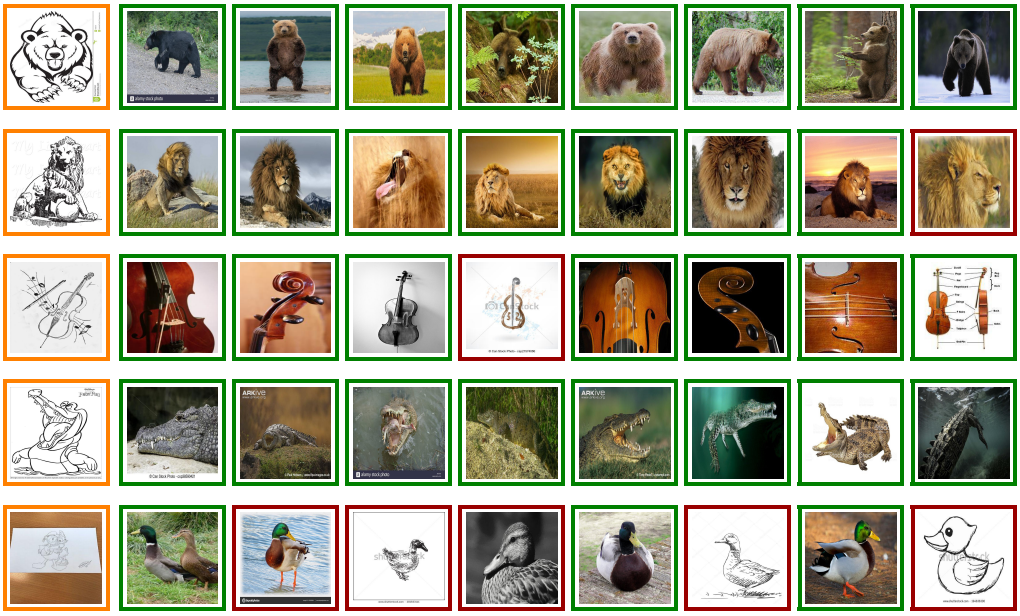


Figure 4.5: *top-k retrieval results of* FreeDom. Sketch-based image retrieval (Sketch → Photo) on MiniDomainNet. **Orange**: image query, **green**: correctly retrieved; **red**: incorrectly retrieved; k=8.

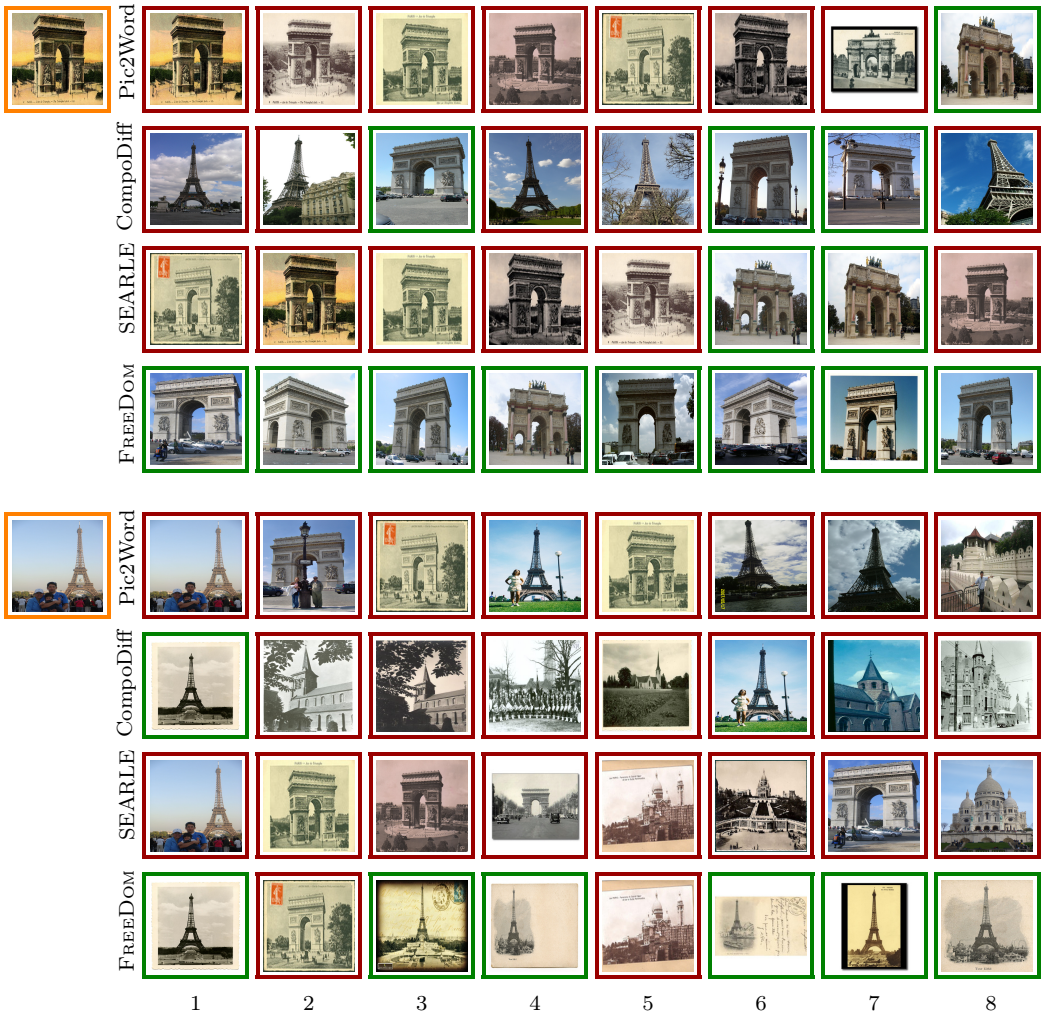


Figure 4.6: *top-k retrieval results.* Competitors vs. FreeDom. Domain conversion (Archive → Today, Today → Archive) on LTLL. **Orange**: image query; **green**: correctly retrieved; **red**: incorrectly retrieved; k=8.

Figure 4.7: *top-k retrieval results.* Competitors vs. FreeDom. Domain conversion (Autumn → Dimlight, Grass → Autumn) on NICO++. **Orange**: image query; **green**: correctly retrieved; **red**: incorrectly retrieved; k=8.

4.5 Conclusion

We have introduced FreeDom, a training-free composed image retrieval method for domain conversion, based on a pre-trained CLIP model. The key component is textual inversion of the query image based on soft assignment to a sparse vocabulary of words. Our detailed ablations show the importance of every component of the method as well as its robustness to the choice of hyper-parameters. We have also introduced three new benchmarks with different domain types, providing a broad testbed for further research in this area. Despite its zero requirements for supervision, data or training, FreeDom outperforms the state-of-the-art methods by a large margin on the task at hand. Composed image retrieval with free-form long sentences remains highly challenging and is still in its early stages. We claim that delving into a challenging sub-task, characterized by more restrictive textual queries as in domain conversion, is beneficial for further understanding and progress.

5

Extracting Multimodal Representations for Remote Sensing Composed Image Retrieval

Contents

5.1 Advancing Remote Sensing with Composed Image Retrieval: A New Era of Multimodal Search

In recent years, earth observation (EO) through remote sensing (RS) has witnessed an enormous growth in data volume, creating a challenge in managing and extracting relevant information. This surge is largely attributed to the proliferation of open satellite data programs, which have democratized access to EO data and broadened the scope of research and applications in various fields. The capacity to efficiently organize extensive archives and quickly *retrieve* specific images is crucial.

Remote sensing image retrieval (RSIR) [65], which aims to search and retrieve images from RS image archives, has emerged as a key solution. RSIR methods can be categorized into *unisource* and *cross-source* [257], where the categorization is based on whether the query image and the retrieved images are from the same source. In the case of unisource, there exists *single-label* [258]–[263] and *multi-label* [264]–[269] retrieval, depending on whether an image is associated with one or multiple labels respectively. In the case of cross-source, the term "source" is used loosely and can correspond to modality, view, etc.

In all cases, RSIR methods encounter a major limitation: the reliance on a query of single modality. This constraint often restricts users from fully expressing their specific requirements, especially given the complex and dynamic nature of Earth's surface as depicted in RS imagery. Ideally, users would benefit from a system that allows them to articulate nuanced modifications or specifications in conjunction with an image-based query. This is where composed image retrieval (CIR) [34]–[39] comes into play. CIR, integrating both image and text in the search query, is designed to retrieve images that are not only visually similar to the *query image* but also relevant to the details of the accompanying *query text*. By incorporating CIR into RS, we aim to offer a more expressive and flexible search capability that closely aligns with the intricate needs of users in this field. Figure 5.1 presents two examples of composed queries, each combining an image with text. We observe the new possibilities that this task unlocks, as the text component allows users to specify desired modifications in the image. Additionally, our method's flexibility facilitates seamless transitions between image and text, enabling a focus on either structural or modification aspects, respectively.

In this paper, we recognize, present and qualitatively evaluate the capabilities and

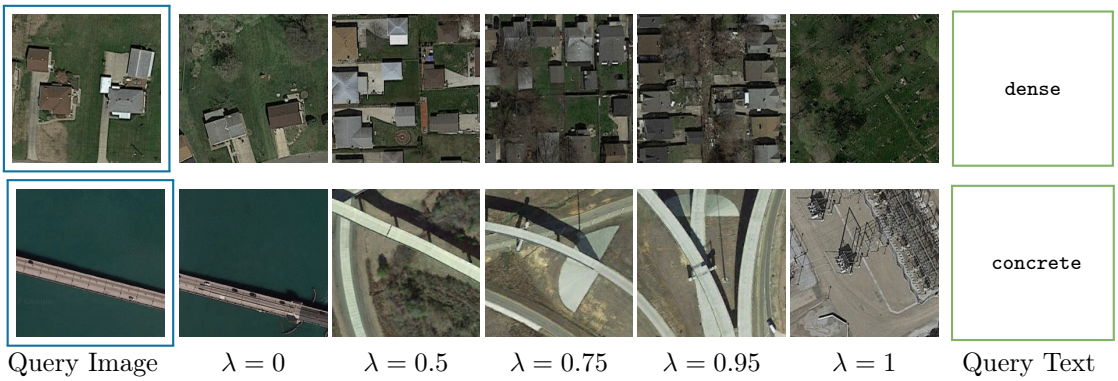


Figure 5.1: We introduce remote sensing composed image retrieval (RSCIR), a novel and expressive remote sensing image retrieval (RSIR) task integrating both image and text in the search query. We also introduce WeiCom, a flexible, training-free method based on vision-language models, utilizing a weighting parameter λ for more image- or text-oriented results, with $\lambda \to 0$ or $\lambda \to 1$ respectively. For each **query image** and **query text**, retrieved images shown for different λ.

challenges that CIR introduces within the RS domain. We demonstrate how users can now pair a query image with a query text specifying modifications related to *color*, *context*, *density*, *existence*, *quantity*, *shape*, *size* or *texture* of one or more classes. Quantitatively, we focus on color, context, density, existence, quantity, and shape modifications, establishing a benchmark and an evaluation protocol. Our approach is training-free by using a frozen vision-language model.

In summary, we make the following contributions:

1. We are the first to introduce composed image retrieval into remote sensing, accompanied with PatternCom, a benchmark dataset.
2. We introduce WeiCom, a training-free method utilizing a modality control parameter for more image- or text-oriented results according to the needs of each search.
3. We evaluate both qualitatively and quantitatively the performance of SimPool, setting the state-of-the-art on remote sensing composed image retrieval.

