GLOBAL AVERAGE PRECISION FOR REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

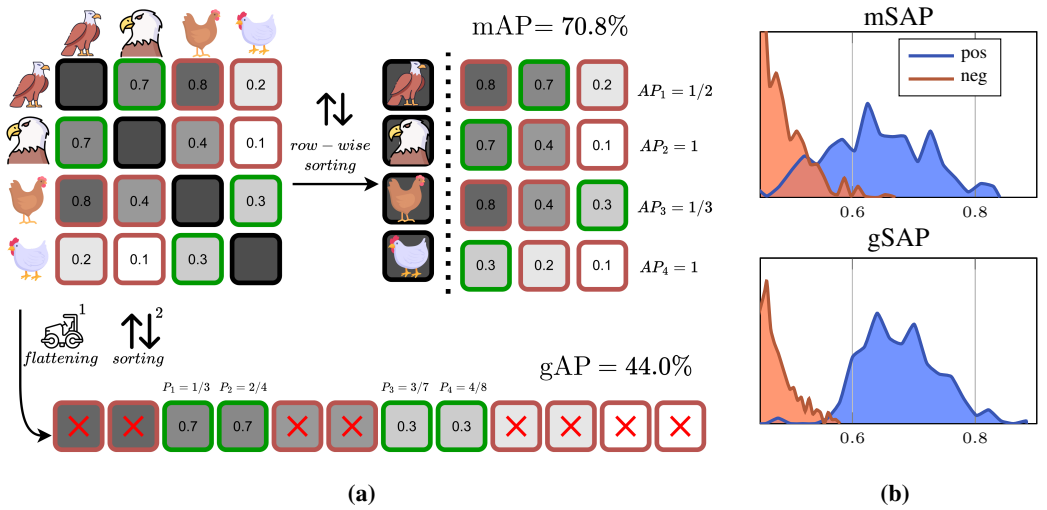
Standard information retrieval metrics, such as mean Average Precision (mAP), assess performance one query at a time, based on how the similarities between a query and its positives compare against those with its negatives. The same holds for common representation learning losses, such as InfoNCE and per-query AP surrogates. None of them considers whether similarities are comparable *across* queries, which any system with a single decision threshold relies on. Global Average Precision (gAP) does, by ranking all query–candidate pairs in one list and computing a single AP. We introduce gSAP, a differentiable surrogate of gAP. It needs only a similarity matrix and a binary matrix marking the positive pairs, the same input as existing losses, so it is a drop-in replacement for them and *agnostic* to the encoder, the modality, and the source of supervision. Since it considers all possible pairwise comparisons in the batch jointly, it also remains trainable at low temperatures, a regime where per-query surrogates run out of gradient. Swapping it into established recipes improves supervised metric learning, cross-modal alignment, and self-supervised pretraining, where, to our knowledge, it is the first ranking loss to replace the community standard InfoNCE in the latter two. Its similarities are more consistent across queries, which drives the gains under a universal threshold. gSAP retrieves up to four times as many positive pairs as the strongest AP surrogate at the same precision, and it degrades the least when queries with no positives in the database are added. Beyond thresholding, models trained with gSAP also learn better representations, with higher transfer, kNN and zero-shot classification accuracy.

1 INTRODUCTION

Average Precision (AP) (Schütze et al., 2008) is the standard metric for ranking quality and is reported as mean AP (mAP) over queries across retrieval and recognition tasks (Radenović et al., 2018; Li et al., 2014; Rolínek et al., 2020). mAP treats each query *separately* (Figure 1), so the distribution of similarities may vary from query to query. Global Average Precision (gAP) (Perronnin et al., 2009), in contrast, ranks all query–database pairs in a single global list and computes a single AP, *i.e.* it evaluates all queries *jointly*. It therefore captures the consistency of similarities across queries as well as ranking quality, and bridges retrieval and detection through a threshold that separates relevant from irrelevant pairs, which is why several benchmarks adopt it (Weyand et al., 2020; Ypsilantis et al., 2021; Pizzi et al., 2022; 2024; Kordopatis-Zilos et al., 2023). gAP is thus the right metric for similarity models and systems operating under a single decision threshold. In Figure 1(b), a per-query surrogate leaves positive and top-negative similarities overlapping across queries, whereas the proposed loss separates them.

Despite its desirable properties, gAP is not used as a training objective, mainly because the ranking at its core is discrete and because it is computed over every pair of the similarity matrix, which is costly when the matrix is large. In this work, we turn gAP into a practical loss for representation learning via pruning negative similarities that do not significantly impact the loss. We derive a smooth surrogate, gSAP, that directly optimizes this global metric. Per-query losses, such as the widely used InfoNCE (Oord et al., 2018) and the existing AP surrogates (Qin et al., 2010; Revaud et al., 2019; Brown et al., 2020; Ramzi et al., 2021), optimize the ranking of each query in isolation and therefore leave the similarity scales of different queries unconstrained. gSAP ranks the whole similarity matrix at once and replaces the loss of any model whose training objective is based on a similarity matrix.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) **(b)**

Figure 1: Overview of mAP and gAP and the effect of optimizing their surrogates. (a) Illustration of the metric computation for a batch of images. For mAP, each row of the similarity matrix is treated as an independent query: AP is computed per row after ranking the similarities, and the final score is obtained by averaging across queries. For gAP, all pairwise similarities are first aggregated into a single list, and a single AP value is computed globally. Consequently, mAP evaluates ranking quality per query and is insensitive to how similarities compare across queries, whereas gAP penalizes inconsistencies in similarity distributions across queries. (b) Distribution of positive and top-negative similarities after overfitting a single batch using mSAP (Brown et al., 2020) and the proposed gSAP loss. The proposed loss produces better separation between positive and negative similarities, allowing them to be distinguished by a single global threshold.

Ranking the whole matrix also eases the trade-off between the fidelity of the surrogate and the flow of its gradient. A small temperature makes the sigmoid follow the step function closely (Qin et al., 2010; Brown et al., 2020), but saturates most comparisons, so their gradients vanish (Ramzi et al., 2021). Since gSAP involves many more comparisons than per-query surrogates, as each positive is compared with every similarity in the batch rather than with its own row only, a sufficient fraction of them remains unsaturated. This allows low temperatures, and thus an accurate AP approximation, without vanishing gradients, which is not possible for the per-query counterparts.

The formulation needs only a similarity matrix and a binary relevance matrix, and is therefore *agnostic* to the encoders, the modality, and the source of supervision. Relevance marks same-class pairs in supervised metric learning, matching pairs in image–text, video–text, or audio–text alignment, and augmented views of the same sample in self-supervised learning. To our knowledge, gSAP is the first ranking loss to replace InfoNCE in CLIP-style cross-modal and self-supervised training.

We evaluate gSAP by replacing only the loss in established recipes for supervised metric learning, CLIP and CLAP cross-modal alignment, MoCo v3 self-supervised pretraining, and tasks with distractor queries. Across all of them, gSAP improves gAP while remaining competitive in mAP and Recall@k, and outperforms InfoNCE and SigLIP, the two standard cross-modal losses, on every retrieval benchmark. It also yields better representations, as the encoders trained with it reach higher transfer, kNN and zero-shot classification accuracy. The gains come from the consistency across queries, which translates into more matches under a single shared threshold and the smallest loss of gAP when distractor queries are added.

In summary, our contributions are outlined below.

- We advocate gAP as a metric for representation learning, since it captures both ranking quality and the consistency of similarities across queries, which per-query metrics ignore.
- We propose gSAP, a smooth surrogate of gAP and a drop-in replacement for existing losses that keeps gradients alive at temperatures where per-query surrogates saturate, and negative dropping, which scales it to very large batches.
- On supervised, cross-modal (image, text, video, audio), self-supervised, and distractor-query tasks, where gSAP is the first ranking loss to replace InfoNCE, it attains the best gAP on every retrieval benchmark, matches or exceeds the best per-query metrics in nearly all cases, and gains by far the most when one threshold serves all queries.

2 Related work

Representation learning losses differ in how much of the similarity matrix they couple. Pairwise losses score each entry on its own; the contrastive loss (Chopra et al., 2005) with an absolute margin and SigLIP (Zhai et al., 2023) with a bias shared by all pairs impose one global threshold, but at a single operating point. Most losses act per query. Triplet (Schroff et al., 2015) and Multi-Similarity (Wang et al., 2019b) weight pairs relative to each anchor, and proxy losses (Deng et al., 2019; Kim et al., 2020) compare each sample with class proxies. InfoNCE (Oord et al., 2018), standard in vision–language (Radford et al., 2021) and self-supervised (Chen et al., 2021) learning, normalizes each row and column with a softmax; RankCLIP (Zhang et al., 2025) adds listwise ordinal-consistency terms across and within modalities. Ranking surrogates directly optimize per-query AP (Rolínek et al., 2020; Revaud et al., 2019; Brown et al., 2020; Ramzi et al., 2021) or Recall@k (Patel et al., 2022); GlobalAP (Liu et al., 2023), despite its name, extends each query's list with a memory bank and still averages AP over queries. These surrogates depend only on within-query orderings and are unchanged by any per-query increasing transformation of the similarities, and each directional InfoNCE term is unchanged by a per-query offset in its own direction. None of these terms constrains how similarities compare *across queries*. Since per-query losses leave this unconstrained, comparable similarities are typically obtained with auxiliary terms (Ramzi et al., 2021; Zhang et al., 2024b; Liu et al., 2022) or post hoc through per-query score normalization (Jégou et al., 2007; Conneau et al., 2017; Pizzi et al., 2022). Batch-wise ranking across groups has also been explored in object detection (Chen et al., 2019), where AP is optimized over all anchors and classes of a batch via gradient approximation. By contrast, our approach trains models by directly optimizing a single surrogate loss that ranks all samples of the batch by their similarities, with every sample acting as a query, and requires neither additional regularization nor post-hoc operations. An extended discussion of related work is provided in Appendix A.

3 Method

3.1 Average Precision

Average Precision (AP) is a standard metric for evaluating the quality of ranking in information retrieval tasks (Schütze et al., 2008). It summarizes the Precision–Recall (PR) curve as the area under the curve, providing a single scalar value.

Let $\mathbf{s} = [s_1, \dots, s_N]$ denote an unsorted list of scalar similarities between items, and $\mathbf{y} = [y_1, \dots, y_N]$, with $y_i \in \{0, 1\}$, the corresponding ground-truth labels, where $y_i = 1$ indicates a positive (*i.e.* same class/semantics) item, and $y_i = 0$ a negative item. We define the index sets of the positive items and of all items as:

$$\mathcal{P}(\mathbf{y}) = \{i \mid y_i = 1\} \quad \text{and} \quad \Omega = \{1, \dots, N\}. \tag{1}$$

AP requires ranking the values in $\mathbf{s}$ in descending order and is defined as the average of precision values computed at the rank of each positive item as

$$\mathrm{AP}(\mathbf{s}, \mathbf{y}) = \frac{1}{|\mathcal{P}(\mathbf{y})|} \sum_{i \in \mathcal{P}(\mathbf{y})} \mathrm{P}_i(\mathbf{s}, \mathbf{y}) \tag{2}$$

Computing precision P_i related to item i requires counting the number of positives and the number of all items that are ranked before it (Qin et al., 2010). This counting process is performed by comparing s_i with the similarities of all positives and all other items. Precision of item i is given by

$$\mathrm{P}_i(\mathbf{s}, \mathbf{y}) = \frac{\sum_{j \in \mathcal{P}(\mathbf{y})} h(s_j - s_i)}{\sum_{j \in \Omega} h(s_j - s_i)} \tag{3}$$

where h is the Heaviside step function:

$$h(\Delta s) = \begin{cases} 1 & \Delta s \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

AP is optimal, equal to 1, when all positives are ranked before all negatives.

3.2 EVALUATION METRICS

Consider a typical retrieval setting with M queries and N database items. The pairwise similarity matrix, comparing each query and database item, and the corresponding ground-truth label matrix are denoted by

$$S = \begin{bmatrix} s_{1,1} & \cdots & s_{1,N} \\ \vdots & \ddots & \vdots \\ s_{M,1} & \cdots & s_{M,N} \end{bmatrix} \in \mathbb{R}^{M \times N} \quad Y = \begin{bmatrix} y_{1,1} & \cdots & y_{1,N} \\ \vdots & \ddots & \vdots \\ y_{M,1} & \cdots & y_{M,N} \end{bmatrix} \in \{0,1\}^{M \times N}. \tag{5}$$

Each row of these matrices corresponds to the similarities and labels of a query, with S_i and Y_i denoting the elements of the i-th row related to the i-th query.

mean Average Precision (mAP). mAP evaluates each query independently, in a row-wise fashion, and averages the AP across all queries:

$$\mathrm{mAP}(S, Y) = \frac{1}{M} \sum_{i=1}^{M} \mathrm{AP}(S_i, Y_i). \tag{6}$$

This metric evaluates the ranking quality separately per query, treating each query equally. There is no interaction between different queries, hence similarity distributions can vary among rows. This metric can be optimal even when similarities are not comparable across queries, *e.g.* the similarities of all positives of one row are smaller than all negatives of another row, despite AP being equal to 1 for both rows. Therefore, mAP does not reflect the consistency of similarities across queries.

global Average Precision (gAP). In the case of gAP, the AP is computed only once, considering similarities across all queries jointly. It is defined as:

$$\mathrm{gAP}(S, Y) = \mathrm{AP}(\mathrm{vec}(S), \mathrm{vec}(Y)), \tag{7}$$

where $\mathrm{vec} : \mathbb{R}^{M \times N} \to \mathbb{R}^{MN}$ is the row-wise flattening operation. This metric evaluates the global ranking quality across all queries jointly. Optimal gAP requires all positives, no matter the query, to have larger similarity than all negatives, no matter the query. Therefore, this metric is sensitive to variations in the similarity distribution across queries. It reflects both the ranking quality per query and the consistency of similarities across queries.

3.3 SURROGATE LOSSES

Standard AP relies on the discrete Heaviside step function (Eq. (4)) and is, therefore, non-differentiable, which prevents direct gradient-based optimization. Prior works address this by smoothing the AP metric. Following Brown et al. (2020), we define the smooth surrogate of the binary indicator function as

$$\tilde{h}(\Delta s) = \sigma(\Delta s / \tau) = \frac{1}{1 + e^{-\Delta s / \tau}}, \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function, and τ is a temperature parameter controlling smoothness. By using this approximation, the AP loss becomes differentiable and is optimized in an end-to-end manner. It is then straightforward to derive from Eq. (6) the surrogate mAP loss as

$$\mathcal{L}_{\mathrm{mSAP}} = 1 - \tilde{\mathrm{mAP}}(S, Y) = 1 - \frac{1}{M} \sum_{i=1}^{M} \tilde{\mathrm{AP}}(S_i, Y_i), \tag{9}$$

where $\tilde{\mathrm{AP}}(\cdot, \cdot)$ is the AP function implemented with the smooth surrogate from Eq. (8). Similarly, we can derive from Eq. (7) the smooth variant of gAP, called gSAP,

$$\mathcal{L}_{\mathrm{gSAP}} = 1 - \tilde{\mathrm{gAP}}(S, Y) = 1 - \tilde{\mathrm{AP}}(\mathrm{vec}(S), \mathrm{vec}(Y)). \tag{10}$$

In that way, we obtain differentiable approximations of the two metrics that can be used to train deep similarity models. Smaller (larger) τ yields a steeper (smoother) sigmoid function and a better (worse) approximation of the exact metric. At the same time, a steep (smooth) sigmoid reaches saturation faster and has a larger (smaller) range where gradients are vanishing, and training is hindered.

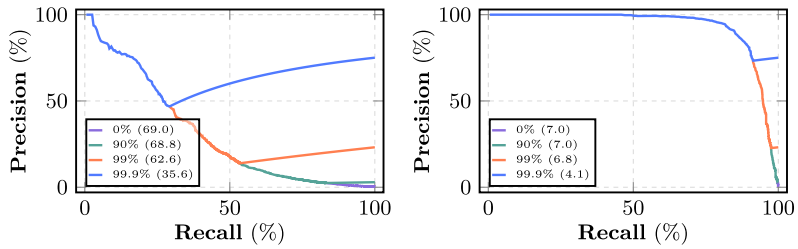


Figure 2: Analysis of negative dropping. Precision–Recall curve computed on a single training batch using DINOv3 before (left) and after (right) training with varying percentages of lowest negative similarity dropping. The legend shows the drop rate α and the corresponding gSAP loss (in parentheses).

3.4 Training with Global Average Precision

Representation learning. Encoders map inputs to a feature space where similarities between samples are estimated via cosine similarity. In the unimodal setting, an image encoder $f_\theta : \mathcal{X} \to \mathbb{R}^d$ maps an image x_i to an ℓ_2-normalized embedding $\mathbf{x}_i = f_\theta(x_i)$, where θ are the learnable parameters. The similarity between two images x_i, x_j with embeddings $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$ is computed as the dot product $s(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$, denoted s_{ij}. In the multimodal setting, separate encoders map inputs from different modalities into a shared embedding space. Given an image x_i and a text description t_j, an image encoder $f_\theta : \mathcal{X} \to \mathbb{R}^d$ and a text encoder $g_\phi : \mathcal{T} \to \mathbb{R}^d$ are used to produce ℓ_2-normalized embeddings $\mathbf{x}_i = f_\theta(x_i)$ and $\mathbf{t}_j = g_\phi(t_j)$. Cross-modal similarity is computed as $s(\mathbf{x}_i, \mathbf{t}_j) = \mathbf{x}_i^\top \mathbf{t}_j$.

Given a batch $\mathcal{B}$ of size B, we compute the pairwise similarity matrix $S^\mathcal{B} \in \mathbb{R}^{B \times B}$ and the corresponding binary label matrix $Y^\mathcal{B} \in \{0,1\}^{B \times B}$ indicating whether pairs match. In the unimodal case, $\mathcal{B} = [x_1, \ldots, x_B]$ is a batch of images and $Y^\mathcal{B}$ encodes class matches between image pairs. In the multimodal case, $\mathcal{B} = [(x_1, t_1), \ldots, (x_B, t_B)]$ consists of image–text pairs and $Y^\mathcal{B}$ is typically the identity matrix indicating aligned image–text pairs. In the unimodal case, the diagonal entries are removed to avoid trivial self-matches. The models are then trained by optimizing either the mSAP or gSAP loss using $S^\mathcal{B}$ and $Y^\mathcal{B}$.

Negative dropping. The cost of gSAP is linear in the number of pairwise similarities in the batch, and nearly all of them are negatives. The smallest negative similarities are already well separated from the positives and barely affect gAP. Hence, to reduce computational and memory requirements in gSAP, we drop the fraction $\alpha \in [0\%, 100\%]$ of the smallest negative similarities. Dropping is implemented by removing these similarities, along with their corresponding labels, from the flattened similarity and label vectors used for the loss. All positives are kept, and the optimization focuses on the negatives that still violate the ranking. The rate α trades memory for fidelity to the full loss. Figure 2 shows its effect on the PR curve of a training batch. Removing up to 90% of the negatives only alters the low-precision tail of the curve, and since the loss is the *area above* the PR curve, a larger α simplifies the optimization and drives the loss towards zero. In practice, large values of α cost minimal performance and under a fixed GPU memory budget they allow larger batches.

