WASSERSTEIN-BARYCENTER PARTS FOR ROBUST VISUAL RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual retrieval has to work under severe nuisance more and more often, where changes in sensor, viewpoint or modality alter the pixels but not the identity of the target. Such changes wash out texture and colour, while the layout of recurring local parts often survives. Part-based methods already find these parts, but they usually summarise each part with a single mean vector, the slot mean. We argue that this is insufficient: After this averaging step, only the part mean is explicitly retained, and thus, not sufficient to preform accurate retrieval. To address this issue, we propose WBPRIOR, a lightweight plug-and-play part representation that brings a layout prior into retrieval. Each patch predicts a mean vector and a positive variance, which together define a diagonal Gaussian; the square root of the variance is a learned scale vector. Each part then becomes the *2-Wasserstein barycenter* of the Gaussians assigned to it: the single Gaussian closest on average to all of them in optimal-transport distance. For diagonal Gaussians this barycenter has an exact closed form, so every part carries a mean vector and a learned scale vector with negligible additional aggregation cost. In a controlled study that changes only this step, the slot mean breaks down on corrupted images and does perform worse than the global image descriptor used alone. The barycenter repairs this failure, and it helps most where appearance is most degraded. The same head makes each of seven published cross-view retrievers more robust. It is also competitive with specialised systems on drone geo-localisation, sketch retrieval and aerial-ground person re-identification. Robust retrieval depends not only on how parts are found, but also on how each part is represented.

1 INTRODUCTION

Visual retrieval finds the gallery image that shows the same place, object or person as a query. It underpins place recognition, drone geo-localisation (locating a drone photo by matching it to satellite tiles), person re-identification (finding the same person in other cameras) and sketch-based search. In practice the query and the gallery rarely look alike. A drone photo and a satellite tile, a ground camera and an aerial camera, or a photo and a hand-drawn sketch can show the same target with very different texture, colour, scale and sensor statistics. We call such a change a *nuisance*: it alters the pixels but not the identity of the target. When it is strong, we speak of *retrieval under severe nuisance*. Under it, the fine appearance cues that global descriptors rely on tend to vanish. What tends to remain is coarser: the recurring local parts of a scene and the way they are arranged, which we call its *layout*.

Part-based methods aim to use this layout signal, but they spend their effort on finding parts, not on representing them. Slot attention (Locatello et al., 2020) and self-supervised vision transformers (ViTs) (Caron et al., 2021; Oquab et al., 2024) can group image patches into parts, and part-matching frameworks align those parts across views. Once a part is found, however, most pipelines summarise it with one pooled vector, the mean of its patch features (the *slot mean*). This significantly reduces the performance of retreival due to loss of important information. As such, a question is *can a single, principled layout prior, designed once, make retrieval robust under severe nuisance, and what should its part representation keep beyond the mean?*

To demonstrate this, consider a drone view of a building corner half in shadow at dusk. The patches of that corner agree roughly on where they are, but they differ in texture, contrast and colour. The slot mean averages their features into one vector and keeps nothing else, so two parts with similar

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.

mean vectors look the same to the retrieval descriptor. When appearance changes, the mean drifts, and the descriptor has no second signal to fall back on.

To address this problem, we propose WBPRIOR, a lightweight plug-and-play head that gives each part this second signal by representing it as a distribution instead of a point. It acts as a *layout prior*, building into a patch-based retriever the assumption that the parts of a scene are more stable than their fine appearance. Each patch predicts a mean vector with a linear head and a positive variance with a Softplus head (a linear layer followed by $\log(1 + e^x)$, which keeps the output positive). Together they define a diagonal Gaussian, with its own variance for every feature dimension. Each part is represented by the *2-Wasserstein barycenter* of the Gaussians assigned to it, which is the distribution closest on average to them in squared 2-Wasserstein distance. For diagonal Gaussians, writing $\mathbf{s} = \sqrt{\mathbf{v}}$ for the learned scale vector (the per-dimension standard deviation), the squared 2-Wasserstein distance splits into a mean term and a standard-deviation term,

$$W_2^2(\nu_a, \nu_b) \;=\; \|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\|_2^2 + \|\mathbf{s}_a - \mathbf{s}_b\|_2^2, \tag{1}$$

where $\nu_a = \mathcal{N}(\boldsymbol{\mu}_a, \mathrm{diag}(\mathbf{v}_a))$ and $\nu_b = \mathcal{N}(\boldsymbol{\mu}_b, \mathrm{diag}(\mathbf{v}_b))$. The barycenter therefore has an exact closed form, which is a weighted average of the mean vectors and of the scale vectors. A slot mean keeps only the first term, as if $\mathbf{s} \equiv \mathbf{0}$. WBPRIOR keeps both, and a learned read-out passes both into the retrieval embedding, so two parts with similar mean vectors can still be told apart. The barycenter is rebuilt from each image's own patches at training and test time alike. The scale vector is then learned for retrieval, and it is not the empirical scatter of the patch means and not a calibrated uncertainty.

The rest of the head is deliberately conventional. A partially fine-tuned DINOv2 backbone (Oquab et al., 2024) (ViT-S/14 by default, ViT-B/14 for scaling) gives patch tokens and a CLS token, a learned token that summarises the whole image. For each image, a spectral assignment groups the patches into parts using the eigenvectors of a patch-similarity graph, so no prototype bank is stored. A part-selection gate keeps the useful parts, and a fusion gate combines the pooled part descriptor with the CLS descriptor into one embedding. Training uses *paired views*, two images of the same target such as a drone photo and its satellite tile. A functional-map term and a debiased Sinkhorn divergence make corresponding parts agree across the pair, next to standard metric-learning and proxy-alignment losses, a masked-patch reconstruction loss and an exponential-moving-average (EMA) teacher. These training terms are dropped at test time, where retrieval is one forward pass and cosine similarity, with no test-time augmentation or re-ranking. The same head can also be attached as a parallel branch to published retrievers, including convolutional ones, without changing their backbone, classifier or original losses.

This design enables a simple, testable prediction. If the learned scale carries useful information, then replacing the slot mean with the barycenter, and changing nothing else, should improve retrieval, most of all when appearance is degraded. A matched five-seed study on drone-to-satellite retrieval confirms it (Table 1). With the backbone, losses, assignment and schedule fixed, the slot mean collapses on corrupted test images and falls below a CLS-only baseline without the part branch. The barycenter repairs this failure, and its gain under corruption is much larger than on clean images. The scale vector matters most on patches with strong local structure, and a full covariance matrix adds little over the diagonal form at a much higher cost. We then test the head on three kinds of appearance change. They are cross-view geometry on the drone benchmarks SUES-200 (Zhu et al., 2023), DenseUAV (Dai et al., 2024) and University-1652 (Zheng et al., 2020); a change of camera platform on the CARGO aerial-ground person re-identification benchmark (Zhang et al., 2024); and a change of modality on the sketch benchmarks FSCOCO (Chowdhury et al., 2022) and SketchyCOCO (Gao et al., 2020). Added to seven published cross-view retrievers, the head makes every one more robust to corruption (Table 2). As a standalone model it is the most robust on DenseUAV and University-1652 and slightly behind the strongest baseline on SUES-200 (App. Tab. 16). On clean images it matches or exceeds recent methods with the smallest parameter and compute budget among those that report it (Table 3). With the larger backbone it is best on both sketch benchmarks and on six of eight CARGO columns (Table 4), and in a sketch retrieval study it improves all 15 pretrained encoders we tested. What transfers is the representation: each task still keeps its own training losses.

Our contributions are:

- **A novel representation principle.** A discovered part should keep a learned scale vector next to its mean vector, not a single mean alone. We package this principle as WBPRIOR, a lightweight plug-and-play layout prior for retrieval under severe nuisance.

- **An exact, cheap instantiation.** Within each image, the 2-Wasserstein barycenter of a part's diagonal Gaussians gives that part a mean vector and a learned scale vector in closed form, and a learned read-out passes both into the descriptor used at test time.
- **Controlled evidence and broad tests.** A matched study that swaps only the aggregation rule isolates the effect of the barycenter. A plug-in study on seven published retrievers, standalone comparisons on three drone benchmarks, and transfer to sketch retrieval and aerial-ground re-identification test how far it carries over.

2 Related Work

Part-based recognition has a rich history (Marr & Nishihara, 1978; Sun et al., 2018); modern variants include slot attention (Locatello et al., 2020; Kori et al., 2024), patch grouping on ViT tokens (Dosovitskiy et al., 2021; Luo et al., 2023), and part-matching frameworks (Wang et al., 2022; Dai et al., 2022). These methods mainly study which patches belong together and how parts are aligned across views. We study a different question: after the patches of a part have been grouped, what information should its representation keep? Most methods use one point vector; WBPrior replaces it with the barycenter of the part's patch distributions.

Probabilistic retrieval is also established. Probabilistic Face Embeddings (Shi & Jain, 2019), DistributionNet (Yu et al., 2019), and probabilistic cross-modal embeddings (PCME and PCME++) (Chun et al., 2021; Chun, 2024) map whole images to Gaussians, while Bayesian Triplet (Warburg et al., 2021) models posterior uncertainty. Other work uses distributions for composed queries (Tang et al., 2025) or models uncertainty for person re-identification at the part level (Sun et al., 2021) or the identity level (Li et al., 2026). Some methods train with Wasserstein losses, or embed labels and concepts as elliptical distributions (a family that includes Gaussians) in a Wasserstein space (Frogner et al., 2015; Muzellec & Cuturi, 2018). These methods embed the whole object as one distribution. WBPrior instead uses the 2-Wasserstein barycenter of one image to summarise each part.

Local descriptor aggregation is a second point of comparison. Many methods pool local features into one global vector: Fisher vectors (Perronnin et al., 2010), NetVLAD (Arandjelović et al., 2016), generalised mean (GeM) pooling (Radenović et al., 2019), second-order covariance pooling (Lin et al., 2015; Li et al., 2018), and recent visual place recognition (VPR) aggregators such as AnyLoc (Keetha et al., 2024) and BoQ (Ali-bey et al., 2024). They pool relative to a codebook, a bank of learned queries, or pairwise feature statistics. WBPrior instead forms parts inside each image, gives each patch a learned Gaussian, and aggregates each part separately. Wasserstein distances and barycenters are established tools (Agueh & Carlier, 2011; Cuturi & Doucet, 2014; Álvarez-Esteban et al., 2016); we use the barycenter to turn the distributions of one part into an exact, closed-form retrieval representation.

3 Wasserstein-Barycenter Parts

WBPrior combines a conventional global descriptor with a part descriptor. A self-supervised ViT backbone, DINOv2 (Oquab et al., 2024), produces N patch tokens and a CLS token; we use ViT-S/14 by default and ViT-B/14 to test scaling, with part of the backbone fine-tuned (App. A2.1). The global branch keeps the CLS token. The part branch groups the patches of each image into parts, summarises each part with a barycenter rather than a mean, and, during training, encourages corresponding parts to agree across paired views. A learned fusion gate combines the global and part descriptors into one ℓ_2-normalised 768-D vector for cosine retrieval.

The methodological contribution is the part representation of Sec. 3.2, which keeps both the mean vector and the learned scale vector of every part. Part assignment (Sec. 3.1) and the training losses (Sec. 3.3) are standard engineering that supports the representation; their full specification is in Appendices A2.2, A2.4 and A2.5. Between Gaussians, the 2-Wasserstein distance is also called the Bures-Wasserstein (BW) distance, and we write *BW barycenter* for our part representation. We defer the description of its geometry to Appendix A1.3.

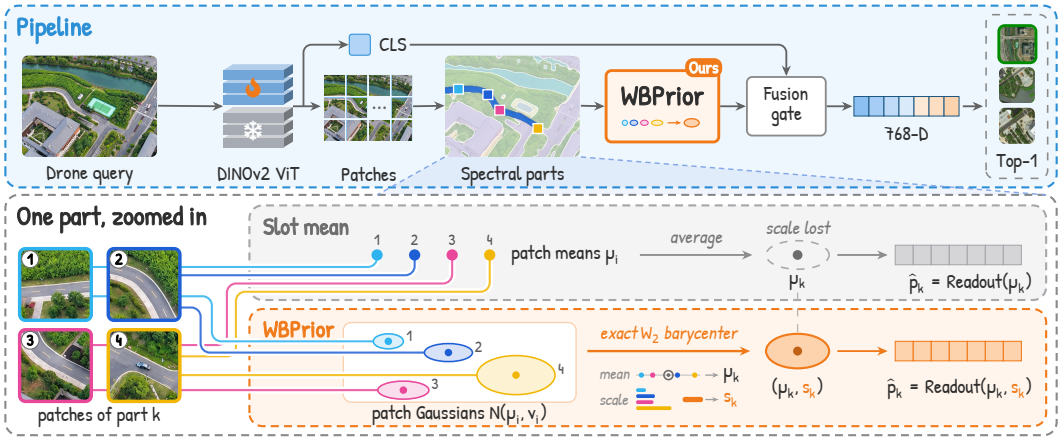


Figure 1: **Distributional part representation.** Both summaries receive the same part assignment, computed from the image itself. The slot mean collapses the assigned patches to one mean vector. WBPRIOR maps every patch to a diagonal Gaussian and combines them with their exact 2-Wasserstein barycenter, which keeps the mean vector together with a learned scale vector. The scale vector of the part is the weighted average of the standard deviations predicted by its patches (Eq. 3); it is not the empirical covariance of the patch means.

3.1 FORMING PARTS

WBPRIOR needs soft weights $\mathbf{A}_{i,k}$ that say how strongly patch i belongs to part k. For each image we build a sparse nearest-neighbour graph over the backbone's own patch tokens and take the top-K eigenvectors $\mathbf{U}$ of its normalised Laplacian. These eigenvectors vary slowly over the graph, so they mark groups of similar patches, as in spectral clustering (von Luxburg, 2007):

$$\mathbf{A}_{i,k} \;=\; \mathrm{softmax}_k\big(|\mathbf{U}_{i,k}|/\tau_a\big), \tag{2}$$

where τ_a is a softmax temperature and the absolute value removes the arbitrary sign of each eigenvector. The graph, $\mathbf{U}$ and $\mathbf{A}$ are rebuilt for every image, at training and test time alike, and then discarded; no prototype bank is stored. The contribution of this paper is the representation applied *after* assignment: the controlled experiments in Sec. 4 keep the assignment fixed and change only how each part is aggregated. Graph construction and the spectral properties of the assignment are in Appendices A2.2 and A1.2.

3.2 WASSERSTEIN-BARYCENTER PART REPRESENTATION

A slot mean keeps only the mean of a group. We instead represent each part by the BW barycenter of its assigned patch Gaussians, which we call the *part prototype*; it keeps the mean vector and a learned scale vector. The scale vector is the per-dimension standard deviation $\mathbf{s} = \sqrt{\mathbf{v}}$ of a learned diagonal variance $\mathbf{v}$; it is a learned feature, not a calibrated uncertainty. The design needs four properties: keep the learned scale, depend only on the current image, allow exact aggregation, and add little retrieval cost. A diagonal Gaussian meets all four, because the diagonal barycenter combines means and standard deviations in closed form.

Definition 1 (Part prototype). *Within a single image, let $\tilde{\mathbf{z}}_i \in \mathbb{R}^{d_p}$ be the projected token of patch i, modulated by the image's CLS token through Feature-wise Linear Modulation (FiLM) (Perez et al., 2018). Patch i is mapped to the diagonal Gaussian ν_i on $\mathbb{R}^{d_b}$ with mean vector $\boldsymbol{\mu}_i$ and positive variance $\mathbf{v}_i$,*

$$\nu_i = \mathcal{N}\big(\boldsymbol{\mu}_i, \mathrm{diag}(\mathbf{v}_i)\big), \qquad \boldsymbol{\mu}_i = \mathbf{W}_\mu \tilde{\mathbf{z}}_i + \mathbf{b}_\mu, \qquad \mathbf{v}_i = \mathrm{Softplus}\big(\mathbf{W}_v \tilde{\mathbf{z}}_i + \mathbf{b}_v\big).$$

Let $\mathbf{s}_i$ be its scale vector and $w_{i,k}$ the normalised weight with which patch i is assigned to part k:

$$\mathbf{s}_i = \sqrt{\mathbf{v}_i}, \qquad w_{i,k} = \frac{\mathbf{A}_{i,k}}{\sum_j \mathbf{A}_{j,k}}.$$

The prototype of part k is the 2-Wasserstein barycenter of $\{\nu_i\}_i$ under the weights $\{w_{i,k}\}_i$,

$$\nu_k = \arg\min_{\nu} \sum_i w_{i,k}\, W_2^2(\nu, \nu_i).$$

Setting every variance to zero gives back the slot mean.

For diagonal Gaussians the distance splits into a mean term and a standard-deviation term (Eq. 1), so the objective separates per dimension and has the exact minimiser $\nu_k = \mathcal{N}(\boldsymbol{\mu}_k, \mathrm{diag}(\mathbf{v}_k))$ with (Agueh & Carlier, 2011; Bhatia et al., 2019)

$$\begin{aligned} \boldsymbol{\mu}_k &= \sum_i w_{i,k}\, \boldsymbol{\mu}_i, \\ \mathbf{s}_k &= \sum_i w_{i,k}\, \mathbf{s}_i, \qquad \mathbf{v}_k = \mathbf{s}_k^2, \\ W_2^2(\nu_k, \nu'_k) &= \|\boldsymbol{\mu}_k - \boldsymbol{\mu}'_k\|_2^2 + \|\mathbf{s}_k - \mathbf{s}'_k\|_2^2. \end{aligned} \tag{3}$$

The barycenter averages standard deviations, not variances. A Euclidean average of the patch variances would give $\sum_i w_{i,k}\mathbf{v}_i \geq \mathbf{s}_k^2$ (Jensen's inequality), so it would make a part look wider whenever its patches predict different variances. The update is computed in one pass per image, with no matrix square root and no iterative solver. The mean vector and the scale vector are then passed together to a learned read-out that maps them back into the d_p-D part space,

$$\hat{\mathbf{p}}_k \;=\; \mathrm{Readout}(\boldsymbol{\mu}_k, \mathbf{s}_k)\,, \tag{4}$$

so the learned scale is not used only by a training loss. In fact, it also enters the retrieval embedding used at test time, before the part descriptors are pooled and fused with the CLS token (Eq. 5). Because patch Gaussians and barycenters are recomputed for every image at test time, the slot-mean control differs from WBPRIOR at test time too. The Gaussians live in a small d_b-D space projected from the d_p-D part space, not in the 768-D retrieval embedding, so the aggregation adds a negligible cost (App. A1.4). No prototype bank, buffer or other state is carried between images (App. A1.5). The general case with full covariance matrices is given in Appendix A1.3, and Appendix A5.3 runs the head with full covariances as a control.

3.3 RETRIEVAL AND TRAINING

Retrieval. A learned part-selection gate keeps the useful parts of each image, and the kept part descriptors $\hat{\mathbf{p}}_k$ are pooled into one part descriptor $\mathbf{f}_{\text{part}}$ (App. A2.2). A fusion gate then combines it with the global CLS descriptor $\mathbf{f}_{\text{cls}}$:

$$\mathbf{f} = \mathrm{L2Norm}(\alpha_p \mathbf{f}_{\text{part}} + \alpha_c \mathbf{f}_{\text{cls}}), \qquad [\alpha_p, \alpha_c] = \mathrm{softmax}(\mathbf{G}_2\, \mathrm{ReLU}(\mathbf{G}_1[\mathbf{f}_{\text{part}}; \mathbf{f}_{\text{cls}}]))\,, \tag{5}$$

where $\mathbf{f}$ is the ℓ_2-normalised 768-D retrieval embedding and $\mathbf{G}_1, \mathbf{G}_2$ are learned weight matrices.

Training. Training uses paired views. Three part-level losses make corresponding parts agree across a pair. A functional-map term (Ovsjanikov et al., 2012) uses a linear map between the eigenvectors of the two views, so that part k means the same region in both. A debiased Sinkhorn divergence (Feydy et al., 2019), an entropy-regularised optimal-transport objective, compares the patches of part k in the two views. The third is the W_2^2 distance of Eq. 3 between the two part prototypes. These sit next to standard retrieval losses (classification, Circle (Sun et al., 2020), proxy-anchor (Kim et al., 2020), InfoNCE (van den Oord et al., 2018) and a patch-level contrastive loss) and training-only auxiliaries: proxy distillation from an EMA teacher with an entropy-adaptive temperature (uncertainty-aware proxy alignment, UAPA), masked-patch reconstruction and an altitude head. The loss groups are balanced by one learned weight each (Kendall et al., 2018). All equations and settings are in Appendices A2.2, A2.4 and A2.5.