5.2 Exploring the Intersection of Remote Sensing and Composed Image Retrieval

Remote Sensing Image Retrieval With the aim to effectively *search* and *retrieve* information from extensive RS image archives, remote sensing image retrieval

(RSIR) can be categorized into *unisource* and *cross-source* [257]. Initially, RSIR methods focus on handcrafted and low-level visual features [270]–[279]. With the advent of deep learning, neural networks are utilized for unisource *single-label* retrieval: (a) as feature extractors [258], [280]–[288], (b) trained from scratch [201], [259], [260], [289]–[293], (c) integrating attention modules [261], [294]–[296] and (d) using metric learning [262], [263], [297]–[300]. Neural networks are also used for unisource *multi-label* [264]–[269], [278], cross-source *cross-sensors* [301]–[304], cross-source *cross-modal* [296], [305]–[310] and cross-source *cross-view* retrieval [311]–[316]. Our work fills a notable gap and enhances user intent expression in RSIR by combining query image with query text.

Composed Image Retrieval Image-to-image [40], [41], [198] and text-to-image [42]–[44] retrieval provide ways to explore large image archives. However, the most accurate and flexible way to express the user intent is a query *composed* of both an image and a text. Composed Image Retrieval (CIR) [34]–[39] aims to retrieve images not only visually similar to the query image, but also altered to align with the specifics of the query text. Traditionally, CIR methods are supervised by *triplets* of the form *query image, query text, target image* [34]–[36], [38], [45], [46], [208], [209]. The labor-intensive process of labeling such triplets limit early works to specific applications in fashion [47]–[49], physical states [50], object attributes and composition [34], [203], [204]. The emergence of vision-language models (VLMs) [52]–[54] led to their integration into CIR, introducing *zero-shot composed image retrieval* (ZS-CIR) [39], [51], [55]. This increases the spectrum of possible applications [51]. Methods are trained using unlabeled images [39], [51], or are not trained at all [55]. Recognizing the unexplored potential of CIR in RS, our work pioneers its introduction in this domain, particularly leveraging ZS-CIR empowered by VLMs.

Vision-Language Models The emergence of VLMs [52]–[54], [225] revolutionizes the field of multimodal learning. Trained on large-scale image-text datasets [217], these models map images and text into a shared embedding space. Apart from zero-shot classification, CLIP [52] can be used for detection [317], segmentation [318] and captioning [319]. CLIP can also be aligned to be used with medical data [320] or satellite data [321], [322]. In this work, we leverage CLIP and RemoteCLIP [321], a vision-language model for remote sensing, in a training-free setting.

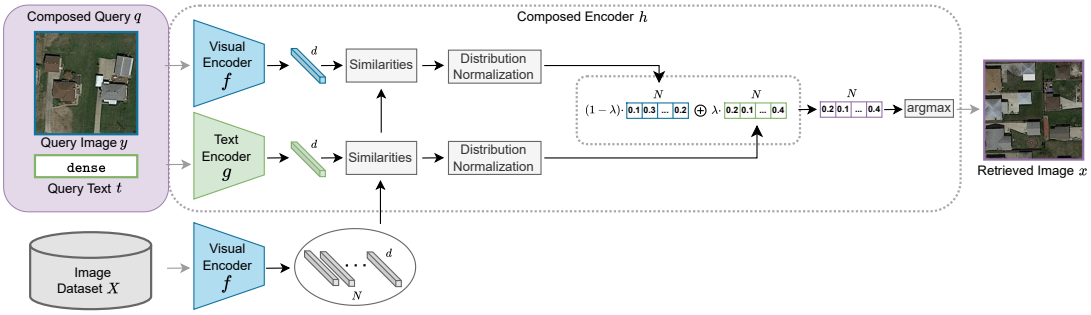


Figure 5.2: *WeiCom: A Weighted Composed Image Retrieval Method.* It utilizes a dual-encoder approach to process both **query image** y and **query text** t. Initially, the **query image** is passed into a visual encoder f and the **query text** into a text encoder g, producing corresponding d-dimensional representations. Subsequently, similarity scores with the representations in the image dataset are calculated. These scores are then normalized and combined using a convex combination controlled by a $\lambda \in [0, 1]$. Finally, an argmax(argsort) operation identifies the most relevant **retrieved image(s)** x.

5.3 WeiCom: A Modality-Control Method for Remote Sensing Composed Image Retrieval

5.3.1 Problem Formulation

In composed image retrieval, the goal is to retrieve images based on a *composed image-text query*, that is, a query that consists of a *visual* part, the query image, and a *textual* part, the query text. In this work, we introduce remote sensing composed image retrieval. To do so, we establish a benchmark and an evaluation protocol.

We denote the query image as y, its class as C_y and an attribute of the depicted class as A_y. We also denote the query text as t, which represents a modified target attribute A_t. We refer to the two queries as the composed query, $q = (y, t)$. Given an image dataset X, our goal is to retrieve images from X that share class with the query image class C_y and have the attribute A_t defined by the text query t. Retrieval aims to rank images $x \in X$ with respect to their composed similarity $s(q, x) \in R$ to the query. The task is extendable to multiple classes and multiple attributes.

To define s, we make use of pre-trained VLMs that consist of a *visual encoder* $f : \mathcal{I} \to \mathbb{R}^d$ and a *text encoder* $g : \mathcal{T} \to \mathbb{R}^d$, which map input images from image space $\mathcal{I}$ and words from the text space $\mathcal{T}$ to the same embedding space with dimension d. We extract the visual embedding $\mathbf{v}_y = f(y) \in \mathbb{R}^d$ and the text embedding

$\mathbf{v}_t = g(t) \in \mathbb{R}^d$ to use as queries. Finally, the embedding of a dataset image $x \in X$ is denoted as $\mathbf{v}_x = f(x) \in \mathbb{R}^d$. All embeddings are ℓ_2-normalized.

5.3.2 Baselines

Unimodal baselines rely solely on a single type of query to determine similarity. We denote: *text-only* by $s_g(q, x) = g(t)^T f(x)$ and *image-only* by $s_f(q, x) = f(y)^T f(x)$. Unimodal baselines are expected to fail since the final similarity cannot embody information from both image and text.

Multimodal combines the two unimodal approaches by averaging their similarities:

$$s_a(q, x) = \frac{s_g(q, x) + s_f(q, x)}{2} \tag{5.1}$$

Note that this baseline is equivalent to averaging the two features $g(t), f(y)$ and then calculating the similarities once. The drawback of this approach is that the features that come from same modalities have similarities significantly greater than the cross-modal similarities, making it an approach biased in favor of the image query.

5.3.3 WeiCom

In our proposed method, WeiCom, we estimate the similarities of the image query $s_f(q, x)$ and the text query $s_g(q, x)$ with the database. Then we perform similarity normalization in order to have a starting point of equal contribution from both modalities and we notate $s'_f(q, x), s'_g(q, x)$. Finally, we use the weighted average of the two similarity sets using a modality control parameter λ:

$$s_{WC}(q, x) = \lambda s'_g(q, x) + (1 - \lambda) s'_f(q, x) \tag{5.2}$$

Similarity Normalization In order to ensure that both image and text queries contribute equally to the retrieval, we normalize their similarities with the database. We first transform the empirical distribution of similarity scores into a standard normal distribution. Subsequently, we apply the cumulative distribution function (CDF) of the standard normal distribution to the standardized data, resulting in values that range between 0 and 1. Assuming the standardized data adhere to a normal distribution, this transformation yields data that approximates a uniform distribution. Transforming data into a uniform distribution diminishes the influence

of outliers and reduces skewness, smoothing any excessively peaked distributions. This approach leads to more robust similarity scores.

The modality control parameter λ After normalizing the similarities, we can control the influence of each modality using a parameter λ as a weight. Here $\lambda = 0$ refers to image-only retrieval, $\lambda = 1$ to text-only retrieval and $\lambda = 0.5$ to equal contribution of image and text. The full WeiCom method is summarized in Figure 5.2.

5.4 Benchmarking WeiCom: Performance and Insights

5.4.1 Datasets, Networks and Evaluation Protocol

Datasets To evaluate quantitatively the methods, we introduce PatternCom, a new benchmark based on PatternNet [323]. PatternNet is a large-scale high-resolution remote sensing image retrieval dataset. There are 38 classes and each class has 800 images of size 256×256 pixels. In PatternCom, we select some classes to be depicted in query images, and add a query text that defines an attribute relevant to that class. For instance, query images of "swimming pools" are combined with text queries defining "shape" as "rectangular", "oval", and "kidney-shaped". In total, PatternCom includes six attributes consisted of up to four different classes each. Each attribute can be associated with two to five values per class. The number of positives ranges from 2 to 1345 and there are more than 21k queries in total. Statistics for all attributes are shown in Table 5.1.

Networks We use the pre-trained CLIP [52] and RemoteCLIP [321], both with a ViT-L/14 image encoder.

Evaluation Protocol We evaluate using mAP. Average Precision (AP) is the average of the precision values obtained for the set of top-k results, up to each relevant item found in the ranking. The mAP is then the mean of these AP values over all queries.

5.4.2 Experimental Results

Qualitative results In Figure 5.3, we present the qualitative results of performing composed image retrieval in PatternCom using WeiCom with RemoteCLIP. Each

Attribute	Class	Value	#Positives	#Queries
color	airplane	white	672	53
		purple	53	672
	nursing home	white	85	383
		gray	383	85
	crosswalk	white	412	388
		yellow	388	412
	tennis court	blue	339	287
		brown	2	624
		gray	50	576
		green	211	415
		red	24	602
context	bridge	concrete	800	800
		water	800	800
density	residential	sparse	800	800
		dense	800	800
existence	parking	with cars	947	653
		without cars	653	947
	pier	with boats	1345	255
		without boats	255	1345
quantity	storage tank	one	356	261
		two	119	498
		three	65	552
		four	77	540
	wast. tr. plant	one	724	78
		two	44	758
		three	10	792
		four	24	778
	basketball court	one	340	383
		two	286	437
		three	21	702
		half	61	662
		two-halfs	15	708
shape	swimming pool	rectangular	261	299
		oval	52	508
		kidney-shaped	247	313
	river	curved	177	623
		straight	623	177
	road	cross	800	800
		round	800	800

Table 5.1: *Statistics of* PatternCom, the first remote sensing composed image retrieval benchmark.

example corresponds to one of the selected attributes with the query text specifying a modification in each attribute value.

Comparison with baselines As shown in Table 5.2, WeiCom outperfoms both unimodal ("Text", "Image") and multimodal ("Text & Image") baselines by a large margin. In particular, it outperforms the second best by 8.95% mAP using CLIP and 15.14% mAP using RemoteCLIP on average. Note that, as expected, the RS specialized RemoteCLIP performs better than the original CLIP on average.