3.5 Analysis by single batch overfitting

To better understand the behavior of the surrogate losses during optimization, we conduct a toy experiment in which the model is deliberately overfitted on a single fixed batch, following the protocol of Figure 1(b). For each loss, we optimize the model for 20 iterations at four temperatures τ, from the same initialization and on the same batch. Figure 3 tracks the mAP and gAP on the batch and the average number of unsaturated comparisons per positive. More precisely, a *comparison* is one term $\tilde{h}(s_j - s_i)$ of the surrogate of Eq. (3), between a positive $i \in \mathcal{P}(\mathbf{y})$ and an item $j \in \Omega$ of the list the loss ranks, *i.e.* a row of S for mSAP and $\mathrm{vec}(S)$ for gSAP. A comparison is *unsaturated* when $|s_j - s_i| < 4\tau$, *i.e.* when it still carries non-zero gradient during back-propagation[1].

We draw the following conclusions: (i) **gSAP improves both metrics, mSAP lags behind in gAP.** For $\tau \leq 0.01$, where the surrogate approximates its metric, mAP and gAP move together under gSAP,

[1]Each comparison contributes to the loss through $\sigma(\Delta s/\tau)$, so its gradient with respect to Δs is proportional to $\sigma'(\Delta s/\tau)$, which peaks at $\sigma'(0) = 0.25$ and decays to $\sigma'(4) \approx 0.018$, *i.e.* 7% of the peak. The cut at 4 is a convention, as comparisons beyond it carry almost no gradient (Ramzi et al., 2021).

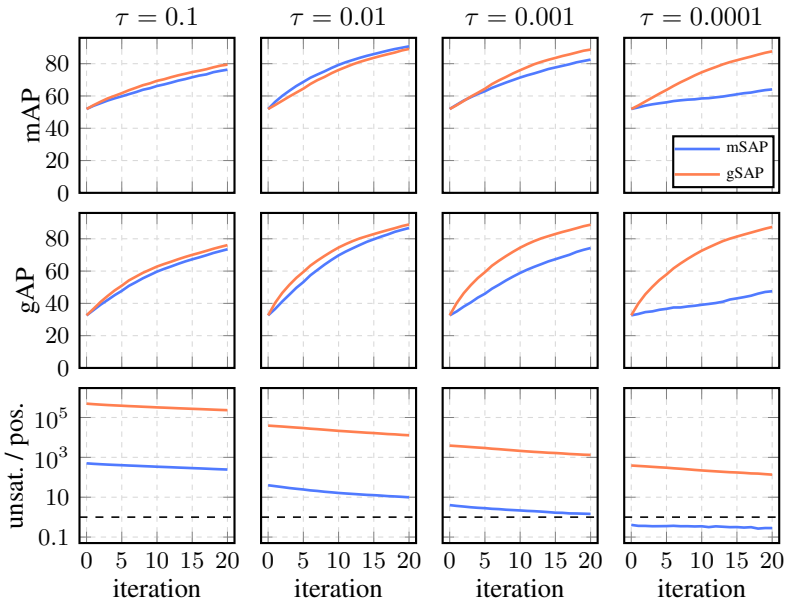


Figure 3: Loss behavior analysis via single batch overfitting. One SOP training batch (250 classes $\times$ 4 images), DINOv3, 20 updates, the same batch and initialization in every panel; columns are temperatures τ, color is the loss (mSAP, gSAP). Top and middle: exact mAP and gAP on the batch, in %. Bottom: average number of unsaturated comparisons per positive ($|\Delta s|/\tau < 4$), *i.e.* their total count divided by the number of positives, log scale. The dashed line indicates one unsaturated comparison per positive.

whereas under mSAP gAP trails mAP by a considerable margin, increasingly so as τ decreases. mSAP is insensitive to the cross-query consistency that gAP depends on, so it improves gAP only as a side effect of better per-query rankings. gSAP, on the other hand, ranks the whole batch and moves both, at no cost in mAP, which is clearly above that of mSAP at three of the four temperatures. (ii) **gSAP is robust to τ because its positives have more unsaturated comparisons.** A small τ makes the surrogate exact but saturates the sigmoid comparisons (Qin et al., 2010; Brown et al., 2020). Empirically, the *share* of unsaturated comparisons is nearly the same for both losses, about 50%, 4%, 0.4% and 0.04% at the four temperatures at initialization, roughly proportional to τ, and it stays within a factor of two between the losses during training. What differs, by three orders of magnitude, is the *number* of comparisons per positive that enter the loss. mSAP scores each positive against the $B - 2 \approx 10^3$ other similarities of its row, gSAP against the $B(B-1) \approx 10^6$ similarities of the whole batch. The bottom row of the figure shows the product of the two, the average number of unsaturated comparisons per positive. At $\tau = 10^{-4}$ an mSAP positive has fewer than one, *i.e.* most positives receive no gradient, and the run stalls well below the other settings. By contrast, a gSAP positive still has more than 150 unsaturated comparisons, every positive keeps a gradient, and the run reaches the same mAP and gAP as at the other values of τ (validated in Figure C). This is why gSAP can operate in the regime of near-exact approximation that mSAP cannot. (iii) **At $\tau = 0.1$ both surrogates are poor proxies of their metric.** Both runs end well below the others. Gradients are plentiful, but the approximation is loose and the optimization noisy. Even in this regime, gSAP reaches slightly higher values of both metrics. In summary, mSAP suffers more from the trade-off between an accurate approximation of the metric and the difficulty of optimizing it.

4 Experiments

Cross-modal learning. We study vision–language representation learning by fine-tuning the released OpenAI CLIP ViT-B/32 (Radford et al., 2021). Starting from the pretrained model, we train for 5 additional epochs on the Conceptual 12M (CC12M) (Changpinyo et al., 2021) dataset using three alternative losses: (i) the original CLIP InfoNCE-based loss (Oord et al., 2018), (ii) the SigLIP sigmoid loss (Zhai et al., 2023), and (iii) our gSAP loss. We evaluate these models in zero-shot retrieval and classification. We follow the CLIP benchmark suite (Cherti & Beaumont, 2022), where applicable. Additional experiments with other vision–language models, further evaluation tasks and analyses, as well as experiments for audio–language alignment, are provided in Appendix B.1.

Table 1: Zero-shot retrieval on image datasets. gAP, mAP, and Recall@5 comparison of gSAP with CLIP and SigLIP losses. All models fine-tune the pretrained OpenAI CLIP ViT-B/32 for 5 epochs on CC12M.

method	Flickr8k					Flickr30k					MS COCO				
	gAP	mAP t→i	mAP i→t	R@5 t→i	R@5 i→t	gAP	mAP t→i	mAP i→t	R@5 t→i	R@5 i→t	gAP	mAP t→i	mAP i→t	R@5 t→i	R@5 i→t
pretrained	31.0	65.0	57.7	79.2	92.0	45.3	68.7	62.7	82.9	94.0	13.1	42.1	36.7	55.0	73.4
+ **CLIP**	41.3	71.9	63.5	85.5	92.4	56.1	75.1	68.3	87.6	94.7	18.7	49.6	41.1	63.6	76.9
+ **SigLIP**	43.2	72.5	64.6	86.0	92.7	58.5	76.3	69.5	88.5	95.7	19.4	50.6	41.8	64.7	77.5
+ **gSAP**	**45.1**	**73.2**	**65.6**	**86.7**	**93.3**	**59.7**	**76.6**	**70.5**	**88.9**	**96.0**	**21.4**	**51.3**	**43.1**	**65.3**	**79.0**

Table 2: Zero-shot retrieval on video datasets. gAP and Recall@k comparison. Same setting as Table 1.

method	ActivityNet							MSR-VTT							MSVD						
	gAP	text→video			video→text			gAP	text→video			video→text			gAP	text→video			video→text		
		R@1	R@5	R@10	R@1	R@5	R@10		R@1	R@5	R@10	R@1	R@5	R@10		R@1	R@5	R@10	R@1	R@5	R@10
pretrained	10.0	22.4	49.8	64.1	21.3	49.4	61.2	4.1	14.3	30.4	38.9	29.4	51.9	61.1	14.5	32.8	60.5	69.9	51.2	71.6	77.8
+ **CLIP**	13.6	27.7	61.0	72.8	26.7	57.6	70.3	4.1	16.2	33.8	42.8	28.0	49.2	59.5	22.0	38.9	67.4	76.9	50.4	73.7	81.8
+ **SigLIP**	13.0	28.8	62.0	74.2	25.9	56.0	69.1	4.2	16.6	34.5	43.5	28.6	50.9	61.1	21.4	39.8	67.8	77.1	52.3	73.6	81.1
+ **gSAP**	**14.9**	**28.9**	**62.2**	**74.6**	**27.5**	**57.8**	**70.5**	**4.7**	**17.0**	**35.0**	**44.0**	**30.2**	**52.3**	**62.0**	**23.2**	**39.9**	**68.4**	**77.4**	**52.7**	**73.9**	**82.1**

Table 3: Zero-shot classification on image and video datasets. Top-1, top-5, and mean per-class recall (mR) comparison of gSAP with CLIP (InfoNCE) and SigLIP losses. Same setting as Table 1.

method	image datasets												video datasets					
	IN-1k			IN-V2			IN-R			IN-Sketch			K-400		K-600		K-700	
	top-1	top-5	mR	top-1	top-5	mR	top-1	top-5	mR	top-1	top-5	mR	top-1	top-5	top-1	top-5	top-1	top-5
pretrained	59.6	86.4	59.6	52.9	80.8	52.9	67.0	87.2	64.9	38.9	66.9	39.0	38.4	64.3	17.8	33.1	27.9	51.1
+ **CLIP**	60.7	87.4	60.7	53.6	82.0	53.6	68.3	87.9	66.8	42.3	70.4	42.3	39.4	65.8	18.6	34.3	29.1	53.5
+ **SigLIP**	61.3	87.8	61.3	54.1	82.1	54.2	69.1	88.2	67.8	43.2	71.6	43.3	39.8	66.3	19.0	34.8	29.5	54.1
+ **gSAP**	**61.8**	**87.9**	**61.8**	**54.3**	**82.5**	**54.3**	**69.4**	**88.3**	**68.1**	**43.7**	**71.7**	**43.7**	**40.2**	**66.5**	**19.2**	**34.9**	**29.8**	**54.4**

Zero-shot retrieval. We evaluate bidirectional retrieval, *i.e.*, text→image/video and image/video→text, on three image datasets, Flickr8k (Hodosh et al., 2013), Flickr30k (Young et al., 2014), and MS COCO Captions (Chen et al., 2015) (Table 1), and three video datasets, ActivityNet (Caba Heilbron et al., 2015), MSR-VTT (Xu et al., 2016), and MSVD (Chen & Dolan, 2011) (Table 2). For image datasets, we report gAP, mAP, and Recall@5; for video datasets, we report gAP and Recall@k. Across all six datasets, gSAP delivers the strongest retrieval performance, with consistent gains over both InfoNCE and SigLIP in every metric and direction.

Zero-shot classification. We evaluate on four image datasets, ImageNet-1k (Deng et al., 2009), ImageNet-V2 (Recht et al., 2019), ImageNet-R (Hendrycks et al., 2021a), and ImageNet-Sketch (Wang et al., 2019a); and three video datasets, Kinetics-400 (Kay et al., 2017), -600 (Carreira et al., 2018), and -700 (Carreira et al., 2019) (Table 3), using the original prompt templates for each dataset. Across all seven datasets, gSAP again provides the most consistent gains, outperforming both InfoNCE and SigLIP in every metric. Beyond these seven datasets, we also evaluate on the full suite of 40 datasets of the benchmark (Cherti & Beaumont, 2022) (37 classification and 3 retrieval), spanning natural, fine-grained, satellite, medical, and synthetic imagery. Aggregating all 108 scores (Table R), gSAP improves on average by +0.49 over InfoNCE and +0.64 over SigLIP, indicating that the benefits of optimizing gAP also translate to improvements in recognition.

Self-supervised learning. In addition, we test whether the benefit survives without labels by following the MoCo v3 (Chen et al., 2021) recipe and replacing InfoNCE by gSAP, with everything else unchanged. Table 4 reports top-1 accuracy of frozen backbones under linear probing (LP) and k-nearest neighbors (kNN). Per-dataset results and settings are in Appendices B.2 and C.2. Three observations follow: (i) The gain carries over to the label-free setting, where gSAP trails only on in-distribution LP at ViT-B. (ii) The gain is consistently larger under kNN, which operates on the learned similarities directly, whereas

Table 4: Image classification with self-supervised pretraining. Top-1 accuracy of models trained with MoCo v3 using InfoNCE or gSAP. Performance of the frozen backbone with linear probing (LP) and k-nearest neighbors (kNN, k=20) on in-distribution (ID, ImageNet-1k), out-of-distribution (OOD, mean of ImageNet-V2/-R/-Sketch), transfer learning (ten datasets), and robustness under corruption (ImageNet-C, 75 sets).

method	model	ID		OOD		transfer		robustness	
		LP	kNN	LP	kNN	LP	kNN	LP	kNN
InfoNCE	ViT-S	73.2	68.4	38.2	31.5	69.8	61.0	44.6	39.4
gSAP	ViT-S	**73.5**	**70.9**	**39.7**	**35.4**	**73.9**	**65.5**	**46.3**	**44.7**
InfoNCE	ViT-B	**76.7**	71.4	**42.0**	36.6	74.8	62.5	49.6	47.0
gSAP	ViT-B	75.6	**72.8**	**42.0**	**38.1**	**77.8**	**66.8**	**49.7**	**48.6**

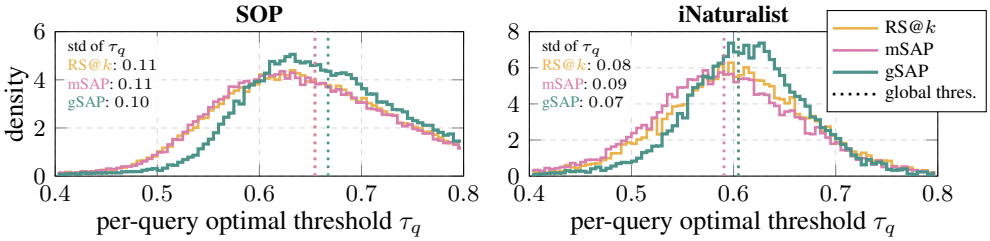


Figure 4: Per-query optimal thresholds. Distribution of the F1-optimal threshold of each evaluation query. Dotted lines indicate the shared threshold selected on the fitting classes. Standard deviation of τ_q per loss in the top-left of each panel.

LP fits a new linear map that can absorb a similarity scale that differs from query to query. (iii) The transfer gain is considerable and concentrated on the six fine-grained datasets (Table H), where hard negatives crowd the positives, which is what a single shared threshold penalizes. Furthermore, the same substitution improves music tagging in audio contrastive learning, so gSAP is not tied to a particular modality (Appendix B.2).

Metric learning. We compare gSAP with competing losses on two standard benchmarks, Stanford Online Products (SOP) (Oh Song et al., 2016) and iNaturalist (Van Horn et al., 2018), following the splits and setup of prior work (Brown et al., 2020; Patel et al., 2022). Evaluation settings and implementation details are provided in Appendix C.3.

Comparison with metric learning losses. Table 5 compares gSAP with pairwise (Contrastive (Chopra et al., 2005), Triplet (Schroff et al., 2015), Multi-Similarity (Wang et al., 2019b)), proxy (ArcFace (Deng et al., 2019), Proxy Anchor (Kim et al., 2020)) and ranking losses (the Recall@k surrogate RS@k (Patel et al., 2022), mSAP (Brown et al., 2020)) under fine-tuning of DINOv3 (Siméoni et al., 2025). gSAP achieves the best gAP on both datasets by a clear margin, while its mAP and R@1 remain close to the best. The ranking losses match gSAP on mAP and R@1, so their lower gAP stems not from worse per-query ranking but from similarity distributions that are less consistent across queries, a property that gAP captures and mAP and R@1 do not. We delve into a theoretical analysis of the losses in Appendix E, in terms of their invariance to per-query similarity and the theoretical best gAP that can be achieved given the per-query ranking.

Table 5: Metric learning losses comparison. Recall@1 (R@1), mAP and gAP on two datasets. gSAP is compared with ranking, pairwise, and proxy losses. Fine-tuning of DINOv3. No negative dropping is applied.

method	SOP			iNaturalist		
	R@1	mAP	gAP	R@1	mAP	gAP
Contrastive	85.3	68.4	53.7	80.6	42.1	28.0
Triplet	84.6	69.1	50.0	77.8	38.6	18.2
ArcFace	82.7	64.8	50.3	80.2	42.4	23.0
Proxy Anchor	84.7	67.7	51.9	78.7	40.9	21.8
Multi-Similarity	86.6	72.0	57.3	**81.6**	45.2	29.9
RS@k	86.7	72.5	56.7	81.0	44.5	26.6
mSAP	**86.9**	**72.9**	57.0	81.3	45.4	27.8
gSAP	86.8	**72.9**	**58.9**	81.5	**45.5**	**31.6**

Table 6: One threshold for all queries. $F1_q$ is the mean per-query F1 when every query uses its own optimal threshold. $F1_g$ is the same mean when all queries share one threshold, *i.e.* it is applied globally, selected on the fitting classes. R@P90 is the recall of that shared threshold set for precision 0.90.

method	SOP			iNaturalist		
	$F1_q$	$F1_g$	R@P90	$F1_q$	$F1_g$	R@P90
RS@k	75.8	57.5	18.3	47.7	35.8	0.6
mSAP	76.2	57.6	18.7	48.5	36.6	0.7
gSAP	**76.2**	**59.3**	**21.0**	**48.7**	**38.3**	**3.0**

Consistency and thresholding. Consistency matters wherever one threshold has to serve every query. Table 6 shows this for the three ranking losses, with the test classes split into fitting classes, on which a threshold is selected, and evaluation classes, on which it is applied (Appendix C.3). When each query is allowed its own optimal threshold, the three losses perform on par on both datasets, in line with mAP and R@1, since the decision is then made within each query's ranking. With a single threshold, selected on the fitting classes and shared by all evaluation queries, gSAP outperforms both other losses on both datasets. Notably, at a precision of 0.90, it retrieves 12% more matches than mSAP, the second best, on SOP and 4× as many on iNaturalist. Figure 4 indicates the reason for its effectiveness. The per-query optimal thresholds of gSAP are more concentrated around the shared one, so a single value fits more queries. The gap between $F1_q$ and $F1_g$ is the cost of inconsistent similarities, and gSAP reduces it without changing what happens inside each query.