Inference. At test time, each image passes once through the backbone and the part branch, independently of all other images. All losses, the EMA teacher and the auxiliary heads are dropped, and retrieval ranks gallery images by cosine similarity of the 768-D embeddings, with no re-ranking (App. Tab. 6).

4 EXPERIMENTS

Experimental setup. We evaluate WBPRIOR on six retrieval benchmarks with three kinds of appearance change: cross-view geometry (SUES-200 (Zhu et al., 2023), DenseUAV (Dai et al., 2024) and University-1652 (Zheng et al., 2020)), a change of camera and viewpoint for person identity

Table 1: **Two controls answer different questions** (SUES-200 drone→satellite, five seeds, %). *(a)* Slot mean versus BW barycenter inside a fixed part branch. *(b)* Complete WBPRIOR model versus a CLS-only baseline trained under the same recipe. *Evals*: query decisions pooled over five seeds (and over the nine corruptions for Corrupt(9)); *lift* in pp. Paired 95% CI of the Corrupt(9) R@1 lift: $[+7.17, +9.24]$ pp in (a) and $[+0.91, +1.82]$ pp in (b).

Condition	Arm	Evals	R@1	R@5	R@10	mAP
(a) Same part branch, aggregation rule changed						
Clean	Slot mean	80,000	94.27	98.43	99.29	96.09
	Diagonal BW	80,000	**97.00**	**99.72**	**99.95**	**98.08**
	lift (pp)		+2.73	+1.29	+0.66	+1.99
Corrupt(9)	Slot mean	720,000	50.93	70.89	81.60	59.15
	Diagonal BW	720,000	**59.12**	**79.02**	**86.67**	**68.43**
	lift (pp)		+8.19	+8.13	+5.08	+9.28
(b) Complete model versus CLS-only baseline						
Clean	CLS-only	80,000	96.02	99.77	99.95	97.74
	Full WBPRIOR	80,000	**97.53**	**99.97**	**100.00**	**98.66**
	lift (pp)		+1.51	+0.21	+0.05	+0.92
Corrupt(9)	CLS-only	720,000	58.94	78.74	86.10	67.75
	Full WBPRIOR	720,000	**60.29**	**79.89**	**87.04**	**69.18**
	lift (pp)		+1.34	+1.16	+0.94	+1.43

(CARGO (Zhang et al., 2024)), and a change of modality from sketch to photo (SketchyCOCO (Gao et al., 2020) and FSCOCO (Chowdhury et al., 2022)). The aim is to test one representation under different kinds of appearance change, not to design a new representation for each task. Every run uses a single forward pass, the full gallery and cosine similarity, with no re-ranking and no test-time augmentation (TTA). The default model, WBPRIOR-S, uses DINOv2 ViT-S/14, and WBPRIOR-B uses ViT-B/14. We report R@K, the share of queries whose correct match is among the top K results, and mean average precision (mAP); differences are in percentage points (pp). *Clean* means unmodified test images, and *Corrupt(9)* is the mean R@1 over nine strong test-time corruptions (Sec. 4.2). The evaluation has four parts: a matched study that changes one component at a time over five seeds (Sec. 4.1); a plug-in study that adds WBPRIOR to seven published retrievers, which we call *hosts* (Tab. 2); a standalone comparison on the three drone benchmarks (Tab. 3 and App. Tab. 16); and a transfer study on sketch retrieval and person re-identification (Tab. 4). Clean results are compared with published numbers; corruption results compare models that we evaluated ourselves under the same corruption protocol. Each study reports its own runs. Table 1 averages five seeds, while the plug-in, standalone and component-ablation tables each use one run of their own protocol, so the scores of WBPRIOR differ slightly between tables and are compared only within a table (App. A4.2). Input sizes and full recipes are in Appendices A2.1 to A3.1.

4.1 DOES THE PART REPRESENTATION MATTER?

Matched intervention. Table 1 answers this question with two controls that ask different things. Panel (a) keeps the whole part branch fixed, including the backbone, the part assignment, the losses, the optimiser, the schedule and the five seeds. It changes only how each part is summarised: by the slot mean or by the BW barycenter of Eq. 3. Panel (b) compares the complete WBPRIOR model with a *CLS-only baseline*, the same backbone trained under the same recipe without the part branch. The barycenter helps in both conditions, and its R@1 gain under corruption is about three times its clean gain, with a paired confidence interval well above zero. The slot-mean arm shows why this matters. Under corruption, the model with slot-mean parts falls below the CLS-only baseline, so adding parts makes retrieval worse. A plausible reading is that a plain mean of damaged patches hands the fusion gate a noisy part vector that pulls the ranking down. The barycenter also passes on the learned scale vector, and this turns the same branch into a useful signal. The complete model then beats the CLS-only baseline on clean queries, and by a small but reliable margin under corruption (Table 1b). The gain is larger at R@1 than at R@10, where clean scores are already near their ceiling, which is consistent with the scale vector separating close candidates whose part means look the same.

Is the gain from the scale vector, its geometry, or extra capacity? When only the aggregation rule changes, the two rules that keep no scale vector, the slot mean and k-means, lose the most, and a Euclidean average of means and variances comes closest to the barycenter without matching it

(App. Tab. 20a). So most of the gain comes from keeping a scale vector at all, and the rest from the 2-Wasserstein geometry, which averages standard deviations rather than variances and so does not inflate the part variance (Sec. 3.2). Extra capacity does not explain the gain: an unconstrained feature of the same width in place of the scale vector trails the deployed model, most of all under corruption (App. A5.3). To see where the scale vector acts, we set its contribution to zero at test time on high-structure patches (those with the strongest corner and edge response), on low-structure patches, or on a random subset of the same size. Removing it from high-structure patches hurts far more than elsewhere, and the paired confidence interval against the random subset excludes zero (App. Tabs. 20b and 24). The scale vector therefore matters most where local structure is strongest, which extra capacity alone would not explain.

4.2 Does it remain robust when appearance degrades?

Robustness protocol. Beyond the appearance change built into each dataset, we also damage the query images at test time. We follow the ImageNet-C protocol (Hendrycks & Dietterich, 2019), which applies common image corruptions at five strength levels, and add task-specific sensor and geometric corruptions. Except MuSeNet (Tab. 2), models are trained on clean images and tested on nine corruptions at the strongest level (severity 5). The corruptions form three groups: *Sensor* (Gaussian, shot, speckle and ISO noise), *Optical* (defocus blur, motion blur, pixelate) and *Geometric* (perspective, $\pm 30^\circ$ rotation); the Sensor, Optical and Geom columns of Table 2 are the mean R@1 within each group. Four more conditions (JPEG, scale, tile shift, haze) did not degrade a reference model, IBN-Net50-a (Pan et al., 2018), and are kept only for the appendix. Besides Corrupt(9), we report two summaries. We define *mCE-Geo*, our variant of the mean corruption error, as follows: each corruption's error $100 - \text{R@1}$ is divided by the error of the reference model on SUES-200, and the nine ratios are averaged, so lower is better. *Rel.Rob* (relative robustness) is Corrupt(9) as a share of clean R@1. Exact formulas and per-dataset recipes are in Appendix A4.3.

Host retrievers. If the representation is useful beyond our own training pipeline, adding the same branch to independently designed retrievers should also help. Table 2 tests this on seven published hosts. Adding the WBPrior branch improves Corrupt(9), mCE-Geo and every corruption group for every host. On average, gains are largest for optical and geometric corruptions and smallest for sensor noise. Rel.Rob falls slightly for CCR and Sample4Geo because the branch raises their clean R@1 proportionally more than their Corrupt(9). The branch is a parallel head that reads the host's own spatial features. It brings no second backbone or teacher, and it leaves the host backbone, classifier and training losses unchanged, so the gain comes from the added part branch, not from an ensemble. Re-training a host under our shared recipe costs it a few points of clean accuracy compared with its original recipe (App. Tab. 17), so the table compares rows only within a host and is not a leaderboard.

Robustness is not confined to SUES-200. When each method runs in its own best configuration, with no WBPrior branch attached to any baseline, WBPrior has the highest Corrupt(9) R@1 on DenseUAV and University-1652, by more than 10 pp over the best baseline on each. It trails Sample4Geo slightly on SUES-200 (App. Tab. 16).

4.3 Does it generalise beyond the default task and encoder?

The part branch keeps clean accuracy while it adds robustness. Table 3 compares WBPrior with published methods on clean images from the three drone benchmarks. On SUES-200 and University-1652, recent methods already place the correct match first for the large majority of queries, so the question here is whether the part branch costs clean accuracy. It does not: WBPrior is competitive with or exceeds recent state of the art on all three benchmarks, with the smallest parameter and compute budget among the methods that report it, and without an external teacher model, TTA or re-ranking. Its ViT-S/14 backbone was pretrained without labels on DINOv2's curated image set (Oquab et al., 2024), just as the ConvNeXt (Liu et al., 2022b) baselines benefit from ImageNet-22k (Deng et al., 2009) pretraining. Per-altitude SUES-200 results are in Appendix A3.2.

The same representation also helps when the change in appearance does not come from viewpoint alone. A sketch keeps shape but loses colour and most texture, while CARGO combines a change of viewpoint with a change of camera. On both sketch benchmarks, WBPrior-B (the same head on the larger ViT-B/14 backbone) is best in every column (Table 4a), ahead of SketchScape, a method built for sketch retrieval with a similar number of parameters. On CARGO, where cameras on the ground

Table 2: **SUES-200 plug-in study: each host with and without WBPRIOR** (drone→satellite, %). Each published retriever (the host) is re-trained by us twice under the same recipe: alone, and with the WBPRIOR branch added (rows marked †). The small number on a † row is its change from the host row above (green: better, red: worse); **bold** marks the better row of the pair. The last pair adds the head to a DINOv2-S model that uses only its CLS token. Metrics are defined in Sec. 4.2. Compare rows only within a pair, not across hosts.

Method	Corrupt(9)↑	mCE-Geo↓	Rel.Rob↑	Sensor↑	Optical↑	Geom↑
MuSeNet (Wang et al., 2024)	35.26	0.865	84.84	35.82	31.40	39.95
+ WBPrior †	**43.89**$_{\uparrow 8.63}$	**0.747**$_{\downarrow 0.118}$	**87.24**$_{\uparrow 2.40}$	**43.56**$_{\uparrow 7.74}$	**42.31**$_{\uparrow 10.91}$	**46.90**$_{\uparrow 6.95}$
MCCG (Shen et al., 2024)	18.53	1.025	29.22	10.85	7.62	50.25
+ WBPrior †	**25.59**$_{\uparrow 7.06}$	**0.923**$_{\downarrow 0.102}$	**34.94**$_{\uparrow 5.72}$	**18.05**$_{\uparrow 7.20}$	**11.94**$_{\uparrow 4.32}$	**61.13**$_{\uparrow 10.88}$
CCR (Du et al., 2024)	37.82	0.765	**49.46**	26.85	31.72	68.90
+ WBPrior †	**41.44**$_{\uparrow 3.62}$	**0.709**$_{\downarrow 0.056}$	47.58$_{\downarrow 1.88}$	**28.82**$_{\uparrow 1.97}$	**34.71**$_{\uparrow 2.99}$	**76.79**$_{\uparrow 7.89}$
MEAN (Chen et al., 2025)	33.43	0.818	40.71	24.47	22.38	67.91
+ WBPrior †	**38.23**$_{\uparrow 4.80}$	**0.743**$_{\downarrow 0.075}$	**42.99**$_{\uparrow 2.28}$	**27.28**$_{\uparrow 2.81}$	**27.93**$_{\uparrow 5.55}$	**75.56**$_{\uparrow 7.65}$
DAC (Xia et al., 2024)	56.46	0.548	67.25	54.36	45.82	76.64
+ WBPrior †	**62.38**$_{\uparrow 5.92}$	**0.450**$_{\downarrow 0.098}$	**67.65**$_{\uparrow 0.40}$	**58.49**$_{\uparrow 4.13}$	**52.97**$_{\uparrow 7.15}$	**84.29**$_{\uparrow 7.65}$
CAMP (Wu et al., 2024)	37.36	0.769	44.72	27.03	29.11	70.41
+ WBPrior †	**45.51**$_{\uparrow 8.15}$	**0.659**$_{\downarrow 0.110}$	**50.71**$_{\uparrow 5.99}$	**36.97**$_{\uparrow 9.94}$	**36.87**$_{\uparrow 7.76}$	**75.54**$_{\uparrow 5.13}$
Sample4Geo (Deuser et al., 2023)	57.34	0.523	**66.97**	53.38	47.26	80.39
+ WBPrior †	**62.34**$_{\uparrow 5.00}$	**0.444**$_{\downarrow 0.079}$	65.68$_{\downarrow 1.29}$	**55.97**$_{\uparrow 2.59}$	**53.96**$_{\uparrow 6.70}$	**87.65**$_{\uparrow 7.26}$
DINOv2-S (CLS-only)	45.38	0.657	50.53	35.12	38.39	76.40
+ WBPrior †	**60.13**$_{\uparrow 14.75}$	**0.460**$_{\downarrow 0.197}$	**61.65**$_{\uparrow 11.12}$	**40.12**$_{\uparrow 5.00}$	**67.04**$_{\uparrow 28.65}$	**89.80**$_{\uparrow 13.40}$

All hosts use the shared SUES-200 recipe of WBPRIOR (App. A4.2), except MuSeNet, which keeps its published recipe that adds corruptions during training. The † rows attach WBPRIOR to each host's own spatial features (last-stage feature maps for ConvNeXt and ResNet, patch tokens for ViT). Three more hosts, LPN (Wang et al., 2022), TransGeo (Zhu et al., 2022) and GeoDTR (Zhang et al., 2023), are reported in App. Tab. 15.

Table 3: **Clean cross-view retrieval on three drone benchmarks.** D→S: drone query, satellite gallery; S→D: the reverse. Cells give R@1 / AP (%), where AP is average precision; SUES-200 cells average four flight altitudes. SDM@1 (DenseUAV) is a spatial distance score that gives partial credit when the top-ranked tile lies close to the true location. **Red**: best; **navy**: second; GFLOPs as reported by each paper. Full tables: App. Tabs. 11, 12, and 13.

Method	Venue	Params↓	GFLOPs↓	SUES-200 D→S R@1 / AP	SUES-200 S→D R@1 / AP	University-1652 D→S R@1 / AP	University-1652 S→D R@1 / AP	DenseUAV (D→S) R@1	DenseUAV (D→S) SDM@1
LPN (Wang et al., 2022)‡	TCSVT'22	62.4 M	36.78	73.57 / 77.88	89.38 / 77.21	77.71 / 80.80	90.30 / 78.78	71.77	77.95
Sample4Geo (Deuser et al., 2023)†	ICCV'23	87.6 M	90.24	96.86 / 97.45	96.84 / **97.45**	92.65 / 93.81	95.14 / 91.39	80.57	-
CAMP (Wu et al., 2024)	TGRS'24	-	90.24	97.60 / 98.11	98.12 / 96.85	94.46 / 95.38	96.15 / 92.72	88.72	-
DAC (Xia et al., 2024)	TCSVT'24	96.5 M	90.24	97.52 / 98.07	98.44 / 96.67	**94.67 / 95.50**	**96.43 / 93.79**	84.47	-
MEAN (Chen et al., 2025)	TGRS'25	36.5 M	26.18	**98.09 / 98.50**	**99.38** / 97.33	93.55 / 94.53	96.01 / 92.08	**90.18**	-
WBPRIOR (Ours)	-	**23.42 M**$_{\downarrow 36\%}$	**22.14**$_{\downarrow 15\%}$	**98.32**$_{\uparrow 0.23}$ / **98.72**	**99.69**$_{\uparrow 0.31}$ / **98.06**	**94.86**$_{\uparrow 0.19}$ / **96.08**	**97.57**$_{\uparrow 1.14}$ / **94.92**	**90.48**$_{\uparrow 0.30}$	**92.55**$_{\uparrow 1.54}$

Baselines are taken *as published*; the WBPRIOR row is re-trained with our shared recipe on each dataset (one RGB image, one forward pass). †Sample4Geo DenseUAV R@1 as reported by Chen et al. (2026); the DenseUAV-protocol table of Xu et al. (2026) lists it at 49.38% (App. Tab. 13). ‡LPN params are reported for ResNet-50 (62.4 M); DenseUAV evaluation uses ViT-S. "-" = not reported or not evaluated.

and in the air must be matched, WBPRIOR-B is best in six of eight columns even after adding the recent ViSA (Zhang et al., 2026) (Table 4b): both metrics on ALL and on aerial-to-ground matching, and mAP on the two same-platform protocols. The architecture is reused, but each task keeps its own training losses, so we claim a reusable part representation and head design, not one set of weights shared by all tasks.

The gain also does not depend on one encoder. On FSCOCO *Unseen*, we attach the head to 15 pretrained encoders and compare each with a matched control that uses only that encoder's own

Table 4: **Transfer to sketch retrieval and aerial-ground person re-identification.** *(a)* Sketch-based image retrieval (SBIR): sketch query, photo gallery. FSCOCO *Normal* is the standard split and *Unseen* tests on sketches from users held out of training; S→I: sketch query, image gallery. *(b)* CARGO person re-identification (re-ID), R@1 and mAP under the four protocols of Zhang et al. (2024): ALL uses every camera; G↔G, A↔A and A↔G match ground to ground, aerial to aerial, and aerial to ground. Baselines are from Li et al. (2025), plus ViSA (Zhang et al., 2026) (CVPR 2026). **Red**: best; **navy**: second.

(a) Sketch-based image retrieval

Method	Params↓	FSCOCO				SketchyCOCO	
		Normal		Unseen		S→I	
		R@1	R@10	R@1	R@10	R@1	R@10
SceneSketcherV2 (Liu et al., 2022a)	∼140 M	-	-	-	-	68.1	95.2
FreestyleRet (Li et al., 2024)	∼150 M*	29.6	56.1	-	-	-	-
SketchScape (Demić & Čehovin Zajc, 2025)	∼89 M	**61.9**	**87.2**	**60.0**	**86.1**	**70.0**	**96.2**
WBPrior-S (Ours)	**23.42 M**	56.27	84.90	55.57	85.17	67.34	93.73
WBPrior-B (Ours)	∼89 M	63.05	88.57	60.77	87.73	70.48	97.42

(b) CARGO: aerial-ground person re-ID

Method	Params↓	ALL		G↔G		A↔A		A↔G	
		R@1	mAP	R@1	mAP	R@1	mAP	R@1	mAP
AGW (Ye et al., 2022)	**24.7 M**	60.26	53.44	81.25	71.66	67.50	56.48	43.57	40.90
DTST (Wang & Pishgar, 2025)	-	64.42	55.73	78.57	72.40	**80.00**	63.31	50.53	43.49
VDT (Zhang et al., 2024)	89.7 M	64.10	55.20	82.14	71.59	82.50	66.83	48.12	42.76
GSAlign (Li et al., 2025)	∼86 M	65.06	57.95	83.04	73.86	**80.00**	65.55	64.89	61.55
ViSA (Zhang et al., 2026)	-	70.51	65.46	**88.39**	**83.90**	82.50	67.78	**71.28**	**69.00**
WBPrior-S (Ours)	**23.42 M**	**74.36**	**68.99**	85.71	82.46	**80.00**	**72.38**	68.13	60.95
WBPrior-B (Ours)	∼89 M	78.53	74.71	**87.50**	**85.11**	77.50	74.41	74.38	70.12

*Approx. CLIP backbone params (visual+text).

global feature; the head improves every metric on every encoder (App. Tab. 14). What carries over between tasks is the part representation, not the training signal: each task still needs its own auxiliary losses, and removing them on CARGO aerial-to-ground matching costs a large drop (App. Tab. 27).

5 Discussion and Scope

The experiments point to a simple account of why the barycenter helps. Under strong corruption, the mean vector of a part moves with the damaged texture, and a slot mean has nothing else to hand to the fusion gate. The learned scale vector gives each part a second coordinate that training can shape, and the matched study shows that this coordinate is what turns the part branch from a liability into a gain. Its effect is largest on patches with strong corners and edges, which is where the layout of a scene lives, so the scale vector behaves less like extra capacity and more like a learned description of local structure.