(a) CLIP [52]

Method	Color	Context	Density	Existence	Quantity	Shape	Avg
Text	13.47	4.83	3.58	4.38	3.31	6.22	5.97
Image	14.66	8.32	13.49	13.50	7.84	15.76	12.26
Text & Image	23.13	11.02	15.87	**13.77**	10.13	21.38	15.88
WeiCom$_{\lambda=0.5}$	46.08	17.45	16.49	9.24	18.15	23.97	21.90
WeiCom$_{\lambda=0.3}$	**46.74**	**20.97**	**22.07**	12.07	**20.96**	**26.22**	**24.83**

(b) RemoteCLIP [321]

Method	Color	Context	Density	Existence	Quantity	Shape	Avg
Text	10.75	8.87	22.16	12.49	8.25	24.12	14.44
Image	14.40	6.62	15.11	9.29	6.99	15.18	11.27
Text & Image	23.67	10.01	18.45	10.56	7.97	19.63	15.05
WeiCom$_{\lambda=0.5}$	**43.68**	31.45	39.94	14.27	20.51	29.78	29.94
WeiCom$_{\lambda=0.6}$	41.04	**31.59**	**41.56**	**14.79**	**20.79**	**31.24**	**30.19**

Table 5.2: *Attribute modification mAP (%)* on PatternCom using CLIP (a) and RemoteCLIP (b); comparison of WeiCom with baselines. For each attribute value of an attribute (e.g. "rectangular" of Shape), average mAP over all the rest attribute values (e.g. "oval" of Shape). AVG: average mAP over all combinations.

5.4.3 Ablations

The impact of λ In Table 5.3 we show the impact of modality control parameter λ on WeiCom using RemoteCLIP. $\lambda = 0$ refers to image-only, $\lambda = 1$ to text-only retrieval. For $\lambda = 0.6$ we get the best average mAP, thus we set this as our method's default. The same study for CLIP gives $\lambda = 0.3$.

5.5 Conclusion

We introduce remote sensing composed image retrieval, a novel task integrating both image and text in the search query, accompanied with PatternCom, a benchmark

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Color	14.5	55.3	53.0	49.6	46.4	43.7	41.0	38.2	35.0	30.4	10.8
Context	6.6	13.3	20.2	25.7	29.5	31.5	31.6	29.6	24.8	16.9	8.9
Density	15.1	23.3	29.5	34.0	37.4	39.9	41.6	42.0	40.7	35.9	22.2
Existence	9.3	10.3	11.1	12.3	13.5	14.3	14.8	15.0	14.8	14.0	12.5
Quantity	7.0	17.6	18.9	19.7	20.2	20.5	20.8	20.9	20.8	20.1	8.3
Shape	15.2	23.8	24.7	26.2	28.0	29.8	31.2	32.0	32.0	31.3	24.1
Average	11.3	23.9	26.2	27.9	29.2	29.9	30.2	29.6	28.0	24.8	14.4

Table 5.3: *The effect of the modality control parameter λ on* WeiCom using RemoteCLIP, measured in attribute modification mAP.

dataset. We demonstrate its versatility through use cases modifying attributes like color or shape and also introduce WeiCom, a flexible and training-free method utilizing a modality control parameter λ, setting the state-of-the-art on the task.

Acknowledgements Bill was supported by the RAMONES H2020 project (grant: 101017808). This work was also supported by the HFRI under the BiCUBES project (grant: 03943), by the Czech Technical University in Prague grant No. SGS23/173/OHK3/3T/13, the Junior Star GACR GM 21-28830M, and the CTU institutional support (Future fund). NTUA thanks NVIDIA for the donation of GPU hardware.

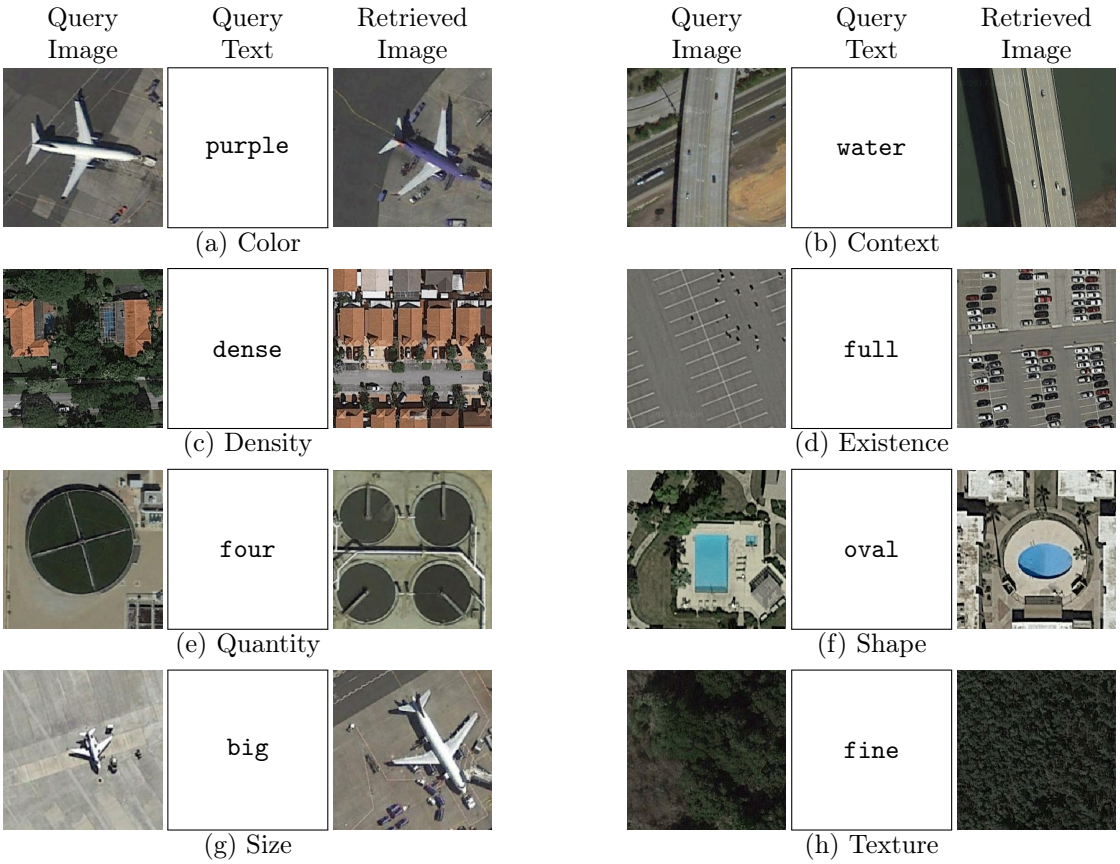


Figure 5.3: *Demonstrating remote sensing composed image retrieval.* Subfigures (a) to (h) depict key attributes of image composition: color, context, density, existence, quantity, shape, size, and texture. Each one illustrates various utilizations of composed image retrieval in remote sensing, demonstrating the wide range of applications and scenarios. Subfigures (b), (d) are examples that extend the task to multiple classes and attributes.

6

Conclusion

Contents

6.1 Summary

In this dissertation, we have explored novel methods for learning both visual and visual-textual (multimodal) representations, focusing on applications in deep metric learning, image classification, and composed image retrieval. Our research has been driven by the need to enhance the quality, robustness, and generalization of models through innovative approaches that address both data-centric and model-centric challenges.

The first part of the dissertation focused on visual representation learning. We introduced *Metrix*, a deep metric learning method utilizing mixup for data augmentation. This method addressed the challenge of interpolating both examples and target labels, overcoming the non-additive nature of traditional metric learning loss functions. Through extensive experiments on four benchmark datasets, we demonstrated that *Metrix* significantly improves robustness and generalization, setting a new state-of-the-art in deep metric learning.

In the second part, we shifted our focus to model architecture, introducing *SimPool*, a simple, attention-based pooling method for convolutional neural networks and vision transformers. We developed a generic pooling framework, allowing for the formulation and comparison of existing pooling methods. The proposed method was shown to generate high-quality attention maps, enhancing object localization and robustness to background changes. Our empirical studies validated its superior performance on standard benchmarks and downstream tasks.

The third part of the dissertation delved into visual-textual representations, starting with *FreeDom*, a training-free method for zero-shot composed image retrieval in open-world domain conversion. Leveraging the descriptive power of a frozen vision-language model and employing discrete-space textual inversion, *FreeDom* demonstrated superior performance across multiple benchmark datasets. This innovative approach highlighted the potential for further applications in generic composed image retrieval.

Finally, we introduced composed image retrieval into the domain of remote sensing with the novel task of remote sensing composed image retrieval. We presented *WeiCom*, a training-free method utilizing a modality control parameter, and established a new benchmark dataset, *PatternCom*. Our method's effectiveness was evaluated through various attribute modifications, showcasing its potential for enhancing search capabilities in remote sensing.

Throughout this dissertation, we have contributed to advancing the state-of-the-art in visual and multimodal representation learning. The methods developed and evaluated herein provide a robust foundation for future research and applications in these domains.

6.2 Future Work

While this dissertation has made significant strides in visual and multimodal representation learning, there are several avenues for future work that can build on our findings and further advance the field.

Advanced data augmentation In the context of deep metric learning, we have explored the use of mixup to enhance the robustness and generalization of visual representations. A promising extension of this work would be to develop a feature mixup method that leverages attention mechanisms to identify and align semantic correspondences between images before interpolation. For example, when interpolating between a cat and a dog, the method could align the eyes of the cat with the eyes of the dog, creating a more meaningful interpolation. This would involve

identifying corresponding features in the respective feature maps, potentially leading to more sophisticated and effective data augmentation strategies. Additionally, this approach could be combined with the extension of our existing work, *Metrix*, to ViTs. In the original work, *Metrix* was integrated into CNNs; however, ViTs, with their self-attention mechanisms, offer a new avenue for exploration. The interplay between self-attention and feature mixup could introduce new challenges, such as handling the global context that ViTs capture, and it could also offer novel insights into the behavior of interpolated features in a transformer-based architecture.

Multimodal data mixing for representation learning Another intriguing direction is the exploration of multimodal data mixing within the framework of representation learning. While mixup has shown promise in the image domain, its extension to multimodal data—such as combining visual and textual inputs—remains underexplored. A multimodal mixup could involve blending not only the pixel-level information from images but also the semantic information from associated text, thereby generating new multimodal training samples. This could be particularly useful in tasks that require a strong understanding of both visual and textual modalities, such as image captioning or VQA. Investigating how multimodal mixing influences the learned representations and how it can be optimized to improve the performance on downstream tasks would be a valuable contribution to the field.

Pooling as probing An interesting direction would be to employ an attention-pooling mechanism like *SimPool* in pre-trained and frozen models, thereby avoiding costly training. This approach could serve as an intermediate step between linear probing and full fine-tuning in self-supervised learning scenarios. By using *SimPool* before the classifier, we could train only the pooling layer and the classifier, thereby reducing the number of learnable parameters compared to full fine-tuning while still capturing more nuanced feature representations than linear probing. This approach, termed attentive probing, could effectively utilize the learned frozen features, potentially achieving competitive performance with a fraction of the computational cost. Future research could explore the balance between complexity and performance in various self-supervised learning tasks, using this method as a flexible and efficient alternative to existing approaches.

Exploring hierarchical and iterative pooling In our exploration of *SimPool*, we tested several extensions, including applying *SimPool* across different layers, using iterative pooling, and generating multiple local representations instead of a single global representation. While these variations performed on par with or worse than the original *SimPool*, they were also more computationally expensive. However, this outcome might be influenced by the characteristics of the ImageNet-1k

dataset, particularly its bias towards images containing a single dominant object. Given this, future research could revisit these extensions, either by further optimizing them within the ImageNet-1k framework or by testing them on datasets with more complex scenes that contain multiple objects. This exploration could help uncover whether these advanced pooling mechanisms can provide benefits in more challenging settings, where capturing multiple local features or iterative refinement might offer a significant advantage.