Table 7: Comparison with the metric learning state-of-the-art. Recall@k (R@k) of the vanilla losses as reported in the original papers on two datasets and on two backbones. R50 stands for ResNet50. Dimensionality of the output representations is indicated in the superscript. **Best** in bold, second best underlined. 4,000 batch size and α=95% drop rate are used.

method	arch.^{dim}	SOP			iNaturalist		
		R@1	R@10	R@100	R@1	R@4	R@16
Triplet SH (Wu et al., 2017)	R50^{512}	72.7	86.2	93.8	58.1	75.5	86.8
NormSoftmax (Zhai & Wu, 2019)	R50^{512}	78.2	90.6	96.2	–	–	–
FastAP (Cakir et al., 2019)	R50^{512}	76.4	89.0	95.1	60.6	77.0	87.2
Blackbox (Rolínek et al., 2020)	R50^{512}	78.6	90.5	96.0	62.9	79.0	88.9
SoftBin (Revaud et al., 2019)	R50^{512}	80.6	91.3	96.1	64.2	77.1	82.7
ProxyNCA (Movshovitz-Attias et al., 2017)	R50^{512}	73.7	–	–	61.6	77.4	87.0
ProxyNCA++ (Teh et al., 2020)	R50^{512}	80.7	92.5	96.7	–	–	–
SAP (Brown et al., 2020)	R50^{512}	80.1	91.5	96.6	67.2	81.8	90.3
ROADMAP (Ramzi et al., 2021)	R50^{512}	83.1	92.7	96.3	69.1	83.1	**91.3**
RS@k (Patel et al., 2022)	R50^{512}	82.8	92.9	**97.0**	71.2	84.0	**91.3**
HIER (Kim et al., 2023)	R50^{512}	80.2	91.5	96.6	–	–	–
Potential Field (Bhatnagar & Ahuja, 2025)	R50^{512}	82.9	92.5	96.8	–	–	–
gSAP	R50^{512}	**83.3**	**93.2**	**97.0**	**71.3**	**84.1**	**91.3**
SAP (Brown et al., 2020)	ViT-B/16^{512}	86.6	95.4	98.4	79.1	89.0	94.2
RS@k (Patel et al., 2022)	ViT-B/16^{512}	88.0	96.1	98.6	83.9	92.1	95.9
gSAP	ViT-B/16^{512}	**88.4**	**96.3**	**98.7**	**84.9**	**92.8**	**96.5**

Table 8: Tasks with distractor queries. gAP: all queries. gAP$^-$: distractor queries removed. drop: relative drop (%); lower is better. SN: after score normalization (Pizzi et al., 2022). **Best** in bold, second best underlined.

method	DISC21						Met			ILIAS			WildlifeReID-10k		
	gAP	gAP$^-$	drop↓	gAP$_{\text{SN}}$	gAP$^-_{\text{SN}}$	drop↓	gAP	gAP$^-$	drop↓	gAP	gAP$^-$	drop↓	gAP	gAP$^-$	drop↓
baseline	51.6	64.6	20.1	65.5	70.1	**6.6**	30.4	52.9	42.5	3.5	17.1	79.5	69.5	73.9	6.0
InfoNCE	50.2	66.0	23.9	67.5	73.3	7.9	56.2	72.2	22.2	7.4	22.0	66.4	77.1	80.0	3.6
mSAP	54.5	68.2	20.1	68.0	73.2	7.1	58.3	**73.0**	20.1	5.6	19.6	71.4	80.2	82.8	3.1
gSAP	**55.9**	**69.0**	**19.0**	**68.5**	**73.7**	7.1	**62.8**	**73.0**	**14.0**	**8.7**	**22.7**	**61.7**	**80.9**	**83.2**	**2.8**

Comparison with state-of-the-art methods. Table 7 compares gSAP with the state of the art on SOP and iNaturalist using ResNet50 (He et al., 2016) and ViT-B/16 (Dosovitskiy et al., 2021) backbones. For these experiments, we leverage the proposed negative dropping strategy with 95% drop rate, to scale the batch size to 4,000 on a single GPU. gSAP ranks among the state of the art across both datasets and architectures, showing that optimizing the gAP surrogate also yields strong performance on the standard retrieval settings.

Tasks with distractor queries. We finally evaluate on four tasks where a large share of queries are *distractor queries*, *i.e.* queries with no relevant item in the database: copy detection (DISC21 (Pizzi et al., 2022)), instance-level recognition (Met (Ypsilantis et al., 2021)) and retrieval (ILIAS (Kordopatis-Zilos et al., 2025)), and animal re-identification (WildlifeReID-10k (Adam et al., 2025)). These are the settings the proposed loss targets, as they traditionally employ gAP for evaluation. The system must decide with one threshold, shared across queries, whether a match exists at all, so a distractor query is handled correctly only if its negatives fall below the positives of other queries. Per-query metrics ignore distractor queries, since a query without positives has no AP, whereas gAP considers all their negatives. Additional results and settings are in Appendices B.4 and C.4. gSAP attains the best gAP on all four tasks and the smallest drop before score normalization. The gain is mostly cross-query. On Met, mSAP and gSAP are identical without distractors and gSAP is 4.5 gAP ahead with them. SN brings the drops of all losses close together but does not close the gap between mSAP and gSAP on DISC21 for gAP, so a post-hoc correction recovers only part of what training for gAP provides. Finally, the second-best method varies across datasets and settings.

5 Conclusions

We revisit global Average Precision (gAP), which evaluates ranking quality together with the consistency of similarities across queries, and propose gSAP, a differentiable surrogate of it. gSAP operates on the same input as the standard losses of established training recipes, a similarity matrix and its binary relevance labels, and swapping it in, and nothing else, improves supervised metric learning, cross-modal alignment, and self-supervised pretraining, surpassing the well-established InfoNCE where it is the first ranking loss to replace it. Its similarities are more consistent across queries, which pays off whenever one threshold serves all queries. Its cost is quadratic in the batch size, which negative dropping mitigates, and part of the achievable consistency remains unrealized, leaving room for complementary approaches, *e.g.* learnable post-hoc score normalization (Pizzi et al., 2022).

AI USE STATEMENT

In this work, we used generative AI tools to automate the execution of experiments, *i.e.* to write and edit scripts that launch training and evaluation runs, collect their outputs, and compile the result tables. We have not used generative AI tools to propose or refine hypotheses, develop the theoretical analysis, formulate mathematical claims or write their proofs, design the research methodology or experiments, implement the proposed method, generate synthetic data, clean or reformat datasets, interpret results, or assist with translation; qualitative or thematic data analysis is not applicable to this work. Additionally, we used generative AI tools to correct grammar and syntax errors in the manuscript. We have reviewed all AI-assisted work: all AI-generated code was inspected by the authors before use, and all reported numbers were checked against the logs of the corresponding runs. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly available datasets and pretrained checkpoints, under their respective licenses. No new data was collected and no human subjects were involved.

REPRODUCIBILITY STATEMENT

The gAP metric and its surrogate, the gSAP loss, are defined in Sections 3.2 to 3.4. Appendix C gives the datasets, implementation details, hyper-parameters and tuning protocol, and evaluation protocol for every task (Appendices C.1 to C.4). Each task builds on the official public repository of its baseline, with only the loss replaced, and reported numbers are averaged over three random seeds. Detailed per-dataset results are in Appendix D. We will release our code upon publication.

REFERENCES

Lukas Adam, Vojtech Cermak, Kostas Papafitsoros, and Lukas Picek. WildlifeReID-10k: Wildlife re-identification dataset with 10k individual animals. In *CVPRW*, 2025. 9, 30

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *SIGKDD*, 2019. 29

Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 2000. 36

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022. 19

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Joshua Tenenbaum, and Boris Katz. ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*, 2019. 27

Charles Beattie, Joel Z. Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Victor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Michael Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. DeepMind Lab. In *arXiv*, 2016. 27

Shubhang Bhatnagar and Narendra Ahuja. Potential field based deep metric learning. In *CVPR*, 2025. 9, 18

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. In *NeurIPS*, 2025. 26, 27, 30

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014. 28

Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *ECCV*, 2020. 1, 2, 3, 4, 6, 8, 9, 19, 24, 25, 29

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *ICML*, 2005. 19

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 7, 27

Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *CVPR*, 2019. 9, 19

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 19

Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about Kinetics-600. In *arXiv*, 2018. 7, 27

Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the Kinetics-700 human action dataset. In *arXiv*, 2019. 7, 27

Vojtěch Čermák, Lukas Picek, Lukáš Adam, and Kostas Papafitsoros. WildlifeDatasets: An open-source toolkit for animal re-identification. In *WACV*, 2024. 30

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 6, 27

David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. 7, 27

Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection. In *ICASSP*, 2022. 20, 28

Kean Chen, Jianguo Li, Weiyao Lin, John See, Ji Wang, Lingyu Duan, Zhibo Chen, Changwei He, and Junni Zou. Towards accurate one-stage object detection with AP-loss. In *CVPR*, pp. 5119–5127, 2019. 3

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 19

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. In *arXiv*, 2015. 7, 27

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 3, 7, 19, 28

Yuxin Chen, Zongyang Ma, Ziqi Zhang, Zhongang Qi, Chunfeng Yuan, Bing Li, Junfu Pu, Ying Shan, Xiaojuan Qi, and Weiming Hu. How to make cross encoder a good teacher for efficient image-text retrieval? In *CVPR*, 2024. 18

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 2017. 27

Mehdi Cherti and Romain Beaumont. CLIP benchmark, 2022. 6, 7, 27, 28, 31

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023. 27

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 3, 8, 18

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 27, 28

Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011. 27

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *ICLR*, 2017. 3

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 7, 27, 28

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 3, 8, 18

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 9, 29

Matthijs Douze, Giorgos Tolias, Ed Pizzi, Zoë Papakipos, Lowik Chanussot, Filip Radenovic, Tomas Jenicek, Maxim Maximov, Laura Leal-Taixé, Ismail Elezi, et al. The 2021 image similarity dataset and challenge. In *arXiv*, 2021. 30

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP*, 2020. 20, 27

Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision transformers: Combining improvements in metric learning. In *CVPR*, 2022. 18

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 2010. 27

Tom Fawcett and Alexandru Niculescu-Mizil. PAV and the ROC convex hull. *Machine Learning*, 2007. 36

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *CVPRW*, 2004. 27

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 27

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017. 20, 28

Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in representation learning: A report on three machine learning contests. In *ICONIP*, 2013. 27

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 19

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 19

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 9, 29

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 19

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 19

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 27

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 22, 28

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021a. 7, 27, 28

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021b. 27

Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 2013. 7, 27

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, 2021. 27

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *TOIS*, 2002. 19

Hervé Jégou, Hedi Harzallah, and Cordelia Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *CVPR*, 2007. 3, 36

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *PAMI*, 2010. 19

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 27

Kaggle and EyePACS. Diabetic retinopathy detection, 2015. 27

Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. In *ECCV*, 2022. 19

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The Kinetics human action video dataset. In *arXiv*, 2017. 7, 27

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. AudioCaps: Generating captions for audios in the wild. In *NAACL*, 2019. 20, 27

Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, 2020. 3, 8, 18

Sungyeon Kim, Boseung Jeong, and Suha Kwak. HIER: Metric learning beyond class labels via hierarchical regularization. In *CVPR*, 2023. 9, 18

Giorgos Kordopatis-Zilos, Giorgos Tolias, Christos Tzelepis, Ioannis Kompatsiaris, Ioannis Patras, and Symeon Papadopoulos. Self-supervised video similarity learning. In *CVPRW*, 2023. 1

Giorgos Kordopatis-Zilos, Vladan Stojnić, Anna Manko, Pavel Šuma, Nikolaos-Antonios Ypsilantis, Nikos Efthymiadis, Zakaria Laskar, Jiří Matas, Ondřej Chum, and Giorgos Tolias. ILIAS: Instance-level image retrieval at scale. In *CVPR*, 2025. 9, 30

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *CVPRW*, 2013. 27, 28

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical Report, University of Toronto, 2009. 27, 28

Edith Law, Kris West, Michael I Mandel, Mert Bay, and J Stephen Downie. Evaluation of algorithms using games: The case of music tagging. In *ISMIR*, 2009. 23, 29

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 27

Yann LeCun, Fu Jie Huang, and Léon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR*, 2004. 27

Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. In *SMC*, 2017. 29

Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. DeepReID: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014. 1

Jiaheng Liu, Zhipeng Yu, Haoyu Qin, Yichao Wu, Ding Liang, Gangming Zhao, and Ke Xu. OneFace: One threshold for all. In *ECCV*, 2022. 3, 19

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. SphereFace: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 18

Yifei Liu, Yaling Liang, Pengfei Wang, Ziheng Chen, and Changxing Ding. Globalap: Global average precision optimization for person re-identification. *Pattern Recognition*, 2023. 3, 19

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 28

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 28, 29

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. In *arXiv*, 2013. 27, 28

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dSprites: Disentanglement testing sprites dataset, 2017. 27

Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017. 9, 18

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPSW*, 2011. 27

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 27, 28

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. 8, 18, 29

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. In *arXiv*, 2018. 1, 3, 6, 28

OpenAI. The Country211 dataset, 2021. 27

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 19

Lasha Otarashvili, Tamilselvan Subramanian, Jason Holmberg, J.J. Levenson, and Charles V. Stewart. Multispecies animal re-id using a large community-curated dataset. In *arXiv*, 2024. 30

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. 27, 28

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 29

Yash Patel, Giorgos Tolias, and Jiří Matas. Recall@k surrogate loss with large batches and similarity mixup. In *CVPR*, 2022. 3, 8, 9, 19, 24, 29

Florent Perronnin, Yan Liu, and Jean-Michel Renders. A family of contextual measures of similarity between distributions with application to image retrieval. In *CVPR*, 2009. 1

Karol J. Piczak. ESC: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 1015–1018, 2015. 20, 27

Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *CVPR*, 2022. 1, 3, 9, 19, 30, 36

Ed Pizzi, Giorgos Kordopatis-Zilos, Hiral Patel, Gheorghe Postelnicu, Sugosh Nagavara Ravindra, Akshay Gupta, Symeon Papadopoulos, Giorgos Tolias, and Matthijs Douze. The 2023 video similarity dataset and challenge. *CVIU*, 2024. 1

Foster Provost and Tom Fawcett. Robust classification for imprecise environments. *Machine Learning*, 2001. 36

Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. SoftTriple loss: Deep metric learning without triplet sampling. In *ICCV*, 2019. 18

Tao Qin, Tie-Yan Liu, and Hang Li. A general approximation framework for direct optimization of information retrieval measures. *Information retrieval*, 2010. 1, 2, 3, 6

Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *PAMI*, 2018. 1

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3, 6, 18, 27, 28

Elias Ramzi, Nicolas Thome, Clément Rambour, Nicolas Audebert, and Xavier Bitot. Robust and decomposable average precision for image retrieval. In *NeurIPS*, 2021. 1, 2, 3, 5, 9, 19, 25

Elias Ramzi, Nicolas Audebert, Nicolas Thome, Clément Rambour, and Xavier Bitot. Hierarchical average precision training for pertinent image retrieval. In *ECCV*, 2022. 18

Elias Ramzi, Nicolas Audebert, Clément Rambour, André Araujo, Xavier Bitot, and Nicolas Thome. Optimization of rank losses for image retrieval. *PAMI*, 2025. 18

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *ICML*, 2019. 7, 27, 28

Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, 2019. 1, 3, 9, 19, 25, 29

Stephen E Robertson. The probability ranking principle in IR. *Journal of Documentation*, 1977. 36

Michal Rolínek, Vít Musil, Anselm Paulus, Marin Vlastelica, Claudio Michaelis, and Georg Martius. Optimizing rank-based metrics with blackbox differentiation. In *CVPR*, 2020. 1, 3, 9, 19, 25

Andrea Rosasco, Stefano Berti, Giulia Pasquale, Damiano Malafronte, Shogo Sato, Hiroyuki Segawa, Tetsugo Inada, and Lorenzo Natale. Concon-chi: Concept-context chimera benchmark for personalized vision-language tasks. In *CVPR*, 2024. 22

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. In *ICLR*, 2019. 19, 25

Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 1041–1044, 2014. 20, 27

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 3, 8, 18

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open dataset of CLIP-filtered 400 million image-text pairs. In *arXiv*, 2021. 20, 27

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008. 1, 3

Jeffrey B Sidney. Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs. *Operations Research*, 1975. 36

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. In *arXiv*, 2025. 8, 19, 26, 27, 29

Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 19

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013. 27

Kihyuk Sohn. Improved deep metric learning with multi-class N-pair loss objective. In *NeurIPS*, 2016. 18

Janne Spijkervet and John Ashley Burgoyne. Contrastive learning of musical representations. In *ISMIR*, 2021. 19, 23, 29

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *IJCNN*, 2011. 27

Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. SoftRank: Optimizing non-smooth rank metrics. In *WSDM*, 2008. 19

Eu Wern Teh, Terrance DeVries, and Graham W Taylor. ProxyNCA++: Revisiting and revitalizing proxy neighborhood component analysis. In *ECCV*, 2020. 9, 18

Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. In *NeurIPS*, 2017. 19

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. In *arXiv*, 2025. 18, 26, 27

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *CVPR*, 2018. 8, 29

Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. In *MICCAI*, 2018. 27

Andreas Veit and Kimberly Wilber. Improving calibration in deep metric learning with cross-example softmax. In *arXiv*, 2020. 19

Shashanka Venkataramanan, Bill Psomas, Ewa Kijak, Laurent Amsaleg, Konstantinos Karantzalos, and Yannis Avrithis. It takes two to tango: Mixup for deep metric learning. In *ICLR*, 2022. 18

Ellen M Voorhees, Dawn M Tice, et al. The TREC-8 question answering track evaluation. In *TREC*, 1999. 19

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 28

Haohan Wang, Songwei Ge, Zachary C. Lipton, and Eric P. Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019a. 7, 27, 28

Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, 2019b. 3, 8, 18

Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google Landmarks Dataset v2 – a large-scale benchmark for instance-level recognition and retrieval. In *CVPR*, 2020. 1, 19

Chao-Yuan Wu, Raghavan Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *ICCV*, 2017. 9

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP*, 2023. 18, 20, 28

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 27, 28

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016. 7, 27

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014. 7, 27

Nikolaos-Antonios Ypsilantis, Noa Garcia, Guangxing Han, Sarah Ibrahimi, Nanne Van Noord, and Giorgos Tolias. The Met dataset: Instance-level recognition for artworks. In *NeurIPS*, 2021. 1, 9, 30

Nikolaos-Antonios Ypsilantis, Kaifeng Chen, Bingyi Cao, Mário Lipovský, Pelin Dogan-Schönberger, Grzegorz Makosa, Boris Bluntschli, Mojtaba Seyedhosseini, Ondřej Chum, and André Araujo. Towards universal image embeddings: A large-scale dataset and challenge for generic image representations. In *ICCV*, 2023. 30

Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. In *BMVC*, 2019. 9, 18

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 3, 6, 18, 28

Qin Zhang, Dongsheng An, Tianjun Xiao, Tong He, Qingming Tang, Ying Nian Wu, Joseph Tighe, and Yifan Xing. Learning for transductive threshold calibration in open-world recognition. In *CVPR*, 2024a. 19

Qin Zhang, Linghan Xu, Qingming Tang, Jun Fang, Ying Nian Wu, Joe Tighe, and Yifan Xing. Threshold-consistent margin loss for open-world deep metric learning. In *ICLR*, 2024b. 3, 19, 25

Yiming Zhang, Zhuokai Zhao, Zhaorun Chen, Zhili Feng, Zenghui Ding, and Yining Sun. RankCLIP: Ranking-consistent language-image pretraining. In *ICCV*, 2025. 3, 18

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. In *ICLR*, 2022. 19

Mu Zhu. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2004. 19

APPENDIX

A Extended related work **18**

B Additional results **20**

B.1 Cross-modal learning 20

B.2 Self-supervised learning 22

B.3 Metric learning 24

B.4 Tasks with distractor queries 25

C Evaluation settings **27**

C.1 Cross-modal learning 27

C.2 Self-supervised learning 28

C.3 Metric learning 29

C.4 Tasks with distractor queries 30

D Detailed results **31**

E Theoretical analysis **34**

A EXTENDED RELATED WORK

Metric learning. Early work employs pairwise and triplet losses, *e.g.*, the contrastive loss (Chopra et al., 2005) and the triplet loss (Schroff et al., 2015), which pull positives together and push negatives beyond a margin. To incorporate more context, listwise (Oh Song et al., 2016) and N-pair (Sohn, 2016) losses sample multiple positives and negatives per anchor, and Multi-Similarity (Wang et al., 2019b) weights all pairs of the batch relative to each anchor. Proxy-based methods reduce complexity by learning class proxies (Movshovitz-Attias et al., 2017; Teh et al., 2020; Kim et al., 2020; Zhai & Wu, 2019) or multiple proxies per class (Qian et al., 2019), while angular-margin losses (Liu et al., 2017; Deng et al., 2019) enforce margin-based separation on the hypersphere. Hierarchical methods (Ramzi et al., 2022; 2025) incorporate error severity into the ranking, while other works train with hyperbolic losses (Ermolov et al., 2022; Kim et al., 2023) or potential fields (Bhatnagar & Ahuja, 2025). Orthogonal to the choice of loss, Metrix (Venkataramanan et al., 2022) brings mixup to metric learning by interpolating both examples and target labels within a generalized formulation of these losses. All these losses sum terms defined per pair, per anchor, or per proxy, whereas gSAP couples all pairs of the batch through a single ranking.

