BEYOND HOLISTIC ALIGNMENT: STRUCTURED SEMANTIC TRANSFER FOR ZERO-SHOT HASHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-shot image hashing aims to learn a compact binary code that generalizes to unseen classes without real training images of those classes, using semantic side information. Recent generative pipelines synthesize pseudo-images for unseen classes and align image features through contrastive objectives, yet the hash space itself is supervised mainly through holistic features and class centers without explicitly exploiting structured semantic relations. Two complementary forms of semantic structure therefore remain underexplored: attribute-related patterns encoded in visual representations and the relational geometry that connects seen and unseen classes. We propose a novel framework termed Structured Semantic Transfer (**SST**) for zero-shot hashing, which preserves semantic structures beyond holistic alignment. A lightweight branch captures attribute-aware representations from local patch features, and a pairwise structure distillation loss aligns the similarities among these representations with those of their classes, enabling the encoder to learn transferable semantic structures. Meanwhile, a geometry alignment loss preserves the attribute-induced relations among hash centers, guiding unseen class centers through structured semantic propagation. We further introduce a per-code-length component configuration that adapts the overall scale of the two loss weights at each operating point while keeping the framework unchanged. Experiments on standard zero-shot hashing benchmarks show that **SST** consistently improves retrieval over strong baselines, and ablations confirm that the two semantic transfer mechanisms are complementary. Our code is available at https://anonymous.4open.science/r/SST-B163.

1 INTRODUCTION

Hashing-based image retrieval compresses images into compact binary codes so that semantically similar images are close in Hamming space, enabling efficient search over web-scale collections Cheng et al. (2026). In open-world scenarios, however, new categories emerge continuously and collecting labeled samples for every novel class is impractical. Zero-shot hashing image retrieval addresses this setting Yang et al. (2016); Guo et al. (2017), where real training images are restricted to seen classes while test-time queries involve unseen classes described by side information such as class attribute vectors or textual descriptions.

Recent methods improve visual–semantic transfer through prompt-based alignment Dong et al. (2024) and attribute modeling Li et al. (2025). PZSH Zhang et al. (2026) generates pseudo-images with stable diffusion Rombach et al. (2022), encodes images with a BLIP encoder Li et al. (2022), aligns image features through a contrastive objective, and uses class centers to supervise hash codes. Our PZSH-based implementation uses learnable centers and a mean squared error objective on continuous codes. While effective, it transfers knowledge at a coarse granularity, since the contrastive alignment operates on holistic image features and the centers are learned purely from hash-code supervision without any explicit connection to the semantic side information. Two rich sources of transferable knowledge therefore remain untapped.

The first source lies in patch-level semantics. Fine-grained categories are distinguished by local appearance, such as beak shapes in CUB or texture layout in SUN, and class attributes describe many of these local cues. Holistic feature alignment ignores where and how discriminative patterns appear, so the learned codes may conflate classes that differ only in local parts. The second source

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.

lies in the geometry of the attribute space, where attribute vectors indicate which seen classes are similar to which unseen classes and can therefore guide the placement of class centers in Hamming space. Randomly initialized centers carry no such structure, which weakens the transfer from seen to unseen classes.

We propose `SST`, a method transferring class information into the hash space from these two complementary angles atop the PZSH pipeline. From the patch angle, a lightweight attribute branch maps pooled patch features to attribute embeddings, and a pairwise structure distillation loss aligns the in-batch cosine similarities of these embeddings with the attribute similarities of the corresponding classes. This structural supervision is invariant to positive rescaling of the embeddings and injects attribute-related discriminative semantics into the shared, trainable encoder. From the attribute angle, a geometry alignment loss matches the pairwise similarity structure of the learnable hash centers to the attribute similarity structure over the unseen–seen block. This jointly regularizes the seen- and unseen-class centers, connecting their geometry as both evolve under hash-code supervision.