Textual inversion for image generation Textual inversion [324] has its origins in the task of personalized image generation, where the goal is to invert an image into a latent pseudo-word vector that can be used for instance-conditioned image generation. Extending this idea, a future direction could involve developing a discrete-space inversion method like FreeDom for image generation. The challenge here lies in describing specific, fine-grained attributes of an object (e.g., the exact appearance of a particular black dog) using a discrete set of tokens. One possible approach could involve inverting various properties of an object into distinct tokens that can be composed to form a detailed description in discrete space. This set of tokens could then serve as an initial conditioning for image generation, which might be further refined through optimization in continuous space. The system would start with a structured phrase describing the object's attributes and then iteratively optimize these tokens to achieve a more accurate and specific instance representation. This dual approach could bridge the gap between discrete and continuous representations, potentially improving the fidelity and controllability of instance-conditioned image generation.

Extending textual inversion to other modalities Another promising direction is to extend the concept of textual inversion beyond images and text to other modalities, such as video, audio, or 3D data. This could involve developing techniques to invert specific features from these modalities into a latent discrete space, enabling new forms of multimodal retrieval and generation. For example, in video, inversion could focus on capturing temporal dynamics or specific actions, while in 3D data, it could involve encoding shape or spatial structure. Exploring how discrete-space inversion can be adapted and optimized for these diverse data types could open up new applications and challenges in multimodal learning.

Advancing composed image retrieval through image-to-image search Traditional composed image retrieval methods typically invert an image into a textual token, which is then combined with a text query for text-to-image retrieval. A future direction worth exploring is to shift from text-to-image to image-to-image retrieval, maintaining the rich visual information inherent in the image modality.

This could be achieved by either generating a synthetic image that visually represents the composed query or by mixing the features of the image and text query directly in the feature space. The goal would be to create a composite feature vector that can be used for efficient image-to-image retrieval, potentially leading to more precise and visually coherent results. This approach would involve advanced feature manipulation techniques, such as attention-based feature blending, to ensure the accurate combination of visual attributes while preserving the integrity of the original image features.

Remote sensing composed image retrieval for change detection One of the most intriguing future directions for our new task is its application to change detection. In scenarios involving time series of high-resolution satellite images, users might query an image depicting a specific scene and use a textual query to specify a particular type of change—such as the appearance or disappearance of an object, or alterations in the scene's background. This approach could significantly enhance the ability to monitor and analyze environmental changes, urban development, or disaster impacts over time. By integrating RSCIR into change detection frameworks, we could develop more intuitive and powerful tools for temporal analysis in remote sensing, potentially leading to more timely and accurate insights into dynamic Earth processes.

Training new method on PatternCom and expanding the benchmark In our work, we introduced a new benchmark dataset, PatternCom, and developed the training-free *WeiCom* method, which was integrated into CLIP and RemoteCLIP. A natural extension of this research would involve designing and training a new method directly on PatternCom to fully explore its potential. This is aligned with our observation that *WeiCom* with RemoteCLIP outperforms its counterpart with the original CLIP. This method should work again on top of pre-trained CLIP or RemoteCLIP models, Additionally, further improvements might be achievable by training on an even larger dataset. Therefore, another direction could be to expand PatternCom or develop a new, larger benchmark that encompasses a wider variety of remote sensing scenes and attributes. This would not only provide a more comprehensive testbed for evaluating RSCIR methods but also push the boundaries of what is achievable with foundation models like CLIP and RemoteCLIP in the remote sensing domain.

Bibliography

[1] Y. Bengio, A. Courville, and P. Vincent, «Representation learning: A review and new perspectives», *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] J. Donahue, Y. Jia, O. Vinyals, *et al.*, «Decaf: A deep convolutional activation feature for generic visual recognition», in *International Conference on Machine Learning*, 2014.

[3] A. Kolesnikov, L. Beyer, X. Zhai, *et al.*, «Big transfer (bit): General visual representation learning», in *European Conference on Computer Vision*, 2020.

[4] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, «Deep clustering for unsupervised learning of visual features», in *European Conference on Computer Vision*, 2018.

[5] K. Sohn, D. Berthelot, N. Carlini, *et al.*, «Fixmatch: Simplifying semi-supervised learning with consistency and confidence», *Advances in Neural Information Processing Systems*, vol. 33, pp. 596–608, 2020.

[6] G. Hinton, O. Vinyals, and J. Dean, «Distilling the knowledge in a neural network», *arXiv preprint arXiv:1503.02531*, 2015.

[7] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, «Autoaugment: Learning augmentation strategies from data», in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.

[8] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, «Mixup: Beyond empirical risk minimization», in *International Conference on Learning Representations*, 2018.

[9] V. Verma, A. Lamb, C. Beckham, *et al.*, «Manifold mixup: Better representations by interpolating hidden states», in *International Conference on Machine Learning*, 2019.

[10] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, «Deep metric learning via lifted structured feature embedding», in *Conference on Computer Vision and Pattern Recognition*, 2016.

[11] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, «Deep image retrieval: Learning global representations for image search», in *European Conference on Computer Vision*, 2016.

[12] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, «Matching networks for one shot learning», *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, «Facenet: A unified embedding for face recognition and clustering», in *Conference on Computer Vision and Pattern Recognition*, 2015.

[14] N. Reimers and I. Gurevych, «Sentence-bert: Sentence embeddings using siamese bert-networks», in *Conference on Empirical Methods in Natural Language Processing*, 2019.

[15] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, «Distance metric learning with application to clustering with side-information», in *Advances in Neural Information Processing Systems*, 2003.

[16] B. Ko and G. Gu, «Embedding expansion: Augmentation in embedding space for deep metric learning», in *Conference on Computer Vision and Pattern Recognition*, 2020.

[17] G. Gu, B. Ko, and H.-G. Kim, «Proxy synthesis: Learning with synthetic classes for deep metric learning», in *Association for the Advancement of Artificial Intelligence*, 2021.

[18] J. G. Daugman, «Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters», *Journal of Optical Society of America*, vol. 2, no. 7, pp. 1160–1169, 1985.

[19] K. Fukushima, «Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position», *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[20] C. Szegedy, W. Liu, Y. Jia, *et al.*, «Going deeper with convolutions», in *Conference on Computer Vision and Pattern Recognition*, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, «Deep residual learning for image recognition», in *Conference on Computer Vision and Pattern Recognition*, 2016.

[22] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, «A convnet for the 2020s», in *Conference on Computer Vision and Pattern Recognition*, 2022.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, «An image is worth 16x16 words: Transformers for image recognition at scale», in *International Conference on Learning Representations*, 2021.

[24] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, «Attention is all you need», in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[25] Z. Liu, Y. Lin, Y. Cao, *et al.*, «Swin transformer: Hierarchical vision transformer using shifted windows», in *International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[26] Q. Zhang and Y.-B. Yang, «Rest: An efficient transformer for visual recognition», *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 475–15 485, 2021.

[27] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, «Scaling local self-attention for parameter efficient visual backbones», in *Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 894–12 904.

[28] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, «Do vision transformers see like convolutional neural networks?», *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 116–12 128, 2021.

[29] M. Caron, H. Touvron, I. Misra, *et al.*, «Emerging properties in self-supervised vision transformers», in *International Conference on Computer Vision*, 2021, pp. 9650–9660.

[30] E. Peruzzo, E. Sangineto, Y. Liu, *et al.*, «Spatial entropy as an inductive bias for vision transformers», *arXiv preprint arXiv:2206.04636*, 2022.

[31] M. M. Naseer, K. Ranasinghe, S. H. Khan, M. Hayat, F. Shahbaz Khan, and M.-H. Yang, «Intriguing properties of vision transformers», *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 296–23 308, 2021.

[32] S. Venkataramanan, A. Ghodrati, Y. M. Asano, F. Porikli, and A. Habibian, «Skip-attention: Improving vision transformers by paying less attention», in *International Conference on Learning Representations*, 2024.

[33] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, «Going deeper with image transformers», in *International Conference on Computer Vision*, 2021, pp. 32–42.

[34] N. Vo, L. Jiang, C. Sun, *et al.*, «Composing text and image for image retrieval-an empirical odyssey», in *Conference on Computer Vision and Pattern Recognition*, 2019.

[35] Y. Chen, S. Gong, and L. Bazzani, «Image search with text feedback by visiolinguistic attention learning», in *Conference on Computer Vision and Pattern Recognition*, 2020.

[36] M. Hosseinzadeh and Y. Wang, «Composed query image retrieval using locally bounded features», in *Conference on Computer Vision and Pattern Recognition*, 2020.

[37] A. Baldrati, M. Bertini, T. Uricchio, and A. Del Bimbo, «Effective conditioned and composed image retrieval combining clip-based features», in *Conference on Computer Vision and Pattern Recognition*, 2022.

[38] S. Lee, D. Kim, and B. Han, «Cosmo: Content-style modulation for image retrieval with text feedback», in *Conference on Computer Vision and Pattern Recognition*, 2021.

[39] K. Saito, K. Sohn, X. Zhang, *et al.*, «Pic2word: Mapping pictures to words for zero-shot composed image retrieval», in *Conference on Computer Vision and Pattern Recognition*, 2023.

[40] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, «Deep image retrieval: Learning global representations for image search», in *European Conference on Computer Vision*, 2016.

[41] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, «Large-scale image retrieval with attentive deep local features», in *International Conference on Computer Vision*, 2017.

[42] N. Sarafianos, X. Xu, and I. A. Kakadiaris, «Adversarial representation learning for text-to-image matching», in *International Conference on Computer Vision*, 2019, pp. 5814–5824.

[43] Q. Zhang, Z. Lei, Z. Zhang, and S. Z. Li, «Context-aware attention network for image-text retrieval», in *Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3536–3545.

[44] A. Frome, G. S. Corrado, J. Shlens, *et al.*, «Devise: A deep visual-semantic embedding model», *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[45] Y. Chen and L. Bazzani, «Learning joint visual semantic matching embeddings for language-guided retrieval», in *European Conference on Computer Vision*, 2020.

[46] M. Yin, Z. Yao, Y. Cao, *et al.*, «Disentangled non-local neural networks», in *European Conference on Computer Vision*, 2020.

[47] X. Han, Z. Wu, P. X. Huang, *et al.*, «Automatic spatially-aware fashion concept discovery», in *International Conference on Computer Vision*, 2017.

[48] T. L. Berg, A. C. Berg, and J. Shih, «Automatic attribute discovery and characterization from noisy web data», in *European Conference on Computer Vision*, Springer, 2010.

[49] H. Wu, Y. Gao, X. Guo, *et al.*, «Fashion iq: A new dataset towards retrieving images by natural language feedback», in *Conference on Computer Vision and Pattern Recognition*, 2021.

[50] P. Isola, J. J. Lim, and E. H. Adelson, «Discovering states and transformations in image collections», in *Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1383–1391.

[51] A. Baldrati, L. Agnolucci, M. Bertini, and A. Del Bimbo, «Zero-shot composed image retrieval with textual inversion», in *International Conference on Computer Vision*, 2023.

[52] A. Radford, J. W. Kim, C. Hallacy, *et al.*, «Learning transferable visual models from natural language supervision», in *International Conference on Machine Learning*, PMLR, 2021, pp. 8748–8763.

[53] C. Jia, Y. Yang, Y. Xia, *et al.*, «Scaling up visual and vision-language representation learning with noisy text supervision», in *International Conference on Machine Learning*, 2021.