Vision–language alignment. Vision–language pretraining is commonly instantiated with CLIP-style contrastive learning (Radford et al., 2021), where image–text alignment is optimized via a softmax-normalized loss over in-batch negatives; the same loss underlies audio–language models such as CLAP (Wu et al., 2023). Recent work revisits this design by introducing new losses. RankCLIP (Zhang et al., 2025) extends pairwise contrastive learning to a listwise formulation and enforces both in-modal and cross-modal ordinal ranking consistency, aiming to capture many-to-many relationships between images and texts. SigLIP (Zhai et al., 2023; Tschannen et al., 2025) adopts a pairwise sigmoid loss that removes the need for row-wise normalization; its bias, shared by all pairs, sets one global threshold, but each pair is scored independently at that single operating point. Beyond pretraining, CPRD (Chen et al., 2024) distills cross-encoder knowledge into vision–language encoders by matching the partial order of hard negatives, demonstrating that relative ranking can be more transferable than raw score regression. In this work, we target gAP directly, replacing these row-, column-, or pair-wise losses with a single AP over all image–text or audio–text pairs under a shared threshold.

Self-supervised learning. Contrastive self-supervised learning treats augmented views of the same sample as positives and the other samples of the batch as negatives, and optimizes InfoNCE per anchor in vision (Chen et al., 2020; He et al., 2020; Chen et al., 2021) and audio (Spijkervet & Burgoyne, 2021). Non-contrastive methods avoid negatives altogether, either by self-distillation between a student and a momentum teacher (Grill et al., 2020; Caron et al., 2021) or by masked image modeling, which reconstructs masked patches in pixel (He et al., 2022) or token space (Bao et al., 2022). iBOT (Zhou et al., 2022) combines the two by distilling masked patch tokens from the teacher, and AttMask (Kakogeorgiou et al., 2022) masks the patches the teacher attends to most, making the distillation task harder. DINOv2 (Oquab et al., 2024) and DINOv3 (Siméoni et al., 2025) scale this recipe into general-purpose backbones; DINOv3 is the starting point of our metric learning and distractor-query experiments. gSAP applies to the contrastive family: since the views define the relevance matrix, it replaces InfoNCE with no other change, whereas distillation and reconstruction losses have no similarity matrix over negatives to rank.

Surrogate ranking losses. To better align training with evaluation, recent work proposes differentiable surrogates for ranking metrics. Early approaches smooth NDCG (Taylor et al., 2008) or learn from pairwise preferences (Burges et al., 2005). Several methods approximate AP for optimizing mAP (Triantafillou et al., 2017; Rolínek et al., 2020; Revaud et al., 2019; Cakir et al., 2019; Brown et al., 2020). BlackBox (Rolínek et al., 2020) estimates gradients through sorting via black-box differentiation, while FastAP (Cakir et al., 2019) improves efficiency by quantizing distances into differentiable histograms, reducing the computational cost of AP gradients. SmoothAP (Brown et al., 2020) further scales AP optimization by applying a sigmoid to similarity differences. ROADMAP (Ramzi et al., 2021) replaces the sigmoid with a different function on positives and negatives, propagating only through the latter. Beyond AP, RS@k (Patel et al., 2022) addresses the non-differentiability of Recall@k with a smooth surrogate. GlobalAP (Liu et al., 2023), despite its name, extends each query's ranking with a memory bank and still averages AP over queries. Surrogates used in representation learning, however, remain query-centric and do not target global criteria such as gAP, which our loss is designed to optimize.

Similarity consistency and thresholding. Prior work aims to make similarities comparable across queries, classes, or domains, so that one threshold serves all of them; the retrieval literature calls this property cross-query calibration (Pizzi et al., 2022), which is distinct from the calibration of scores as match probabilities (Guo et al., 2017). Consistency can be encouraged during training via modified objectives that incorporate cross-example similarities (Veit & Wilber, 2020), regularizers that constrain pairwise distances within bounds (Ramzi et al., 2021; Zhang et al., 2024b; Liu et al., 2022), or dispersion losses that push nearest negatives apart (Sablayrolles et al., 2019). Beyond training time, similarity scores can be mapped post hoc to a common scale, *e.g.* by per-query score normalization against a background set (Pizzi et al., 2022), Platt scaling or isotonic regression (Guo et al., 2017), or learned or transductive mapping modules (Zhang et al., 2024a). In contrast, gSAP needs no auxiliary regularizer or post-hoc mapping. By directly optimizing gAP, it encourages similarities that are consistent across queries as a direct effect of the loss, and adding a consistency regularizer on top brings no further gain.

Evaluation metrics. The predominant evaluation is query-centric; mAP (Zhu, 2004) measures the ranking of the relevant samples according to the query, and Recall@k (Jegou et al., 2010) measures the proportion of queries for which at least one positive is ranked in the first k positions. Other metrics are classification-based, *e.g.* precision, recall, F1-score (Sivic & Zisserman, 2003), and retrieval-based, *e.g.* normalized discounted cumulative gain (NDCG) (Järvelin & Kekäläinen, 2002), and mean reciprocal rank (MRR) (Voorhees et al., 1999). However, these metrics emphasize per-query top-rank performance and are insensitive to dataset-level retrieval quality. gAP, also reported as micro-AP (μAP) in landmark recognition and copy detection (Weyand et al., 2020; Pizzi et al., 2022), instead aggregates all positive and negative pairs across queries to produce a single global precision–recall curve, effectively casting retrieval as a detection problem under a shared similarity threshold. We therefore adopt gAP as a primary metric and derive a surrogate loss to optimize it directly.

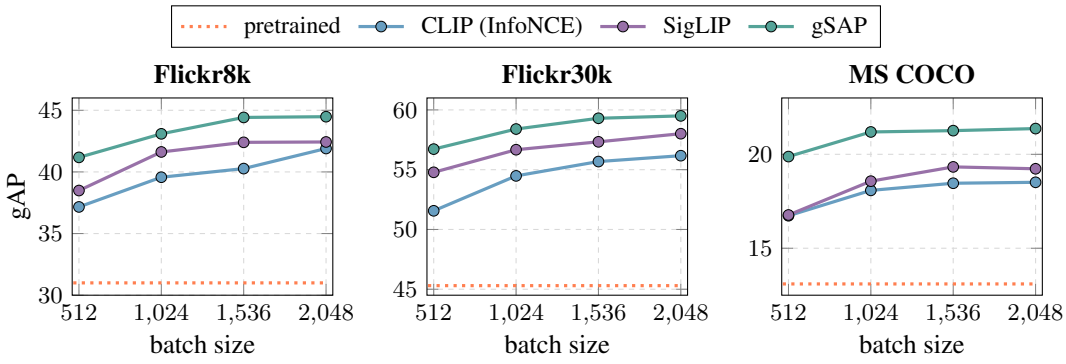


Figure A: Effect of batch size on zero-shot retrieval gAP. gAP comparison of gSAP with InfoNCE (CLIP) and SigLIP losses for varying batch sizes. All models fine-tune the pretrained OpenAI CLIP ViT-B/32 for 5 epochs on CC12M.

B Additional Results

B.1 Cross-modal learning

Audio–language. We assess whether the benefits of optimizing gAP extend beyond vision by repeating the protocol in the audio–language domain. We initialize LAION-CLAP (Wu et al., 2023) with an HTS-AT (Chen et al., 2022) base audio encoder that is pretrained on music, speech, and AudioSet (Gemmeke et al., 2017) and fine-tune for 10 epochs on AudioCaps (Kim et al., 2019) with the same three losses as in the main paper. Table A reports zero-shot audio–text retrieval on the Clotho (Drossos et al., 2020) validation and evaluation splits, together with zero-shot classification on ESC-50 (Piczak, 2015) and UrbanSound8K (Salamon et al., 2014). gSAP outperforms both CLIP and SigLIP on every column, with a clear margin in text-to-audio retrieval on the Clotho evaluation split, while SigLIP falls below the pretrained model on ESC-50.

Table A: Zero-shot retrieval and classification on audio datasets. Recall@1 on Clotho and top-1 accuracy on ESC-50 and UrbanSound8K, comparison of gSAP with the CLIP (InfoNCE) and SigLIP losses. All models fine-tune the pretrained LAION-CLAP HTS-AT base on AudioCaps.

method	Clotho val		Clotho eval		classification	
	a→t	t→a	a→t	t→a	ESC	US8K
pretrained	14.3	13.9	13.1	12.4	90.5	82.3
+ **CLIP**	16.3	15.5	15.2	15.5	90.9	83.3
+ **SigLIP**	16.0	15.2	15.0	14.4	90.0	83.3
+ **gSAP**	**16.7**	**15.9**	**15.7**	**16.7**	**91.7**	**83.7**

Vision–language. We extend the vision–language results of Section 4 with the same fine-tuning protocol. We first analyze the effect of batch size and of negative dropping with OpenAI CLIP ViT-B/32, then repeat the zero-shot retrieval and classification evaluations starting from two other models, OpenAI CLIP ViT-B/16 and LAION-400M (Schuhmann et al., 2021) CLIP ViT-B/32, and finally evaluate on an additional task, zero-shot composed image retrieval. gSAP is the best of the three losses in every setting.

Batch size sensitivity. We fine-tune OpenAI CLIP ViT-B/32 on CC12M for 5 epochs with varying batch sizes and compare InfoNCE and SigLIP with gSAP on zero-shot image–text retrieval on Flickr8k, Flickr30k, and MS COCO (Figure A). gSAP achieves the highest gAP on all three datasets at every batch size, with the largest margin at the smallest batch. At 512, it already exceeds InfoNCE trained with a four times larger batch on Flickr30k and MS COCO, so its gain does not rely on large batches.

Negative dropping. Table B evaluates negative dropping on zero-shot retrieval on the three image datasets with OpenAI CLIP ViT-B/32. At the same batch size, gSAP outperforms InfoNCE and SigLIP in every metric on all three datasets. Dropping 95% of the negatives (†) costs at most 0.9 gAP, and gSAP still outperforms both losses in gAP. The saved memory allows increasing the batch size from 2,304 to 4,096, which gives the best result in every metric. In contrast, training InfoNCE and SigLIP with the largest batch size that fits in GPU memory (3,072, ∗) improves InfoNCE, which still remains below gSAP at 2,304, and even drops the gAP of SigLIP.

Table B: Zero-shot retrieval with negative dropping on image datasets. gAP, mAP, and Recall@5 comparison of gSAP with InfoNCE (CLIP) and SigLIP losses. All models fine-tune the pretrained OpenAI CLIP ViT-B/32 for 5 epochs on CC12M. †: negative dropping with $\alpha = 95\%$. $*$: largest batch size that fits in GPU memory.

method	batch	Flickr8k					Flickr30k					MS COCO				
		gAP	mAP t→i	mAP i→t	R@5 t→i	R@5 i→t	gAP	mAP t→i	mAP i→t	R@5 t→i	R@5 i→t	gAP	mAP t→i	mAP i→t	R@5 t→i	R@5 i→t
pretrained	-	31.0	65.0	57.7	79.2	92.0	45.3	68.7	62.7	82.9	94.0	13.1	42.1	36.7	55.0	73.4
+ **CLIP**	2304	41.3	71.9	63.5	85.5	92.4	56.1	75.1	68.3	87.6	94.7	18.7	49.6	41.1	63.6	76.9
+ **SigLIP**	2304	43.2	72.5	64.6	86.0	92.7	58.5	76.3	69.5	88.5	95.7	19.4	50.6	41.8	64.7	77.5
+ **gSAP**	2304	45.1	73.2	65.6	86.7	93.3	59.7	76.6	70.5	88.9	96.0	21.4	51.3	43.1	65.3	79.0
+ **CLIP***	3072	42.3	72.1	63.9	85.8	92.7	57.0	75.6	68.7	88.1	95.0	19.9	50.0	41.7	64.0	77.7
+ **SigLIP***	3072	42.7	72.5	64.0	86.1	92.4	57.5	76.0	69.0	88.4	96.0	18.7	50.4	41.3	64.4	77.1
+ **gSAP**†	2304	44.2	73.0	65.6	86.5	92.6	59.2	76.6	70.4	88.6	96.0	20.7	51.2	42.9	65.3	79.0
+ **gSAP**$^{\dagger *}$	4096	**45.3**	**73.6**	**66.4**	**86.8**	**93.6**	**59.9**	**76.9**	**71.0**	**89.1**	**96.5**	**22.3**	**51.6**	**43.6**	**65.6**	**79.8**

Table C: Zero-shot retrieval on image datasets with other backbones and pretraining data. gAP, mAP, and Recall@5 comparison of gSAP with the CLIP (InfoNCE) and SigLIP losses. All models fine-tune the pretrained OpenAI CLIP ViT-B/16 or LAION-400M CLIP ViT-B/32 for 5 epochs on CC12M.

method	Flickr8k					Flickr30k					MS COCO				
	gAP	mAP t→i	mAP i→t	R@5 t→i	R@5 i→t	gAP	mAP t→i	mAP i→t	R@5 t→i	R@5 i→t	gAP	mAP t→i	mAP i→t	R@5 t→i	R@5 i→t
OpenAI CLIP ViT-B/16															
pretrained	30.6	65.6	56.8	80.4	90.2	44.7	69.8	64.5	83.2	94.7	12.2	42.3	37.6	55.0	74.0
+ **CLIP**	42.7	75.3	66.5	87.9	93.8	61.0	79.4	72.9	90.7	96.7	21.7	52.9	44.0	66.4	79.8
+ **SigLIP**	46.4	76.2	67.7	**88.7**	94.4	64.0	80.7	74.5	91.5	97.4	22.7	53.9	44.9	67.4	80.0
+ **gSAP**	**47.3**	**76.6**	**68.8**	**88.7**	**94.5**	**64.6**	**81.1**	**74.7**	**91.7**	**97.7**	**24.6**	**54.7**	**46.0**	**68.0**	**81.2**
LAION-400M CLIP ViT-B/32															
pretrained	38.9	67.6	61.1	81.4	91.8	52.1	70.5	64.9	83.8	93.7	17.3	46.4	39.4	60.0	76.3
+ **CLIP**	38.5	68.0	60.1	82.7	91.3	53.7	72.6	65.4	85.4	93.6	16.6	47.7	39.1	61.3	75.0
+ **SigLIP**	39.9	68.9	61.4	83.3	90.8	54.5	73.2	66.5	85.9	93.8	17.2	48.5	39.9	62.1	76.1
+ **gSAP**	**41.5**	**69.7**	**62.6**	**84.0**	**92.2**	**55.7**	**73.3**	**66.6**	**86.3**	**94.4**	**18.5**	**49.0**	**40.9**	**62.6**	**77.2**

Table D: Zero-shot retrieval on video datasets with other backbones and pretraining data. gAP and Recall@k comparison. Same setting as Table C.

method	ActivityNet							MSR-VTT							MSVD						
	gAP	text→video			video→text			gAP	text→video			video→text			gAP	text→video			video→text		
		R@1	R@5	R@10	R@1	R@5	R@10		R@1	R@5	R@10	R@1	R@5	R@10		R@1	R@5	R@10	R@1	R@5	R@10
OpenAI CLIP ViT-B/16																					
pretrained	11.7	25.2	55.4	69.0	22.7	50.0	62.0	5.0	15.6	32.6	41.4	31.8	52.8	63.9	17.1	36.5	63.9	73.3	55.0	74.8	82.1
+ **CLIP**	14.7	29.9	63.6	76.9	29.2	61.4	73.7	4.5	18.2	36.9	45.9	30.2	52.4	62.2	26.3	42.5	70.9	79.8	54.9	76.3	83.6
+ **SigLIP**	15.3	31.3	65.2	77.6	29.9	60.8	74.0	4.6	18.7	37.5	46.6	30.9	53.6	63.5	26.8	43.2	71.2	80.0	55.0	77.1	**85.2**
+ **gSAP**	**15.6**	**31.9**	**66.2**	**77.7**	**30.9**	**61.8**	**74.1**	**5.3**	**19.1**	**38.1**	**47.2**	**34.2**	**56.2**	**65.5**	**27.5**	**43.5**	**71.7**	**80.4**	**56.7**	**78.3**	**85.2**
LAION-400M CLIP ViT-B/32																					
pretrained	10.9	23.7	49.8	61.3	21.4	49.4	61.8	4.2	13.9	29.1	37.4	**28.7**	**48.7**	**59.3**	15.5	33.4	59.6	68.6	47.0	67.2	75.7
+ **CLIP**	10.3	24.2	54.1	68.2	21.7	51.5	64.7	3.1	14.6	31.0	40.1	22.5	43.1	53.1	19.1	36.3	65.8	75.1	47.5	69.4	78.7
+ **SigLIP**	10.5	24.0	54.2	68.1	22.1	51.7	65.0	3.4	14.9	31.7	40.6	23.8	45.8	55.8	**19.7**	37.3	66.2	75.7	49.0	69.8	78.1
+ **gSAP**	**11.4**	**25.1**	**55.2**	**68.5**	**23.2**	**51.9**	**65.1**	**4.8**	**15.4**	**32.2**	**41.2**	26.3	47.9	57.5	**19.7**	**37.6**	**66.7**	**76.0**	**50.1**	**72.2**	**80.0**