Two design choices keep this benefit cheap, and in both the costlier option buys little. Each part keeps one variance per dimension, a diagonal Gaussian, instead of a full covariance matrix: at the same width, the full matrix is only slightly more accurate under corruption and makes the head almost three times as costly (App. Tab. 21). The spectral assignment computes its eigenvectors on a coarse 8×8 grid of pooled patches and then upsamples them, so assignment and aggregation still use every patch; the next finer grid barely changes accuracy but makes this step about nine times slower (App. Tab. 8).

The same design suggests where the idea can grow. A single diagonal Gaussian describes a part with one mode; parts whose patches form several separate groups could be described by a small mixture, and our diagnostics bound how much such an extension can still add (App. A6). Transfer across tasks currently reuses the representation and the head design while each task keeps its own training losses, so one set of weights for all three kinds of appearance change is a natural next step.

6 Conclusion

We asked what a part representation should keep once its patches have been found, and answered with WBPrior, a lightweight plug-and-play layout prior for retrieval under severe nuisance. Each patch predicts a mean vector and a scale vector, and each part becomes their exact 2-Wasserstein barycenter, a closed-form summary that keeps a learned scale vector next to the mean vector with negligible additional aggregation cost. A matched study that changes only this summary shows that the slot mean fails under corruption while the barycenter restores a useful part branch. The same head makes seven published retrievers more robust and carries over to sketch retrieval and aerial-ground person re-identification. These results suggest that part-based retrieval has a second design axis next to part discovery: robust retrieval depends not only on how parts are found, but also on how each part is represented.

Ethics, Reproducibility, and AI Use Statements

The ethics, reproducibility and AI-use statements are in Appendix A6 (Sections A6.1, A6.2 and A6.3).

References

Martial Agueh and Guillaume Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011. doi: 10.1137/100805741. URL https://epubs.siam.org/doi/abs/10.1137/100805741.

Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. BoQ: A place is worth a bag of learnable queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17794–17803, 2024. doi: 10.1109/CVPR52733.2024.01685.

Pedro C. Álvarez-Esteban, Eustasio del Barrio, Juan A. Cuesta-Albertos, and Carlos Matrán. A fixed-point approach to barycenters in Wasserstein space. *Journal of Mathematical Analysis and Applications*, 441(2):744–762, 2016. doi: 10.1016/j.jmaa.2016.04.045. URL https://www.sciencedirect.com/science/article/pii/S0022247X16300907.

Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Arandjelovic_NetVLAD_CNN_Architecture_CVPR_2016_paper.html.

Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the Bures–Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2):165–191, 2019. doi: 10.1016/j.exmath.2018.01.002.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021. doi: 10.1109/ICCV48922.2021.00951. URL https://openaccess.thecvf.com/content/ICCV2021/html/Caron_Emerging_Properties_in_Self-Supervised_Vision_Transformers_ICCV_2021_paper.html.

Zhongwei Chen, Zhao-Xu Yang, and Hai-Jun Rong. Multilevel embedding and alignment network with consistency and invariance learning for cross-view geo-localization. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–15, 2025. doi: 10.1109/TGRS.2025.3572775.

Zhongwei Chen, Zhao-Xu Yang, Hai-Jun Rong, and Guoqi Li. Without paired labeled data: End-to-end self-supervised learning for drone-view geo-localization. *IEEE Transactions on Neural Networks and Learning Systems*, 2026. doi: 10.1109/TNNLS.2026.3696684.

Pinaki Nath Chowdhury, Aneeshan Sain, Ayan Kumar Bhunia, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. FS-COCO: Towards understanding of freehand sketches of common objects in context. In *European Conference on Computer Vision (ECCV)*, pp. 253–270, 2022. doi: 10.1007/978-3-031-20074-8_15.

Sanghyuk Chun. Improved probabilistic image-text representations. In *International Conference on Learning Representations (ICLR)*, 2024.

Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio de Rezende, Yannis Kalantidis, and Diane Larlus. Probabilistic embeddings for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8415–8424, 2021. doi: 10.1109/CVPR46437.2021.00831.

Marco Cuturi and Arnaud Doucet. Fast computation of Wasserstein barycenters. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014. doi: 10.48550/arXiv.1310.4375.

Ming Dai, Jianhong Hu, Jiedong Zhuang, and Enhui Zheng. A transformer-based feature segmentation and region alignment method for UAV-view geo-localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(7):4376–4389, 2022. doi: 10.1109/TCSVT.2021.3135013.

Ming Dai, Enhui Zheng, Zhenhua Feng, Lei Qi, Jiedong Zhuang, and Wankou Yang. Vision-based UAV self-positioning in low-altitude urban environments. *IEEE Transactions on Image Processing*, 33:493–508, 2024. doi: 10.1109/TIP.2023.3346279.

Emil Demić and Luka Čehovin Zajc. Back to the drawing board: Rethinking scene-level sketch-based image retrieval. In *36th British Machine Vision Conference 2025, BMVC 2025, Sheffield, UK, November 24-27, 2025*. BMVA, 2025. URL https://bmva-archive.org.uk/bmvc/2025/assets/papers/Paper_373/paper.pdf.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Fabian Deuser, Konrad Habel, and Norbert Oswald. Sample4Geo: Hard negative sampling for cross-view geo-localisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16847–16856, 2023. doi: 10.1109/ICCV51070.2023.01545.

Lirong Ding, Ji Zhou, Lingxuan Meng, and Zhiyong Long. A practical cross-view image matching method between UAV and satellite for UAV-based geo-localization. *Remote Sensing*, 13(1):47, 2021. doi: 10.3390/rs13010047.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. doi: 10.48550/arXiv.2010.11929.

Haolin Du, Jingfei He, and Yuanqing Zhao. CCR: A counterfactual causal reasoning-based method for cross-view geo-localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(11):11630–11643, 2024. doi: 10.1109/TCSVT.2024.3425509.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and MMD using Sinkhorn divergences. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019. doi: 10.48550/arXiv.1810.08278.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. Learning with a Wasserstein loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. SketchyCOCO: Image generation from freehand scene sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5173–5182, 2020. doi: 10.1109/CVPR42600.2020.00522. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Gao_SketchyCOCO_Image_Generation_From_Freehand_Scene_Sketches_CVPR_2020_paper.html.

Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, pp. 147–151, 1988. doi: 10.5244/C.2.23.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. AnyLoc: Towards universal visual place recognition. *IEEE Robotics and Automation Letters*, 9(2):1286–1293, 2024. doi: 10.1109/LRA.2023.3343602.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7482–7491, 2018. doi: 10.1109/CVPR.2018.00781. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Kendall_Multi-Task_Learning_Using_CVPR_2018_paper.html.

Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3238–3247, 2020. doi: 10.48550/arXiv.2003.13911. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Kim_Proxy_Anchor_Loss_for_Deep_Metric_Learning_CVPR_2020_paper.html.

Avinash Kori, Francesco Locatello, Ainkaran Santhirasekaram, Francesca Toni, Ben Glocker, and Fabio De Sousa Ribeiro. Identifiable object-centric representation learning via probabilistic slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. doi: 10.52202/079017-2960. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a9a3f0e4a95cb273867931369c8fc3b1-Abstract.html.

Hao Li, Yanhao Jia, Peng Jin, Zesen Cheng, Kehan Li, Jialu Sui, Chang Liu, and Li Yuan. FreestyleRet: Retrieving images from style-diversified queries. In *European Conference on Computer Vision (ECCV)*, pp. 258–274. Springer, 2024. doi: 10.1007/978-3-031-73337-6_15.

Peihua Li, Jiangtao Xie, Qilong Wang, and Zilin Gao. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 947–955, 2018. doi: 10.1109/CVPR.2018.00105.

Qiao Li, Jie Li, Yukang Zhang, Lei Tan, Jing Chen, and Jiayi Ji. GSAlign: Geometric and semantic alignment network for aerial-ground person re-identification. In *Advances in Neural Information Processing Systems*, volume 38, pp. 93785–93805. Curran Associates, Inc., 2025. doi: 10.52202/085713-3137. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/86c17de05579cde52025f9984e6e2ebb-Abstract-Conference.html.

Yanping Li, Yizhang Liu, Hongyun Zhang, Cairong Zhao, Zhihua Wei, and Duoqian Miao. UD-Gaussian: Uncertainty-driven Gaussian modeling for occluded person re-identification. *IEEE Transactions on Image Processing*, 35:2889–2900, 2026. doi: 10.1109/TIP.2026.3672380. URL https://ieeexplore.ieee.org/document/11435281.

Jinliang Lin, Zhedong Zheng, Zhun Zhong, Zhiming Luo, Shaozi Li, Yi Yang, and Nicu Sebe. Joint representation learning and keypoint detection for cross-view geo-localization. *IEEE Transactions on Image Processing*, 31:3780–3792, 2022. doi: 10.1109/TIP.2022.3175601.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1449–1457, 2015.

Fang Liu, Xiaoming Deng, Changqing Zou, Yu-Kun Lai, Keqi Chen, Ran Zuo, Cuixia Ma, Yong-Jin Liu, and Hongan Wang. SceneSketcher-v2: Fine-grained scene-level sketch-based image retrieval using adaptive GCNs. *IEEE Transactions on Image Processing*, 31:3737–3751, 2022a. doi: 10.1109/TIP.2022.3175403.

Xiangzeng Liu, Ziyao Wang, Yue Wu, and Qiguang Miao. SeGCN: A semantic-aware graph convolutional network for UAV geo-localization. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:6055–6066, 2024. doi: 10.1109/JSTARS.2024.3370612.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986, 2022b.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. doi: 10.48550/arXiv.2006.15055.

Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. SegCLIP: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23033–23044. PMLR, 2023. URL https://proceedings.mlr.press/v202/luo23a.html.

David Marr and H. Keith Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 200(1140):269–294, 1978. doi: 10.1098/rspb.1978.0020.

Boris Muzellec and Marco Cuturi. Generalizing point embeddings using the Wasserstein space of elliptical distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 10258–10269, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://mlanthology.org/tmlr/2024/oquab2024tmlr-dinov2/.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (SIGGRAPH)*, 31(4):30:1–30:11, 2012. doi: 10.1145/2185520.2185526. URL https://dl.acm.org/doi/abs/10.1145/2185520.2185526.

Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via IBN-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 464–479, 2018.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. doi: 10.1609/aaai.v32i1.11671.

Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed Fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3384–3391, 2010. doi: 10.1109/CVPR.2010.5540009.

Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2019. doi: 10.1109/TPAMI.2018.2846566.

Tianrui Shen, Yingmei Wei, Lai Kang, Shanshan Wan, and Yee-Hong Yang. MCCG: A ConvNeXt-based multiple-classifier method for cross-view geo-localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(3):1456–1468, 2024. doi: 10.1109/TCSVT.2023.3296074.

Yichun Shi and Anil K. Jain. Probabilistic face embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6902–6911, 2019.

Jian Sun, Kangdao Liu, Chi Zhang, Chuangquan Chen, Junge Shen, C. L. Philip Chen, and Chi-Man Vong. MobileGeo: Exploring hierarchical knowledge distillation for resource-efficient cross-view drone geo-localization. *arXiv preprint arXiv:2510.22582*, 2025.

Wenyu Sun, Jiyang Xie, Jiayan Qiu, and Zhanyu Ma. Part uncertainty estimation convolutional neural network for person re-identification. In *IEEE International Conference on Image Processing (ICIP)*, pp. 2304–2308, 2021.

Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 480–496, 2018. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Yifan_Sun_Beyond_Part_Models_ECCV_2018_paper.html.

Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. doi: 10.1109/CVPR42600.2020.00643. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Sun_Circle_Loss_A_Unified_Perspective_of_Pair_Similarity_Optimization_CVPR_2020_paper.html.

Haomiao Tang, Jinpeng Wang, Yuang Peng, Guanghao Meng, Ruisheng Luo, Bin Chen, Long Chen, Yaowei Wang, and Shu-Tao Xia. Modeling uncertainty in composed image retrieval via probabilistic embeddings. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1210–1222, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.61. URL https://aclanthology.org/2025.acl-long.61/.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. doi: 10.48550/arXiv.1807.03748.

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

Tingyu Wang, Zhedong Zheng, Chenggang Yan, Jiyong Zhang, Yaoqi Sun, Bolun Zheng, and Yi Yang. Each part matters: Local patterns facilitate cross-view geo-localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(2):867–879, 2022. doi: 10.1109/TCSVT.2021.3061265.

Tingyu Wang, Zhedong Zheng, Yaoqi Sun, Chenggang Yan, Yi Yang, and Tat-Seng Chua. Multiple-environment self-adaptive network for aerial-view geo-localization. *Pattern Recognition*, 152:110363, 2024. doi: 10.1016/j.patcog.2024.110363.

Yuhai Wang and Maryam Pishgar. Dynamic token selective transformer for aerial-ground person re-identification. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2025. doi: 10.1109/ICME59968.2025.11210054.

Frederik Warburg, Martin Jørgensen, Javier Civera, and Søren Hauberg. Bayesian triplet loss: Uncertainty quantification in image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Qiong Wu, Yi Wan, Zhi Zheng, Yongjun Zhang, Guangshuai Wang, and Zhenyang Zhao. CAMP: A cross-view geo-localization method using contrastive attributes mining and position-aware partitioning. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–14, 2024. doi: 10.1109/TGRS.2024.3448499.

Panwang Xia, Yi Wan, Zhi Zheng, Yongjun Zhang, and Jiwei Deng. Enhancing cross-view geo-localization with domain alignment and scene consistency. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(12):13271–13281, 2024. doi: 10.1109/TCSVT.2024.3443510.

Yuanze Xu, Ming Dai, and Wankou Yang. Precise GPS-denied UAV self-positioning via context-enhanced cross-view geo-localization. In *Pattern Recognition and Computer Vision*, volume 16286 of *Lecture Notes in Computer Science*, pp. 374–388. Springer Singapore, 2026. doi: 10.1007/978-981-95-5628-1_26. Proceedings of PRCV 2025.

Jiaqiang Yang, Danyang Qin, Huapeng Tang, Sili Tao, Haoze Bie, and Lin Ma. DINOv2-based UAV visual self-localization in low-altitude urban environments. *IEEE Robotics and Automation Letters*, 10(2):2080–2087, 2025. doi: 10.1109/LRA.2025.3527762.

Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C. H. Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2872–2893, 2022. doi: 10.1109/TPAMI.2021.3054775.

Tianyuan Yu, Da Li, Yongxin Yang, Timothy M. Hospedales, and Tao Xiang. Robust person re-identification by modelling feature uncertainty. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 552–561, 2019.

Quan Zhang, Lei Wang, Vishal M. Patel, Xiaohua Xie, and Jianhuang Lai. View-decoupled transformer for person re-identification under aerial-ground camera network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22000–22009, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Zhang_View-decoupled_Transformer_for_Person_Re-identification_under_Aerial-ground_Camera_Network_CVPR_2024_paper.html.

Quan Zhang, Zeqiang Cai, Peiming Zhao, Jingze Wu, Cailun Wu, Hongbo Chen, and Jianhuang Lai. View-aware semantic alignment for aerial-ground person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4383–4392, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhang_View-Aware_Semantic_Alignment_for_Aerial-Ground_Person_Re-Identification_CVPR_2026_paper.html.

Xiaohan Zhang, Xingyu Li, Waqas Sultani, Yi Zhou, and Safwan Wshah. Cross-view geo-localization via learning disentangled geometric layout correspondence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 3480–3488, 2023. doi: 10.1609/aaai.v37i3.25457.

Zhedong Zheng, Yunchao Wei, and Yi Yang. University-1652: A multi-view multi-source benchmark for drone-based geo-localization. In *Proceedings of the 28th ACM International Conference on Multimedia (ACM MM)*, pp. 1395–1403, 2020. doi: 10.1145/3394171.3413896.

Runzhe Zhu, Ling Yin, Mingze Yang, Fei Wu, Yuncheng Yang, and Wenbo Hu. SUES-200: A multi-height multi-scene cross-view image benchmark across drone and satellite. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9):4825–4839, 2023. doi: 10.1109/TCSVT.2023.3249204.

Sijie Zhu, Mubarak Shah, and Chen Chen. TransGeo: Transformer is all you need for cross-view image geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1162–1171, 2022. doi: 10.1109/CVPR52688.2022.00123. URL https://openaccess.thecvf.com/content/CVPR2022/html/Zhu_TransGeo_Transformer_Is_All_You_Need_for_Cross-View_Image_Geo-Localization_CVPR_2022_paper.html.

The appendix has six parts: theory and geometry (§A1); the method recipe (§A2); datasets and per-benchmark detail (§A3); corruption-robustness details (§A4); method diagnostics and sensitivity (§A5); and limitations with broader impact (§A6). The results come from four studies: the matched five-seed studies (Study A), the standalone benchmark and robustness comparison (Study B), the plug-in study on published hosts with one run per pair (Study C), and the component and alternative-aggregation ablations (Study D). Each study reports its own runs, so absolute scores are compared only within a study.

A1 Theory and Geometry

A1.1 Symbol Reference

Table 5: Key symbols and the sections that define them.

Symbol	Meaning	Section
$\boldsymbol{\mu}_i, \mathbf{v}_i$	Learned diagonal Gaussian of patch i: mean vector from a linear head, positive learned scale vector from a Softplus head. Recomputed every forward pass; never stored.	3.2
$\nu_k = \mathcal{N}(\boldsymbol{\mu}_k, \mathrm{diag}(\mathbf{v}_k))$	Part prototype for part k of one image: the diagonal BW barycenter of the assigned patch Gaussians (Eq. 3), with scale vector $\mathbf{s}_k = \sqrt{\mathbf{v}_k}$. Recomputed for every image; never stored.	3.2
$w_{i,k}$	Assignment weight of patch i to part k, normalised over patches: $w_{i,k} = \mathbf{A}_{i,k} / \sum_j \mathbf{A}_{j,k}$.	3.1
$W_2(\nu_1, \nu_2)$	2-Wasserstein distance between two probability measures. For Gaussians this is the Bures-Wasserstein distance: $W_2^2 = \lVert\mu_1 - \mu_2\rVert^2 + \mathrm{tr}\big(\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}\big)$.	3.2
$S_\varepsilon(\nu_d, \nu_s)$	Debiased Sinkhorn divergence with regularisation ε; satisfies $S_\varepsilon \to W_2^2$ as $\varepsilon \to 0$ (Feydy et al., 2019).	A1.6
$\mathbf{U} \in \mathbb{R}^{N \times K}$	Top-K eigenvectors of the normalised Laplacian of the affinity graph over the backbone's own patch tokens; rows give the soft patch-to-part assignment $\mathbf{A} \propto \mathrm{softmax}(\lvert\mathbf{U}\rvert/\tau_a)$.	A2.2
$\mathbf{C} = \mathbf{U}_d^\top \mathbf{U}_s$	Functional map between drone- and satellite-view eigenbases; cross-view consistency penalises $\lVert CC^\top - \mathbf{I}\rVert^2$ (Ovsjanikov et al., 2012).	A2.2
$\hat{\mathbf{p}}_k$	Part descriptor obtained by reading out $(\boldsymbol{\mu}_k, \mathbf{s}_k)$, the mean vector and the scale vector, followed by the residual part head.	3.2
$g_k \in [0, 1]$	Part-selection gate over the $K_{\max}$=12 part channels. Training uses a binary-concrete (logistic-sigmoid) relaxation of the keep-or-drop decision, with a hard threshold in the forward pass, a straight-through gradient and a floor of $K_{\min}$=4 kept parts; inference uses the noise-free threshold with no floor.	A2.2
$[\alpha_p, \alpha_c]$	Two-way fusion gate (softmax MLP with weight matrices $\mathbf{G}_1, \mathbf{G}_2$) combining $\mathbf{f}_{\text{part}}$ and $\mathbf{f}_{\text{cls}}$ into the retrieval embedding.	3.3
γ, β	CLS-conditioned FiLM modulation predicted from the same view's CLS token; it modulates the projected tokens $\tilde{\mathbf{z}}_i$ that feed the Gaussian heads, not the graph input.	A2.2
$\mathcal{L}_{\text{bw}}$	Cross-view 2-Wasserstein consistency loss between corresponding part prototypes (mean and scale vectors) of the two views (part-quality group).	A2.5
$\mathcal{L}_{\text{sink}}$	Debiased Sinkhorn divergence between the drone and satellite patch distributions of each part.	A2.5
$\mathcal{L}_{\text{uapa}}$	Uncertainty-aware proxy alignment, modulating only the KL temperature.	A2.4
θ^{ema}	EMA teacher: a training-only, stop-gradient exponential moving average of the student weights, used as a distillation target and discarded at inference.	A1.5
s_g	Learned weight parameter of loss group $g \in \{\text{align}, \text{distill}, \text{part}, \text{alt}\}$ in the multi-task loss weighting of Kendall et al. (2018); it is a log-variance in that formulation, not a patch variance.	A2.5

A1.2 Assignment Invariance and What the Barycenter Adds

This subsection states two formal properties of the head. The first concerns the spectral assignment of Eq. 14: it is invariant to orthogonal transformations of the feature space, because such transformations leave the affinity graph unchanged. The second shows what the barycenter keeps that the slot mean discards.