[54] J. Li, D. Li, C. Xiong, and S. Hoi, «Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation», in *International Conference on Machine Learning*, 2022.

[55] S. Karthik, K. Roth, M. Mancini, and Z. Akata, «Vision-by-language for training-free compositional image retrieval», in *International Conference on Learning Representations*, 2023.

[56] T. Brown, B. Mann, N. Ryder, *et al.*, «Language models are few-shot learners», *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[57] X. Zhou, X. Han, H. Li, J. Wang, and X. Liang, «Cross-domain image retrieval: Methods and applications», *International Journal of Multimedia Information Retrieval*, 2022.

[58] J. Song, Q. Yu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, «Deep spatial-semantic attention for fine-grained sketch-based image retrieval», in *International Conference on Computer Vision*, 2017.

[59] J. Huang, R. S. Feris, Q. Chen, and S. Yan, «Cross-domain image retrieval with a dual attribute-aware ranking network», in *International Conference on Computer Vision*, 2015.

[60] C. Hu, Y. Yang, Y. Li, T. M. Hospedales, and Y.-Z. Song, «Towards unsupervised sketch-based image retrieval», *BMVC*, 2021.

[61] D. Kim, K. Saito, T.-H. Oh, B. A. Plummer, S. Sclaroff, and K. Saenko, «Cds: Cross-domain self-supervised pre-training», in *International Conference on Computer Vision*, 2021.

[62] X. Wang, D. Peng, M. Yan, and P. Hu, «Correspondence-free domain alignment for unsupervised cross-domain image retrieval», 2023.

[63] C. Hu and G. H. Lee, «Feature representation learning for unsupervised cross-domain image retrieval», in *European Conference on Computer Vision*, Springer, 2022.

[64] S. Paul, T. Dutta, and S. Biswas, «Universal cross-domain retrieval: Generalizing across classes and domains», in *International Conference on Computer Vision*, 2021.

[65] P. Agouris, J. Carswell, and A. Stefanidis, «An environment for content-based image retrieval from large spatial databases», *ISPRS Journal of Photogrammetry and Remote Sensing*, 1999.

[66] K. Musgrave, S. Belongie, and S.-N. Lim, «A metric learning reality check», in *European Conference on Computer Vision*, 2020.

[67] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krähenbühl, «Sampling matters in deep embedding learning», in *International Conference on Computer Vision*, 2017.

[68] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, «Contrastive learning with hard negative samples», *International Conference on Learning Representations*, 2021.

[69] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, «Multi-similarity loss with general pair weighting for deep metric learning», in *Conference on Computer Vision and Pattern Recognition*, 2019.

[70] T. Wang and P. Isola, «Understanding contrastive representation learning through alignment and uniformity on the hypersphere», in *International Conference on Machine Learning*, 2020.

[71] B. Kulis *et al.*, «Metric learning: A survey», *Foundations and trends in machine learning*, vol. 5, no. 4, pp. 287–364, 2012.

[72] R. Hadsell, S. Chopra, and Y. LeCun, «Dimensionality reduction by learning an invariant mapping», in *Conference on Computer Vision and Pattern Recognition*, 2006.

[73] J. Wang, Y. Song, T. Leung, *et al.*, «Learning fine-grained image similarity with deep ranking», in *Conference on Computer Vision and Pattern Recognition*, 2014.

[74] K. Q. Weinberger and L. K. Saul, «Distance metric learning for large margin nearest neighbor classification.», *Journal of Machine Learning Research*, 2009.

[75] A. Hermans, L. Beyer, and B. Leibe, «In defense of the triplet loss for person re-identification», *arXiv preprint arXiv:1703.07737*, 2017.

[76] W. Chen, X. Chen, J. Zhang, and K. Huang, «Beyond triplet loss: A deep quadruplet network for person re-identification», in *Conference on Computer Vision and Pattern Recognition*, 2017.

[77] K. Sohn, «Improved deep metric learning with multi-class n-pair loss objective», in *Advances in Neural Information Processing Systems*, 2016.

[78] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, «No fuss distance metric learning using proxies», in *International Conference on Computer Vision*, 2017.

[79] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin, «Softtriple loss: Deep metric learning without triplet sampling», in *International Conference on Computer Vision*, 2019.

[80] S. Kim, D. Kim, M. Cho, and S. Kwak, «Proxy anchor loss for deep metric learning», in *Conference on Computer Vision and Pattern Recognition*, 2020.

[81] E. W. Teh, T. DeVries, and G. W. Taylor, «Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis», in *European Conference on Computer Vision*, 2020.

[82] Y. Zhu, M. Yang, C. Deng, and W. Liu, «Fewer is more: A deep graph metric learning perspective using fewer proxies», *Advances in Neural Information Processing Systems*, 2020.

[83] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, «Cutmix: Regularization strategy to train strong classifiers with localizable features», in *International Conference on Computer Vision*, 2019.

[84] J.-H. Kim, W. Choo, and H. O. Song, «Puzzle mix: Exploiting saliency and local statistics for optimal mixup», in *International Conference on Machine Learning*, 2020.

[85] J.-H. Kim, W. Choo, H. Jeong, and H. O. Song, «Co-mixup: Saliency guided joint mixup with supermodular diversity», in *International Conference on Learning Representations*, 2021.

[86] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, «Augmix: A simple data processing method to improve robustness and uncertainty», *International Conference on Learning Representations*, 2020.

[87] T. DeVries and G. W. Taylor, «Improved regularization of convolutional neural networks with cutout», *arXiv preprint arXiv:1708.04552*, 2017.

[88] J. Qin, J. Fang, Q. Zhang, W. Liu, X. Wang, and X. Wang, «Resizemix: Mixing data with preserved object information and true labels», *arXiv preprint arXiv:2012.11101*, 2020.

[89] A. F. M. S. Uddin, S. M. Mst., W. Shin, T. Chung, and S.-H. Bae, «Saliencymix: A saliency guided data augmentation strategy for better regularization», in *International Conference on Learning Representations*, 2021.

[90] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow, «Understanding and improving interpolation in autoencoders via an adversarial regularizer», *arXiv preprint arXiv:1807.07543*, 2018.

[91] X. Liu, Y. Zou, L. Kong, *et al.*, «Data augmentation via latent space interpolation for image classification», in *International Conference on Pattern Recognition*, 2018.

[92] C. Beckham, S. Honari, V. Verma, *et al.*, «On adversarial mixup resynthesis», *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[93] J. Zhu, L. Shi, J. Yan, and H. Zha, «Automix: Mixup networks for sample interpolation via cooperative barycenter learning», in *European Conference on Computer Vision*, 2020.

[94] S. Venkataramanan, E. Kijak, L. Amsaleg, and Y. Avrithis, «Alignmixup: Improving representations by interpolating aligned features», in *Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 174–19 183.

[95] W. Zheng, Z. Chen, J. Lu, and J. Zhou, «Hardness-aware deep metric learning», in *Conference on Computer Vision and Pattern Recognition*, 2019.

[96] G. Gu and B. Ko, «Symmetrical synthesis for deep metric learning», in *Association for the Advancement of Artificial Intelligence*, 2020.

[97] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, «Hard negative mixing for contrastive learning», *Advances in Neural Information Processing Systems*, 2020.

[98] K. Lee, Y. Zhu, K. Sohn, C.-L. Li, J. Shin, and H. Lee, «I-mix: A domain-agnostic strategy for contrastive representation learning», in *International Conference on Learning Representations*, 2021.

[99] S. Kim, G. Lee, S. Bae, and S.-Y. Yun, «Mixco: Mix-up contrastive learning for visual representation», *Advances in Neural Information Processing Systems Workshop*, 2020.

[100] D. Yi, Z. Lei, and S. Z. Li, «Deep metric learning for practical person re-identification», *arXiv preprint arXiv:1703.07737*, 2014.

[101] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, «Neighbourhood components analysis», in *Advances in Neural Information Processing Systems*, 2005.

[102] H. Xuan, A. Stylianou, and R. Pless, «Improved embeddings with easy positive triplet mining», in *Winter Conference on Applications of Computer Vision*, 2020.

[103] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, «The Caltech-UCSD Birds-200-2011 Dataset», California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[104] J. Krause, M. Stark, J. Deng, and F.-F. Li, «3d object representations for fine-grained categorization», *International Conference on Computer Vision Workshop*, 2013.

[105] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, «Deepfashion: Powering robust clothes recognition and retrieval with rich annotations», in *Conference on Computer Vision and Pattern Recognition*, 2016.

[106] A. Zhai and H.-Y. Wu, «Classification is a strong baseline for deep metric learning», *arXiv preprint arXiv:1811.12649*, 2018.

[107] O. Russakovsky, J. Deng, H. Su, *et al.*, «Imagenet large scale visual recognition challenge», *International Journal of Computer Vision*, 2015.

[108] A. Sanakoyeu, V. Tschernezki, U. Buchler, and B. Ommer, «Divide and conquer the embedding space for metric learning», in *Conference on Computer Vision and Pattern Recognition*, 2019.

[109] I. Loshchilov and F. Hutter, «Decoupled weight decay regularization», in *International Conference on Learning Representations*, 2018.

[110] K. He, X. Zhang, S. Ren, and J. Sun, «Deep residual learning for image recognition», in *Conference on Computer Vision and Pattern Recognition*, 2016.

[111] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, «Bert: Pre-training of deep bidirectional transformers for language understanding», *arXiv preprint arXiv:1810.04805*, 2018.

[112] A. Babenko and V. Lempitsky, «Aggregating local deep features for image retrieval», in *International Conference on Computer Vision*, 2015.

[113] F. Radenović, G. Tolias, and O. Chum, «Fine-Tuning CNN Image Retrieval with No Human Annotation», IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[114] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, «Imagenet: A large-scale hierarchical image database», in *Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[115] G. Tolias, R. Sicre, and H. Jégou, «Particular object retrieval with integral max-pooling of CNN activations», in *International Conference on Learning Representations*, 2016.

[116] P. O. Pinheiro and R. Collobert, «From image-level to pixel-level labeling with convolutional networks», in *Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1713–1721.

[117] G. Tolias, T. Jenicek, and O. Chum, «Learning and aggregating deep local descriptors for instance-level recognition», in *European Conference on Computer Vision*, 2020.

[118] G. Mialon, D. Chen, A. d'Aspremont, and J. Mairal, «A trainable optimal transport embedding for feature aggregation and its relationship to attention», *arXiv preprint arXiv:2006.12065*, 2020.

[119] F. Locatello, D. Weissenborn, T. Unterthiner, *et al.*, «Object-centric learning with slot attention», *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[120] J. Hu, L. Shen, and G. Sun, «Squeeze-and-excitation networks», in *Conference on Computer Vision and Pattern Recognition*, 2018.

[121] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, «Cbam: Convolutional block attention module», in *European Conference on Computer Vision*, 2018, pp. 3–19.

[122] J. G. Daugman, «Two-dimensional spectral analysis of cortical receptive field profiles», *Vision Research*, vol. 20, no. 10, pp. 847–856, 1980.

[123] A. Oliva and A. Torralba, «Building the gist of a scene: The role of global image features in recognition», *Visual Perception*, 2006.

[124] D. G. Lowe, «Distinctive image features from scale-invariant keypoints», *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[125] N. Dalal and B. Triggs, «Histograms of oriented gradients for human detection», in *Computer Vision and Pattern Recognition*, vol. 1, 2005.