Table E: Zero-shot composed image retrieval on ConCon-Chi. gAP, mAP, mAP@k, and Recall@k comparison of gSAP with InfoNCE (CLIP) and SigLIP losses. All models fine-tune the pretrained OpenAI CLIP ViT-B/32 (top) or ViT-B/16 (bottom) for 5 epochs on CC12M.

method			mAP@k					Recall@k				
	gAP	mAP	@1	@5	@10	@25	@50	@1	@5	@10	@25	@50
OpenAI CLIP ViT-B/32												
pretrained	24.6	20.4	6.5	13.9	16.2	18.2	19.1	6.5	20.5	28.5	42.0	53.3
+ **CLIP**	30.6	25.2	8.7	17.8	20.7	22.9	23.8	8.7	24.4	34.1	46.7	57.4
+ **SigLIP**	31.5	26.4	9.1	18.9	21.8	24.1	25.0	9.1	25.7	35.0	48.5	59.6
+ **gSAP**	**31.7**	**26.9**	**9.5**	**19.3**	**22.3**	**24.6**	**25.5**	**9.5**	**26.2**	**35.6**	**49.2**	**60.0**
OpenAI CLIP ViT-B/16												
pretrained	28.6	24.7	8.3	18.0	20.4	22.4	23.3	8.3	24.5	32.6	45.3	55.4
+ **CLIP**	36.1	30.2	10.2	22.0	25.6	27.9	28.9	10.2	29.0	39.8	52.7	62.8
+ **SigLIP**	37.4	32.5	11.0	24.3	27.8	30.2	31.1	11.0	32.0	42.6	55.6	64.8
+ **gSAP**	**39.0**	**33.4**	**11.5**	**24.9**	**28.8**	**31.1**	**32.0**	**11.5**	**32.1**	**43.1**	**56.1**	**65.7**

Table F: Zero-shot classification on image and video datasets with other backbones and pretraining data. Top-1, top-5, and mean per-class recall (mR) comparison of gSAP with the CLIP (InfoNCE) and SigLIP losses. All models fine-tune the pretrained model in *italics* for 5 epochs on CC12M.

method	image datasets												video datasets					
	IN-1k			IN-V2			IN-Rendition			IN-Sketch			K-400		K-600		K-700	
	top-1	top-5	mR	top-1	top-5	mR	top-1	top-5	mR	top-1	top-5	mR	top-1	top-5	top-1	top-5	top-1	top-5
OpenAI CLIP ViT-B/16																		
pretrained	64.4	89.7	64.4	57.8	85.0	57.8	73.5	91.0	71.7	44.2	72.1	44.3	43.4	69.3	20.0	35.8	32.2	56.5
+ **CLIP**	65.3	90.7	65.3	58.8	86.1	58.8	76.0	91.9	74.1	47.2	75.7	47.3	45.0	71.4	21.7	37.3	33.8	59.7
+ **SigLIP**	66.3	91.1	66.3	59.4	86.5	59.5	77.0	**92.3**	**75.5**	48.3	76.8	48.4	45.4	71.4	21.8	37.2	34.5	59.8
+ **gSAP**	**66.6**	**91.2**	**66.6**	**60.0**	**86.6**	**60.1**	**77.1**	**92.3**	**75.5**	**49.0**	**77.0**	**49.0**	**45.8**	**71.9**	**22.0**	**37.5**	**35.0**	**60.4**
LAION-400M CLIP ViT-B/32																		
pretrained	60.2	85.9	60.2	52.3	79.4	52.3	70.8	88.6	69.1	46.5	72.6	46.5	35.5	60.0	32.7	56.7	25.3	47.2
+ **CLIP**	59.3	86.0	59.3	51.2	79.8	51.2	70.8	89.0	69.5	46.8	74.0	46.8	36.3	61.8	34.2	58.7	26.4	49.0
+ **SigLIP**	60.9	86.7	60.9	52.9	80.5	52.9	71.9	**89.3**	70.7	48.3	75.2	48.4	36.9	62.2	34.7	59.5	27.2	50.0
+ **gSAP**	**61.0**	**86.8**	**61.0**	**53.0**	**80.7**	**53.1**	**72.1**	**89.3**	**70.9**	**48.7**	**75.4**	**48.8**	**37.2**	**62.6**	**35.1**	**59.7**	**27.4**	**50.2**

Zero-shot retrieval. Tables C and D repeat the image and video retrieval evaluation starting from OpenAI CLIP ViT-B/16 and LAION-400M CLIP ViT-B/32. The ViT-B/32 trends hold: for both models, gSAP is the best or tied-best objective in every metric and direction on every dataset. With ViT-B/16, it doubles the gAP of the pretrained model on MS COCO. The LAION-400M model is harder to improve by fine-tuning on CC12M. gAP drops for InfoNCE on Flickr8k and MS COCO, and both InfoNCE and SigLIP degrade on ActivityNet and MSR-VTT, whereas gSAP improves gAP on all six datasets and is the only loss that improves every image retrieval metric. MSR-VTT is the hardest case. With ViT-B/16, InfoNCE and SigLIP fall below the pretrained model in gAP and video→text R@1, whereas gSAP improves both, outperforming SigLIP by 3.3 R@1. With LAION-400M, no loss recovers the pretrained video→text recall, but gSAP loses the least.

Zero-shot composed image retrieval. Table E reports results on ConCon-Chi (Rosasco et al., 2024), an instance-level composed image retrieval (CIR) benchmark where each query pairs a reference image of a specific object with a text modifier describing the target scene. Following the standard zero-shot CIR protocol, the retrieval score is the product of the image-to-image and text-to-image cosine similarities. All three losses improve over the pretrained model, and gSAP is best in every metric with both backbones; SigLIP stays close with ViT-B/32, but the gap widens with ViT-B/16.

Zero-shot classification. Table F repeats the zero-shot classification evaluation starting from the same two models, on the same four image and three video datasets and with the original prompt templates of each dataset. For both models, gSAP is the best or tied-best loss in every metric on all seven datasets. gSAP outperforms SigLIP by a small margin and InfoNCE by a clearer one. Starting from the LAION-400M checkpoint, InfoNCE even degrades top-1 accuracy of the pretrained model on ImageNet-1k and ImageNet-V2.

B.2 Self-supervised learning

In- and out-of-distribution performance. Table G breaks down the OOD mean. With ViT-S, gSAP is ahead on every set under both probes. With ViT-B, the two models tie under LP, each leading on one shifted set, while gSAP leads on every set under kNN. The OOD ordering follows the in-distribution ordering per probe: the better model in distribution is at least as good under shift, so neither loss trades in-distribution accuracy for robustness.

Transfer learning. Table H shows that the transfer gain comes mostly from fine-grained datasets. On the six sub-category datasets, gSAP is clearly ahead under both probes and at both scales, with the largest margins on Stanford Cars, FGVC Aircraft, and CUB-200, whereas on the four coarse object, scene, and texture sets the two losses are close, with InfoNCE slightly ahead at ViT-B. In fine-grained classes, hard negatives crowd the positives, which is exactly what a single shared threshold penalizes.

Robustness to corruptions. Table I reports ImageNet-C (Hendrycks & Dietterich, 2019) results by corruption family and severity. With ViT-S, gSAP leads under both probes. With ViT-B, the two models tie under LP on every family, while under kNN gSAP leads on every family and severity, by a margin that shrinks as corruptions grow stronger. gSAP also retains a slightly larger share of its clean accuracy under both probes, so its robustness is not merely inherited from higher clean accuracy.

Table G: In- and out-of-distribution classification with self-supervised pretraining. Top-1 accuracy on ImageNet-1k (ID) and three OOD datasets (ImageNet-V2/-R/-Sketch) of MoCo v3 models trained with InfoNCE or gSAP. Linear probing (LP) and kNN on the frozen backbone; LP is trained on ImageNet-1k and applied to every test set without dataset-specific training.

backbone	probe	loss	ID	OOD			
			IN-1k	IN-V2	IN-R	IN-Sketch	**avg**
ViT-S	LP	InfoNCE	73.2	61.1	32.0	21.3	38.2
		gSAP	**73.5**	**61.9**	**34.3**	**23.1**	**39.7**
	kNN	InfoNCE	68.4	54.9	23.0	16.5	31.5
		gSAP	**70.9**	**58.4**	**27.4**	**20.5**	**35.4**
ViT-B	LP	InfoNCE	**76.7**	**65.2**	**35.9**	24.9	**42.0**
		gSAP	75.6	63.9	35.8	**26.1**	**42.0**
	kNN	InfoNCE	71.4	58.6	29.6	21.8	36.6
		gSAP	**72.8**	**60.4**	**29.7**	**24.2**	**38.1**

Table H: Transfer learning with self-supervised pretraining. Top-1 accuracy on six fine-grained and four coarse-grained datasets of MoCo v3 models trained with InfoNCE or gSAP. Linear probing (LP) and kNN on the frozen backbone, using each dataset's own training split.

backbone	probe	loss	fine-grained (6)							coarse-grained (4)					
			Aircraft	CUB	Cars	Pets	Flowers	Food	**avg**	C-10	C-100	DTD	SUN	**avg**	**all (10)**
ViT-S	LP	InfoNCE	37.6	64.4	33.3	88.3	88.1	75.8	64.6	93.9	79.9	**73.3**	**63.1**	77.6	69.8
		gSAP	**45.2**	**76.4**	**48.3**	**91.7**	**89.6**	**76.2**	**71.2**	**94.9**	**81.0**	73.1	62.6	**77.9**	**73.9**
	kNN	InfoNCE	29.3	47.6	20.6	82.3	78.2	63.6	53.6	92.0	73.6	**69.3**	53.3	72.1	61.0
		gSAP	**35.1**	**65.0**	**28.5**	**88.6**	**81.5**	**66.0**	**60.8**	**93.7**	**75.8**	65.7	**55.0**	**72.5**	**65.5**
ViT-B	LP	InfoNCE	44.9	70.5	49.9	91.4	91.5	**79.3**	71.3	95.3	**83.2**	**76.6**	**65.7**	**80.2**	74.8
		gSAP	**53.9**	**78.9**	**64.3**	**93.1**	**92.9**	78.6	**76.9**	**96.0**	82.7	74.1	63.5	79.1	**77.8**
	kNN	InfoNCE	27.5	47.4	23.1	84.5	79.9	64.6	54.5	93.3	76.2	**72.7**	**56.2**	**74.6**	62.5
		gSAP	**35.6**	**64.2**	**33.7**	**89.9**	**83.6**	**66.3**	**62.2**	**94.8**	**77.4**	67.0	55.8	73.8	**66.8**

Table I: Robustness to corruptions with self-supervised pretraining. Top-1 accuracy on ImageNet-C of MoCo v3 models trained with InfoNCE or gSAP, with linear probing (LP, trained on ImageNet-1k) and kNN on the frozen backbone. Corruption families average over their corruptions and the five severities; severities average over the 15 corruptions. rel.: ImageNet-C accuracy as a share of the ImageNet-1k (ID) accuracy.

backbone	probe	loss	corruption family				severity						
			noise	blur	weather	digital	1	2	3	4	5	**all**	**rel.** (%)
ViT-S	LP	InfoNCE	41.4	36.2	46.0	53.9	62.1	53.5	46.7	36.2	24.5	44.6	60.9
		gSAP	**41.5**	**38.5**	**48.1**	**56.0**	**63.9**	**55.6**	**48.7**	**37.9**	**25.4**	**46.3**	**63.0**
	kNN	InfoNCE	35.3	31.2	40.8	49.4	58.2	49.0	41.1	29.9	18.9	39.4	57.7
		gSAP	**38.5**	**36.7**	**46.4**	**55.6**	**63.1**	**54.5**	**47.1**	**35.7**	**23.2**	**44.7**	**63.1**
ViT-B	LP	InfoNCE	45.5	41.9	**51.0**	**58.9**	66.6	58.6	52.0	41.6	**29.1**	49.6	64.7
		gSAP	**45.7**	**42.3**	50.9	58.7	**66.8**	**58.7**	**52.1**	**41.7**	**29.1**	**49.7**	**65.7**
	kNN	InfoNCE	41.9	39.6	49.3	56.1	64.1	56.4	49.6	38.7	26.3	47.0	65.9
		gSAP	**43.9**	**40.7**	**49.8**	**58.9**	**66.0**	**58.1**	**51.3**	**40.3**	**27.5**	**48.6**	**66.8**

Audio-based SSL. We also experiment with audio contrastive learning using the CLMR (Spijkervet & Burgoyne, 2021) framework, with the settings of Appendix C.2. Substituting InfoNCE with the proposed loss improves music tagging on MagnaTagATune (Law et al., 2009), as shown in Table J, with the larger gain in PR-AUC. PR-AUC, the area under the precision–recall curve, is the tagging analogue of gAP, as gSAP directly optimizes a closely related objective.

Table J: Music tagging with self-supervised pretraining. ROC-AUC and PR-AUC on MagnaTagATune of CLMR models trained with InfoNCE or gSAP, with linear probing on the frozen encoder.

method	ROC-AUC	PR-AUC
InfoNCE	88.7	35.6
gSAP	**89.3**	**37.1**

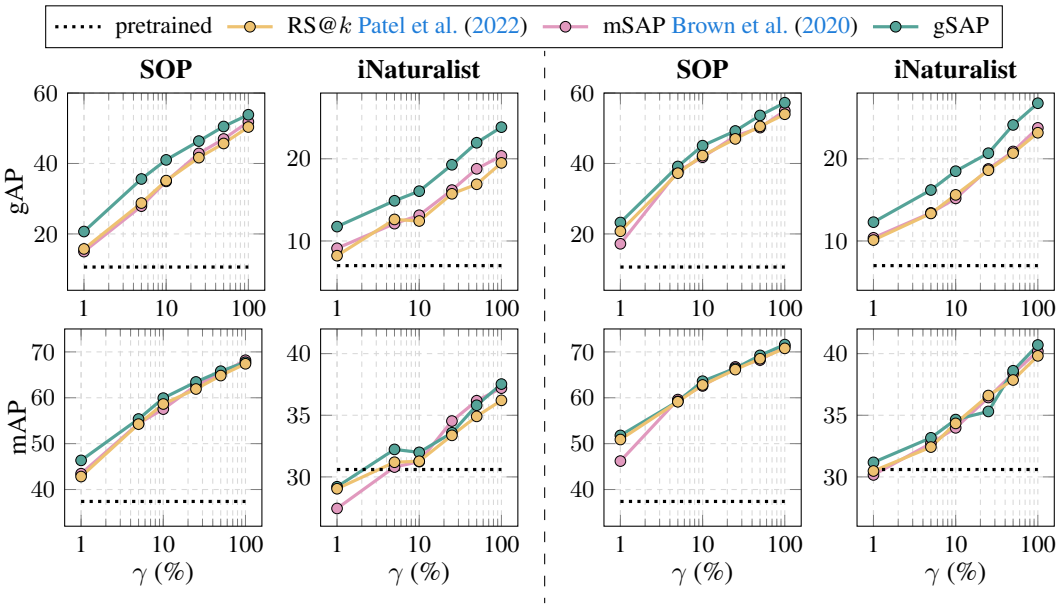


Figure B: Training with scarce annotated data. gAP (top) and mAP (bottom) of RS@k, mSAP, and gSAP on SOP and iNaturalist, fine-tuning DINOv3 under the scarce-data protocol of Appendix C.3, which varies the percentage of classes γ and the number of samples per class ρ (ρ=2: left pair, ρ=4: right pair). The full training sets have 11,318 (SOP) and 5,690 (iNaturalist) classes. Dotted: pretrained model.

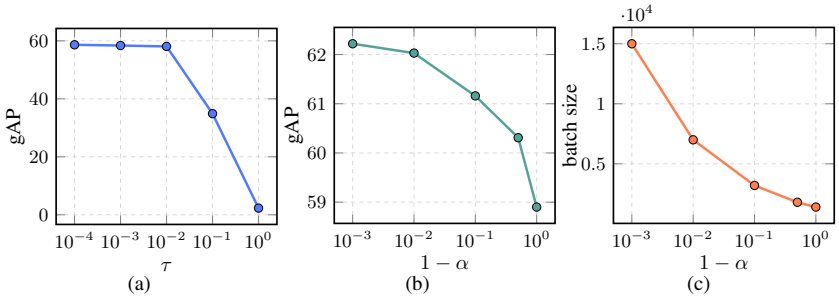


Figure C: Impact of hyper-parameters. (a) gAP for different temperatures τ, without negative dropping. (b), (c) Impact of negative dropping on gAP and on the batch size, set to the largest number of images that fits on one H200 GPU. All results on SOP, fine-tuning DINOv3 with full-scale training.

B.3 METRIC LEARNING

Comparison on limited data settings. We further evaluate the methods in a scarce data regime controlled by two parameters: γ, the percentage of training classes, and ρ, the number of samples per class. Evaluation settings are described in Appendix C.3. Figure B reports the performance of gSAP compared with the other two ranking surrogates (mSAP and RS@k) under different settings. gSAP is consistently the best in terms of gAP, with the largest margins at ρ=2, reaching up to 6 gAP over the second-best method. When training with a small γ, gSAP often surpasses competitors trained with more than twice the data. We attribute this to its formulation over all pairwise similarities, whereas competing losses operate row-wise and observe a single positive per row in this setting. Similarly, for ρ=4, gSAP remains the best, and trained models generally surpass the pretrained model, reflecting the benefit of additional annotations and multiple positives per row. The bottom row of the figure shows the mAP results, where all three losses perform similarly, achieving nearly identical mAP scores, in line with the results from the main paper. The pretrained model already has good mAP.

Impact of hyper-parameters. Figure C presents the gAP performance of the model trained with gSAP under different settings. Performance is very low for large τ values greater than 0.1; however, it peaks and plateaus as τ decreases below 0.01. Additionally, we assess the impact of the drop rate α on the batch size and, consequently, on performance. It is evident that the larger the batch size that fits in memory, which is determined by the drop rate, the better the performance.

Table K: Combination with similarity-scale regularizers. Recall@1 (R@1), mAP, and gAP when TCM or KoLeo is added to each loss. DINOv3, full-scale training, no negative dropping.

method	regularizer	SOP			iNaturalist		
		R@1	mAP	gAP	R@1	mAP	gAP
RS@k	–	86.7	72.5	56.7	81.0	44.5	26.6
mSAP	–	**86.9**	**72.9**	57.0	81.3	45.4	27.8
gSAP	–	86.8	**72.9**	**58.9**	**81.5**	**45.5**	**31.6**
RS@k	TCM	86.9	72.6	57.9	81.7	44.9	30.2
mSAP	TCM	86.5	72.0	56.7	**81.9**	**45.9**	31.0
gSAP	TCM	**87.0**	**72.8**	**58.7**	81.5	45.3	**31.4**
RS@k	KoLeo	83.5	66.4	49.7	79.9	41.0	24.6
mSAP	KoLeo	85.3	69.9	53.6	80.5	42.5	25.9
gSAP	KoLeo	**85.4**	**70.1**	**55.5**	**81.5**	**45.1**	**31.2**

Table L: AP approximations as mAP or gAP surrogates. Recall@1 (R@1), mAP, and gAP on SOP and iNaturalist. Each AP approximation is optimized as a surrogate of either mAP or gAP; the gAP surrogate with Smooth-AP is gSAP.