The contributions of this paper can be summarized in three points. ❶ **Overlooked transfer angles.** We identify two underexploited sources of transferable knowledge for zero-shot hashing, namely patch-level attribute semantics and attribute-space geometry, and we realize both through dedicated losses. ❷ **Per-code-length component configuration.** We show that the two angles are effective to different degrees across hash lengths and datasets, and propose to select the component configuration at each operating point rather than fixing one setting globally, which keeps the framework uniform while adapting the strength of structural supervision at each code length. ❸ **Extensive experiments.** We validate `SST` on AWA2, CUB, and SUN across four hash lengths and demonstrate consistent improvements over the strong PZSH baseline, with ablations confirming that the two transfer angles play distinct and complementary roles.

2 Related Work

Zero-shot hashing. Early methods enable unseen-class transfer through semantic embeddings (TSK Yang et al. (2016)) or pairwise similarity (SitNet Guo et al. (2017)). Later work introduces orthogonal projections Zhang et al. (2019), while PIXEL Dong et al. (2024) aligns visual and textual semantics through prompts. PZSH Zhang et al. (2026) adds Stable Diffusion pseudo-images and proxy-based hash supervision, without explicitly imposing attribute relations on hash centers. Our attributes supervise visual representations and center geometry.

Center-based hashing. CSQ Yuan et al. (2020) constructs class centers from Hadamard matrices and Bernoulli sampling; other objectives address discrete optimization and semantic similarity Su et al. (2018); Cao et al. (2017). SHC Chen et al. (2025) estimates class similarities from visual data and optimizes semantic centers before hash-network training. Our geometry term constrains continuous unseen–seen center similarities using attributes alongside image-code supervision. This differs in relation source, constrained class pairs, and joint optimization.

Patch- and attribute-level supervision. Prior hashing methods use semantic similarity Shi et al. (2023) and part alignment Jiang et al. (2025). COMAE Li et al. (2025) combines attribute prototypes with point/pair/class-wise constraints; DA2H Wang et al. (2026) localizes regions via an attribute-conditioned decoder. For interpretable classification, ALBM Zhang et al. (2025) organizes concepts by shared attributes and learns visual attribute prompts. Our learnable attribute queries attend to patches and match batchwise similarities, combining fine-grained aggregation with image-level relational supervision, without region annotations or coordinate-wise regression. Hash-code supervision integrates this branch with learned centers for generative zero-shot hashing.

3 Methodology

3.1 Problem Definition

Let $\mathcal{C}_s$ and $\mathcal{C}_u$ be disjoint seen and unseen class sets. Training uses real images from $\mathcal{C}_s$, Stable Diffusion pseudo-images from $\mathcal{C}_u$, and class attributes $\boldsymbol{a}_c \in \mathbb{R}^{d_a}$. We learn a hash function h that maps images to K-bit codes for semantic retrieval, with evaluation on queries from unseen classes. Training uses continuous codes in $(-1, +1)^K$; retrieval binarizes them with the sign function.