Setup. Let $\mathbf{X} \in \mathbb{R}^{N\times d}$ be the matrix of ℓ_2-normalised patch tokens of the backbone for a single image (rows are the N patch tokens). The affinity-graph weight matrix is $\mathbf{W}_{ij} = \mathrm{ReLU}(\langle \mathbf{x}_i, \mathbf{x}_j\rangle)$ retained over the top-k neighbours and symmetrised, with $\mathbf{D}$ the degree matrix of $\mathbf{W}$ and $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ the normalised Laplacian. Let $\mathbf{U} \in \mathbb{R}^{N\times K_{\max}}$ be the top-$K_{\max}$ eigenvectors of $\mathbf{L}$, and $\mathbf{A}_{i,k} = \mathrm{softmax}_k(|\mathbf{U}_{i,k}|/\tau_a)$ the soft assignment of Eq. 14; the part-selection gate then chooses which of these candidate part channels stay active.

Theorem 1 (Orthogonal invariance of spectral graph assignment under simple spectrum)**.** *Assume the selected Laplacian eigenvalues of* $\mathbf{L}$ *are simple and separated by a positive eigengap* $\delta > 0$*. Then for any orthogonal transformation* $\mathbf{Q} \in O(d)$*, the assignment computed from rotated patch tokens* $\mathbf{X}\mathbf{Q}^\top$ *equals the assignment computed from* $\mathbf{X}$*:*

$$\mathbf{A}(\mathbf{X}\mathbf{Q}^\top) \;=\; \mathbf{A}(\mathbf{X}). \tag{6}$$

Proof. Cosine similarity is invariant under orthogonal transformation: $\langle \mathbf{x}_i\mathbf{Q}^\top, \mathbf{x}_j\mathbf{Q}^\top\rangle = \mathbf{x}_i\mathbf{Q}^\top\mathbf{Q}\mathbf{x}_j^\top = \mathbf{x}_i\mathbf{x}_j^\top = \langle \mathbf{x}_i, \mathbf{x}_j\rangle$. Hence $\mathbf{W}(\mathbf{X}\mathbf{Q}^\top) = \mathbf{W}(\mathbf{X})$, and consequently $\mathbf{D}$ and $\mathbf{L}$ are unchanged. Because the selected eigenvalues are simple and separated by a positive eigengap, each corresponding one-dimensional eigenspace is uniquely determined up to a scalar sign ± 1. Taking the elementwise absolute value $|\mathbf{U}_{i,k}|$ removes this sign ambiguity, ensuring that the soft assignment $\mathbf{A}_{i,k} = \mathrm{softmax}_k(|\mathbf{U}_{i,k}|/\tau_a)$ satisfies $\mathbf{A}(\mathbf{X}\mathbf{Q}^\top) = \mathbf{A}(\mathbf{X})$. □ □

Scope. The eigengap assumption is encouraged during training by the eigengap regulariser $\mathcal{L}_{\text{eigengap}}$ (Tab. 9). Theorem 1 is a property of the assignment rule: the grouping does not depend on the coordinate system of the patch tokens. Different encoders differ in architecture, tokenisation and training, so their tokens are not rotations of one another, and the gains on all 15 encoders of Tab. 14 are empirical evidence rather than a consequence of the theorem.

The second result is the decomposition of the Bures-Wasserstein distance into a mean term and a covariance term. It underlies Sec. 3.2 and shows exactly what the slot mean removes.

Theorem 2 (BW versus slot-mean cross-view signal)**.** *Let* $\nu_d = \mathcal{N}(\mu_d, \Sigma_d)$ *and* $\nu_s = \mathcal{N}(\mu_s, \Sigma_s)$ *on* $\mathbb{R}^{d_b}$ *with* $\Sigma_d, \Sigma_s \succeq \varepsilon\mathbf{I}$ *(*$\varepsilon > 0$*). The BW cross-view consistency loss splits as*

$$\mathcal{L}_{bw}(\nu_d, \nu_s) \;=\; \underbrace{\|\mu_d - \mu_s\|_2^2}_{\mathcal{L}_\mu} \;+\; \underbrace{\mathrm{tr}\big(\Sigma_d + \Sigma_s - 2(\Sigma_d^{1/2}\Sigma_s\Sigma_d^{1/2})^{1/2}\big)}_{\mathcal{L}_\Sigma \geq 0}. \tag{7}$$

The slot-mean reduction ($\Sigma_d = \Sigma_s = \mathbf{0}$*) collapses Eq. 7 to* $\mathcal{L}_\mu$*, removing* $\mathcal{L}_\Sigma$*. Two structural properties hold:* (a) Strict separation: $\mathcal{L}_\Sigma \geq 0$ *with equality iff* $\Sigma_d = \Sigma_s$*, so the BW loss is strictly larger than its slot-mean reduction whenever the learned covariances differ.* (b) Standard-deviation geometry: *for commuting* (Σ_d, Σ_s) *with eigenvalues* $\{\sigma_d^{(i)}\}, \{\sigma_s^{(i)}\}$ *in a common basis (the diagonal case of our deployed head),* $\mathcal{L}_\Sigma = \sum_i(\sqrt{\sigma_d^{(i)}} - \sqrt{\sigma_s^{(i)}})^2$*, which is exactly the squared Euclidean distance between their scale vectors. Thus BW separates cross-view pairs that have identical mean vectors but different learned covariances, which the slot mean cannot separate.*

Proof. Eq. 7 is the standard Bures-Wasserstein decomposition (Bhatia et al., 2019). Non-negativity of $\mathcal{L}_\Sigma$ and the equality condition follow from $\mathcal{L}_\Sigma = \|\Sigma_d^{1/2} - (\Sigma_d^{1/2}\Sigma_s\Sigma_d^{1/2})^{1/2}\Sigma_d^{-1/2}\|_F^2$ on SPD_{d_b} (Álvarez-Esteban et al., 2016), zero iff $\Sigma_d = \Sigma_s$. For (b), commutativity reduces $(\Sigma_d^{1/2}\Sigma_s\Sigma_d^{1/2})^{1/2}$ to $\mathrm{diag}(\sqrt{\sigma_d^{(i)}\sigma_s^{(i)}})$ and $\sigma_d + \sigma_s - 2\sqrt{\sigma_d\sigma_s} = (\sqrt{\sigma_d} - \sqrt{\sigma_s})^2$. □ □

Link to the experiments. Theorem 2 identifies the signal that the slot mean discards, the covariance term $\mathcal{L}_\Sigma$. Whether this term helps retrieval is an empirical question, which the matched comparisons

of Tab. 1 and Tab. 19 (Block A) answer. Appendix A5.7 checks that the prototype variance stays non-degenerate and anisotropic (it differs across feature dimensions) during training.

Corollary 3 (Strict covariance-sensitive tie breaking under fixed assignment)**.** *With a fixed (data-independent) per-prototype assignment, define* $d^2_{\mathrm{BW}}(I_q, I_g) = \sum_k W_2^2(\nu_q^{(k)}, \nu_g^{(k)})$ *and* $d^2_{\mathrm{SM}}(I_q, I_g) = \sum_k \|\mu_q^{(k)} - \mu_g^{(k)}\|_2^2$*, with* $\Delta = d^2_{\mathrm{BW}} - d^2_{\mathrm{SM}} = \sum_k \mathcal{L}_\Sigma^{(k)} \geq 0$*. For any query* I_q *and gallery pair* (I_g^+, I_g^-) *with* $d^2_{\mathrm{SM}}(I_q, I_g^+) = d^2_{\mathrm{SM}}(I_q, I_g^-)$*, the slot mean cannot rank the pair, while the BW distance strictly breaks the tie whenever* $\Delta(I_q, I_g^+) \neq \Delta(I_q, I_g^-)$*. Proof:* $d^2_{\mathrm{BW}} = d^2_{\mathrm{SM}} + \Delta$.

Why retrieval uses cosine similarity rather than pairwise W_2. Pairwise 2-Wasserstein distances between all query and gallery parts are too slow for galleries of 10^4 to 10^5 images and do not fit standard inner-product search indexes such as FAISS. WBPRIOR therefore uses W_2 during training (Eq. 7) to shape the part representations, and the read-out projects $(\boldsymbol{\mu}_k, \mathbf{s}_k)$ into a single embedding that such indexes can search. Corollary 3 holds for a fixed assignment; the deployed model uses the data-dependent assignment of Eq. 14, so its effect on ranking is measured (Tab. 19) rather than derived.

Diagnostic: W_2 re-ranking of a shortlist. This diagnostic is not part of the reported protocol; every other result in the paper is single-pass cosine retrieval without re-ranking. It asks whether the explicit 2-Wasserstein distance (Eq. 7) still carries retrieval signal at test time. On SUES-200 we retrieve the top-M gallery candidates (M=50) by cosine similarity and re-rank them by $D_{\mathrm{BW}}(I_q, I_g) = \sum_k W_2^2\big(\nu_q^{(k)}, \nu_g^{(k)}\big)$ over the $K_{\max}$ part channels, computed from the predicted diagonal parameters $(\boldsymbol{\mu}_k, \mathbf{v}_k)$. Re-ranking raises Corrupt(9) R@1 from 60.29% to 61.18% ($+0.89$ pp) and clean R@1 from 97.53% to 97.88% ($+0.35$ pp). This is consistent with Corollary 3: among close candidates, the part-level distance separates some that the pooled embedding ranks wrongly.

A1.3 BURES-WASSERSTEIN GEOMETRY FOR FULL COVARIANCES

The deployed head uses diagonal Gaussians, whose barycenter has the closed form of Eq. 3. The general case with symmetric positive-definite (SPD) covariance matrices is needed only for the full-covariance control of Appendix A5.3. This subsection restates the standard results for that case (Álvarez-Esteban et al., 2016; Agueh & Carlier, 2011; Bhatia et al., 2019) and shows how the diagonal case follows from them.

The fixed-point characterisation. Given SPD covariances $\{\Sigma_b\}_{b=1}^B$ with weights $\{\lambda_b\}$ on the simplex, the Bures barycenter $\Sigma^\star = \arg\min_{\Sigma \in \mathrm{SPD}_{d_b}} \sum_b \lambda_b W_2^2(\Sigma, \Sigma_b)$ is the unique SPD matrix that satisfies $\Sigma^\star = \sum_{b=1}^B \lambda_b \big((\Sigma^\star)^{1/2} \Sigma_b (\Sigma^\star)^{1/2}\big)^{1/2}$. For general non-commuting covariances, Álvarez-Esteban et al. (2016) compute it with the update

$$\Sigma_{t+1} \;=\; \Sigma_t^{-1/2} \left[\sum_{b=1}^{B} \lambda_b \big(\Sigma_t^{1/2} \Sigma_b \Sigma_t^{1/2}\big)^{1/2} \right]^2 \Sigma_t^{-1/2}, \tag{8}$$

and prove existence, uniqueness on SPD_{d_b}, and global convergence of $\{\Sigma_t\}_{t \geq 0}$ from any initialisation $\Sigma_0 \succ 0$. When all $\{\Sigma_b\}$ commute, as the diagonal covariances $\Sigma_b = \mathrm{diag}(\mathbf{v}_b)$ of our variance head do (Eq. 10), the problem separates per coordinate and has the exact closed-form minimiser

$$\mathbf{s}^\star \;=\; \sum_{b=1}^{B} \lambda_b \, \mathbf{s}_b, \qquad \mathbf{v}^\star = (\mathbf{s}^\star)^2, \quad \mathbf{s}_b = \sqrt{\mathbf{v}_b}, \tag{9}$$

with no matrix-valued solver. In the deployed head, the weights λ_b are the normalised assignment weights $w_{i,k}$ of one image (Eq. 3); the iteration of Eq. 8 is used only in the full-covariance control.

Reduction to the slot mean. If $\Sigma_b = \sigma^2 \mathbf{I}$ for all b with a common $\sigma > 0$, then, using $\sum_b \lambda_b = 1$ and $(\sigma^2 \Sigma)^{1/2} = \sigma\, \Sigma^{1/2}$, the fixed-point condition reduces to $\Sigma^\star = \sum_b \lambda_b \big((\Sigma^\star)^{1/2} \sigma^2 \mathbf{I} \, (\Sigma^\star)^{1/2}\big)^{1/2} = \sigma\, (\Sigma^\star)^{1/2}$, whose unique SPD solution is $\Sigma^\star = \sigma^2 \mathbf{I}$. The barycenter is then the same matrix for every part and every view, the covariance term carries no signal, and the prototype is determined by $\boldsymbol{\mu}^\star$ alone. Letting $\sigma \to 0$ recovers the slot-mean prototype $(\boldsymbol{\mu}^\star, \mathbf{0})$ of Definition 1. The slot mean is therefore a special case of the barycenter prototype; Appendix A5.2 compares the two empirically.

A1.4 Deployed BW Aggregation: Equations, Numerics and Cost

This subsection states what the deployed head computes. The same computation runs at training and test time, and every part prototype is formed from one image's own patches and discarded after the forward pass.

What is computed. For each image and each patch i, two heads on the FiLM-modulated projected token $\tilde{\mathbf{z}}_i \in \mathbb{R}^{d_p}$ produce a mean and a diagonal variance,

$$\boldsymbol{\mu}_i \;=\; \mathbf{W}_\mu \tilde{\mathbf{z}}_i + \mathbf{b}_\mu \;\in\; \mathbb{R}^{d_b}, \qquad\qquad \mathbf{v}_i \;=\; \mathrm{Softplus}\big(\mathbf{W}_v \tilde{\mathbf{z}}_i + \mathbf{b}_v\big) \;\in\; \mathbb{R}^{d_b}_{>0}, \tag{10}$$

so that $\nu_i = \mathcal{N}(\boldsymbol{\mu}_i, \mathrm{diag}(\mathbf{v}_i))$, with the Softplus supplying positivity. Write $\mathbf{s}_i = \sqrt{\mathbf{v}_i}$ coordinate-wise. Given the normalised assignment weights $w_{i,k} = \mathbf{A}_{i,k} / \sum_j \mathbf{A}_{j,k}$ of Eq. 14, the part-k prototype of that image is the diagonal Bures-Wasserstein barycenter

$$\boldsymbol{\mu}_k \;=\; \sum_i w_{i,k}\, \boldsymbol{\mu}_i, \qquad\qquad \mathbf{s}_k \;=\; \sum_i w_{i,k}\, \mathbf{s}_i, \qquad \mathbf{v}_k = \mathbf{s}_k^2. \tag{11}$$

Because the squared 2-Wasserstein distance between diagonal Gaussians splits per coordinate into a mean term and a standard-deviation term (Eq. 1), the objective $\sum_i w_{i,k} \big[\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_i\|_2^2 + \|\mathbf{s}_k - \mathbf{s}_i\|_2^2\big]$ is strictly convex and separable across coordinates, and Eq. 11 is its exact minimiser (Agueh & Carlier, 2011; Bhatia et al., 2019). The cross-view consistency loss uses the same distance between the paired part prototypes:

$$\mathcal{L}_{\text{bw}} \;=\; W_2^2\big(\nu_k^{(d)}, \nu_k^{(s)}\big) \;=\; \|\boldsymbol{\mu}_k^{(d)} - \boldsymbol{\mu}_k^{(s)}\|_2^2 \;+\; \|\mathbf{s}_k^{(d)} - \mathbf{s}_k^{(s)}\|_2^2. \tag{12}$$

Eq. 11 is evaluated per image in the forward pass, at training and at inference, with no sum over minibatches and no state carried between images.

How the scale vector reaches the retrieval embedding. Both barycenter outputs, the mean vector $\boldsymbol{\mu}_k$ and the scale vector $\mathbf{s}_k$, enter the learned read-out $\mathrm{Readout}(\boldsymbol{\mu}_k, \mathbf{s}_k)$ of Eq. 4, so the learned scale reaches the part descriptor through $\mathbf{s}_k$. The descriptors are then weighted by the part-selection gate, pooled, and fused with the CLS embedding by the fusion gate of Eq. 5. Zeroing the variance contribution removes this signal and leaves the assignment, the part-selection gate, the pooling and the fusion untouched; the attribution study of Appendix A5.5 (Tab. 24) uses this intervention.

Numerics. The diagonal head needs no matrix square root. Two operations are forced to float32: the Laplacian eigendecomposition of Eq. 14, which runs in `torch.linalg.eigh` under `autocast(enabled=False)`, and the Sinkhorn iteration of Eq. 13 (Feydy et al., 2019). Cosine affinities are clamped at zero before symmetrisation, so the affinity matrix is non-negative and the normalised Laplacian is well defined.

Cost. With $K_{\max}$=12 part channels, Gaussian-head dimension d_b=32 (projected from a d_p=256 part space), and PK batches of $P{\times}K{=}16{\times}4{=}64$ samples (each contributing $32{\times}32{=}1024$ patches at 448^2 input, average-pooled to $8{\times}8{=}64$ nodes for the Laplacian `eigh`), the aggregation of Eq. 11 costs $\mathcal{O}(K_{\max} \cdot N \cdot d_b)$ element-wise multiply-accumulates and needs no matrix factorisation. This is cheaper than the $\mathcal{O}(K_{\max} \cdot d_b^3)$ full SPD update and under 0.01% of the backbone forward cost ($\sim 2.2{\cdot}10^{10}$ flops at 448^2 on DINOv2 ViT-S/14). Training wall-clock time on A100 is $1.07\times$ that of the slot-mean baseline ($+7\%$) under an otherwise identical recipe. The spectral assignment and the aggregation also run on every query and gallery image at test time; Appendix A1.5 reports that cost.

A1.5 Train-Time and Inference-Time Lifecycle

This subsection separates the quantities computed per image from the persistent model weights, and the deployed inference path from the training-only regularisers (see also Figure 1). Tab. 6 gives the lifetime of every quantity in the head, and Algorithms 1 and 2 give the training and inference paths separately.

No persistent state. No state of the part head survives from one image to the next. Each image builds a k-nearest-neighbour graph from the similarities of its own backbone patch tokens, a within-image affinity structure rather than a retrieval index over the dataset, and takes the top-$K_{\max}$ eigenvectors of its normalised Laplacian (App. A2.2). These eigenvectors, the columns of $\mathbf{U}$, give the patch-to-part assignment, and the graph, eigenvectors and assignment are then discarded. The part-selection gate

makes its keep-or-drop decision per image, and its top-$K_{\min}$ floor applies during training only; the continuous part-selection weights then scale the kept parts during pooling. The only EMA in the system is the training-only teacher, a stop-gradient copy of the model weights used as a distillation target.

Measured inference cost. The part branch runs on every query and gallery image. The deployed model has 23.42 M parameters and costs 22.14 GFLOPs at 448^2 (7.23 GFLOPs at 256^2 on DenseUAV; Table 3), the smallest budget among the compared methods. Within the head, the dominant operation at test time is the Laplacian eigendecomposition, which takes 5.30 ms at the deployed 8×8 grid (Appendix A2.3, Table 8). The barycenter aggregation adds only element-wise operations (App. A1.4).