AP approx.	surrogate	SOP			iNaturalist		
		R@1	mAP	gAP	R@1	mAP	gAP
SoftBin	mAP	**87.0**	**73.0**	57.3	81.2	45.3	27.4
	gAP	86.7	72.7	58.6	81.3	44.8	30.6
BlackBox	mAP	86.5	72.3	56.6	80.7	44.5	26.2
	gAP	86.4	72.4	58.2	80.7	44.6	29.5
ROADMAP	mAP	85.5	70.1	54.2	80.5	44.0	29.5
	gAP	86.8	72.6	58.8	81.2	44.9	31.1
Smooth-AP	mAP	86.9	72.9	57.0	81.3	45.4	27.8
	gAP	86.8	72.9	**58.9**	**81.5**	**45.5**	**31.6**

Combination with similarity-scale regularizers. We add two regularizers proposed to constrain the similarity scale, TCM (Zhang et al., 2024b) and KoLeo (Sablayrolles et al., 2019), as auxiliary losses to each ranking loss (Table K). TCM helps in general the per-query losses and brings them closer to gSAP, but not consistently, *e.g.* performance of mSAP on SOP drops. Also, it adds nothing on top of gSAP, while KoLeo hurts gAP everywhere. TCM constrains the similarity scale across queries, which gSAP already achieves through its loss alone, whereas KoLeo spreads the embeddings apart and does not target this consistency. An explicit regularizer thus recovers part of what gSAP provides, at the price of extra hyper-parameters, and without consistent effectiveness.

Performance with other AP approximations. Table L replaces the default smooth approximation (Brown et al., 2020) with SoftBin (Revaud et al., 2019), BlackBox (Rolínek et al., 2020) and ROADMAP (Ramzi et al., 2021), each implemented both as an mAP and as a gAP surrogate, with its own hyper-parameters tuned based on the values recommended in the original work. Whatever the approximation, the gAP surrogate beats its mAP counterpart in gAP by a clear margin at similar mAP and R@1. The gain therefore comes from ranking the whole similarity matrix at once, not from the particular AP approximation.

Impact of negative dropping. Table M analyzes the effect of negative dropping by varying the drop percentage α. Performance remains remarkably stable even when discarding a large fraction of negatives. In particular, removing up to 99.9% of negatives, reducing their number by three orders of magnitude, has only a marginal impact. Performance degrades only when the remaining negatives become extremely few, suggesting that a minimal number is still necessary for effective training. Increasing the batch size while keeping $\alpha = 99\%$ improves gAP, showing that the proposed loss benefits from additional pairwise interactions even when most easy negatives are discarded.

Table M: Negative dropping and batch size. gAP, mAP, and R@1 of gSAP on SOP for different drop rates α, the fraction of the lowest negative similarities dropped during training. A higher drop rate allows a larger batch size on the same GPU, which improves gAP.

α	batch size	neg sim.	gAP	mAP	R@1
0%	1,000	1M	58.9	72.9	86.8
90%	1,000	100k	58.9	72.9	86.7
99%	1,000	10k	59.0	72.8	86.6
99.9%	1,000	1k	58.0	71.9	86.1
99.99%	1,000	100	48.0	62.5	81.0
99.999%	1,000	10	40.4	56.1	77.5
99.9999%	1,000	1	27.5	45.2	69.7
99%	7,000	500k	60.0	72.8	87.1

B.4 Tasks with distractor queries

Per-query metrics. Table N reports the per-query metrics of the four tasks with distractor queries, *i.e.* each benchmark's own metric, which by construction ignores those queries. All three fine-tuned losses improve over the baseline on every column, and gSAP is best or on par with the best on every column. The three fine-tuned losses differ far less on these measures than in gAP, which points to the consistency across queries as the source of the gAP differences, as discussed in the main paper.

Table N: Per-query metrics of the tasks with distractor queries. Each benchmark's own metric, which does not depend on the distractor queries. ACC: top-1 accuracy. BAKS: balanced accuracy on known samples. **Best** in bold, second best underlined.

method	DISC21	Met	ILIAS	WildlifeReID-10k	
	ACC	ACC	mAP	ACC	BAKS
baseline	71.6	56.3	33.2	79.2	76.8
InfoNCE	75.0	74.6	33.3	84.0	82.3
mSAP	74.6	**74.7**	33.4	86.3	**84.9**
gSAP	**75.1**	74.5	**34.6**	**86.4**	84.8

Operating point on DISC21. Table O reports R@P90, the metric of the SSCD benchmark, which is the recall obtained when one threshold, shared by all queries, is set for 90% precision over all query–reference pairs. It is a global rather than a per-query measure, which is why it is not in Table N. Under one threshold, gSAP retrieves the largest share of the copies, ahead of mSAP and well ahead of InfoNCE. SN lifts every loss and closes the gap between the two AP surrogates in terms of R@P90, so a post-hoc correction recovers much of the operating-point gap.

Table O: Recall at 90% precision on DISC21. Share of the true copies retrieved under one global threshold set for 90% precision, without and with score normalization (SN), in %. Same models as Table 8. **Best** in bold, second best underlined.

method	R@P90	$R@P90_{SN}$
InfoNCE	21.5	54.9
mSAP	26.4	57.1
gSAP	**28.6**	**57.1**

Table P: Open-set recognition on WildlifeReID-10k. A query of an unknown individual has to be rejected. AUROC separates known from unknown queries by their top-1 similarity. The other three columns use one threshold per source dataset, set for a 95% true positive rate on its known queries, and report the true negative rate (TNR) on the unknown queries, and the balanced accuracy on the known (BAKS) and on the unknown (BAUS) queries at that same threshold. All in %. Same models as Table 8. **Best** in bold, second best underlined.

method	AUROC	at 95% TPR		
		TNR	BAKS	BAUS
baseline	78.9	28.2	74.9	29.0
InfoNCE	84.8	46.8	79.8	47.1
mSAP	87.4	53.5	**82.2**	53.0
gSAP	**88.2**	**55.8**	81.9	**55.7**

Open-set recognition on WildlifeReID-10k. The queries of unknown individuals in WildlifeReID-10k are the distractor queries of the task, and a deployed system rejects them with a threshold on the similarity to the nearest database image. Table P evaluates this with one threshold per source dataset, set for a 95% true positive rate on its known queries. gSAP separates known from unknown queries best and, at the operating point, rejects the largest share of the unknown queries, well ahead of InfoNCE, while the two AP surrogates are on par on the known queries at the same threshold. gSAP thus gains on the side of the decision that depends on the similarity scale, while matching mSAP on the side that depends on the per-query ranking.

Backbones on ILIAS. Table Q repeats the ILIAS experiment, run with PE-L (Bolya et al., 2025), with two further frozen backbones, DINOv3 ViT-L/16 (Siméoni et al., 2025) and SigLIP2 ViT-L/16 (Tschannen et al., 2025), under the same protocol. The conclusions from the main paper hold for every backbone. gSAP has the best gAP with and without distractors and also the best mAP, so the gain is tied neither to one backbone nor to gAP alone. With SigLIP2, InfoNCE and gSAP are on par without the distractor queries and gSAP is clearly ahead with them, the same cross-query pattern as on Met. gAP falls sharply once distractor queries are added, for every loss and backbone. Score normalization raises gAP^{-} for two of the three backbones but never raises gAP with distractors, for any loss or backbone. On ILIAS a post-hoc correction thus helps only in the absence of distractor queries. Therefore, score normalization is not a straightforward solution for every task and setting.

Table Q: ILIAS with different backbones. Linear adaptation on frozen features with the protocol of Appendix C.4 for every backbone. mAP: per-query metric of the benchmark. gAP: all queries. gAP^{-}: distractor queries removed. SN: after score normalization. **Best** per backbone in bold, second best underlined.

backbone	method	mAP	gAP	gAP^{-}	gAP_{SN}	gAP^{-}_{SN}
PE-L@336 (Bolya et al., 2025)	**baseline**	33.2	3.5	17.1	2.7	20.3
	InfoNCE	33.3	7.4	22.0	**5.4**	22.7
	mSAP	33.4	5.6	19.6	3.9	21.2
	gSAP	**34.6**	**8.7**	**22.7**	4.3	**24.4**
DINOv3-L (Siméoni et al., 2025)	**baseline**	28.3	3.7	15.0	0.8	15.6
	InfoNCE	28.0	4.1	16.5	1.2	15.0
	mSAP	27.0	3.2	14.1	1.3	13.7
	gSAP	**29.5**	**4.5**	**17.3**	**1.5**	**16.4**
SigLIP2-L@512 (Tschannen et al., 2025)	**baseline**	31.3	3.8	17.6	3.8	20.9
	InfoNCE	31.8	6.1	21.0	5.0	25.7
	mSAP	31.6	4.7	19.3	4.6	23.6
	gSAP	**32.8**	**7.4**	**21.1**	**5.2**	**26.3**

C Evaluation settings

C.1 Cross-modal learning

Datasets. *Vision–language.* We follow the standard CLIP benchmark suite (Cherti & Beaumont, 2022), covering both cross-modal retrieval and zero-shot recognition across a broad range of domains. For zero-shot retrieval, we evaluate on three image datasets, Flickr8k (Hodosh et al., 2013), Flickr30k (Young et al., 2014), and MS COCO Captions (Chen et al., 2015), and three video datasets, ActivityNet (Caba Heilbron et al., 2015), MSR-VTT (Xu et al., 2016), and MSVD (Chen & Dolan, 2011) (Table 2). For zero-shot classification, the benchmark includes generic object and scene recognition datasets such as ImageNet-1k (Deng et al., 2009) and its OOD variants ImageNet-V2 (Recht et al., 2019), ImageNet-Sketch (Wang et al., 2019a), ImageNet-A and ImageNet-O (Hendrycks et al., 2021b), and ImageNet-R (Hendrycks et al., 2021a); Caltech101 (Fei-Fei et al., 2004); CIFAR-10 and CIFAR-100 (Krizhevsky, 2009); SUN397 (Xiao et al., 2010); STL10 (Coates et al., 2011); DTD (Cimpoi et al., 2014); FGVC Aircraft (Maji et al., 2013); Flowers-102 (Nilsback & Zisserman, 2008); Oxford-IIIT Pets (Parkhi et al., 2012); Stanford Cars (Krause et al., 2013); GTSRB (Stallkamp et al., 2011); SVHN (Netzer et al., 2011); VOC2007 (Everingham et al., 2010) (single- and multi-label); and FER2013 (Goodfellow et al., 2013). It further includes robustness, bias, and geographic benchmarks such as ObjectNet (Barbu et al., 2019) and Country211 (OpenAI, 2021); remote-sensing datasets such as EuroSAT (Helber et al., 2019) and RESISC45 (Cheng et al., 2017); medical datasets such as Diabetic Retinopathy (Kaggle & EyePACS, 2015) and PatchCamelyon (Veeling et al., 2018); and synthetic or structured reasoning benchmarks such as CLEVR (Johnson et al., 2017), dSprites (Matthey et al., 2017), DeepMind Lab (Beattie et al., 2016), SmallNORB (LeCun et al., 2004), KITTI distance (Geiger et al., 2012), MNIST (LeCun et al., 1998), and Rendered SST2 (Socher et al., 2013; Radford et al., 2021). For benchmarks derived from a common source dataset, such as ImageNet-A/O/R/Sketch or the different attribute splits of CLEVR, dSprites, and SmallNORB, we use the standard task definitions from the CLIP benchmark. Additionally, we evaluate zero-shot classification on three video datasets, Kinetics-400 (Kay et al., 2017), -600 (Carreira et al., 2018), and -700 (Carreira et al., 2019). *Audio–language.* We fine-tune on the AudioCaps (Kim et al., 2019) training split (45,178 audio–caption pairs) and evaluate zero-shot on datasets not used for fine-tuning: audio–text retrieval on the Clotho (Drossos et al., 2020) validation (1,045 clips) and evaluation split (1,045 clips), and classification on ESC-50 (Piczak, 2015) (2,000 clips, 50 classes) and UrbanSound8K (Salamon et al., 2014) (8,732 clips, 10 classes).

Implementation details. *Vision–language.* All experiments are implemented in OpenCLIP (Ilharco et al., 2021; Cherti et al., 2023)[2]. Our main experiments fine-tune the released OpenAI CLIP ViT-B/32 checkpoint (Radford et al., 2021); in the appendix we additionally fine-tune the OpenAI ViT-B/16 (Radford et al., 2021) and the OpenCLIP ViT-B/32 trained on LAION-400M (Schuhmann et al., 2021; Cherti et al., 2023). Fine-tuning is conducted on CC12M (Changpinyo et al., 2021) (10.97M image–text pairs, one pass per epoch without resampling), for 5 epochs with learning rate 10^{-5}, 2,000 warm-up steps followed by cosine decay, on 2 H200 GPUs (141 GB each). The global batch size is 1,792 for ViT-B/16 and 2,304 for ViT-B/32. We adopt the default OpenCLIP optimization setup:

[2] https://github.com/mlfoundations/open_clip

AdamW (Loshchilov & Hutter, 2019) with weight decay 0.2, $(\beta_1, \beta_2) = (0.9, 0.98)$ and $\epsilon = 10^{-6}$, fp16 automatic mixed precision, no gradient clipping, and no gradient accumulation, with the default OpenCLIP image preprocessing at 224×224. Both image and text encoders are updated jointly, without tower freezing, locking, or architectural changes, and we evaluate the final checkpoint without checkpoint selection. We compare three losses: the original CLIP InfoNCE loss (Oord et al., 2018; Radford et al., 2021), the SigLIP sigmoid loss (Zhai et al., 2023), and our gSAP loss. In all three, the logit scale is learnable and initialized from the pretrained checkpoint ($t = 100$); for SigLIP, the learnable bias is initialized to 0, since CLIP checkpoints do not provide one. For gSAP, we apply our differentiable surrogate to the same gathered, scaled similarity matrix, with $\tau = 10^{-3}$ and no negative dropping unless stated otherwise. *Audio–language.* We fine-tune the released LAION-CLAP (Wu et al., 2023)[3] checkpoint, which pairs an HTS-AT base audio encoder (Chen et al., 2022) with a RoBERTa text encoder (Liu et al., 2019) and is pretrained on music, speech, and AudioSet (Gemmeke et al., 2017). Both towers are fine-tuned jointly for 10 epochs. Audio is resampled to 48 kHz and cropped or zero-padded to 10 s, without feature fusion. We use AdamW with learning rate 10^{-5}, weight decay 0.2, $(\beta_1, \beta_2) = (0.9, 0.98)$, and $\epsilon = 10^{-6}$, 60 warm-up steps followed by cosine decay, gradient clipping at 1.0, bf16 mixed precision, and a global batch size of 1,024. The logit scale is learnable and initialized to $1/0.07$; for SigLIP, the bias is initialized to -10. *Both domains.* In all cases, we use distributed gathering so that the loss is computed from similarities over all samples in the global batch. Reported numbers are the mean over three seeds.

Evaluation protocol. *Vision–language.* Image benchmarks follow the CLIP benchmark (Cherti & Beaumont, 2022)[4] with its default class names and prompt templates. For video, we encode a single frame per video: the frame at 3 s for MSR-VTT, MSVD, and Kinetics, and the temporal midpoint for the much longer ActivityNet videos. Kinetics class embeddings average four action prompts ("a photo of a person {}.", "a video of a person {}.", "a photo of people {}.", "a video of {}."). We use the full MSR-VTT test split (2,990 videos, 59,800 captions), the 436 available MSVD test videos (18,098 captions), and paragraph-to-video retrieval on ActivityNet val_1, where each of the 1,063 available videos is paired with the concatenation of its captions. Kinetics-400, -600, and -700 are evaluated on their validation sets (19,881, 29,739, and 33,966 clips; 400, 599, and 700 classes). For retrieval we report gAP, i.e., average precision over the flattened query–item similarity matrix, together with Recall@k; for classification, top-1/top-5 accuracy and mean per-class recall. *Audio–language.* For retrieval, each Clotho clip is paired with its first reference caption, and we report R@1 in both directions against the full gallery. For classification, class embeddings average four prompts ("the sound of {}", "a sound of {}", "an audio recording of {}", "{} sounds"), and we report accuracy over all clips of each dataset.

C.2 Self-supervised learning

Datasets. Pretraining uses ImageNet-1k (Deng et al., 2009) without labels. In-distribution accuracy is measured on the ImageNet-1k validation set; distribution shift on ImageNet-V2 (Recht et al., 2019), ImageNet-R (Hendrycks et al., 2021a) and ImageNet-Sketch (Wang et al., 2019a), reported as their mean; corruption robustness on ImageNet-C (Hendrycks & Dietterich, 2019), 15 corruptions at 5 severities. Transfer uses ten datasets: six fine-grained ones—FGVC Aircraft (Maji et al., 2013), CUB-200 (Wah et al., 2011), Stanford Cars (Krause et al., 2013), Oxford-IIIT Pets (Parkhi et al., 2012), Flowers-102 (Nilsback & Zisserman, 2008) and Food-101 (Bossard et al., 2014)—and four coarse-grained object, scene and texture sets—CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), DTD (Cimpoi et al., 2014) and SUN397 (Xiao et al., 2010).

Implementation details. We take the MoCo v3 (Chen et al., 2021) recipe unchanged[5]. We train a ViT-S/16 and a ViT-B/16 on ImageNet-1k for 300 epochs with batch size 4,096 and learning rate $4 \cdot 10^{-3}$. We replace the InfoNCE loss by gSAP computed over the same gathered query–key similarity matrix, with the positives on the diagonal and temperature $\tau = 0.2$. Nothing else changes: augmentations, momentum encoder, projection and prediction heads, optimizer and schedule are those of the official code. The baselines are the released 300-epoch MoCo v3 checkpoints with their published linear heads. Re-training the official recipe reproduces them to within seed variance.

[3] https://github.com/LAION-AI/CLAP
[4] https://github.com/LAION-AI/CLIP_benchmark
[5] https://github.com/facebookresearch/moco-v3

Evaluation protocol. All numbers are top-1 accuracy on frozen features. *Linear probing* (LP) on ImageNet-1k, its OOD variants and ImageNet-C uses the official 90-epoch linear head on the CLS features of the frozen backbone (the published MoCo v3 heads for the baselines and heads trained with the identical recipe for gSAP). Every OOD set is classified with the ImageNet-1k head and no dataset-specific training. ImageNet-R covers 200 classes and the softmax is restricted to them. Transfer datasets have no published head, so LP there is a linear classifier per dataset on frozen features, mean of three seeds (seed standard deviation ≤ 0.11 on every set). *k-Nearest Neighbors* (kNN) is a weighted vote over the k=20 nearest training features, with k fixed and not tuned (cosine similarity, weights $\exp(s/0.07)$) on the full 1.28M-image ImageNet-1k train set, or on each transfer set's own training split. Neither the choice of k nor of the linear head affects any comparison: k=20 is within 0.15 of every model's best $k \in \{5, \ldots, 200\}$ with the same model ranking at every k, and the official head is 0.2–0.7 points above a short probe on the same features, again with an identical ranking.