[126] J. G. Daugman, «Complete discrete 2-d gabor transforms by neural networks for image analysis and compression», *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, 1988.

[127] M. R. Turner, «Texture discrimination by gabor functions», *Biological Cybernetics*, vol. 55, no. 2-3, pp. 71–82, 1986.

[128] J. Malik, S. Belongie, J. Shi, and T. Leung, «Textons, contours and regions: Cue integration in image segmentation», in *International Conference on Computer Vision*, vol. 2, 1999, p. 918.

[129] J. Sivic and A. Zisserman, «Video google: A text retrieval approach to object matching in videos», in *International Conference Computer Vision*, vol. 2, 2003.

[130] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, «Visual categorization with bags of keypoints», in *ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[131] Y.-L. Boureau, F. Bach, Y. Lecun, and J. Ponce, «Learning mid-level features for recognition», in *Computer Vision and Pattern Recognition*, 2010.

[132] F. Perronnin and C. Dance, «Fisher kernels on visual vocabularies for image categorization», in *Computer Vision and Pattern Recognition*, 2006.

[133] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, «Aggregating local image descriptors into compact codes», IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[134] L. Bo and C. Sminchisescu, «Efficient match kernel between sets of features for visual recognition», in *Advances in Neural Information Processing Systems 22*, 2009.

[135] K. He, X. Zhang, S. Ren, and J. Sun, «Spatial pyramid pooling in deep convolutional networks for visual recognition», IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[136] Y. Jia, C. Huang, and T. Darrell, «Beyond spatial pyramids: Receptive field learning for pooled image features», in *Computer Vision and Pattern Recognition*, 2012.

[137] S. Sukhbaatar, T. Makino, and K. Aihara, «Auto-pooling: Learning to improve invariance of image features from image sequences», *arXiv preprint arXiv:1301.3323*, 2013.

[138] H. Jegou, M. Douze, and C. Schmid, «On the burstiness of visual elements», in *Computer Vision and Pattern Recognition*, 2009.

[139] N. Murray and F. Perronnin, «Generalized max pooling», in *Computer Vision and Pattern Recognition*, 2014.

[140] H. Jegou and A. Zisserman, «Triangulation embedding and democratic aggregation for image search», in *Computer Vision and Pattern Recognition*, 2014.

[141] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, «Deep sets», in *Advances in Neural Information Processing Systems*, 2017.

[142] T. Serre, L. Wolf, and T. Poggio, «Object recognition with features inspired by visual cortex», in *Computer Vision and Pattern Recognition*, 2005.

[143] Y.-L. Boureau, J. Ponce, and Y. LeCun, «A theoretical analysis of feature pooling in visual recognition», in *International Conference on Machine Learning*, 2010, pp. 111–118.

[144] H. Jegou, M. Douze, and C. Schmid, «Hamming embedding and weak geometric consistency for large scale image search», in *European Conference on Computer Vision*, 2008.

[145] G. Tolias, Y. Avrithis, and H. Jégou, «To aggregate or not to aggregate: Selective match kernels for image search», in *International Conference on Computer Vision*, 2013, pp. 1401–1408.

[146] Y. LeCun, B. Boser, J. Denker, *et al.*, «Handwritten digit recognition with a back-propagation network», *Advances in Neural Information Processing Systems*, 1989.

[147] A. Krizhevsky, I. Sutskever, and G. E. Hinton, «Imagenet classification with deep convolutional neural networks», in *Advances in Neural Information Processing Systems 25*, 2012.

[148] K. Simonyan and A. Zisserman, «Very deep convolutional networks for large-scale image recognition», in *International Conference on Learning Representations*, 2015.

[149] M. Lin, Q. Chen, and S. Yan, «Network in network», *arXiv preprint arXiv:1312.4400*, 2013.

[150] R. Girshick, «Fast R-CNN», in *International Conference on Computer Vision*, 2015.

[151] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, «Learning deep features for discriminative localization», in *Computer Vision and Pattern Recognition*, 2016.

[152] Y. Kalantidis, C. Mellina, and S. Osindero, «Cross-dimensional weighting for aggregated deep convolutional features», in *European Conference on Computer Vision Workshops*, 2016.

[153] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, «Large-scale image retrieval with attentive deep local features», in *International Conference on Computer Vision*, 2017, pp. 3456–3465.

[154] B. Cao, A. Araujo, and J. Sim, «Unifying deep local and global features for image search», in *European Conference on Computer Vision*, Springer, 2020, pp. 726–743.

[155] T. Ng, V. Balntas, Y. Tian, and K. Mikolajczyk, «Solar: Second-order loss and attention for image retrieval», in *European Conference on Computer Vision*, Springer, 2020, pp. 253–270.

[156] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, «Gather-excite: Exploiting feature context in convolutional neural networks», *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[157] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, «Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations», in *Conference on Computer Vision and Pattern Recognition*, 2017.

[158] X. Wang, R. Girshick, A. Gupta, and K. He, «Non-local neural networks», in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[159] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, «Attention augmented convolutional networks», in *International Conference on Computer Vision*, 2019, pp. 3286–3295.

[160] H. Zhao, J. Jia, and V. Koltun, «Exploring self-attention for image recognition», in *Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 076–10 085.

[161] T. Plötz and S. Roth, «Neural nearest neighbors networks», *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[162] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, «Aˆ 2-nets: Double attention networks», *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[163] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, «ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks», in *Conference on Computer Vision and Pattern Recognition*, 2020.

[164] K. Wu, H. Peng, M. Chen, J. Fu, and H. Chao, «Rethinking and improving relative position encoding for vision transformer», in *International Conference on Computer Vision*, 2021.

[165] B. Graham, A. El-Nouby, H. Touvron, *et al.*, «LeViT: A vision transformer in convnet's clothing for faster inference», in *International Conference on Computer Vision*, 2021.

[166] H. Wu, B. Xiao, N. Codella, *et al.*, «CvT: Introducing convolutions to vision transformers», in *International Conference on Computer Vision*, 2021.

[167] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, «Convit: Improving vision transformers with soft convolutional inductive biases», in *International Conference on Machine Learning*, 2021, pp. 2286–2296.

[168] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, «Rethinking spatial dimensions of vision transformers», in *International Conference on Computer Vision*, 2021.

[169] W. Wang, E. Xie, X. Li, *et al.*, «Pyramid vision transformer: A versatile backbone for dense prediction without convolutions», in *International Conference on Computer Vision*, 2021, pp. 568–578.

[170] W. Wang, E. Xie, X. Li, *et al.*, «Pvt v2: Improved baselines with pyramid vision transformer», *Computational Visual Media*, vol. 8, no. 3, pp. 415–424, 2022.

[171] W. Yu, M. Luo, P. Zhou, *et al.*, «Metaformer is actually what you need for vision», in *Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 819–10 829.

[172] S.-i. Amari, «Integration of stochastic models by minimizing α-divergence», *Neural Computation*, vol. 19, no. 10, pp. 2780–2796, 2007.

[173] S. Lloyd, «Least squares quantization in pcm», *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[174] P. A. Knight, «The Sinkhorn-Knopp algorithm: Convergence and applications», *SIAM Journal on Matrix Analysis and Applications*, 2008.

[175] M. Cuturi, «Sinkhorn distances: Lightspeed computation of optimal transport», *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[176] J. L. Ba, J. R. Kiros, and G. E. Hinton, «Layer normalization», *arXiv preprint arXiv:1607.06450*, 2016.

[177] K. Cho, B. van Merrienboer, Ç. Gülçehre, *et al.*, «Learning phrase representations using RNN encoder-decoder for statistical machine translation», in *Empirical Methods in Natural Language Processing*, 2014.

[178] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, «Unsupervised feature learning via non-parametric instance discrimination», in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3733–3742.

[179] A. Krizhevsky, G. Hinton, *et al.*, «Learning multiple layers of features from tiny images», 2009.

[180] M.-E. Nilsback and A. Zisserman, «Automated flower classification over a large number of classes», in *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

[181] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata, and H. Shim, «Evaluating weakly supervised object localization methods right», in *Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3133–3142.

[182] O. Siméoni, G. Puy, H. V. Vo, *et al.*, «Localizing objects with self-supervised transformers and no labels», in *British Machine Vision Conference*, 2021.
[183] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, «The pascal visual object classes (voc) challenge», *International Journal of Computer Vision*, vol. 88, pp. 303–308, 2009.
[184] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, «Microsoft coco: Common objects in context», in *European Conference on Computer Vision*, Springer, 2014, pp. 740–755.
[185] T. Deselaers, B. Alexe, and V. Ferrari, «Localizing objects while learning their appearance», in *European Conference on Computer Vision*, Springer, 2010, pp. 452–466.
[186] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, «Scene parsing through ade20k dataset», in *Conference on Computer Vision and Pattern Recognition*, 2017, pp. 633–641.
[187] K. Xiao, L. Engstrom, A. Ilyas, and A. Madry, «Noise or signal: The role of image backgrounds in object recognition», in *International Conference on Learning Representations*, 2021.
[188] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, «Revisiting oxford and paris: Large-scale image retrieval benchmarking», in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5706–5715.
[189] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, «Object retrieval with large vocabularies and fast spatial matching», in *Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
[190] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, «Lost in quantization: Improving particular object retrieval in large scale image databases», in *Conference on Computer Vision and Pattern Recognition*, 2008.
[191] *Flickr*, https://www.flickr.com.
[192] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, «Deepfashion: Powering robust clothes recognition and retrieval with rich annotations», in *Conference on Computer Vision and Pattern Recognition*, 2016.
[193] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, «Deep metric learning via lifted structured feature embedding», in *Conference on Computer Vision and Pattern Recognition*, 2016.
[194] C.-Y. Lee, P. W. Gallagher, and Z. Tu, «Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree», in *Artificial Intelligence and Statistics*, PMLR, 2016, pp. 464–472.
[195] J. Zhou, C. Wei, H. Wang, *et al.*, «Ibot: Image bert pre-training with online tokenizer», in *International Conference on Learning Representations*, 2022.
[196] K. Musgrave, S. Belongie, and S.-N. Lim, «A metric learning reality check», in *European Conference on Computer Vision*, 2020.

[197] I. Kakogeorgiou, S. Gidaris, B. Psomas, *et al.*, «What to hide from your students: Attention-guided masked image modeling», in *European Conference on Computer Vision*, 2022.

[198] F. Radenović, G. Tolias, and O. Chum, «Fine-Tuning CNN Image Retrieval with No Human Annotation», IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[199] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, «Deepfashion: Powering robust clothes recognition and retrieval with rich annotations», in *Conference on Computer Vision and Pattern Recognition*, 2016.

[200] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, «Arcface: Additive angular margin loss for deep face recognition», in *Conference on Computer Vision and Pattern Recognition*, 2019.

[201] U. Chaudhuri, B. Banerjee, and A. Bhattacharya, «Siamese graph convolutional network for content based remote sensing image retrieval», *CVIU*, 2019.

[202] L. R. Nair, K. Subramaniam, and G. Prasannavenkatesan, «A review on multiple approaches to medical image retrieval system», *Intelligent Computing in Engineering*, 2019.

[203] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, «Microsoft coco: Common objects in context», in *European Conference on Computer Vision*, 2014.

[204] A. Neculai, Y. Chen, and Z. Akata, «Probabilistic compositional embeddings for multimodal image retrieval», in *Conference on Computer Vision and Pattern Recognition*, 2022.