Gradient path. Every patch Gaussian and every diagonal BW part prototype is recomputed in the student forward pass. The aggregation, the read-out of Eq. 4, the part-selection weighting, the pooling and the CLS fusion stay on the autograd graph, with no `detach` and no EMA applied to prototypes, so retrieval and consistency gradients reach the Gaussian heads of Eq. 10, the descriptor projections and the unfrozen backbone blocks. Three steps are not differentiable in the usual sense: the within-image k-nearest-neighbour selection that builds the graph is a discrete choice; the symmetric eigendecomposition of $\mathbf{L}$ is differentiable only away from repeated eigenvalues; and the hard part-selection gate uses a straight-through estimator.

Table 6: Lifetime of every WBPRIOR quantity. The top block is the deployed forward pass and runs identically at training and test time; the bottom block is training-only. Nothing persists across images: the graph, Laplacian, eigenvectors, assignments, patch Gaussians and barycenters are rebuilt from each image's own patches and then discarded. The top-$K_{\min}$ floor of the part-selection gate is training-only; at inference the gate is the noise-free threshold.

Operation or quantity	Training	Inference	Lifetime or role
CLS and patch extraction	Yes	Yes	Per-image inputs
Graph, Laplacian, eigenvectors, assignments	Yes	Yes	Per image, then discarded
Patch Gaussians and BW parts	Yes	Yes	Descriptor construction
Part-selection gate, FiLM, pooling, fusion	Yes	Yes	Retrieval embedding
Functional map, Sinkhorn and BW consistency	Yes	No	Paired-view regularisation
EMA teacher	Yes	No	Stop-gradient target
Decoder, task heads, auxiliary losses	Yes	No	Training supervision

A1.6 DEBIASED SINKHORN DIVERGENCE

The training-only part-quality term $\mathcal{L}_{\text{sink}}$ is the debiased Sinkhorn divergence (Feydy et al., 2019) between the patch distributions of each part in the two paired views, not the plain entropic OT cost. This subsection explains the choice and gives the settings.

Three properties. Let $\mathrm{OT}_\varepsilon(\nu_d, \nu_s)$ be the entropy-regularised OT cost with regularisation $\varepsilon \geq 0$ and S_ε the debiased divergence

$$S_\varepsilon(\nu_d, \nu_s) \;=\; \mathrm{OT}_\varepsilon(\nu_d, \nu_s) - \tfrac{1}{2}\big(\mathrm{OT}_\varepsilon(\nu_d, \nu_d) + \mathrm{OT}_\varepsilon(\nu_s, \nu_s)\big). \tag{13}$$

Then Feydy et al. (2019, Thm. 1, 3, 5) prove that S_ε is (P1) *positive*: $S_\varepsilon(\nu, \nu) = 0$ and $S_\varepsilon(\nu_d, \nu_s) \geq 0$; (P2) *consistent as* $\varepsilon \to 0$: $S_\varepsilon(\nu_d, \nu_s) \to W_2^2(\nu_d, \nu_s)$ on bounded support, with rate $O(\varepsilon)$ on quadratic costs; and (P3) *metrising weak convergence*: $S_\varepsilon(\nu_n, \nu) \to 0 \Leftrightarrow \nu_n \rightharpoonup \nu$, so the divergence goes to zero exactly when the distributions converge weakly. Plain OT_ε lacks these properties at fixed $\varepsilon > 0$: in particular $\mathrm{OT}_\varepsilon(\nu, \nu) \neq 0$, an entropic self-bias that the debiasing in Eq. 13 removes, and it does not metrise weak convergence.

Role in the loss. S_ε enters the part-quality group $\mathcal{G}_{\text{part}}$ next to the BW consistency loss $\mathcal{L}_{\text{bw}}$ (Eq. 12). The two terms are complementary: $\mathcal{L}_{\text{bw}}$ constrains the part prototype (Eq. 11), and S_ε constrains the matching of the patches assigned to that part. Property (P2) links the value optimised at $\varepsilon = 0.1$ (Appendix A5.8) to the limiting W_2^2 objective, and property (P3) makes S_ε a soft constraint on distributions rather than on individual samples.

Implementation. We compute S_ε per active part on the patch features assigned to that part in the two views. The cost matrix holds squared Euclidean distances between patch features projected through

Algorithm 1 WBPRIOR **training** step, one view of one image. Lines marked *[T]* exist only at training time; lines 1 to 7 are shared verbatim with inference (Alg. 2), and line 8 is shared with the gate reduced to its noise-free threshold. No quantity produced here outlives the image.

Require: Patch tokens $\mathbf{X} \in \mathbb{R}^{N \times d}$ and CLS token $\mathbf{c}$ from the student backbone; channel ceiling $K_{\max}$, active floor $K_{\min}$, temperatures τ_a, τ_g, neighbour count κ.

1: $\mathbf{Z} \leftarrow \mathrm{Proj}(\mathbf{X})$; $(\boldsymbol{\gamma}, \boldsymbol{\beta}) \leftarrow \mathrm{FiLM}(\mathbf{c})$; $\tilde{\mathbf{Z}} \leftarrow \boldsymbol{\gamma} \odot \mathbf{Z} + \boldsymbol{\beta}$ ▷ per image

2: $\mathbf{W} \leftarrow \mathrm{sym}\big(\text{top-}\kappa\ \max(\bar{\mathbf{X}}\bar{\mathbf{X}}^\top, 0)\big)$ on the row-ℓ_2-normalised native patch tokens $\bar{\mathbf{X}}$, self-loops removed; $\mathbf{L} \leftarrow \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ ▷ **not differentiable**: discrete k-NN selection

3: $\mathbf{U} \leftarrow$ top-$K_{\max}$ eigenvectors of $\mathbf{L}$ (`eigh`, float32) ▷ differentiable away from repeated eigenvalues

4: $\mathbf{A}_{i,k} \leftarrow \mathrm{softmax}_k\big(|\mathbf{U}_{i,k}|/\tau_a\big)$; $w_{i,k} \leftarrow \mathbf{A}_{i,k} / \sum_j \mathbf{A}_{j,k}$

5: $\boldsymbol{\mu}_i, \mathbf{v}_i \leftarrow$ linear and Softplus heads on $\tilde{\mathbf{z}}_i$ ▷ Eq. 10; learned diagonal Gaussian

6: $\boldsymbol{\mu}_k \leftarrow \sum_i w_{i,k}\boldsymbol{\mu}_i$; $\mathbf{s}_k \leftarrow \sum_i w_{i,k}\sqrt{\mathbf{v}_i}$; $\mathbf{v}_k \leftarrow \mathbf{s}_k^2$ ▷ Eq. 11

7: $\hat{\mathbf{p}}_k \leftarrow \mathrm{Readout}(\boldsymbol{\mu}_k, \mathbf{s}_k)$ ▷ Eq. 4; both coordinates enter

8: $g_k \leftarrow \mathbb{1}\big[\sigma((\ell_k + \epsilon_k)/\tau_g) > \frac{1}{2}\big]$ with a straight-through gradient, top-$K_{\min}$ fallback (training only); $\mathbf{f} \leftarrow \mathrm{L2Norm}\big(\mathrm{Fuse}(\mathrm{Pool}(\{g_k\hat{\mathbf{p}}_k\}), \mathbf{f}_{\mathrm{cls}})\big)$ ▷ **not differentiable**: STE; Eq. 5

9: *[T]* With the paired view: $\mathbf{C} \leftarrow \mathbf{U}_d^\top \mathbf{U}_s$; accumulate $\mathcal{L}_{\text{fmap}}$, $\mathcal{L}_{\text{fmap-iso}}$ into $\mathcal{G}_{\text{align}}$

10: *[T]* Accumulate $\mathcal{L}_{\text{bw}} = W_2^2\big(\nu_k^{(d)}, \nu_k^{(s)}\big)$ and $\mathcal{L}_{\text{sink}} = S_\varepsilon$ per active k into $\mathcal{G}_{\text{part}}$

11: *[T]* Accumulate $\mathcal{L}_{\text{mar}}$, the altitude auxiliary, the metric-learning terms, and the distillation terms whose targets come from the stop-gradient weight teacher θ^{ema}

12: *[T]* Backpropagate the total loss, weighted by the learned multi-task loss weights of Kendall et al. (2018), into the Gaussian heads $\{\mathbf{W}_\mu, \mathbf{W}_v\}$, the read-out, the projections, the gate logits, FiLM, the fusion gate and the unfrozen backbone blocks; then $\theta^{\text{ema}} \leftarrow m\,\theta^{\text{ema}} + (1-m)\,\theta$

13: **Discard** $\mathbf{W}, \mathbf{L}, \mathbf{U}, \mathbf{A}, \{\nu_i\}, \{(\boldsymbol{\mu}_k, \mathbf{v}_k)\}, \mathbf{C}$ ▷ no prototype, variance or graph state is retained

Algorithm 2 WBPRIOR **inference** for a single image (query or gallery). Identical to lines 1 to 8 of Alg. 1 with the gate reduced to its noise-free threshold (no top-$K_{\min}$ fallback). No partner view, no teacher, no loss term, no persistent state, and no re-ranking or test-time augmentation.

Require: Patch tokens $\mathbf{X}$ and CLS token $\mathbf{c}$; trained parameters only.

1: $\mathbf{Z} \leftarrow \mathrm{Proj}(\mathbf{X})$; $(\boldsymbol{\gamma}, \boldsymbol{\beta}) \leftarrow \mathrm{FiLM}(\mathbf{c})$; $\tilde{\mathbf{Z}} \leftarrow \boldsymbol{\gamma} \odot \mathbf{Z} + \boldsymbol{\beta}$

2: Build $\mathbf{W}$, $\mathbf{L}$ and $\mathbf{U}$ from *this image's* native patch tokens; $\mathbf{A}$ and w as in Alg. 1

3: $\boldsymbol{\mu}_i, \mathbf{v}_i \leftarrow$ Gaussian heads (Eq. 10); $\boldsymbol{\mu}_k, \mathbf{s}_k \leftarrow$ diagonal BW barycenter (Eq. 11)

4: $\hat{\mathbf{p}}_k \leftarrow \mathrm{Readout}(\boldsymbol{\mu}_k, \mathbf{s}_k)$ (Eq. 4); $g_k \leftarrow \mathbb{1}[\sigma(\ell_k/\tau_g) > \frac{1}{2}]$ with no sampled noise and no $K_{\min}$ floor

5: $\mathbf{f} \leftarrow \mathrm{L2Norm}\big(\mathrm{Fuse}(\mathrm{Pool}(\{g_k\hat{\mathbf{p}}_k\}), \mathbf{f}_{\mathrm{cls}})\big)$

6: **return** $\mathbf{f} \in \mathbb{R}^{768}$ ▷ retrieval is cosine similarity between such vectors

a shared linear head, $C_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|^2$, with regularisation $\varepsilon = 0.1$ and 20 Sinkhorn iterations. The value of ε is one of the few hyperparameters of the part head that we tune; the sweep is in Appendix A5.8. Sinkhorn inputs are forced to float32: entropic OT in float16 produces NaN gradients on A100 once a column probability drops below $\sim 10^{-3}$, the numerical issue that Feydy et al. (2019) also report.

A2 METHOD RECIPE

A2.1 HYPERPARAMETER CONFIGURATION

Table 7 lists the hyperparameters of the deployed model.

A2.2 PART HEAD: SPECTRAL ASSIGNMENT, GATE, PAIRED-VIEW LOSSES AND FILM

This subsection specifies the part head in enough detail to reimplement it. The spectral assignment, the part-selection gate and FiLM run at inference; the debiased Sinkhorn divergence, the functional map, the altitude head and the masked-patch reconstruction are training only (Tab. 6).

Patch-affinity graph and spectral assignment. For each view we build a sparse affinity graph over the $32\times32{=}1024$ backbone patch tokens (448^2 input, ViT-S/14). The token grid is first downsampled by a factor of 4 to an $8\times8{=}64$-node grid, on which the graph is built and `eigh` is stable. The resulting

Table 7: Hyperparameters of the deployed model. The EMA teacher is training-only (Tab. 6). The five-seed matched studies of Appendix A5.3 use their own configuration.

Category	Parameter	Value
Architecture	Backbone	DINOv2 ViT-S/14
	Unfrozen blocks	6 (of 12)
	Maximum part channels $K_{\max}$	12
	Min. active parts $K_{\min}$ (train only)	4
	Part dimension d_p	256
	Embedding dim D	768
	Teacher dim	768
	Cluster temperature τ	0.07
BW part head	Spectral-assignment temp. τ_a	0.07
	Affinity graph k-NN size	8
	Gaussian-head dimension d_b	32
	Sinkhorn regularisation ε	0.1
	Sinkhorn iterations	20
Training	Input size	448×448 (256×256 on DenseUAV)
	Epochs	120
	PK sampler (P, K)	(16, 4)
	Head LR	3×10^{-4}
	Backbone LR	3×10^{-5}
	Weight decay	0.05
	Warmup epochs	5
	LR floor	1% of peak
Loss	Proxy margin δ	0.1
	Proxy scale s	32
	EMA teacher decay m (training only)	0.996
	Masked-patch (MAR) ratio	0.30
	Reconstruction warm-up	10 epochs
	Distillation temp	4.0
	Num loss groups	4
	Label smoothing	0.1
Hardware	GPU	NVIDIA A100 80 GB
	Mixed precision	torch.amp (FP16)

eigenvectors are bilinearly upsampled back to the 32×32 grid, so the assignment $\mathbf{A}$ of Eq. 14 is defined on all $N{=}1024$ patch tokens. Only the eigendecomposition is coarse: the assignment and the BW aggregation both operate on the full 32×32 patch grid, whose features are never blurred or pooled. The accuracy-compute trade-off of this choice is measured in App. A2.3. Edge weights are the cosine similarities of the ℓ_2-normalised backbone patch tokens; the projected and FiLM-modulated tokens feed only the Gaussian heads. Negative similarities are clamped to zero, we keep the top-8 neighbours per node, and we symmetrise, so the affinity is non-negative and the normalised Laplacian $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ is well defined. The top-$K_{\max}$ eigenvectors are computed by `torch.linalg.eigh` in float32; in float16 the assignment becomes rank-deficient, because nearby Laplacian eigenvalues collide under the float16 epsilon. These eigenvectors form the columns of $\mathbf{U}\in\mathbb{R}^{N\times K_{\max}}$, and its rows give the candidate soft assignment

$$\mathbf{A}_{i,k} \;=\; \mathrm{softmax}_k\big(|\mathbf{U}_{i,k}|/\tau_a\big), \qquad \tau_a = 0.07. \tag{14}$$

The absolute value removes the arbitrary sign of each eigenvector. It folds the two halves of each spectral cut onto the same part index, giving up the bipartition property of the signed eigenvector in exchange for sign stability. The softmax is over part indices, not over patch positions, so the assignment is a competition among parts rather than a re-weighting of patches. The orthogonality regulariser $\mathcal{L}_{\text{orth}}$ (part-quality group, Tab. 9) and the temperature $\tau_a{=}0.07$ keep the part channels well separated. Because part indices follow the eigenvalue order of each image's own Laplacian, they are not learned parameters and no prototype bank is stored: part k is the k-th of the top-$K_{\max}$ eigenvectors of that image's graph. During training the functional map (Eq. 16) makes this ordering

agree across paired views; at test time each image is processed alone. The part-selection gate then selects the active channels.

Part-selection gate. A binary-concrete (logistic-sigmoid) gate over the $K_{\max}$=12 part channels decides which parts are active for the image. It is a sigmoid of a learned logit plus logistic noise, which relaxes the keep-or-drop choice into a differentiable one:

$$g_k = \sigma\big((\ell_k + \epsilon_k)/\tau_g\big), \qquad \epsilon_k \sim \mathrm{Logistic}(0,1), \qquad \tau_g = 0.5, \tag{15}$$

where $\epsilon_k = \log u_k - \log(1-u_k)$ with $u_k \sim \mathrm{Uniform}(0,1)$ is symmetric Logistic noise and ℓ_k is the learned gate logit. At training time the continuous $g_k \in (0,1)$ is binarised by a straight-through estimator, which uses the hard decision (threshold 0.5) in the forward pass but passes the gradient of the soft sigmoid backward. At inference it is binarised by the deterministic threshold $g_k > 0.5$. A training-time floor of $K_{\min}$=4 active parts (top-$K_{\min}$ override if the gate fires fewer) prevents collapse onto a single part channel early in training; it is not applied at inference. The eigendecomposition of Eq. 14 is computed for the fixed $K_{\max}$ candidate channels; the gate then selects the active subset.

Debiased Sinkhorn divergence. For each active part, the debiased Sinkhorn divergence S_ε of Eq. 13 compares the patch features assigned to that part in the two paired views. It is chosen because it is positive and consistent with W_2^2 as $\varepsilon \to 0$; App. A1.6 gives these properties together with the cost matrix and settings. It is added to the part-quality group $\mathcal{G}_{\text{part}}$ together with the BW consistency loss $\mathcal{L}_{\text{bw}} = W_2^2(\nu_k^{(d)}, \nu_k^{(s)})$ between the paired part prototypes (Eq. 12).

Functional map. The eigenbases $\mathbf{U}_d, \mathbf{U}_s$ of the two paired views have arbitrary sign and ordering. A functional map (Ovsjanikov et al., 2012), a linear map between the two spectral bases, aligns them:

$$\mathbf{C} \;=\; \mathbf{U}_d^\top \mathbf{U}_s \in \mathbb{R}^{K_{\max} \times K_{\max}}. \tag{16}$$

We add two soft constraints to the alignment group (Table 9):

$$\mathcal{L}_{\text{fmap}} = \|\mathbf{C}\,\mathbf{F}_d - \mathbf{F}_s\|_F^2 \quad \text{(descriptor transfer)}, \tag{17}$$

$$\mathcal{L}_{\text{fmap-iso}} = \|\mathbf{C}\mathbf{C}^\top - \mathbf{I}\|_F^2 \quad \text{(map bijectivity)}, \tag{18}$$

where $\mathbf{F}_d = \mathbf{U}_d^\top \mathbf{X}_d$ and $\mathbf{F}_s = \mathbf{U}_s^\top \mathbf{X}_s$ are the spectral coefficients of the backbone patch tokens in their own bases. These terms make the same part index correspond to the same region in both views without part-level labels. $\mathbf{C}$ maps drone-basis coordinates to satellite-basis coordinates, which makes $\|\mathbf{C}\mathbf{F}_d - \mathbf{F}_s\|_F^2$ the natural residual. The product $\mathbf{U}_d^\top \mathbf{U}_s$ sums over the shared spatial grid, so the term assumes that the two views are coarsely registered, as the nadir-aligned drone and satellite tiles of SUES-200, University-1652 and DenseUAV are. This condition binds at training time only, since no functional-map solver runs at inference, and the term would not transfer to severely unaligned ground-to-aerial pairs; App. A5.6 (Tab. 25) measures this dependence. Eigenvectors are sign-arbitrary: the assignment uses the sign-invariant $|\mathbf{U}|$, but $\mathbf{C}$ is computed from the raw $\mathbf{U}$, so its rows can flip between batches. The bijectivity term is unaffected because $(\pm\mathbf{C})(\pm\mathbf{C})^\top = \mathbf{C}\mathbf{C}^\top$, and the eigenvalue ordering resolves row and column permutations.

CLS-conditioned FiLM. The head modulates the projected tokens that feed the Gaussian heads as $\tilde{\mathbf{z}}_i = \boldsymbol{\gamma} \odot \mathbf{z}_i + \boldsymbol{\beta}$, where $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ are predicted from the same view's CLS token by a 2-layer MLP. The affinity graph and the assignment use the unmodulated backbone patch tokens. During training, an auxiliary 4-way altitude head supervises the CLS encoder (cross-entropy weight 0.3); it is discarded at inference, where FiLM depends only on the CLS token and needs no altitude metadata. The same operator therefore applies to SUES-200 (altitude bins), University-1652 (no altitude metadata) and DenseUAV (height triple).

Auxiliary masked-patch reconstruction. A small auxiliary loss reconstructs 30% of masked patch features from their part-aggregated descriptor, $\hat{\mathbf{z}}_i = \mathrm{MLP}(\sum_k \mathbf{A}_{i,k}\hat{\mathbf{p}}_k)$, with a cosine-similarity loss; it is training only and belongs to the part-quality group. Its effect on retrieval is small (removing it costs ~ 0.4 pp R@1 on SUES-200), but it stabilises the early-epoch eigendecomposition by preventing patches from collapsing onto one part.