Audio. For audio we follow CLMR (Spijkervet & Burgoyne, 2021) unchanged[6]: a SampleCNN encoder (Lee et al., 2017) is pretrained on the raw waveforms of MagnaTagATune (Law et al., 2009) with the augmentation chain, batch size, optimizer and schedule of the official code. We replace InfoNCE by gSAP on the same similarity matrix between the two augmented views of a batch, with temperature $\tau = 0.05$, and keep every other setting of the InfoNCE run. Evaluation is linear probing on the frozen encoder for the 50 tags of MagnaTagATune, scored by ROC-AUC and PR-AUC. We pretrain three seeds per loss, each with its own probe, and report the mean.

C.3 Metric learning

Datasets. We conduct experiments on two widely used datasets: Stanford Online Products (SOP) (Oh Song et al., 2016) and iNaturalist (Van Horn et al., 2018). We follow the dataset split used in prior work (Brown et al., 2020; Patel et al., 2022). SOP consists of products. The training split has 59,551 images from 11,318 classes, and the test split contains 60,502 images from 11,316 classes. iNaturalist consists of natural images of wildlife and plants. The training split has 325,846 images across 5,690 classes, while the test split contains 136,093 images across 2,452 classes. For SOP, all test images serve as queries. Due to the large size of the iNaturalist test set, for mAP and gAP computation, we sample 4 images per test class as queries, resulting in 9,808 queries.

Evaluation protocol. For all runs, we tune hyper-parameters with Optuna (Akiba et al., 2019), optimizing gAP. Tuning on a different metric, *i.e.* mAP, yields the same results in several verification runs. Specifically, we split the training set into 70% for training and 30% for validation and run 10 trials to search for the learning rate $\eta \in [10^{-5}, 5 \cdot 10^{-4}]$ and the temperature $\tau \in [10^{-4}, 5 \cdot 10^{-2}]$. For the other metric learning losses, we tune their main hyper-parameter and keep the rest fixed to the recommended settings of the original papers, and for the regularizers of Table K we also tune their weight. After tuning, the final model is trained with the selected hyper-parameters on the whole training set and evaluated on the test set. The full process, tuning, training and evaluation, is run for three random seeds.

Implementation details. For the comparison of metric learning losses, we use the ViT-S variant of the DINOv3 (Siméoni et al., 2025) vision encoders. Our implementation is in PyTorch (Paszke et al., 2019) and builds on the open-source repository of Recall@k[7]. Unless specified otherwise, we use a batch size of 1,000 and a chunk size of 200 for the multi-stage backpropagation of Revaud et al. (2019); Patel et al. (2022). We train for 100 epochs with the AdamW (Loshchilov & Hutter, 2019) optimizer and fix the embedding dimension d to 512 in all experiments. For the comparison with the state of the art, we experiment with ResNet50 (He et al., 2016) and ViT-B/16 (Dosovitskiy et al., 2021), trained for 200 epochs following the implementation details of Patel et al. (2022). Experiments are conducted on Nvidia A100 and H200 GPUs with 40 GB and 141 GB of memory, respectively. Each run is executed on a single GPU.

[6] https://github.com/Spijkervet/CLMR
[7] https://github.com/yashvarpatel/RecallatK_surrogate

Thresholding protocol. For Table 6 and Figure 4, the test classes are split in half, identically for every model, into *fitting* classes, used only to select a threshold, and *evaluation* classes, on which it is applied. The evaluation queries are the test images of the evaluation classes, and the gallery is the whole test split minus the query. On the fitting classes we select one threshold for maximal F1 and one as the lowest threshold with precision 0.90. $F1_g$ is the mean F1 of the evaluation queries at the first, and R@P90 the recall over all evaluation pairs at the second. $F1_q$ uses the F1-optimal threshold of each evaluation query instead. Thresholds are searched with a step of 5×10^{-4}.

Scarce-data protocol. For Figure B, we simulate limited annotations. From the training classes with at least 4 samples, we keep a random $\gamma\%$ of the classes and draw ρ images per class, which gives a reduced dataset $\mathcal{D}^{\gamma,\rho}$. Its classes are split into $\mathcal{D}^{\gamma,\rho}_{\text{train}}$ and $\mathcal{D}^{\gamma,\rho}_{\text{val}}$ for hyper-parameter tuning, the model with the best validation performance at the end of training is selected, and the final model is trained with these hyper-parameters on all classes of $\mathcal{D}^{\gamma,\rho}$ and evaluated on the full test set. We use $\gamma \in \{1, 5, 10, 25, 50, 100\}$ and $\rho \in \{2, 4\}$, and run the whole process with three random seeds for each pair.

C.4 Tasks with distractor queries

Datasets. DISC21 (Douze et al., 2021) is an image copy detection benchmark: queries are edited copies of reference images, and the query set is dominated by distractor queries that have no source in the reference set. The Met dataset (Ypsilantis et al., 2021) is instance-level recognition of artworks, where a query is either a photograph of an exhibit in the database or a distractor photograph of no exhibit. ILIAS (Kordopatis-Zilos et al., 2025) is instance-level image retrieval at scale, retrieving images of particular object instances from a database dominated by distractor images. WildlifeReID-10k (Adam et al., 2025) is re-identification of individual animals across many species. Each task keeps the training and evaluation code of its official repository, with only the loss replaced.

Implementation details. Unless tuned on the validation split of a task, gSAP uses $\tau = 0.001$ and negative dropping with $\alpha = 90\%$, and every task trains with batches of 4,096 samples. All tuned hyper-parameters are selected by grid search. InfoNCE and mSAP are trained with exactly the same recipes and their hyper-parameters are tuned in the same way. All results are averaged over three random seeds. On DISC21 we follow the SSCD repository[8] and fine-tune a DINOv3 ViT-B/16 at 224 px for 30 epochs with AdamW. The absolute learning rate, the temperature τ and the weight λ of the entropy regularizer of SSCD are tuned on the dev queries. On Met we follow the official repository[9] and fine-tune a DINOv3 ViT-B/16 at 512 px for 10 epochs with AdamW, one warm-up epoch and cosine decay. The learning rate and the temperature τ are tuned on the validation set, as are the k and the temperature of the kNN classifier, in accordance with the original work. On ILIAS we follow the official repository[10] and train a linear adaptation layer on UnED (Ypsilantis et al., 2023), on top of publicly available extracted features. We report PE-L@336 (Bolya et al., 2025) for the main comparison as it achieves the best performance. We follow the default training settings from the original work, with the only difference that we draw four samples per class in every batch. The DINOv3-L and SigLIP2-L@512 runs of Table Q use exactly the same protocol. On WildlifeReID-10k we follow the WildlifeDatasets toolkit (Čermák et al., 2024)[11] and fine-tune a DINOv3 ViT-B/16 at 224 px for 100 epochs with AdamW, learning rate $5 \cdot 10^{-5}$ and 10 warm-up epochs, again with four samples per class in every batch. Score normalization (SN) (Pizzi et al., 2022) reduces every similarity of a query by β times the mean similarity of the query to its k nearest neighbors in a background database. On DISC21 we use the setting of the original work, $\beta = 1$ and $k = 3$ with the DISC21 training set as background, and on ILIAS $\beta = 0.4$ and $k = 1$ with UnED as background.

Baselines. On DISC21 the baseline is the SSCD-large model (Pizzi et al., 2022), on Met the R50-SWSL backbone with a kNN classifier, the best result reported in the original work (Ypsilantis et al., 2021), on ILIAS the method of the original work (Kordopatis-Zilos et al., 2025), and on WildlifeReID-10k MiewID-msv3 (Otarashvili et al., 2024), the reference model of the benchmark.

[8] https://github.com/facebookresearch/sscd-copy-detection
[9] https://github.com/nikosips/met
[10] https://github.com/ilias-vrg/ilias
[11] https://github.com/WildlifeDatasets/wildlife-tools

Metrics. gAP is computed over all query–database pairs, including those of the distractor queries, and gAP^- after removing the distractor queries. The relative drop $(\text{gAP}^- - \text{gAP})/\text{gAP}^-$ is the share of gAP lost to their negatives. Per-query metrics (accuracy, mAP, and the balanced accuracy on known samples, BAKS) are those of each benchmark and ignore the distractor queries. ILIAS has no distractor queries of its own, so we simulate them. Every query retrieves its top-$1{,}000$ images, and queries are processed in chunks of 100. For a chunk, gAP^- ranks the lists of its queries, and gAP the lists of all queries restricted to the positives of the chunk's instances and the YFCC100M distractors, so that every query outside the chunk acts as a distractor query. Both are averaged over the chunks. On WildlifeReID-10k the distractor queries are those of individuals with no image in the database, and we also report open-set metrics, where such a query has to be rejected by a threshold on its top-1 similarity. On DISC21 we also report R@P90, the recall at the global threshold that gives 90% precision.

D Detailed results

Detailed results. Tables R and S report per-dataset zero-shot classification results on the full CLIP benchmark (Cherti & Beaumont, 2022) for ViT-B/32 and ViT-B/16, respectively: top-1, top-5, and mean per-class recall (mR) on 38 tasks (37 classification datasets, with VOC2007 evaluated as single- and multi-label), each averaged over three runs. Results on the three image–text retrieval benchmarks are reported in Tables 1 and C. Fine-tuning with gSAP yields the strongest overall performance on both backbones. Averaged over all 108 classification scores, it improves over the pretrained model, the CLIP InfoNCE loss, and SigLIP by 1.16, 0.49, and 0.64 for ViT-B/32, and by 1.43, 0.65, and 0.77 points for ViT-B/16. On retrieval, the task that gSAP directly optimizes, the gains over the pretrained model and InfoNCE are considerably larger. Where gSAP is not the best, two patterns account for most cases. First, on some datasets fine-tuning on CC12M brings little or no benefit regardless of the loss, and the pretrained model remains competitive (*e.g.* ObjectNet, MNIST); the structured dSprites and SmallNORB tasks also stay close to chance level for all models. Second, CLIP or SigLIP occasionally perform slightly better on a specific metric (*e.g.* CLEVR distance, Flowers, or Pets), but the differences are typically small. In contrast, cases where gSAP underperforms by a clear margin are rare.

	top-1				top-5				mean per-class recall			
dataset	**Pre.**	**CLIP**	**SigLIP**	**gSAP**	**Pre.**	**CLIP**	**SigLIP**	**gSAP**	**Pre.**	**CLIP**	**SigLIP**	**gSAP**
ImageNet and robustness												
ImageNet-1k	59.6	60.7	61.3	**61.8**	86.4	87.4	87.8	**87.9**	59.6	60.7	61.3	**61.8**
ImageNet-V2	52.9	53.6	54.1	**54.3**	80.8	82.0	82.1	**82.5**	52.9	53.6	54.2	**54.3**
ImageNet-Sketch	38.9	42.3	43.2	**43.7**	66.9	70.4	71.6	**71.7**	39.0	42.3	43.3	**43.7**
ImageNet-A	28.1	27.8	**28.4**	27.9	60.3	60.0	**60.7**	60.1	**29.4**	29.1	**29.4**	29.0
ImageNet-O	44.9	**48.1**	47.8	**48.1**	75.0	77.6	**78.3**	78.2	46.1	**49.5**	48.8	49.3
ImageNet-R	67.0	68.3	69.1	**69.4**	87.2	87.9	88.2	**88.3**	64.9	66.8	67.8	**68.1**
ObjectNet	**43.4**	42.5	41.8	42.2	**69.4**	68.3	68.3	68.5	**41.7**	41.6	41.0	41.3
Country211	**15.7**	15.0	15.3	**15.7**	**37.7**	36.2	37.0	37.4	**15.7**	15.0	15.3	**15.7**
Generic objects and scenes												
Caltech101	80.5	82.5	83.0	**83.2**	94.0	96.1	96.1	**96.7**	85.7	86.6	87.8	**88.0**
CIFAR-10	86.5	89.3	89.4	**90.0**	99.2	99.6	99.5	**99.7**	86.5	89.4	89.4	**90.0**
CIFAR-100	62.4	64.8	65.8	**66.0**	87.1	88.3	**89.2**	**89.2**	62.4	64.8	65.8	**66.0**
SUN397	61.4	64.7	64.9	**65.9**	89.9	92.1	92.3	**92.7**	61.3	64.6	65.1	**65.5**
STL10	95.7	96.6	96.5	**96.9**	99.8	**100.0**	**100.0**	**100.0**	95.7	96.6	96.5	**96.9**
DTD	41.5	44.8	44.6	**46.0**	73.0	76.6	77.3	**77.4**	41.5	44.7	44.6	**46.0**
FGVC Aircraft	**16.0**	13.5	14.2	13.6	42.3	41.3	**43.2**	41.8	**16.0**	13.4	14.1	13.6
Flowers-102	60.2	59.2	59.4	**61.2**	82.7	83.7	**84.0**	**84.0**	**60.4**	57.8	58.4	60.1
Oxford-IIIT Pets	84.0	**85.5**	85.0	85.1	98.0	**99.3**	**99.3**	**99.3**	83.7	**85.4**	84.9	84.9
Stanford Cars	51.5	50.8	52.8	**53.1**	86.0	87.3	88.6	**89.1**	51.8	50.7	52.6	**53.0**
GTSRB	26.6	30.8	31.9	**33.6**	65.1	63.8	65.1	**67.5**	25.9	27.7	26.8	**30.1**
SVHN	9.4	23.6	19.7	**24.6**	61.8	69.7	70.7	**72.6**	13.2	18.7	20.3	**20.6**
VOC2007	74.1	75.4	75.1	**76.6**	**95.1**	94.2	93.7	94.8	78.2	**79.3**	78.9	79.0
VOC2007 (multi-label)†	75.1	81.6	**82.1**	80.6	–	–	–	–	–	–	–	–
FER2013	38.9	**41.7**	39.6	40.6	88.9	93.3	91.7	**94.6**	35.9	38.4	**38.8**	37.5
Remote sensing and medical												
EuroSAT	45.3	52.4	49.8	**52.7**	90.1	88.5	90.1	**90.6**	44.1	52.3	48.9	**52.6**
RESISC45	50.9	50.7	**51.0**	50.0	84.5	84.5	**85.5**	84.8	51.5	51.4	**51.7**	50.8
Diabetic Retinopathy	10.2	**20.5**	18.4	8.0	–	–	–	–	21.8	**23.9**	21.4	22.4
PatchCamelyon	**58.3**	55.5	53.9	56.3	–	–	–	–	**58.3**	55.6	53.9	56.3
Synthetic and structured												
CLEVR (count)	**25.2**	19.3	18.6	19.2	**80.7**	70.7	70.5	73.0	**24.8**	19.1	18.3	18.8
CLEVR (distance)	15.7	**16.5**	15.8	16.3	92.2	89.8	87.5	**94.2**	16.3	16.1	**16.7**	16.6
dSprites (orientation)	**2.4**	1.9	1.9	2.1	10.6	11.1	10.5	**11.2**	**2.4**	1.9	1.9	2.1
dSprites (x position)	**3.4**	3.0	3.3	3.1	15.7	15.4	**16.2**	14.9	**3.4**	3.0	3.2	3.1
dSprites (y position)	**3.2**	**3.2**	**3.2**	**3.2**	**15.6**	**15.6**	15.5	15.4	**3.1**	**3.1**	**3.1**	**3.1**
DMLab	**18.5**	17.9	16.2	17.5	84.0	84.0	82.3	**84.5**	17.0	**17.7**	16.6	**17.7**
KITTI (distance)	22.8	**27.5**	24.1	26.3	–	–	–	–	35.7	34.4	**35.8**	34.1
SmallNORB (azimuth)	**6.6**	5.7	5.6	5.8	**28.4**	27.8	28.2	27.6	**6.7**	5.8	5.7	5.9
SmallNORB (elevation)	11.1	12.9	11.8	**13.1**	54.1	**58.1**	57.4	57.2	11.0	12.8	11.7	**12.9**
MNIST	**50.9**	30.7	30.7	33.0	85.9	83.0	78.6	**87.0**	**50.6**	31.4	31.2	34.2
Rendered SST2	59.3	**61.1**	56.6	59.3	–	–	–	–	59.2	**61.1**	56.6	59.2
Average	42.1	43.2	42.8	**43.3**	71.8	72.2	72.3	**73.2**	42.0	42.3	42.2	**42.8**

Table R: Zero-shot classification on the full CLIP benchmark with ViT-B/32. Top-1, top-5, and mean per-class recall of the pretrained OpenAI CLIP ViT-B/32 (Pre.) and of its fine-tuning on CC12M with the CLIP (InfoNCE), SigLIP, and gSAP losses; mean over three seeds. †Mean average precision in place of top-1. Top-5 is omitted (–) for tasks with five or fewer classes. Averages are taken over the tasks with a value in each group (38, 33, and 37).