[205] D. Hendrycks, S. Basart, N. Mu, *et al.*, «The many faces of robustness: A critical analysis of out-of-distribution generalization», in *International Conference on Computer Vision*, 2021.

[206] X. Zhang, Y. He, R. Xu, H. Yu, Z. Shen, and P. Cui, «Nico++: Towards better benchmarking for domain generalization», in *Conference on Computer Vision and Pattern Recognition*, 2023.

[207] B. Fernando, T. Tommasi, and T. Tuytelaars, «Location recognition over large time lags», *Computer Vision and Image Understanding*, vol. 139, pp. 21–28, 2015.

[208] S. Ren, K. He, R. Girshick, and J. Sun, «Faster r-cnn: Towards real-time object detection with region proposal networks», *Advances in Neural Information Processing Systems*, 2015.

[209] G. Delmas, R. S. de Rezende, G. Csurka, and D. Larlus, «Artemis: Attention-based retrieval with text-explicit matching and implicit similarity», in *International Conference on Learning Representations*, 2022.

[210] X. Jiang, Y. Wang, Y. Wu, M. Wang, and X. Qian, «Dual relation alignment for composed image retrieval», *arXiv preprint arXiv:2309.02169*, 2023.

[211] R. Bommasani, D. A. Hudson, E. Adeli, *et al.*, «On the opportunities and risks of foundation models», *arXiv preprint arXiv:2108.07258*, 2021.

[212] Z. Liu, C. Rodriguez-Opazo, D. Teney, and S. Gould, «Image retrieval on real-life images with pre-trained vision-and-language models», in *International Conference on Computer Vision*, 2021.

[213] S. Goenka, Z. Zheng, A. Jaiswal, *et al.*, «Fashionvlp: Vision language transformer for fashion retrieval with feedback», in *Conference on Computer Vision and Pattern Recognition*, 2022.

[214] X. Li, X. Yin, C. Li, *et al.*, «Oscar: Object-semantics aligned pre-training for vision-language tasks», in *European Conference on Computer Vision*, 2020.

[215] P. Zhang, X. Li, X. Hu, *et al.*, «Vinvl: Revisiting visual representations in vision-language models», in *Conference on Computer Vision and Pattern Recognition*, 2021.

[216] Z. Liu, W. Sun, Y. Hong, D. Teney, and S. Gould, «Bi-directional training for composed image retrieval via text prompt learning», in *Winter Conference on Applications of Computer Vision*, 2024, pp. 5753–5762.

[217] C. Schuhmann, R. Beaumont, R. Vencu, *et al.*, «Laion-5b: An open large-scale dataset for training next generation image-text models», *Advances in Neural Information Processing Systems*, 2022.

[218] Y. Liu, J. Yao, Y. Zhang, Y. Wang, and W. Xie, «Zero-shot composed text-image retrieval», *arXiv preprint arXiv:2306.07272*, 2023.

[219] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, «Making the v in vqa matter: Elevating the role of image understanding in visual question answering», in *Conference on Computer Vision and Pattern Recognition*, 2017.

[220] M. Levy, R. Ben-Ari, N. Darshan, and D. Lischinski, «Data roaming and early fusion for composed image retrieval», *arXiv preprint arXiv:2303.09429*, 2023.

[221] G. Gu, S. Chun, W. Kim, H. Jun, Y. Kang, and S. Yun, «Compodiff: Versatile composed image retrieval with latent diffusion», *arXiv preprint arXiv:2303.11916*, 2023.

[222] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, «High-resolution image synthesis with latent diffusion models», in *Conference on Computer Vision and Pattern Recognition*, 2022.

[223] Y. Bai, X. Xu, Y. Liu, *et al.*, «Sentence-level prompts benefit composed image retrieval», *arXiv preprint arXiv:2310.05473*, 2023.

[224] Y. Du, M. Wang, W. Zhou, S. Hui, and H. Li, «Image2sentence based asymmetrical zero-shot composed image retrieval», in *International Conference on Learning Representations*, 2023.

[225] J. Li, D. Li, S. Savarese, and S. Hoi, «Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models», in *International Conference on Machine Learning*, PMLR, 2023, pp. 19 730–19 742.

[226] C. Zhou, C. C. Loy, and B. Dai, «Extract free dense labels from clip», in *European Conference on Computer Vision*, Springer, 2022.

[227] M. Wysoczańska, M. Ramamonjisoa, T. Trzciński, and O. Siméoni, «Clip-diy: Clip dense inference yields open-vocabulary semantic segmentation for-free», in *Winter Conference on Applications of Computer Vision*, 2024.

[228] T. Lu, X. Zhang, J. Gu, *et al.*, «Fuse your latents: Video editing with multi-source latent diffusion models», *arXiv preprint arXiv:2310.16400*, 2023.

[229] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, «Prompt-to-prompt image editing with cross attention control», *arXiv preprint arXiv:2208.01626*, 2022.

[230] M. Chen, I. Laina, and A. Vedaldi, «Training-free layout control with cross-attention guidance», *arXiv preprint arXiv:2304.03373*, 2023.

[231] T. Reiss, B. Cavia, and Y. Hoshen, «Detecting deepfakes without seeing any», *arXiv preprint arXiv:2311.01458*, 2023.

[232] S. Lu, Y. Liu, and A. W.-K. Kong, «Tf-icon: Diffusion-based training-free cross-domain image composition», in *International Conference on Computer Vision*, 2023.

[233] J. Li, G. Shakhnarovich, and R. A. Yeh, «Adapting clip for phrase localization without further training», *arXiv preprint arXiv:2204.03647*, 2022.

[234] A. Fuentes and J. M. Saavedra, «Sketch-qnet: A quadruplet convnet for color sketch-based image retrieval», in *Conference on Computer Vision and Pattern Recognition*, 2021.

[235] R. He, X. Wu, Z. Sun, and T. Tan, «Wasserstein cnn: Learning invariant features for nir-vis face recognition», IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[236] R. Hu and J. Collomosse, «A performance evaluation of gradient field hog descriptor for sketch based image retrieval», *CVIU*, 2013.

[237] Y. Li, T. M. Hospedales, Y.-Z. Song, and S. Gong, «Fine-grained sketch-based image retrieval by matching deformable part models», in *BMVC*, 2014.

[238] J. M. Saavedra, «Sketch based image retrieval using a soft computation of the histogram of edge local orientations (s-helo)», in *2014 IEEE international conference on image processing (ICIP)*, IEEE, 2014, pp. 2998–3002.

[239] J. M. Saavedra, J. M. Barrios, and S. Orand, «Sketch based image retrieval using learned keyshapes (lks).», in *BMVC*, vol. 1, 2015, p. 7.

[240] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, «The sketchy database: Learning to retrieve badly drawn bunnies», *TOG*, 2016.

[241] Q. Yu, F. Liu, Y.-Z. Song, T. Xiang, T. M. Hospedales, and C.-C. Loy, «Sketch me that shoe», in *Conference on Computer Vision and Pattern Recognition*, 2016.

[242] X. Ji, W. Wang, M. Zhang, and Y. Yang, «Cross-domain image retrieval with attention modeling», in *ACM Multimedia*, 2017.

[243] S. Ibrahimi, N. van Noord, Z. Geradts, and M. Worring, «Deep metric learning for cross-domain fashion instance retrieval», in *ICCVW*, 2019.

[244] S. Dey, P. Riba, A. Dutta, J. Llados, and Y.-Z. Song, «Doodle to search: Practical zero-shot sketch-based image retrieval», in *Conference on Computer Vision and Pattern Recognition*, 2019.

[245] A. Dutta and Z. Akata, «Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval», in *Conference on Computer Vision and Pattern Recognition*, 2019.

[246] S. K. Yelamarthi, S. K. Reddy, A. Mishra, and A. Mittal, «A zero-shot framework for sketch based image retrieval», in *European Conference on Computer Vision*, 2018.

[247] A. Kuznetsova, H. Rom, N. Alldrin, *et al.*, «The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale», *International Journal of Computer Vision*, 2020.

[248] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, «Domain generalization with mixstyle», *International Conference on Learning Representations*, 2021.

[249] W. Dong, R. Socher, L. Li-Jia, K. Li, and L. Fei-Fei, «ImageNet: A large-scale hierarchical image database», in *Conference on Computer Vision and Pattern Recognition*, 2009.

[250] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, «Moment matching for multi-source domain adaptation», in *International Conference on Computer Vision*, 2019.

[251] B. Fernando, T. Tommasi, and T. Tuytelaars, «Location recognition over large time lags», *CVIU*, vol. 139, pp. 21–28, 2015.

[252] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, «An image is worth 16x16 words: Transformers for image recognition at scale», in *International Conference on Learning Representations*, 2021.

[253] N. Cohen, R. Gal, E. A. Meirom, G. Chechik, and Y. Atzmon, «"this is my unicorn, fluffy": Personalizing frozen vision-language representations», in *European Conference on Computer Vision*, 2022.

[254] T. Brooks, A. Holynski, and A. A. Efros, «Instructpix2pix: Learning to follow image editing instructions», in *CVPR*, 2023, pp. 18 392–18 402.

[255] C. Schuhmann, R. Vencu, R. Beaumont, *et al.*, «Laion-400m: Open dataset of clip-filtered 400 million image-text pairs», *NeurIPS Workshop*, 2021.

[256] B. Psomas, I. Kakogeorgiou, N. Efthymiadis, *et al.*, «Composed image retrieval for remote sensing», in IEEE *International Geoscience and Remote Sensing Symposium*, 2024.

[257] W. Zhou, H. Guan, Z. Li, Z. Shao, and M. R. Delavar, «Remote sensing image retrieval in the past decade: Achievements, challenges, and future directions», *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2023.

[258] D. Hou, Z. Miao, H. Xing, and H. Wu, «Exploiting low dimensional features from the mobilenets for remote sensing image retrieval», *Earth Science Informatics*, vol. 13, pp. 1437–1443, 2020.

[259] Y. Wang, S. Ji, and Y. Zhang, «A learnable joint spatial and spectral transformation for high resolution remote sensing image retrieval», *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 8100–8112, 2021.

[260] G. Sumbul and B. Demir, «Plasticity-stability preserving multi-task learning for remote sensing image retrieval», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[261] S. Wang, D. Hou, and H. Xing, «A novel multi-attention fusion network with dilated convolution and label smoothing for remote sensing image retrieval», *International Journal of Remote Sensing*, vol. 43, no. 4, pp. 1306–1322, 2022.

[262] H. Zhao, L. Yuan, H. Zhao, and Z. Wang, «Global-aware ranking deep metric learning for remote sensing image retrieval», *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[263] Q. Cheng, D. Gan, P. Fu, H. Huang, and Y. Zhou, «A novel ensemble architecture of residual attention-based deep metric learning for remote sensing image retrieval», *Remote Sensing*, vol. 13, no. 17, p. 3445, 2021.

[264] J. Kang, R. Fernandez-Beltran, D. Hong, J. Chanussot, and A. Plaza, «Graph relation network: Modeling relations between scenes for multilabel remote-sensing image classification and retrieval», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4355–4369, 2020.

[265] G. Sumbul and B. Demir, «A novel graph-theoretic deep representation learning method for multi-label remote sensing image retrieval», in *IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2021, pp. 266–269.

[266] G. Sumbul, M. Ravanbakhsh, and B. Demir, «Informative and representative triplet selection for multilabel remote sensing image retrieval», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2021.