A2.3 Eigendecomposition Resolution: Accuracy and Compute

The spectral assignment of §A2.2 solves its eigendecomposition on a coarse 8×8 grid. The coarse grid is used only for that step: eigenvectors are upsampled before assignment, and both the assignment

(Eq. 14) and the BW aggregation (Eq. 3) operate on the full 32×32 patch grid. Table 8 re-runs the deployed model with two finer eigendecomposition grids, holding the rest of the head, the recipe and the seeds fixed, and times the eigendecomposition alone.

Table 8: **Eigendecomposition grid: accuracy and compute** (SUES-200 D→S, five-seed matched study; Clean N=80,000, Corrupt(9) N=720,000 paired decisions). Only the grid on which `eigh` is solved changes; assignment and BW aggregation always use the 32×32 patch grid. Time is the latency of the eigendecomposition alone; Memory is peak GPU memory during full-model batch evaluation. The deployed configuration is the coarsest grid (8×8).

Eigendecomposition grid	Clean R@1	Corrupt(9) R@1	Δ Corrupt	Time	Memory
Deployed (coarsest)	**97.53**	**60.29**	reference	**5.30 ms** (1.00×)	**11.80 GB**
Finer	97.60	60.52	+1,649/720,000 (+0.23 pp)	46.70 ms (8.81×)	18.40 GB
Finest	97.49	60.43	+1,047/720,000 (+0.15 pp)	583.40 ms (110.08×)	39.70 GB

Reading the table. Clean R@1 stays between 97.49 and 97.60 across the whole range and is not monotone in resolution. Under corruption the finer grids are slightly ahead (+0.23 pp and +0.15 pp on 720,000 paired decisions), at 8.81× and 110.08× the eigendecomposition time and 18.40 GB and 39.70 GB of peak memory against 11.80 GB. We therefore keep the coarsest grid. It sets the resolution of the grouping, not of the descriptor.

A2.4 Uncertainty-Aware Proxy Alignment (UAPA): Formulation and Forward-KL Ablation

Formulation. Uncertainty-aware proxy alignment (UAPA) sets the temperature of the proxy-distillation KL term from the entropy gap between the two views, so that a less certain query receives a softer alignment target. Let $\mathbf{p}_q, \mathbf{p}_g$ be the softmax probabilities of the query and gallery views over the $|C|$ proxy classes (one proxy per training location: $|C|$=120 for SUES-200, 701 for University-1652, 2,256 for DenseUAV). On the drone benchmarks, the query is the drone view and the gallery is the satellite view. UAPA sets the KL temperature as

$$T = T_0\big(1 + \max(0, \Delta H)/H_{\max}\big), \qquad \Delta H = H(\mathbf{p}_q) - H(\mathbf{p}_g), \tag{19}$$

with $H(\cdot)$ Shannon entropy, T_0=4, and $H_{\max} = \log|C|$ normalizing the gap. The $\max(0,\cdot)$ clamp ensures the temperature can only rise relative to T_0. When the query view is broader than the gallery view ($\Delta H > 0$), the alignment target is softened. When the query is sharper ($\Delta H < 0$), we keep T=T_0 rather than sharpen further, because the gallery proxy is the reference target. The effective range is therefore $[T_0, 2T_0] = [4, 8]$. The proxy-classification head reuses the $|C|$ proxy embeddings from metric learning; each logit is the cosine similarity to a proxy. Entropy is computed over these $|C|$ probabilities. Only the KL temperature is modulated; gradient magnitude is preserved and no whole-sample re-weighting is applied.

We use reverse KL (drone‖satellite) rather than forward KL. Reverse KL is mode-seeking: it concentrates on the dominant target mode, so the drone branch concentrates on the most confident satellite peak. Forward KL is mode-covering: it spreads mass to cover lower-probability modes. A forward-KL variant plateaus roughly 1 pp lower in R@1, while the learned group weights of the grouped multi-task objective (App. A2.5) converge to similar ratios under either choice (alignment 0.97, distill 1.01, part 1.03, alt 1.04 for forward KL). This suggests that self-calibration is robust to the KL direction.

The closest prior is the uncertainty-aware prediction alignment of MobileGeo (Sun et al., 2025), which re-weights samples by a per-sample uncertainty $\hat{u}(x)$. UAPA differs in three respects. It modulates only the KL temperature and so preserves gradient magnitude; it conditions on the relative entropy gap between the two views; and it sits inside a loss group with a learned weight in the grouped multi-task objective rather than following a standalone schedule.

A2.5 Grouped Multi-Task Objective: Training Details

The grouped multi-task objective of Tab. 9 gives each loss group one learned weight (Kendall et al., 2018). The learned weights $\exp(-s_g)$ change during training rather than staying at a fixed split: from

Table 9: Loss groups of the grouped multi-task objective. The entry $\mathcal{L}_{\text{uapa}}$ is the proxy-distillation KL term of §3.3, not a separate penalty: UAPA is the entropy-gap rule that sets that term's temperature T (Eq. 19, Appendix A2.4), so switching UAPA off only fixes $T{=}T_0$.

Group	Constituent losses	Weighting
Alignment	$\mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{nce}} + \mathcal{L}_{\text{proxy}} + \mathcal{L}_{\text{circle}} + \mathcal{L}_{\text{pnce}} + \mathcal{L}_{\text{vicreg}} + \mathcal{L}_{\text{uapa}} + \mathcal{L}_{\text{fmap}} + \mathcal{L}_{\text{fmap-iso}}$	$\exp(-s_{\text{align}})$
Distillation	$\mathcal{L}_{\text{cross}} + \mathcal{L}_{\text{ema}}$	$\exp(-s_{\text{distill}})$
Part quality	$\mathcal{L}_{\text{mar}} + \mathcal{L}_{\text{sparse}} + \mathcal{L}_{\text{sink}} + \mathcal{L}_{\text{bw}} + \mathcal{L}_{\text{orth}} + \mathcal{L}_{\text{eigengap}}$	$\exp(-s_{\text{part}})$
Altitude	$\mathcal{L}_{\text{alt}} + \mathcal{L}_{\text{alt-aux}}$	$\exp(-s_{\text{alt}})$
Regularizer	$+\sum_g s_g$ (Kendall uncertainty)	automatic

uniform initialisation they diverge within the first 20 epochs and settle by epoch 120 at approximately alignment 0.94, distillation 1.05, part quality 1.12 and altitude 1.18. Alignment dominates early, while the embedding geometry forms, and relaxes as metric learning saturates, while the part-quality and altitude weights rise. The regulariser $+\sum_g s_g$ keeps every group weight from collapsing to zero. The patch-level NCE term in the alignment group uses as positives the patch pairs assigned to the same part index in both views and as negatives all other patches in the batch.

A3 Datasets and Per-Benchmark Detail

A3.1 Dataset Statistics and Evaluation Protocol

Table 10: Training split sizes and test query/gallery counts for both retrieval directions on the three benchmarks.

Dataset	Training phase		Test phase			
			Drone→Satellite		Satellite→Drone	
	Drone	Satellite	Query	Gallery	Query	Gallery
University-1652 (Zheng et al., 2020)	37,854	701	37,855	951	701	51,355
SUES-200 (Zhu et al., 2023)	24,000	120	16,000	200	50	40,000
DenseUAV (Dai et al., 2024)	6,768	13,536	2,331	18,198	4,662	9,099

A3.2 Cross-View Geo-Localization: Full per-Benchmark Detail

This subsection expands Table 3 into its full form: per-altitude results on SUES-200 (Table 11) and further published baselines on University-1652 (Table 12) and DenseUAV (Table 13). Baselines are taken as published, and the colours follow the main table.

Table 11: **SUES-200 per-altitude comparison.** R@1/AP (%) per altitude, both directions. Mean column = mean R@1 across the four altitudes. **Red**: best; **navy**: second.

Method	Venue	Params↓	GFLOPs↓	Drone→Satellite									Satellite→Drone								
				150m		200m		250m		300m		Mean	150m		200m		250m		300m		Mean
				R@1	AP	R@1	AP	R@1	AP	R@1	AP	R@1	R@1	AP	R@1	AP	R@1	AP	R@1	AP	R@1
Baseline (Zhu et al., 2023)	TCSVT'23	-	-	55.65	61.92	66.78	71.55	72.00	76.43	74.05	78.26	67.12	75.00	55.46	85.00	66.05	86.25	69.94	88.75	74.46	83.75
LPN (Wang et al., 2022)	TCSVT'22	62.4 M	36.78	61.58	67.23	70.85	75.96	80.38	83.80	81.47	84.53	73.57	83.75	66.78	88.75	75.01	92.50	81.34	92.50	85.72	89.38
MCCG (Shen et al., 2024)	TCSVT'24	56.6 M	51.04	82.22	85.47	89.38	91.41	93.82	95.04	95.07	96.20	90.12	93.75	89.72	93.75	92.21	96.25	96.14	98.75	96.64	95.62
SeGCN (Liu et al., 2024)	JSTARS'24	**28.0 M**	**24.58**	90.80	92.32	91.93	93.41	92.53	93.90	93.33	94.61	92.15	93.75	92.45	95.00	93.65	96.25	94.39	97.50	94.55	95.62
CCR (Du et al., 2024)	TCSVT'24	156.6 M	160.6	87.08	89.55	93.57	94.90	95.42	96.28	96.82	97.39	93.22	92.50	88.54	97.50	95.22	97.50	97.10	97.50	97.49	96.25
Sample4Geo (Deuser et al., 2023)	ICCV'23	87.6 M	90.24	92.60	94.00	97.38	97.81	98.28	98.64	99.18	99.36	96.86	**97.50**	93.63	**98.75**	96.70	**98.75**	**98.28**	**98.75**	98.05	98.44
CAMP (Wu et al., 2024)	TGRS'24	-	90.24	95.40	96.38	97.63	98.16	98.05	98.45	99.33	99.46	97.60	96.25	93.69	97.50	96.76	98.75	98.10	**100.00**	98.85	98.12
DAC (Xia et al., 2024)	TCSVT'24	96.5 M	90.24	**96.80**	**97.54**	97.48	97.97	98.20	98.62	97.58	98.14	97.52	**97.50**	94.06	**98.75**	96.66	**98.75**	98.09	**98.75**	97.87	98.44
MEAN (Chen et al., 2025)	TGRS'25	36.5 M	26.18	95.50	96.46	**98.38**	**98.72**	**98.95**	**99.17**	**99.52**	**99.63**	**98.09**	**97.50**	**94.75**	**100.00**	**97.09**	**100.00**	**98.28**	**100.00**	**99.21**	**99.38**
WBPrior (Ours)	-	**23.42 M**$_{\downarrow 16\%}$	**22.14**$_{\downarrow 10\%}$	**96.58**	**97.42**	**98.23**	**98.61**	**99.13**$_{\uparrow 0.18}$	**99.34**$_{\uparrow 0.17}$	**99.35**	**99.52**	**98.32**$_{\uparrow 0.23}$	**98.75**$_{\uparrow 1.25}$	**96.08**$_{\uparrow 1.33}$	**100.00**	**98.12**$_{\uparrow 1.03}$	**100.00**	**98.71**$_{\uparrow 0.43}$	**100.00**	**99.34**$_{\uparrow 0.13}$	**99.69**$_{\uparrow 0.31}$

A3.3 Matched-Control Cross-Backbone Study (FS-COCO Unseen)

A ranking by absolute accuracy cannot tell whether a backbone is strong or whether the head helps on it. This study answers the second question with a matched global control for every encoder.

Table 12: **University-1652 state-of-the-art comparison.** R@1/AP (%) reported separately for both retrieval directions: **Drone→Satellite** (left block) and **Satellite→Drone** (right block). Single-scale 448×448, full confusion gallery. Cell shading scales with R@1 (green = high). **Red**: best; **navy**: second; "-" = not reported.

			Drone→Satellite		Satellite→Drone	
Method	**Venue**	**Params**	**R@1**	**AP**	**R@1**	**AP**
LPN (Wang et al., 2022)	TCSVT'22	62.4 M	77.71	80.80	90.30	78.78
MCCG (Shen et al., 2024)	TCSVT'24	56.6 M	89.40	91.07	95.01	89.93
CEUSP (Xu et al., 2026)	PRCV'25	-	90.14	91.54	93.30	89.35
CCR (Du et al., 2024)	TCSVT'24	156.6 M	92.54	93.78	95.15	91.80
Sample4Geo (Deuser et al., 2023)	ICCV'23	87.6 M	92.65	93.81	95.14	91.39
MEAN (Chen et al., 2025)	TGRS'25	36.5 M	93.55	94.53	96.01	92.08
CAMP (Wu et al., 2024)	TGRS'24	-	94.46	95.38	96.15	92.72
DAC (Xia et al., 2024)	TCSVT'24	96.5 M	**94.67**	**95.50**	**96.43**	**93.79**
WBPRIOR (Ours)	-	**23.42 M**$_{\downarrow 36\%}$	**94.86**$_{\uparrow 0.19}$	**96.08**$_{\uparrow 0.58}$	**97.57**$_{\uparrow 1.14}$	**94.92**$_{\uparrow 1.13}$

Table 13: **DenseUAV state-of-the-art comparison** (D→S, 256×256, dataset convention). R@1, R@5, R@top1, AP, SDM@$\{1, 3, 5\}$. SDM@K is DenseUAV's spatial distance score at rank K, which also rewards retrieving tiles close to the true location (Dai et al., 2024). **Red**: best; **navy**: second; "-" = not reported. Baselines from (Xu et al., 2026). Sample4Geo is listed at 49.38% as reported by Xu et al. (2026); Chen et al. (2026) report 80.57% for it, the value used in Tab. 3. LPN is evaluated with both ResNet-50 (62.4 M params) and ViT-S.

Method	**Venue**	**Backbone**	**Params**	**R@1**	**R@5**	**R@top1**	**AP**	**SDM@1**	**SDM@3**	**SDM@5**
Instance Loss (Zheng et al., 2020)	MM'20	ResNet-50	-	13.00	35.78	-	-	23.61	-	-
LCM (Ding et al., 2021)	RS'21	ResNet-50	52.7 M	25.37	50.92	-	-	35.52	-	-
LPN (Wang et al., 2022)	TCSVT'22	ResNet-50	62.4 M	32.43	56.80	-	-	40.26	-	-
RK-Net (Lin et al., 2022)	TIP'22	ResNet-50	-	38.74	62.85	-	-	45.78	-	-
Sample4Geo (Deuser et al., 2023)	ICCV'23	ConvNeXt-B	87.6 M	49.38	78.29	99.40	35.93	61.72	-	-
LPN (Wang et al., 2022)	TCSVT'22	ViT-S	-	71.77	90.13	-	-	77.95	-	-
DenseUAV (Dai et al., 2024)	TIP'23	ViT-S	-	83.01	95.58	**99.91**	72.10	86.50	84.50	**80.44**
MCCG (Shen et al., 2024)	TCSVT'24	ConvNeXt-T	56.6 M	83.14	93.39	99.74	72.60	85.94	84.32	80.14
CAMP (Wu et al., 2024)	TGRS'24	ConvNeXt-B	-	88.72	-	-	-	-	-	-
Yang et al. (Yang et al., 2025)	RA-L'25	DINOv2-B	-	86.27	**96.83**	-	-	88.87	-	-
CEUSP (Xu et al., 2026)	PRCV'25	ConvNeXt-T	-	89.45	96.05	**100.00**	**79.62**	**91.01**	**89.42**	**85.34**
MEAN (Chen et al., 2025)	TGRS'25	ConvNeXt-T	36.5 M	**90.18**	-	-	-	-	-	-
WBPRIOR (Ours)	-	**ViT-S/14**	**23.42 M**$_{\downarrow 36\%}$	**90.48**$_{\uparrow 0.30}$	**97.17**$_{\uparrow 0.34}$	99.31	**91.24**$_{\uparrow 11.62}$	**92.55**$_{\uparrow 1.54}$	**91.11**$_{\uparrow 1.69}$	78.56

For every encoder we train a matched *global control*. It keeps the backbone, recipe, data, schedule, seeds and checkpoint-selection rule of the WBPRIOR arm and replaces only the part branch with the encoder's own global read-out: the CLS/global token for the ViTs, the pooled global representation for SigLIP-2, and final-map global average pooling for the ConvNeXt variants. Both arms are five-seed runs on exactly N=15,000 FS-COCO *Unseen* pairs (Chowdhury et al., 2022). Lifts are computed from decision counts before rounding, so a lift need not equal the difference of the two rounded R@1 columns.

The head improves every metric on every encoder (60/60 contrasts), with a macro lift of +4.35/ + 3.58/ + 2.69/ + 3.72 pp on R@1/R@5/R@10/mAP. The largest R@1 lift is on DINOv3 ViT-L/16 (+6.40 pp) and the smallest on SigLIP-2 Base (+2.39 pp). Absolute accuracy still varies widely across encoders; the two DINOv2 models give the two highest absolute R@1 scores, which is why DINOv2 is the deployed backbone.

A3.4 THREE FURTHER PLUG-IN HOSTS

Table 2 adds the WBPRIOR branch to seven published hosts. Three further hosts, LPN, TransGeo and GeoDTR, were run under a separate five-seed SUES-200 protocol, each over N=720,000 query-condition decisions, and are reported here as lifts. As in the main plug-in study, the host backbone, classifier and supervision are left unchanged.

All three hosts improve under corruption, by +6.99 to +7.51 pp.

Table 14: **Matched-control cross-backbone study on FS-COCO *Unseen*.** Control and WBPRIOR columns are R@1 (%); Δ columns are the within-backbone lift in pp on R@1/R@5/R@10/mAP. Every arm is five seeds over exactly N=15,000 Unseen pairs and shares recipe, data and checkpoint rule with its partner. The macro row averages the lifts over the 15 backbones. **Red**: best per column; **navy**: second.

Family	Variant	Control	WBPRIOR	Lift (Δ, pp)			
		R@1	R@1	R@1	R@5	R@10	mAP
DINOv1 (Caron et al., 2021)	ViT-S/16	23.89	29.11	+5.22	+4.29	+3.16	+3.91
	ViT-B/16	16.49	20.90	+4.41	+3.57	+2.61	+3.53
DINOv2 (Oquab et al., 2024)	**ViT-S/14**	**49.51**	**55.55**	+6.05	+4.94	+3.83	+5.21
	ViT-B/14	**54.44**	**60.77**	**+6.33**	**+5.31**	**+4.09**	**+5.59**
DINOv3	ViT-S/16	23.48	28.46	+4.98	+4.08	+3.11	+4.11
	ViT-B/16	27.59	32.91	+5.33	+4.48	+3.37	+4.43
	ViT-L/16	37.43	43.83	**+6.40**	**+5.17**	**+3.92**	**+5.33**
	ConvNeXt-S	19.11	22.83	+3.71	+3.09	+2.23	+3.22
	ConvNeXt-B	21.43	26.00	+4.57	+3.69	+2.78	+3.81
	ConvNeXt-L	22.07	26.79	+4.72	+3.92	+2.89	+3.94
CLIP	ViT-B/32	34.21	36.70	+2.49	+2.15	+1.59	+2.43
	ViT-L/14	34.06	36.85	+2.79	+2.40	+1.71	+2.71
SigLIP-2	Base	8.29	10.67	+2.39	+1.87	+1.50	+2.24
	Large	10.51	13.24	+2.73	+2.27	+1.59	+2.51
	SO400M	19.48	22.59	+3.11	+2.51	+1.91	+2.83
Macro (15 equal-N backbones)		-	-	**+4.35**	**+3.58**	**+2.69**	**+3.72**

Table 15: **Three further plug-in hosts** (five seeds, SUES-200 D→S, each N=720,000 query-condition decisions). ΔCorrupt(9) R@1 is the change in mean R@1 over the nine severity-5 corruptions when the WBPRIOR branch is attached to the published host. These runs use a separate protocol from Tab. 2; absolute R@1 is not reported for them, and their lifts are not compared with that table.

Host	ΔCorrupt(9) R@1 (pp)
LPN (Wang et al., 2022)	+6.99
TransGeo (Zhu et al., 2022)	+7.23
GeoDTR (Zhang et al., 2023)	**+7.51**

A4 CORRUPTION-ROBUSTNESS DETAILS

A4.1 STANDALONE ROBUSTNESS ON THREE DRONE BENCHMARKS

Main-text Table 2 covers SUES-200 only and pairs each host with its WBPRIOR plug-in variant. Table 16 gives the standalone comparison (Study B): four published baselines and WBPRIOR on SUES-200, DenseUAV and University-1652, each method in its own best available configuration and with no WBPRIOR branch attached to any baseline. WBPRIOR has the highest Corrupt(9) R@1 on DenseUAV (72.12 against 61.80 for Sample4Geo, +10.32 pp) and on University-1652 (49.02 against 36.91 for MuSeNet, +12.11 pp), and trails Sample4Geo on SUES-200 (60.13 against 61.70, −1.57 pp).