	top-1				top-5				mean per-class recall			
dataset	**Pre.**	**CLIP**	**SigLIP**	**gSAP**	**Pre.**	**CLIP**	**SigLIP**	**gSAP**	**Pre.**	**CLIP**	**SigLIP**	**gSAP**
ImageNet and robustness												
ImageNet-1k	64.4	65.3	66.3	**66.6**	89.7	90.7	91.1	**91.2**	64.4	65.3	66.3	**66.6**
ImageNet-V2	57.8	58.8	59.4	**60.0**	85.0	86.1	86.5	**86.6**	57.8	58.8	59.5	**60.1**
ImageNet-Sketch	44.2	47.2	48.3	**49.0**	72.1	75.7	76.8	**77.0**	44.3	47.3	48.4	**49.0**
ImageNet-A	44.5	44.3	**44.8**	44.7	75.3	76.1	**76.9**	**76.9**	**43.4**	42.7	43.2	42.5
ImageNet-O	41.0	41.9	42.3	**42.6**	67.8	72.7	72.8	**73.5**	42.9	43.1	**43.9**	**43.9**
ImageNet-R	73.5	76.0	77.0	**77.1**	91.0	91.9	**92.3**	**92.3**	71.7	74.1	**75.4**	**75.4**
ObjectNet	**54.7**	51.6	51.3	51.0	**78.3**	76.2	76.1	76.3	**52.3**	50.6	50.5	50.1
Country211	20.2	19.5	20.1	**20.3**	44.5	44.1	44.8	**45.0**	20.2	19.5	20.1	**20.3**
Generic objects and scenes												
Caltech101	81.7	82.8	83.5	**84.0**	95.4	95.7	96.6	**97.0**	86.9	88.3	**88.8**	88.5
CIFAR-10	88.2	91.7	91.2	**92.0**	98.9	**99.7**	99.5	**99.7**	88.2	91.7	91.2	**92.0**
CIFAR-100	62.0	68.4	68.5	**70.6**	85.4	90.2	90.5	**91.4**	62.0	68.4	68.5	**70.6**
SUN397	63.2	64.5	65.9	**66.4**	91.3	92.7	93.5	**93.6**	62.5	65.8	66.7	**67.5**
STL10	97.6	97.7	97.6	**97.9**	**100.0**	**100.0**	99.9	99.9	97.6	97.7	97.6	**97.9**
DTD	40.4	46.6	46.3	**47.3**	71.6	75.5	76.0	**76.2**	40.4	46.6	46.3	**47.3**
FGVC Aircraft	**20.0**	17.5	18.5	19.3	49.0	47.3	49.9	**50.7**	**20.0**	17.5	18.5	19.2
Flowers-102	65.3	62.0	64.8	**65.5**	82.2	83.5	**84.9**	84.1	63.9	61.1	62.7	**64.1**
Oxford-IIIT Pets	85.2	88.2	**89.7**	89.4	98.8	**99.5**	**99.5**	99.4	85.0	88.1	**89.7**	89.4
Stanford Cars	57.5	57.7	59.1	**59.9**	91.2	92.0	92.2	**92.6**	57.5	57.4	58.9	**59.7**
GTSRB	41.1	45.8	43.5	**46.7**	65.9	71.7	72.8	**74.1**	33.8	35.8	36.8	**38.7**
SVHN	**38.2**	29.1	22.2	30.4	**77.6**	69.1	65.6	73.5	**37.8**	29.3	25.5	30.0
VOC2007	**78.0**	76.0	75.7	75.3	**95.5**	91.4	92.7	92.3	**81.0**	79.5	79.1	80.5
VOC2007 (multi-label)†	78.5	83.7	**83.9**	**83.9**	–	–	–	–	–	–	–	–
FER2013	43.6	47.1	**47.7**	46.9	**96.0**	93.8	92.0	92.9	40.0	41.5	**43.8**	42.9
Remote sensing and medical												
EuroSAT	43.0	50.8	**55.5**	54.9	86.5	93.4	**94.2**	93.6	43.3	49.8	**54.8**	53.5
RESISC45	53.5	56.3	56.2	**57.7**	85.9	90.2	89.9	**90.9**	54.1	56.7	56.6	**58.3**
Diabetic Retinopathy	3.8	**52.7**	28.1	32.5	–	–	–	–	20.2	**28.7**	26.1	26.1
PatchCamelyon	60.2	**61.1**	52.3	58.3	–	–	–	–	60.2	**61.1**	52.3	58.3
Synthetic and structured												
CLEVR (count)	20.8	19.9	19.5	**21.3**	**79.8**	71.5	69.4	76.0	20.4	19.6	19.1	**20.9**
CLEVR (distance)	15.9	**16.1**	15.7	16.0	**80.1**	80.0	**80.1**	80.0	16.5	**17.7**	16.7	17.3
dSprites (orientation)	2.1	**2.7**	2.4	2.4	10.8	11.8	**13.1**	11.9	2.0	**2.7**	2.4	2.5
dSprites (x position)	**3.4**	3.1	3.1	3.1	**16.4**	15.4	16.1	15.4	**3.3**	3.0	3.1	3.1
dSprites (y position)	**3.1**	**3.1**	**3.1**	**3.1**	15.5	15.6	**16.0**	15.6	**3.1**	**3.1**	**3.1**	**3.1**
DMLab	**17.3**	14.6	14.8	14.5	**83.7**	82.7	82.6	83.1	**17.4**	16.1	16.4	15.9
KITTI (distance)	**31.6**	15.0	15.2	14.3	–	–	–	–	**37.6**	26.8	25.2	28.4
SmallNORB (azimuth)	4.6	**5.6**	5.3	5.3	26.3	27.4	**27.7**	27.5	4.7	**5.7**	5.4	5.4
SmallNORB (elevation)	11.4	11.3	**11.6**	11.2	**55.8**	53.9	54.7	55.2	11.4	11.2	**11.5**	11.1
MNIST	**69.6**	59.2	59.9	63.1	92.5	90.3	**93.0**	91.4	**69.4**	59.3	60.4	63.5
Rendered SST2	60.7	**63.0**	60.9	62.5	–	–	–	–	60.8	**63.0**	60.9	62.5
Average	45.8	47.3	46.6	**47.6**	73.8	74.2	74.5	**75.0**	45.4	45.8	45.8	**46.7**

Table S: Zero-shot classification on the full CLIP benchmark with ViT-B/16. Top-1, top-5, and mean per-class recall of the pretrained OpenAI CLIP ViT-B/16 (Pre.) and of its fine-tuning on CC12M with the CLIP (InfoNCE), SigLIP, and gSAP losses; mean over three seeds. †Mean average precision in place of top-1. Top-5 is omitted (–) for tasks with five or fewer classes. Averages are taken over the tasks with a value in each group (38, 33, and 37).

E THEORETICAL ANALYSIS

The per-query rankings of a model fix mAP and Recall@k, but not gAP, which also depends on how the similarities of different queries are interleaved in the global ranking. Our goal is to find the interleaving that gives the *theoretical best* gAP given the fixed per-query rankings, and to measure how close each loss comes to it. To make this precise, let G be the group of transformations that apply a *different* strictly increasing bijection of $\mathbb{R}$ to every row of S,

$$(\psi \cdot S)_{i,j} = \psi_i(s_{i,j}), \qquad \psi = (\psi_1, \dots, \psi_M), \quad \psi_i : \mathbb{R} \to \mathbb{R} \text{ strictly increasing bijection}, \tag{11}$$

i.e. a per-query transformation of the similarities that leaves each query's own ranking untouched. We call the set of all matrices $\psi \cdot S$ with $\psi \in G$ the *orbit* of S. Its members have the same per-query rankings as S and differ only in how the rows are interleaved.

Lemma 1. *Any function of S that depends only on the ranking within each row takes the same value on every member of the orbit of S. mAP and Recall@k are such functions, whereas gAP is not.*

A loss that reads only within-row rankings is thus constant on every orbit and cannot tell one interleaving of the rows from another, which is exactly what gAP varies over. mSAP is such a loss (in the limit $\tau \to 0$), since it reads only differences of similarities within a row. gSAP, on the other hand, by flattening S, is the minimal change to mSAP that makes the loss depend on the interleaving.

Decomposing gAP into rankings and consistency. By the lemma, gAP depends on the per-query rankings, which fix the orbit of S, and on the consistency of the similarities across queries, which fixes where on the orbit S lies. The *ceiling* $C(S, Y)$ is the largest gAP on the orbit of S, *i.e.* the gAP the model would reach if its similarities were made perfectly consistent across queries with its per-query rankings unchanged. The *efficiency* $\mathrm{eff} = \mathrm{gAP}/C$ is the share of the ceiling the model actually realizes, and measures how consistent its similarities are across queries. Since $\mathrm{gAP} = \mathrm{eff} \cdot C$, two models can differ in gAP because their rankings allow different ceilings or because they realize different shares of them. Differencing the product separates the two exactly,

$$\Delta\mathrm{gAP} \;=\; \underbrace{\overline{\mathrm{eff}}\,\Delta C}_{\text{ranking}} \;+\; \underbrace{\overline{C}\,\Delta\mathrm{eff}}_{\text{consistency}}, \tag{12}$$

where Δ is the difference between the two models and bars denote the mean of their values. The *ranking* term is the part of the difference due to better per-query rankings, *i.e.* a higher ceiling, at the average efficiency, and the *consistency* term the part due to more consistent similarities across queries, *i.e.* a higher efficiency, at the average ceiling. The proof of the lemma and the computation of C follow the results.

Results. In Table T, where Δ is gSAP minus the loss of the row, the ceiling C of gSAP is within 0.6 points of those of mSAP and RS@k, so the three losses learn rankings that allow the same gAP, and what gSAP adds is a higher efficiency on both datasets. Accordingly, Eq. (12) attributes its whole gain over mSAP to the consistency term. Among the pairwise and proxy losses, Multi-Similarity has the highest ceiling and, with Contrastive, an efficiency close to that of gSAP, whereas the others have ceilings well below it. Contrastive loses mainly on rankings, ArcFace and Proxy Anchor mainly on rankings on SOP but on consistency on iNaturalist, and Triplet has by far the lowest efficiency. Optimizing the global ranking thus buys consistency at no cost in per-query ranking. Nonetheless, even gSAP leaves much of the achievable consistency unrealized, a gap that per-query losses cannot act on, since they are blind to the interleaving of the queries.

Table T: Rankings *vs*. consistency across queries. gAP, ceiling C and efficiency eff of the losses of Table 5 on two datasets. rank and cons are the terms of Eq. (12) relative to gSAP on each row, in gAP points.

method	SOP					iNaturalist				
	gAP	C	eff	rank	cons	gAP	C	eff	rank	cons
Contrastive	53.7	76.3	70.4	+4.1	+1.1	28.0	48.2	58.1	+3.3	+0.3
Triplet	50.0	78.5	63.7	+2.4	+6.5	18.2	47.3	38.5	+3.2	+10.2
ArcFace	50.3	72.8	69.1	+6.5	+2.1	23.0	48.9	47.0	+2.6	+6.0
Proxy Anchor	51.9	75.9	68.4	+4.3	+2.7	21.8	48.9	44.6	+2.5	+7.3
Multi-Similarity	57.3	80.3	71.4	+1.2	+0.4	29.9	51.8	57.7	+1.2	+0.5
RS@k	56.7	81.7	69.4	+0.2	+2.0	26.6	53.2	50.0	+0.3	+4.7
mSAP	57.0	82.0	69.5	+0.0	+1.9	27.8	54.1	51.4	-0.2	+4.0
gSAP	58.9	82.0	71.8	–	–	31.6	53.8	58.7	–	–

Proof of Lemma 1. Let $\psi \in G$ act on row i by ψ_i. Since ψ_i is strictly increasing,

$$\psi_i(s_{i,l}) \geq \psi_i(s_{i,j}) \iff s_{i,l} \geq s_{i,j}, \qquad \text{so} \qquad h\big(\psi_i(s_{i,l}) - \psi_i(s_{i,j})\big) = h(s_{i,l} - s_{i,j})$$

for all $j, l \in \Omega$, with h the step function of Eq. (4). The precision of Eq. (3) and the AP of a row are built from such within-row comparisons only, so they take the same value at S and $\psi \cdot S$, and so does mAP, their average over rows. Recall@k is the fraction of queries with a positive among their k highest similarities,

$$\text{Recall@}k(S, Y) = \frac{1}{M} \sum_{i=1}^{M} h\Big(k - \min_{j \in \mathcal{P}(Y_i)} \sum_{l \in \Omega} h(s_{i,l} - s_{i,j})\Big), \tag{13}$$

and is again invariant for the same reason. So is any function that reads S only through within-row comparisons, hence every per-query ranking metric and every loss built from within-row rank estimates.

gAP is different. It computes AP on $\mathrm{vec}(S)$, so its sums also contain comparisons $h(s_{i',j'} - s_{i,j})$ between similarities of different queries, $i' \neq i$. Since ψ_i and $\psi_{i'}$ are chosen independently, there is no guarantee that $h\big(\psi_{i'}(s_{i',j'}) - \psi_i(s_{i,j})\big)$ is equal to $h(s_{i',j'} - s_{i,j})$. Two examples show that gAP indeed varies on an orbit.

Example 1. Two queries a and b with one positive and one negative each, $a^+ = 0.8$, $a^- = 0.3$, $b^+ = 0.6$, $b^- = 0.2$. Keeping row a and shifting row b by $\psi_b(s) = s - 0.5$ changes the global ranking from

$$a^+,\, b^+,\, a^-,\, b^- \quad (\text{gAP} = 1) \qquad \text{to} \qquad a^+,\, a^-,\, b^+,\, b^- \quad \big(\text{gAP} = \tfrac{1}{2}\big(1 + \tfrac{2}{3}\big) = 0.833\big),$$

while both per-query rankings, and with them mAP $= 1$ and Recall@$1 = 1$, are unchanged.

Example 2. M queries with N items each, every query having a single positive ranked first in its row, so mAP $= 1$ on the whole orbit. If every positive is ranked above every negative, gAP $= 1$. Now stack the rows one below the other, *e.g.* by $\psi_i(s) = s - 2(i-1)$ for similarities in $[-1, 1]$, so that the whole of row 1 comes first in the global ranking, then the whole of row 2, and so on. Row i occupies global positions $(i-1)N + 1$ to iN, and its positive, the first item of the row, sits at position $(i-1)N + 1$. The positives above it are those of rows 1 to $i - 1$, so the precision at the positive of row i is $i/((i-1)N + 1)$, and averaging over the M positives gives

$$\text{gAP} = \frac{1}{M} \sum_{i=1}^{M} \frac{i}{(i-1)N + 1}.$$

The first term is 1. For $i \geq 2$, dropping the $+1$ and using $i/(i-1) \leq 2$ gives $i/((i-1)N+1) \leq 2/N$, so the remaining $M - 1$ terms sum to at most $2(M-1)/N$ and

$$\text{gAP} \;\leq\; \frac{1}{M}\Big(1 + \frac{2(M-1)}{N}\Big),$$

which approaches $1/M$ as $N \to \infty$. On this one orbit, gAP thus ranges from 1 down to about $1/M$ while mAP stays at 1. □

Computing C. The maps ψ_i of Eq. (11) keep the order within each row and are otherwise arbitrary, so the global rankings of the members of the orbit of S are exactly the interleavings of the M per-query ranked lists that keep the order within each list, and $C(S, Y)$ is the largest gAP over these interleavings. Let query i have $K_i = |\mathcal{P}(Y_i)|$ positives and let $K = \sum_i K_i$. As in the denominator of Eq. (3), the rank of a positive $j \in \mathcal{P}(Y_i)$ within its own list is

$$\rho_{i,j} = \sum_{l \in \Omega} h(s_{i,l} - s_{i,j}), \tag{14}$$

and the position of a positive entry $t \in \mathcal{P}(\mathrm{vec}(Y))$ in the global list is

$$\pi_t = \sum_{t'=1}^{MN} h\big(\mathrm{vec}(S)_{t'} - \mathrm{vec}(S)_t\big). \tag{15}$$

The ranks ρ are fixed on the orbit, whereas the positions π change with the interleaving. Index the positives of query i so that $\rho_{i,1} < \cdots < \rho_{i,K_i}$, and all K positives of the matrix so that $\pi_1 < \cdots < \pi_K$. Exactly k positives are at or above position π_k, so the precision at the k-th positive of the global ranking can also be computed as k/π_k, and gAP $= (1/K)\sum_{k=1}^{K} k/\pi_k$. C is the maximum of this mean over the interleavings, and we compute it in three steps. An optimal interleaving is built from blocks of each list, the blocks are ordered by a scheduling rule, and C is bounded from both sides, as the rule need not be optimal for gAP.

Blocks. Consider the consecutive negatives immediately before a positive. If some of them come from other queries, moving them to just after the positive keeps the order within every list and raises gAP, since that positive moves up and no other positive moves. In an optimal interleaving, the consecutive negatives immediately before a positive all come from the query of that positive, so each list enters as a chain of blocks, where block j of query i consists of the negatives between positives $j-1$ and j followed by positive j. Its length is $\rho_{i,j} - \rho_{i,j-1}$, with $\rho_{i,0} = 0$, and its density, positives per item, is the reciprocal of its length, since each block holds exactly one positive. The negatives after the last positive of a query go to the end of the global list, below every positive, where they cost nothing.

Ordering the blocks. The blocks of a query must keep their order, so the task is to interleave M chains of blocks, and we use the ratio rule of Sidney (1975) for it. Along each chain, consecutive blocks are merged into segments until the density is non-increasing along the chain, and the segments of all queries are then ordered by decreasing density, ties kept in chain order. The merging step is the pool-adjacent-violators algorithm (Provost & Fawcett, 2001; Fawcett & Niculescu-Mizil, 2007), and the ordering step is the probability ranking principle (Robertson, 1977) applied to all pairs jointly.

Bounds. Every feasible interleaving has gAP at most C. The rule gives one, and a second is built greedily, without merging, by appending at each step the densest of the blocks that come next in their chains. We report the larger of their two gAP values as C, which can only be too low, and bound how far it is from the true maximum as follows. Take the segments in the order of the rule and pretend that the positives of each segment are spread evenly over its items. Its gAP is an upper bound on C, since at every depth it holds at least as many positives as any feasible interleaving and moving positives up can only raise gAP. The true C thus lies between the reported value and this upper bound, whether or not the rule is optimal, and on the full-size runs of Table T the two differ by at most 7×10^{-7} (SOP) and 6×10^{-6} (iNaturalist), far below the 0.1 points at which C is reported.

Checks. We also compared the rule with an exhaustive search over all order-respecting interleavings on instances small enough for it. On 400 random synthetic instances of two to four lists, the rule found the optimum in 382 and the reported value, the better of the rule and the greedy schedule, in 393, with mean deviations from the optimum of 4×10^{-4} and 1×10^{-4}. On 100 instances per dataset of three real query rows truncated to their top 40 items, the rule found the optimum in 99 (SOP) and 82 (iNaturalist) and the reported value in 100 and 86, with a mean deviation of about 10^{-4}. On the full-size runs the rule never fell below the greedy schedule.

Example. Two queries a and b with six items each. Query a has positives at ranks $1, 2, 6$ and b at ranks $2, 3$, so their APs are 0.833 and 0.583 and mAP is 0.708 on the whole orbit. The blocks of a have lengths $1, 1, 4$ and densities $1, 1, \frac{1}{4}$, already non-increasing, so none are merged. The blocks of b have lengths $2, 1$ and densities $\frac{1}{2}, 1$. The denser block comes second, so the two merge into one segment of density $\frac{2}{3}$. Ordering by density gives the first two blocks of a, the segment of b, then the last block of a, *i.e.* the global list $++\,|\,-++\,|\,---+$ followed by the trailing negatives of b, with positives at positions $1, 2, 4, 5, 9$ and

$$\text{gAP} = \tfrac{1}{5}\left(\tfrac{1}{1} + \tfrac{2}{2} + \tfrac{3}{4} + \tfrac{4}{5} + \tfrac{5}{9}\right) = 0.821,$$

which an exhaustive search over all 924 order-respecting interleavings confirms to be C. Putting the whole list of a before that of b gives positions $1, 2, 6, 8, 9$ and gAP $= 0.711$, and the list of b first gives $2, 3, 7, 8, 12$ and gAP $= 0.502$. All three lie on one orbit, so mAP stays at 0.708 while gAP ranges from 0.50 to 0.82.

Remark. The maps in G act on the rows independently, so when the queries are themselves database items, as in Table T, the orbit contains asymmetric similarity matrices, which are infeasible with ℓ_2-normalized embeddings. C is therefore an upper bound on what an embedding could reach, by post-hoc score normalization (Auckenthaler et al., 2000; Jégou et al., 2007; Pizzi et al., 2022) or by a loss on all similarities jointly such as gSAP, and it bounds every loss alike.