[267] Q. Cheng, H. Huang, L. Ye, P. Fu, D. Gan, and Y. Zhou, «A semantic-preserving deep hashing model for multi-label remote sensing image retrieval», *Remote Sensing*, vol. 13, no. 24, p. 4965, 2021.

[268] R. Imbriaco, C. Sebastian, E. Bondarev, and P. H. de With, «Toward multi-label image retrieval for remote sensing», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[269] Z. Shao, W. Zhou, X. Deng, M. Zhang, and Q. Cheng, «Multilabel remote sensing image retrieval based on fully convolutional network», *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 318–328, 2020.

[270] Y. Mamatha and A. Ananth, «Content based image retrieval of satellite imageries using soft query based color composite techniques», *International Journal of Computer Applications*, vol. 7, no. 5, pp. 0975–8887, 2010.

[271] C. Ma, Q. Dai, J. Liu, S. Liu, and J. Yang, «An improved svm model for relevance feedback in remote sensing image retrieval», *International Journal of Digital Earth*, vol. 7, no. 9, pp. 725–745, 2014.

[272] J. A. Piedra-Fernandez, G. Ortega, J. Z. Wang, and M. Canton-Garbin, «Fuzzy content-based image retrieval for oceanic remote sensing», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 52, no. 9, pp. 5422–5431, 2013.

[273] J. Li and R. M. Narayanan, «Integrated spectral and spatial information mining in remote sensing imagery», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 42, no. 3, pp. 673–685, 2004.

[274] S. Bhagavathy and B. S. Manjunath, «Modeling and detection of geospatial objects using texture motifs», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 44, no. 12, pp. 3706–3715, 2006.

[275] M. Wang and T. Song, «Remote sensing image retrieval by scene semantic matching», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 51, no. 5, pp. 2874–2886, 2012.

[276] Z. Shao, W. Zhou, and Q. Cheng, «Remote sensing image retrieval with combined features of salient region», *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, pp. 83–88, 2014.

[277] M. Wang, Q. Wan, L. Gu, and T. Song, «Remote-sensing image retrieval by combining image visual and semantic features», *International Journal of Remote Sensing*, vol. 34, no. 12, pp. 4200–4223, 2013.

[278] B. Chaudhuri, B. Demir, S. Chaudhuri, and L. Bruzzone, «Multilabel remote sensing image retrieval using a semisupervised graph-theoretic method», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 1144–1158, 2017.

[279] O. E. Dai, B. Demir, B. Sankur, and L. Bruzzone, «A novel system for content based retrieval of multi-label remote sensing images», in *IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2017, pp. 1744–1747.

[280] Y. Li, Y. Zhang, C. Tao, and H. Zhu, «Content-based high-resolution remote sensing image retrieval via unsupervised feature learning and collaborative affinity metric fusion», *Remote Sensing*, vol. 8, no. 9, p. 709, 2016.

[281] F. Hu, X. Tong, G.-S. Xia, and L. Zhang, «Delving into deep representations for remote sensing image retrieval», in *International Conference on Signal Processing*, IEEE, 2016, pp. 198–203.

[282] Y. Boualleg and M. Farah, «Enhanced interactive remote sensing image retrieval with scene classification convolutional neural networks model», in *IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2018, pp. 4748–4751.

[283] F. Ye, H. Xiao, X. Zhao, M. Dong, W. Luo, and W. Min, «Remote sensing image retrieval using convolutional neural network features and weighted distance», *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 10, pp. 1535–1539, 2018.

[284] P. Napoletano, «Visual descriptors for content-based retrieval of remote-sensing images», *International Journal of Remote Sensing*, vol. 39, no. 5, pp. 1343–1376, 2018.

[285] Y. Ge, S. Jiang, Q. Xu, C. Jiang, and F. Ye, «Exploiting representations from pre-trained convolutional neural networks for high-resolution remote sensing image retrieval», *Multimedia Tools and Applications*, vol. 77, pp. 17 489–17 515, 2018.

[286] X. Tang, X. Zhang, F. Liu, and L. Jiao, «Unsupervised deep feature learning for remote sensing image retrieval», *Remote Sensing*, vol. 10, no. 8, p. 1243, 2018.

[287] R. Imbriaco, C. Sebastian, E. Bondarev, and P. H. de With, «Aggregated deep local features for remote sensing image retrieval», *Remote Sensing*, vol. 11, no. 5, p. 493, 2019.

[288] P. Sadeghi-Tehran, P. Angelov, N. Virlet, and M. J. Hawkesford, «Scalable database indexing and fast image retrieval based on deep learning and hierarchically nested structure applied to remote sensing and plant biology», *Journal of Imaging*, vol. 5, no. 3, p. 33, 2019.

[289] W. Zhou, S. Newsam, C. Li, and Z. Shao, «Learning low dimensional convolutional neural networks for high-resolution remote sensing image retrieval», *Remote Sensing*, vol. 9, no. 5, p. 489, 2017.

[290] M. Zhang, Q. Cheng, F. Luo, and L. Ye, «A triplet nonlocal neural network with dual-anchor triplet loss for high-resolution remote sensing image retrieval», *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2711–2723, 2021.

[291] Z. Zhuo and Z. Zhou, «Remote sensing image retrieval with gabor-ca-resnet and split-based deep feature transform network», *Remote Sensing*, vol. 13, no. 5, p. 869, 2021.

[292] Y. Liu, Y. Liu, C. Chen, and L. Ding, «Remote-sensing image retrieval with tree-triplet-classification networks», *Neurocomputing*, vol. 405, pp. 48–61, 2020.

[293] Y. Wang, L. Zhang, X. Tong, *et al.*, «A three-layered graph-based learning approach for remote sensing image retrieval», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6020–6034, 2016.

[294] Y. Wang, S. Ji, M. Lu, and Y. Zhang, «Attention boosted bilinear pooling for remote sensing image retrieval», *International Journal of Remote Sensing*, vol. 41, no. 7, pp. 2704–2724, 2020.

[295] W. Xiong, Y. Lv, Y. Cui, X. Zhang, and X. Gu, «A discriminative feature learning approach for remote sensing image retrieval», *Remote Sensing*, vol. 11, no. 3, p. 281, 2019.

[296] U. Chaudhuri, B. Banerjee, A. Bhattacharya, and M. Datcu, «Attention-driven graph convolution network for remote sensing image retrieval», *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[297] R. Cao, Q. Zhang, J. Zhu, *et al.*, «Enhancing remote sensing image retrieval using a triplet deep metric learning network», *International Journal of Remote Sensing*, vol. 41, no. 2, pp. 740–751, 2020.

[298] Y. Liu, Z. Han, C. Chen, L. Ding, and Y. Liu, «Eagle-eyed multitask cnns for aerial image retrieval and scene classification», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 58, no. 9, pp. 6699–6721, 2020.

[299] L. Fan, H. Zhao, and H. Zhao, «Global optimization: Combining local loss with result ranking loss in remote sensing image retrieval», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 7011–7026, 2020.

[300] Y. Liu, L. Ding, C. Chen, and Y. Liu, «Similarity-based unsupervised deep transfer learning for remote sensing image retrieval», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 58, no. 11, pp. 7872–7889, 2020.

[301] Y. Li, Y. Zhang, X. Huang, and J. Ma, «Learning source-invariant deep hashing convolutional neural networks for cross-source remote sensing image retrieval», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 56, no. 11, pp. 6521–6536, 2018.

[302] J. Ma, D. Shi, X. Tang, X. Zhang, X. Han, and L. Jiao, «Cross-source image retrieval based on ensemble learning and knowledge distillation for remote sensing images», in *IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2021, pp. 2803–2806.

[303] W. Xiong, Z. Xiong, Y. Cui, and Y. Lv, «A discriminative distillation network for cross-source remote sensing image retrieval», *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1234–1247, 2020.

[304] W. Xiong, Y. Lv, X. Zhang, and Y. Cui, «Learning to translate for cross-source remote sensing image retrieval», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4860–4874, 2020.

[305] F. Xu, W. Yang, T. Jiang, S. Lin, H. Luo, and G.-S. Xia, «Mental retrieval of remote sensing images via adversarial sketch-image feature learning», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 58, no. 11, pp. 7801–7814, 2020.

[306] Y. Sun, S. Feng, Y. Ye, *et al.*, «Multisensor fusion and explicit semantic preserving-based deep hashing for cross-modal remote sensing image retrieval», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[307] G. Sumbul, A. De Wall, T. Kreuziger, *et al.*, «Bigearthnet-mm: A large-scale, multimodal, multilabel benchmark archive for remote sensing image classification and retrieval [software and data sets]», *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 3, pp. 174–180, 2021.

[308] Y. Lv, W. Xiong, X. Zhang, and Y. Cui, «Fusion-based correlation learning model for cross-modal remote sensing image retrieval», *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[309] Z. Yuan, W. Zhang, C. Tian, *et al.*, «Remote sensing cross-modal text-image retrieval based on global and local information», IEEE *Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[310] Z. Yuan, W. Zhang, K. Fu, *et al.*, «Exploring a fine-grained multiscale method for cross-modal remote sensing image retrieval», *arXiv preprint arXiv:2204.09868*, 2022.

[311] S. Hu, M. Feng, R. M. Nguyen, and G. H. Lee, «Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization», in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7258–7267.

[312] Z. Zeng, Z. Wang, F. Yang, and S. Satoh, «Geo-localization via ground-to-satellite cross-view image retrieval», *IEEE Transactions on Multimedia*, 2022.

[313] Y. Tian, X. Deng, Y. Zhu, and S. Newsam, «Cross-time and orientation-invariant overhead image geolocalization using deep local features», in *Winter Conference on Applications of Computer Vision*, 2020, pp. 2512–2520.

[314] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, «Learning deep representations for ground-to-aerial geolocalization», in *Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5007–5015.

[315] N. Khurshid, T. Hanif, M. Tharani, and M. Taj, «Cross-view image retrieval-ground to aerial image retrieval through deep learning», in *International Conference on Neural Information Processing*, Springer, 2019, pp. 210–221.

[316] Y. Shi and H. Li, «Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image», in *Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 010–17 020.

[317] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, «Open-vocabulary object detection via vision and language knowledge distillation», in *International Conference on Learning Representations*, 2021.

[318] F. Liang, B. Wu, X. Dai, *et al.*, «Open-vocabulary semantic segmentation with mask-adapted clip», in *Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7061–7070.

[319] Y. Tewel, Y. Shalev, I. Schwartz, and L. Wolf, «Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic», in *Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 918–17 928.

[320] Z. Wang, Z. Wu, D. Agarwal, and J. Sun, «Medclip: Contrastive learning from unpaired medical images and text», *arXiv preprint arXiv:2210.10163*, 2022.

[321] F. Liu, D. Chen, Z. Guan, *et al.*, «Remoteclip: A vision language foundation model for remote sensing», *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[322] K. Klemmer, E. Rolf, C. Robinson, L. Mackey, and M. Rußwurm, «Satclip: Global, general-purpose location embeddings with satellite imagery», *arXiv preprint arXiv:2311.17179*, 2023.

[323] W. Zhou, S. Newsam, C. Li, and Z. Shao, «Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval», *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, pp. 197–209, 2018.

[324] R. Gal, Y. Alaluf, Y. Atzmon, *et al.*, «An image is worth one word: Personalizing text-to-image generation using textual inversion», in *International Conference on Learning Representations*.