A4.2 PLUG-IN AND STANDALONE PROTOCOLS

The corruption results come from two studies that answer different questions. Both use the nine severity-5 conditions and the metrics of Appendix A4.3, but they obtain the comparison methods differently, so their absolute values are not interchangeable.

Table 16: **Standalone robustness on three drone benchmarks** (Study B, D→S, %). Four published baselines and the deployed WBPRIOR model; no WBPRIOR branch is attached to any baseline. Same nine severity-5 corruptions as Table 2. Corrupt(9) is the mean over all nine conditions, so it equals the Sensor, Optical and Geom group means weighted by group size (four, three and two conditions) up to rounding. **Red**: best; **navy**: second.

Dataset	Method	Normal	Corrupt(9)	mCE-Geo↓	Rel.Rob	Sensor	Optical	Geom
SUES-200	IBN-Net50-a (Pan et al., 2018)	63.81	21.02	1.000	32.94	13.13	12.35	49.80
	MuSeNet (Wang et al., 2024)	44.82	38.02	0.823	**84.84**	38.73	34.04	42.58
	CCR (Du et al., 2024)	79.72	40.75	0.727	51.12	29.56	34.91	71.88
	Sample4Geo (Deuser et al., 2023)	**92.87**	**61.70**	**0.457**	66.44	**57.87**	**51.40**	**84.83**
	WBPRIOR (Ours)	**97.54**$_{\uparrow 4.67}$	**60.13**	**0.460**	61.65	40.12	**67.04**	**89.80**$_{\uparrow 4.97}$
DenseUAV	IBN-Net50-a (Pan et al., 2018)	12.96	10.72	1.222	**82.71**	9.70	15.42	5.71
	MuSeNet (Wang et al., 2024)	13.30	13.49	1.144	**101.43**	12.54	16.80	10.42
	CCR (Du et al., 2024)	48.61	36.07	0.841	74.22	38.52	39.51	26.02
	Sample4Geo (Deuser et al., 2023)	**75.20**	**61.80**	**0.516**	82.18	**67.60**	**69.38**	**38.85**
	WBPRIOR (Ours)	**85.11**$_{\uparrow 9.91}$	**72.12**	**0.390**$_{\downarrow 0.126}$	84.73	**79.83**	**80.82**	**43.65**$_{\uparrow 4.80}$
University-1652	IBN-Net50-a (Pan et al., 2018)	68.33	13.61	1.079	19.92	3.62	0.71	52.95
	MuSeNet (Wang et al., 2024)	52.99	36.91	0.787	**69.66**	**37.24**	28.60	48.72
	CCR (Du et al., 2024)	73.10	20.38	0.909	27.88	11.28	5.65	60.66
	Sample4Geo (Deuser et al., 2023)	**88.05**	36.88	**0.702**	41.88	29.29	24.41	**70.78**
	WBPRIOR (Ours)	**92.09**$_{\uparrow 4.04}$	**49.02**	**0.564**$_{\downarrow 0.138}$	**53.23**	**35.74**	**48.04**	**77.05**$_{\uparrow 6.27}$

Each baseline uses its own standalone configuration on each dataset. MuSeNet keeps its published training recipe, which adds corruptions during training because its style classifier requires corruption labels; test-time corruption is therefore partly in-domain for it, and on DenseUAV its Corrupt(9) R@1 (13.49) slightly exceeds its clean R@1 (13.30), which puts its Rel.Rob above 100. The clean score of IBN-Net50-a on DenseUAV (12.96) is in line with the global-pooling ResNet-50 baseline Instance Loss (13.00, Tab. 13). Metric definitions are in Appendix A4.3.

The plug-in study (Study C, Table 2) re-trains every pair ourselves. Within a pair, the host alone and the host with the WBPRIOR branch share one recipe: the shared SUES-200 recipe of WBPRIOR for every host except MuSeNet, which keeps its published recipe that adds corruptions during training, because its style classifier requires corruption labels. The two rows of a pair therefore differ only by the WBPRIOR branch. Most hosts are not run under the recipes their authors tuned, so they sit below their standalone accuracy, and the table is read within a pair, not against the literature.

The standalone comparison (Study B, Table 16) evaluates each method from its own best available configuration on all three benchmarks, with no WBPRIOR branch attached to any baseline, and places the methods on a common corruption scale across datasets.

Table 17 gives the clean accuracy that the two studies yield for the same method. The gap is the expected effect of re-training a published method under a recipe different from its original one.

Table 17: Clean (*Normal*) SUES-200 D→S R@1 of the same method in the plug-in and standalone studies. Standalone values are from Table 16. Re-training a host under the shared plug-in recipe costs it three to seven points of clean accuracy, which is why the two studies are reported separately.

Method	Plug-in	Standalone	Difference
MuSeNet (Wang et al., 2024)	41.57	44.82	−3.25
CCR (Du et al., 2024)	76.46	79.72	−3.26
Sample4Geo (Deuser et al., 2023)	85.62	92.87	−7.25
WBPRIOR (Ours)	97.54	97.54	0.00

WBPRIOR scores the same in both studies because the shared plug-in recipe is its own recipe, so it is never re-trained under a different one. The plug-in study is therefore read through the difference within each pair, not through absolute scores.

A4.3 Robustness Metric Definitions

Tables 2 and 16 report seven robustness columns between them; Table 2 omits the *Normal* column. Let R_0 denote the clean (*Normal*) R@1, and let R_c denote the R@1 on the c-th of the nine severity-5 corruption conditions $\mathcal{C} = \{c_1, \ldots, c_9\}$ defined in Sec. 4.2. The conditions are partitioned into three groups: $\mathcal{S}$={gaussian, shot, speckle, iso noise} (Sensor, $|\mathcal{S}|$=4), $\mathcal{O}$={defocus blur, motion blur, pixelate} (Optical, $|\mathcal{O}|$=3), $\mathcal{G}$={perspective, rotation} (Geometric, $|\mathcal{G}|$=2). All quantities are computed on D→S retrieval. The seven columns are:

$$\textbf{Normal} = R_0, \qquad \textbf{Corrupt(9)} = \tfrac{1}{9}\sum_{c\in\mathcal{C}} R_c, \tag{20}$$

$$\textbf{Rel.Rob} = \tfrac{100}{9}\sum_{c\in\mathcal{C}} R_c / \max(R_0, 1), \tag{21}$$

$$\textbf{Sensor} = \tfrac{1}{|\mathcal{S}|}\sum_{c\in\mathcal{S}} R_c, \quad \textbf{Optical} = \tfrac{1}{|\mathcal{O}|}\sum_{c\in\mathcal{O}} R_c, \quad \textbf{Geom} = \tfrac{1}{|\mathcal{G}|}\sum_{c\in\mathcal{G}} R_c, \tag{22}$$

$$\textbf{mCE-Geo} = \tfrac{1}{9}\sum_{c\in\mathcal{C}} \frac{100-R_c}{\max(100-R_c^{\text{IBN,SUES}},\ \epsilon_{\text{den}})}, \quad \epsilon_{\text{den}} = 5, \tag{23}$$

with $R_c^{\text{IBN,SUES}}$ the per-condition R@1 of the reproduced IBN-Net50-a row on SUES-200 (cross-dataset denominator, see below). *Higher* is better for Normal/Corrupt(9)/Rel.Rob/Sensor/Optical/Geom; *lower* is better for mCE-Geo. In the paired-format Table 2 the coloured subscript on each † row is not a separate column but the per-metric difference against the host row immediately above it (green ↑ = gain, red ↓ = loss), computed on the same definitions. All R@1 are percentages; mCE-Geo is unitless.

The standard mean corruption error divides a method's per-condition error by the error of a fixed reference model *on the same dataset*, which makes the reference exactly 1.000 on every dataset and prevents comparison across datasets. mCE-Geo instead uses one common reference for all three benchmarks, the per-condition errors of our reproduced IBN-Net50-a on SUES-200, with the floor ϵ_{den}=5 pp to avoid division blow-up when the reference nearly clears a condition. The reference then scores 1.000 on SUES-200, 1.222 on DenseUAV and 1.079 on University-1652, which reflects the relative difficulty of the datasets. Gallery size, resolution and the clean ceilings of the baselines still differ across benchmarks, so mCE-Geo is a secondary index, and raw Corrupt(9) R@1 (Table 16) remains the primary measure.

A4.4 Four Dropped Corruptions

The corruption protocol degrades the drone queries at test time and keeps the satellite gallery clean. In a pilot screening, fixed before WBPrior was evaluated, four further conditions (JPEG compression, scale jitter, tile shift and additive haze) were tested on IBN-Net50-a (Pan et al., 2018), chosen because its recipe of random resized crops, horizontal flips and colour jitter is the most generic among the baselines. JPEG, scale and tile shift raised the reference score and haze left it almost unchanged (Table 18), so none of them was kept in the primary set.

Table 18: Pilot screening on IBN-Net50-a (SUES-200, D→S, one severity-5 evaluation per condition). Clean R@1 is 63.81; Δ is the change from it. The four conditions are kept only as appendix diagnostics.

Condition	Corrupt R@1	Δ	Reason for exclusion
JPEG (q=15)	72.49	+8.68	Acts as a low-pass filter on the drone query.
Scale (±25%)	66.52	+2.71	Subsumed by RandomResizedCrop.
Tile shift (±10%)	65.10	+1.29	Subsumed by random crop.
Additive haze (low-freq.)	63.93	+0.12	No measurable degradation under the recipe.

Severe JPEG compression ($q = 15$) quantises 8×8 DCT blocks coarsely and so acts as a strong low-pass filter. On the drone queries it plausibly suppresses high-frequency sensor noise and grain, which makes them easier to match to the clean satellite tiles (+8.68 pp), so it behaves as a regulariser

rather than a degradation. All retained conditions except rotation degrade IBN-Net50-a by at least 20 pp at severity 5. Rotation by $\pm 30^\circ$ lowers R@1 by about 3 pp and is kept as a mild geometric stress, so that the geometric group has two conditions.

A5 Method Diagnostics and Sensitivity

A5.1 Component Ablation

This subsection tests which components of WBPrior matter. Table 19 gives the component ablation (Study D) on SUES-200 (D→S): it removes or replaces each architectural component, cross-view training term and pathway in turn, under the unified corruption protocol.

Table 19: **Component ablation (Study D) on SUES-200** (D→S). Variants share recipe, seed and schedule with the full model. Group **A** swaps the *part aggregation*, **B** the cross-view *supervision*, **C1** the pathways and **C2** the auxiliary objective. Columns: Clean and Corrupt(9) R@1 (%), one run per row. Values are compared only within this table, not with the five-seed study of Tab. 1 or the plug-in study of Tab. 2.

Configuration	Clean	Corrupt
WBPrior (full: BW aggregation + Sinkhorn + functional map + UAPA)	**98.32**	**60.36**
A. Part aggregation *(does the W_2 barycenter matter?)*		
replace BW barycenter → slot mean	94.27	50.91
replace BW barycenter → sliced-Wasserstein ($L = 64$ projections)	95.90	58.94
replace BW barycenter → Euclidean (Frobenius)	97.10	59.71
B. Cross-view supervision *(do Sinkhorn / functional map matter?)*		
replace debiased Sinkhorn → patch-NCE	94.26	54.77
remove functional-map term	94.14	53.45
remove the entire part-quality loss group	94.06	52.19
C1. Pathway controls		
CLS-only pathway (part branch off)	96.03	58.96
part-only pathway (CLS branch off)	79.59	31.73
C2. Auxiliary objective control		
without UAPA (alignment temperature fixed)	93.85	53.95

Reading the ablation. Replacing the BW barycenter with the slot mean causes the largest drop in the table (-4.05 pp clean and -9.45 pp under corruption), and the sliced-Wasserstein and Euclidean rules also trail the barycenter; the controlled version of this comparison is the matched five-seed study of Tab. 1. Replacing the debiased Sinkhorn divergence, removing the functional map or removing the whole part-quality loss group lowers both columns, and so does removing UAPA, which costs 6.41 pp under corruption. Each of these variants with weaker part supervision falls below the CLS-only pathway of this ablation on both columns, as the slot mean does. A plausible reading is the one given for the slot mean in Sec. 4.1: a part branch that is poorly aligned across views hands the fusion gate a noisy part vector, which pulls the joint embedding down. The part-only pathway is far weaker than either full pathway (79.59 clean), so the part branch is useful as a complement to the CLS token, not as a replacement for it. With the full supervision and the BW barycenter, the part branch lifts clean R@1 to 98.32, 2.29 pp above the CLS-only pathway of this ablation.

A5.2 Alternative Barycenter and Pooling Rules

We replace the BW aggregation of Eq. 11 with five alternative rules and re-train end to end under an otherwise identical recipe. The runs belong to the component-ablation setting (Study D), so the BW row equals the full model of Tab. 19. Table 20a reports SUES-200 D→S clean R@1 (%), averaged over the four altitudes, on the full 200-tile gallery at a single scale of 448^2 with no re-ranking.

What the comparison shows. Every alternative rule trails the BW barycenter, and the order is informative. The two rules that keep no variance, the slot mean and the k-means centroid, lose the most (4.05 and 2.48 pp clean R@1), which is the price of dropping the learned scale vector. The other three rules keep a variance but combine it differently. The sliced-Wasserstein barycenter approximates W_2 on the full distribution with L=64 random projections instead of the Gaussian summary, at about three times the per-step cost of Algorithm 1. The Gromov-Wasserstein barycenter compares distance matrices within each distribution instead of transporting one distribution onto another. The Euclidean average of mean and variance comes closest (-1.11 pp) and differs from the BW rule only in its

Table 20: **Where the effect comes from.** *(a)* SUES-200 clean R@1 (%) when only the aggregation rule changes (Study D, App. A5.2); Δ is the change relative to the BW barycenter. *(b)* Corrupt(9) R@1 (%) of the deployed model when the contribution of its learned scale is set to zero at test time on the named patch subset, from a separate paired five-seed study (App. Tab. 24). The random subset has the same size as the high-structure subset. The two panels come from different experiments, so their numbers should not be subtracted from each other.

(a) Aggregation rule

BW barycenter (ours)	**98.32**	-
Euclidean average of mean and variance	97.21	−1.11
Gromov-Wasserstein barycenter	96.42	−1.90
Sliced-Wasserstein barycenter	95.90	−2.42
k-means centroid (no variance)	95.84	−2.48
Slot mean (no variance)	94.27	−4.05

(b) Variance contribution zeroed on

Nothing (deployed)	**60.29**	-
High-structure patches	58.43	−1.85
Low-structure patches	59.96	−0.33
Random patches (count-matched)	59.71	−0.57

geometry. The BW barycenter averages standard deviations, $\mathbf{s}_k = \sum_i w_{i,k}\sqrt{\mathbf{v}_i}$, at linear $\mathcal{O}(d_b)$ cost, whereas the Euclidean rule averages variances; by Jensen's inequality, $\sum_i w_{i,k}\mathbf{v}_i \geq \mathbf{s}_k^2$, so the Euclidean rule makes a part look wider whenever its patches disagree. In the component ablation the Euclidean rule also trails the barycenter under corruption (59.71 against 60.36, Tab. 19). The Euclidean entries of the two tables (97.10 and 97.21 clean) come from two separate training runs of the same rule. Most of the gain therefore comes from keeping a learned scale vector at all, and the rest from the BW geometry.

A5.3 Additional Matched Study: Covariance Representation

This subsection asks whether the diagonal restriction, or merely extra model capacity, explains the gain. Both studies use the matched five-seed protocol of Tab. 1 (Study A; seeds 42, 123, 2024, 7 and 1337), with identical queries, schedules and checkpoint-selection rules across arms.

(i) Does the deployed diagonal representation cost accuracy? The deployed head uses a learned diagonal Gaussian with Gaussian head width d_b=32 (Eq. 10). To test whether the diagonal restriction, and not the parameter budget, limits the head, we re-train two width-matched variants at d_b=16. One is diagonal; the other uses full covariance and runs the general SPD operator of Eq. 8, with its two matrix square roots per part. Only the two width-16 rows of Tab. 21 are matched to each other, so only they isolate the covariance *representation*; the deployed row also differs in width and is shown for reference.

Table 21: **Covariance representation** (SUES-200 drone→satellite, Study A, five seeds, %). Corrupt(9) averages the nine severity-5 corruptions. Relative cost is measured against the deployed diagonal row; only the two width-16 rows are matched to each other.

Representation	Width	Clean R@1	Corrupt(9) R@1	Rel. cost
Diagonal, deployed	32	**97.53**	**60.29**	**1.00**×
Diagonal, matched	16	97.36	59.86	**0.82**×
Full covariance, matched	16	**97.80**	**60.70**	2.30×

At matched width, full covariance beats the width-matched diagonal variant by 0.44 pp Clean and 0.84 pp Corrupt(9) R@1, but it costs 2.30× the deployed diagonal head, or 2.80× the width-matched diagonal variant it should be compared against. The deployed diagonal head is therefore a choice on the accuracy-compute trade-off, not the best available covariance model.

(ii) Learned scale or extra capacity? The scale vector $\mathbf{s}_k \in \mathbb{R}^{d_b}$ could help through its role as a positive second-order quantity, or merely because it adds d_b extra feature channels (widening the Gaussian head from d_b to $2d_b$). To test the second explanation, we evaluate an unconstrained control of the same capacity. Instead of predicting positive variances $\mathbf{v}_i$ and forming the standard-deviation barycenter of Eq. 3, the part branch gets an unconstrained linear projection head of the same dimension $d_b = 32$. It outputs an arbitrary auxiliary feature $\mathbf{h}_i \in \mathbb{R}^{d_b}$, which is aggregated by standard linear pooling, $\mathbf{h}_k = \sum_i A_{ik}\mathbf{h}_i$, and concatenated as $[\boldsymbol{\mu}_k; \mathbf{h}_k]$. This control reaches 96.85% Clean R@1 and 55.42% Corrupt(9) R@1, 0.68 pp and 4.87 pp below the deployed diagonal BW

model (97.53% and 60.29%; Tab. 21). Extra unconstrained capacity therefore does not substitute for the positive scale vector, and the gap is largest under corruption.

A5.4 Is One Gaussian Enough? BIC Diagnostic and a Matched Mixture Control

The head summarises the patches of each part with a single diagonal Gaussian. This Gaussian is not a model of the nuisance process, such as viewpoint, time of day, sensor or weather; it is a learned summary of the patches already assigned to a part. The spectral assignment (Eq. 14) defines the part, and the BW barycenter adds a learned scale vector to its mean vector. The useful questions are therefore how often a *single* component is an inadequate summary of a part, and what a richer summary would buy.

(i) How often is one component inadequate? We fit mixture models to the assigned patch features of each part prototype and apply the Bayesian information criterion (BIC) on SUES-200 drone-to-satellite, across five checkpoints, with 10,000 prototypes sampled per group. Table 22 reports the share of prototypes for which the criterion prefers a mixture over a single component.

Table 22: **BIC model-selection diagnostic** on SUES-200 (D→S; five checkpoints; 10,000 prototypes sampled per group). Share of prototypes for which BIC prefers a mixture over a single Gaussian summary.

Group	Prototypes where BIC prefers a mixture	Share
Clean	1,217/10,000	12.17%
Sensor	2,189/10,000	21.89%
Optical	1,806/10,000	18.06%
Geometric	1,981/10,000	19.81%

(ii) What would a richer summary buy? Table 23 answers this with a paired five-seed matched study (Study A) on SUES-200 Corrupt(9): the part prototype is replaced by a mixture, with everything else (backbone, assignment, alignment losses, schedule, seeds) held fixed.

Table 23: **Matched mixture control** (SUES-200 Corrupt(9), Study A, five seeds, N=720,000 paired decisions). Only the prototype measure changes. The two mixture rows use progressively larger mixtures.

Prototype measure	Corrupt(9) R@1	Δ vs. one Gaussian	Cost
One Gaussian (deployed)	434,059/720,000 (**60.29**)	reference	1.00×
Larger mixture	437,652/720,000 (60.79)	+3,593/720,000 (+0.50 pp)	1.63×
Still larger mixture	436,972/720,000 (60.69)	+2,913/720,000 (+0.40 pp)	2.18×

Reading the two tables together. Multimodal parts are real, and corruption makes them more common: BIC prefers a mixture for 12.17% of prototypes on clean inputs and for 18.06% to 21.89% under corruption (Tab. 22). The retrieval consequence is small. Paying 1.63× the cost for a mixture recovers 0.50 pp of Corrupt(9) R@1, and paying 2.18× recovers only 0.40 pp. We therefore keep the single Gaussian as the deployed summary and treat multimodal parts as the boundary case of the construction; the 0.50 pp gain estimates what a richer mixture could still add under this protocol.

A5.5 Does the Learned Variance Suppress Sharp Structure? A Matched Intervention

One might worry that representing each patch as a Gaussian blurs sharp structure, such as the building corner that motivated the construction. It does not do so directly, because the learned scale is a statistic *in feature space*, not a spatial filter. Nothing in the head convolves, blurs or pools the patch map: the affinity graph is built on the backbone's own patch tokens, and the variance enters only through the read-out of Eq. 4 and the BW terms (Eqs. 3, 7). Whether the variance contributes more or less at structural locations is an empirical question, so we test it with a matched intervention.

Intervention. On each image we compute a Harris-style corner and edge response (Harris & Stephens, 1988), a score that is high where image intensity changes strongly in more than one direction. We split

patches into structure quartiles by this response and, at inference, zero *only* the variance contribution to Eq. 4 for a chosen quartile. Everything else, including the mean path and the assignment, is untouched. A random-patch control is matched to the high-structure condition by per-image patch count, so the two conditions differ in *which* patches lose their variance contribution and not in how many.

Table 24: **Structure-quartile variance intervention** (SUES-200 Corrupt(9), Study A, five seeds, N=720,000 paired decisions). Only the variance contribution is zeroed, on the patch subset named in each row; the random subset is matched by count, image by image, to the high-structure subset.

Intervention	Corrupt(9) R@1	Loss vs. full
Full (deployed)	**60.29**	reference
Variance contribution zeroed on high-structure patches	58.43	13,328/720,000 (1.85 pp)
Variance contribution zeroed on low-structure patches	59.96	0.33 pp
Variance contribution zeroed on matched random patches	59.71	4,118/720,000 (0.57 pp)
High-structure vs. matched random (difference)		9,210/720,000 (1.28 pp)
paired 95% CI		[+0.94, +1.62] pp

What the intervention shows. If the variance contribution acted like a smoothing operation that destroyed structural detail, removing it at high-structure locations would be harmless or helpful. The opposite holds. Removing it costs 1.85 pp at high-structure patches against 0.33 pp at low-structure ones, and 1.28 pp more than removing it from a count-matched random subset, with a paired 95% confidence interval of $[+0.94, +1.62]$ pp that excludes zero. The learned scale therefore contributes most where sharp structure is. This is consistent with Corollary 3: two parts with equal mean vectors and different learned scale vectors are separable under BW but not under a slot mean.

A5.6 Functional-Map Dependence: Train-Time Correspondence

Functional maps carry a well-known caveat: they behave well given a reasonable initial alignment and degrade when the two shapes are far apart. It matters, then, where in our pipeline the functional map sits, and it sits in training only. At inference there is no functional-map solver, no paired view and no initialisation: a query image is embedded by a single forward pass through the backbone and the head, its own Laplacian eigenbasis orders its own prototypes, and $\mathbf{C}$ is never formed. Eq. 16 is a supervision term that shapes the eigenbasis during training so that the ordering of parts in each image agrees across views; the retrieval path never evaluates it. The dependence therefore lies in the quality of the training correspondences, which the following matched study varies.

Degraded training correspondences. We displace the training correspondences used by the alignment terms by a fixed pixel budget and re-train, changing nothing else and in particular nothing at inference.

Table 25: **Training-correspondence displacement** (SUES-200 Corrupt(9), Study A, five seeds, N=720,000 paired decisions). Displacement perturbs only the training correspondences used for alignment; the inference path is unchanged in every row.

Training intervention	Corrupt(9) R@1	Loss vs. 0 px
0 px (deployed)	**60.29**	reference
8 px	59.98	0.31 pp
16 px	59.30	7,065/720,000 (0.98 pp)
32 px	57.39	20,843/720,000 (2.89 pp)
Alignment removed	55.81	32,237/720,000 (4.48 pp)

When the method should apply. Training-side correspondence error degrades the model gradually (0.31 pp at 8 px, 2.89 pp at 32 px), while removing cross-view alignment altogether costs 4.48 pp, so the supervision has a large effect but is not brittle. Because the term needs coarsely registered training pairs, we state coarse cross-view registration as a condition under which the method is expected to work. It holds for the nadir-aligned drone and satellite tiles of SUES-200, University-1652 and DenseUAV (§A2.2); we make no claim for severely off-nadir or ground-to-aerial image pairs.

A5.7 Barycenter Diagnostics: Variance Spectrum and Cross-View Residual

This subsection looks inside the barycenter instead of at retrieval accuracy. The deployed rule is the closed-form diagonal barycenter of Eq. 3. The two diagnostics below read the learned prototype scale vectors and the cross-view residuals directly; neither has an analogue in a slot-mean head, whose variance is identically zero.

Variance spectrum. Diagnostic D1 records, at the end of every fifth epoch and for each active prototype on a fixed validation minibatch, the entries of its variance $\mathbf{v}_k$, which are the eigenvalues of the diagonal covariance $\mathrm{diag}(\mathbf{v}_k)$. Two patterns are stable across seeds. The smallest entry stays bounded away from zero ($\lambda_{\min} \geq 10^{-3}$ for all k once the part-selection gate has stabilised, around epoch 20), so the positive Softplus head keeps every prototype non-degenerate. The largest entry grows monotonically over training, and by epoch 120 it is about 4.7 times the smallest. The prototypes therefore become anisotropic in feature space instead of collapsing to an isotropic Gaussian, for which this ratio would be 1; we never observe that case.

Cross-view residual. Diagnostic D2 is the cross-view BW residual $W_2^2(\nu_k^{(d)}, \nu_k^{(s)})$, which measures how closely the drone-view and satellite-view Gaussians of the same prototype match. Averaged over the 10 active prototypes on the SUES-200 validation split, it falls from about 0.34 at epoch 5 to about 0.04 by epoch 120. Under a slot-mean prototype (mean only, $\Sigma = 0$) the same residual reduces to $\|\mu_k^{(d)} - \mu_k^{(s)}\|^2$ and ignores the learned scale vector, which is the additional quantity the barycenter is trained to align.

A5.8 Hyperparameter Sensitivity Sweep: ε and $K_{\min}$

Two hyperparameters are specific to the part head: the Sinkhorn regularisation ε and the training-only floor $K_{\min}$ of the part-selection gate. The remaining hyperparameters (learning rate, weight decay, prototype dimension) are shared with the slot-mean baseline and listed in Tab. 7. We sweep each of the two separately on SUES-200 (D→S, 448^2, full 200-tile gallery, single scale, no re-ranking).

Table 26: Sensitivity sweep on SUES-200 D→S R@1 (%). Default configuration in **bold**. Each row varies one hyperparameter with the other held at its default. The values come from an earlier training run than the deployed model (98.32 in Tab. 3), so only the changes across settings within this table are meaningful.

	Setting				
Hyperparameter			default		
ε (Sinkhorn)	0.01	0.05	**0.10**	0.50	1.00
R@1	97.83	98.31	**98.74**	97.62	96.04
$K_{\min}$ (gate floor)	1	2	**4**	6	8
R@1	96.45	98.02	**98.74**	98.51	98.27

Reading the sweep. R@1 peaks at the default ε=0.10 and falls on both sides: by 0.43 and 0.91 pp at 0.05 and 0.01, and by 1.12 and 2.70 pp at 0.50 and 1.00. The steeper fall at large ε matches the entropic bias of the divergence, which moves it away from W_2^2 as ε grows (§A1.6). For the gate floor, $K_{\min}$=1 costs the most (2.29 pp), as the gate can then concentrate on too few prototypes early in training. Higher floors cost less (0.23 and 0.47 pp at 6 and 8), consistent with keeping weakly specialised prototypes active. The default $K_{\min}$=4 gives the best R@1 in the sweep.

A5.9 Convergence and Seed Stability

WBPrior exceeds 90% R@1 by epoch 30 on SUES-200 and stabilises above 97% by epoch 85. Across the five random seeds (42, 123, 2024, 7, 1337) under the fixed 120-epoch schedule, the standard deviation of Clean R@1 is below 0.15 pp on both SUES-200 and University-1652. For matched paired tests such as Tab. 1, confidence intervals use 10,000 query-level paired bootstrap replicates clustered across the five seeds, so that the evaluations of one query across corruption conditions and seeds are not treated as independent evidence.

A5.10 Auxiliary Bundle: Architecture Reuse Versus Task Supervision

The transfer results of Tab. 4 reuse the architecture of WBPrior, not its weights or its training objective: FSCOCO and CARGO each keep a task-specific auxiliary objective. This subsection measures how much of each result those auxiliaries account for. The intervention removes only the auxiliary bundle and keeps the architecture, the principal retrieval objective and the evaluation protocol.

Table 27: **Auxiliary-bundle ablation on the two transfer tasks** (%). Architecture, principal objective and evaluation protocol are retained; only the task-specific auxiliary bundle is removed. Drop is R@1/R@5/R@10/mAP in pp.

Setting	N	Full				Auxiliaries off				Drop (pp)
		R@1	R@5	R@10	mAP	R@1	R@5	R@10	mAP	
FSCOCO *Unseen*‡	15,000	**60.76**	**80.95**	**87.74**	**69.99**	56.71	76.26	84.21	65.07	$-4.05/-4.69/-3.53/-4.92$
CARGO WBPrior-S A→G	800	**68.38**	**88.13**	**94.63**	**61.02**	40.38	67.38	79.25	37.38	$-28.00/-20.75/-15.38/-23.64$

‡A separate five-seed run from the cross-backbone study of Tab. 14, whose DINOv2 ViT-B/14 row reads 60.77. Exact decision counts: FSCOCO $9{,}114/15{,}000 \rightarrow 8{,}506/15{,}000$, a loss of $608/15{,}000$; CARGO $547/800 \rightarrow 323/800$, a loss of $224/800$. The CARGO row is a separate $N{=}800$ run and uses a different protocol from the result in Tab. 4 ($109/160 = 68.13\%$).

Reading. Removing the auxiliaries costs 4.05 pp R@1 on FSCOCO *Unseen* and 28.00 pp on CARGO A→G. Aerial-ground re-identification therefore needs its identity-level supervision, which a part representation alone does not supply. What carries over across the three kinds of appearance change is the architecture of WBPrior and its BW aggregation rule; the training objective is chosen per task, and these numbers show how much that choice is worth.

A5.11 Extended Related Work

This subsection compares WBPrior in more detail with the closest work on probabilistic slots, probabilistic embeddings, second-order pooling and foundation-model retrieval.

Probabilistic slot attention. Kori et al. (2024) introduce Probabilistic Slot Attention, which models the slots of an image as the components of a per-image Gaussian mixture. The mixture is updated by expectation-maximisation (EM), in place of the GRU update of Slot Attention (Locatello et al., 2020), and slot identifiability is proved up to an equivalence relation through an aggregate-posterior mixture prior. Both methods give visual components Gaussian parameters, but they differ in goal, mechanism and use of the variance. Probabilistic Slot Attention targets unsupervised object-centric representation learning with a reconstruction objective, evaluated on toy 2D data, SpriteWorld, CLEVR, ObjectsRoom and Pascal VOC, whereas WBPrior targets discriminative retrieval across large viewpoint and appearance changes. Probabilistic Slot Attention fits slot means, variances and mixing weights by EM over the input features; WBPrior takes its assignment from a spectral partition of the patch affinity graph (Theorem 1) and combines learned patch Gaussians with the diagonal 2-Wasserstein barycenter, $\mathbf{s}_k = \sum_i w_{i,k}\sqrt{\mathbf{v}_i}$ (Eq. 9). Finally, in Probabilistic Slot Attention the variances parameterise the per-image mixture that underlies slot identifiability, while in WBPrior the learned scale vector is carried into the retrieval embedding, where it separates candidates that share part means (Theorem 2, Corollary 3).

Probabilistic cross-modal embeddings and Wasserstein losses. PCME (Chun et al., 2021) and PCME++ (Chun, 2024) represent whole images and captions as Gaussian distributions in $\mathbb{R}^D$ to address one-to-many cross-modal ambiguity; PCME estimates match probabilities by Monte Carlo sampling, and PCME++ replaces this with a closed-form probabilistic distance. Frogner et al. (2015) propose learning with an entropic Wasserstein loss over fixed ground metrics on label spaces, while Muzellec & Cuturi (2018) embed discrete entities into Wasserstein spaces of elliptical distributions. WBPrior differs in two respects. PCME maps entire images to global distributions, whereas WBPrior discovers local parts inside each image and computes a closed-form Wasserstein barycenter for each part. PCME and PCME++ also score a pair of distributions, which does not fit plain inner-product search, while WBPrior maps the part mean vectors and scale vectors $(\boldsymbol{\mu}_k, \mathbf{s}_k)$ into a single deterministic ℓ_2-normalised vector for cosine retrieval.

Second-order pooling and covariance networks. Bilinear pooling (Lin et al., 2015) and covariance pooling such as iSQRT-COV (Li et al., 2018) form a single global second-order descriptor per image; the matrix power normalisation of iSQRT-COV uses Newton-Schulz iterations whose cost is cubic in the channel dimension. GeM (Radenović et al., 2019) is a global first-order power-mean pooling. Both families summarise the whole image and do not keep part-level structure. WBPRIOR instead partitions patches into parts via spectral eigenvectors and computes a diagonal barycenter per part at linear $\mathcal{O}(d_b)$ cost, with no matrix square root.

Foundation models for place recognition and cross-view geo-localisation. Recent VPR and geo-localisation approaches build on strong pretrained backbones. AnyLoc (Keetha et al., 2024) aggregates DINOv2 patch features with VLAD or GeM for universal place recognition; BoQ (Alibey et al., 2024) aggregates backbone features with learnable queries through cross-attention; and Sample4Geo (Deuser et al., 2023) trains a ConvNeXt encoder with hard-negative sampling. These methods aggregate first-order features into a global descriptor. WBPRIOR instead keeps a learned scale vector for each discovered part, and within a fixed part branch this variance gives the large corruption gain of Tab. 1a.

A6 LIMITATIONS AND BROADER IMPACT

Every result in this paper comes with a scope, and stating it clearly also shows where the idea can go next. We describe four such boundaries, then the broader impact of the method.

The first boundary is geographic. Our drone benchmarks split training and test data within the same regions, so they measure robustness to changes in sensor, viewpoint and modality rather than to a change of continent. The inference path does not depend on the training locations, since retrieval is cosine similarity between embeddings, which makes cross-region training and testing a direct and informative next experiment.

The second boundary concerns what transfers. Across cross-view geometry, aerial-ground re-identification and sketch retrieval we reuse the architecture and the BW aggregation rule, while each task keeps its own auxiliary objective; removing that objective on CARGO costs a large drop (Appendix A5.10, Tab. 27). The representation is therefore shared, and the supervision is task specific. A single checkpoint for all three settings is the natural next goal.

The third boundary is the shape of each part. A diagonal Gaussian summarises a part with one mode. In cluttered scenes a part can mix two kinds of texture, for example vegetation and shadow, and a single Gaussian then smooths over both. We measured how often this happens rather than assuming it away. A per-prototype BIC diagnostic prefers a mixture for about one prototype in eight on clean inputs and about one in five under corruption, and a matched mixture control recovers at most 0.50 pp of Corrupt(9) R@1 at $1.63\times$ the cost (Appendix A5.4, Tabs. 22 and 23). The single Gaussian captures most of the benefit, and the mixture result gives a concrete estimate of what a richer model could still add.

The fourth boundary is how parts are indexed. The head uses at most $K_{\max}{=}12$ parts, and the part-selection gate keeps about ten per image on average; a sweep of the training-time floor $K_{\min}$ degrades gracefully around the deployed setting (Appendix A5.8, Tab. 26). Part indices are not guaranteed to carry human-readable meanings, and the losses are built so that they do not need to: the patch-level contrastive loss pairs patches by their part index within each view pair, and the orthogonality term (Tab. 9) helps keep the part channels distinct during training. Rotation augmentation is used only on DenseUAV, so robustness to rotation there is learned from data rather than built into the architecture. Choosing $K_{\max}$ automatically for each dataset would remove the last manual setting of the head.

WBPRIOR is a retrieval method with dual use, and two of our settings differ in kind. Cross-view geo-localisation retrieves geographic tiles from drone or satellite imagery; it supports navigation, mapping and search where satellite positioning is unavailable, and it can also support persistent monitoring of places. CARGO (Zhang et al., 2024) evaluates aerial-ground person re-identification, which matches people by construction and can support surveillance directly. We therefore describe the two settings separately and state their risks plainly.

Our experiments establish retrieval performance under the stated benchmarks and protocols. They do not establish privacy protection, consent, legal compliance or the adequacy of any deployment

safeguard, and they should not be read as evidence that a particular deployment is safe, lawful or beneficial. Such deployments need their own assessment of these questions.

A6.1 Ethics Statement

This work uses six publicly released retrieval benchmarks (SUES-200, University-1652, DenseUAV, CARGO, FSCOCO and SketchyCOCO) under the terms set by their authors. We collected no new data, ran no human-subject study, and employed no crowdworkers. CARGO, FSCOCO and SketchyCOCO contain imagery of people and of everyday scenes; we use them only as retrieval benchmarks, report only aggregate retrieval metrics, and release no images, identities or derived per-person artefacts. No personally identifying information is used at any stage of training or evaluation.

The application we target, cross-view geo-localisation for navigation in GPS-denied settings such as search-and-rescue, disaster response and infrastructure inspection, is dual-use. The same capability can support unauthorised aerial surveillance, and the aerial-ground re-identification results in particular describe a person-matching capability. Our experiments establish retrieval performance on the benchmarks above and nothing more: they are not evidence of privacy protection, of consent, of legal compliance, or of social benefit. What our results do and do not license is discussed in Appendix A6. The submission is anonymised and we adhere to the ICLR Code of Ethics.

A6.2 Reproducibility Statement

The method is specified so that it can be reimplemented from the paper alone. The complete hyperparameter configuration of the deployed model is Table 7 (Appendix A2.1). The exact quantities the head computes, their numerics and their cost are in Appendix A1.4, and separate training and inference algorithms, together with a table giving the lifetime of every quantity, are in Appendix A1.5. The remaining part-head mechanics (affinity graph, spectral assignment, part-selection gate, part pooling, Sinkhorn cost, functional-map terms and CLS-conditioned FiLM) are written out in Appendix A2.2. The loss groups and their Kendall weighting are in Table 9 (Appendix A2.5), and UAPA's temperature rule is Eq. 19 (Appendix A2.4). Dataset splits and query/gallery counts for both retrieval directions are in Table 10 (Appendix A3.1). The corruption protocol, the exact formulas for every robustness column, the cross-dataset mCE-Geo denominator and the pilot screening that dropped four conditions are in Appendices A4.3 to A4.4. Sensitivity to the barycenter-specific hyperparameters is swept in Appendix A5.8, and seed stability in Appendix A5.9. Results reported as *additional matched* studies (Study A) were run on a separate paired five-seed protocol whose configuration is stated where those results appear; they are never mixed with the primary benchmark numbers. We will release the training and evaluation code together with the per-condition result files.

A6.3 Use of Large Language Models and Generative AI

Generative AI tools were used for manuscript editing, adversarial review of mathematical arguments, feedback on experimental framing and methodology, consistency checking, and literature discovery and verification. All AI-assisted suggestions were independently evaluated and checked by the authors against the source material, mathematical derivations, codebase, experimental outputs, and cited literature. Generative AI was not used to fabricate, alter, or select experimental data or results; every number reported comes from the authors' own training and evaluation runs. The research ideas, theoretical formulations, empirical design, and scientific claims are the authors' original work, and the authors take full responsibility for the entire contents of the paper.