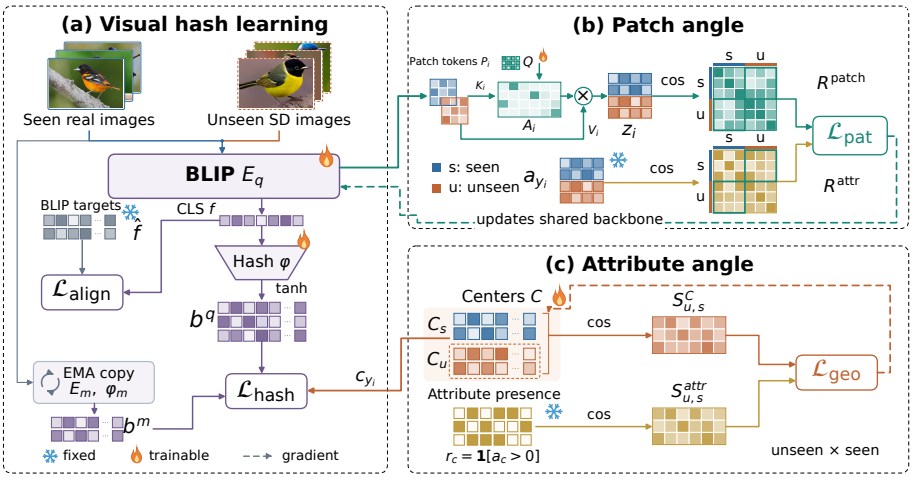


Figure 1: Overview of SST. Distillation from pooled patch tokens supervises the trainable BLIP backbone, while geometry alignment constrains relations between continuous centers of unseen and seen classes. The hash loss connects both paths. BLIP target features and attributes are fixed, and the momentum copy uses EMA without gradients. Both transfer losses apply only during training.

3.2 Framework Overview

SST transfers class-attribute relations to visual representations and hash centers through two semantic transfer modules. Patch-angle distillation supervises relations among pooled patch embeddings, while attribute-angle alignment constrains the geometry between unseen and seen class centers. These modules connect semantic structure to visual hash learning through a shared encoder and learnable centers. The trainable backbone is the visual encoder of pretrained BLIP Li et al. (2022), which produces a CLS feature $\boldsymbol{f} \in \mathbb{R}^{768}$ and patch tokens for each image. A two-layer hash head maps the CLS feature to $\boldsymbol{b}^q = \tanh(\phi(\boldsymbol{f})) \in (-1,+1)^K$, while an auxiliary branch processes pooled patch tokens for relational supervision. A learnable matrix $\boldsymbol{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k]^\top \in \mathbb{R}^{k \times K}$, where $k = |\mathcal{C}_s| + |\mathcal{C}_u|$, represents the class centers. Centers are independently initialized by Gaussian sampling on AWA2 and Bernoulli $\{-1,+1\}$ sampling on CUB and SUN, then learned from the corresponding real or pseudo-images. Fixed, precomputed BLIP features provide visual targets, and an EMA copy of the query network supplies momentum codes during training. Both transfer modules are optimized jointly with the visual alignment and hash objectives described in Sec. 3.5. An overview of the proposed framework SST is illustrated in Figure 1.

3.3 Fine-grained Semantic Transfer via Patch-level Relational Distillation

We transfer class-attribute relations to the shared BLIP backbone through attribute-query attention over patch tokens. Real seen images and unseen pseudo-images form a single training list, shuffled to construct mini-batches without a fixed ratio between the two groups. A training batch $\mathcal{B} = \mathcal{B}_s \cup \mathcal{B}_u$ contains B labeled samples $(\boldsymbol{x}_i, y_i)$, where $\mathcal{B}_s$ and $\mathcal{B}_u$ denote its seen real images and unseen pseudo-images, respectively; either subset may be empty. Unseen real images are unavailable during training, whereas unseen pseudo-images have known class labels and attributes.

For sample $i \in \{1, \ldots, B\}$, the query visual encoder E_q, with trainable parameters θ, produces $(\boldsymbol{f}_i, \boldsymbol{P}_i) = E_q(\boldsymbol{x}_i)$. Here $\boldsymbol{f}_i \in \mathbb{R}^{768}$ is the CLS feature and $\boldsymbol{P}_i \in \mathbb{R}^{N \times 768}$ contains patch tokens $\boldsymbol{p}_{i,n} \in \mathbb{R}^{768}$ for $n = 1, \ldots, N$. Each attribute dimension $a \in \{1, \ldots, d_a\}$ has a learnable query $\boldsymbol{q}_a$. These queries are shared across images and attend to the $N = 196$ patch tokens of each 224×224 input, excluding the CLS token. Writing the cross-attention operation in single-head notation, let $\boldsymbol{k}_{i,n} = W_k \boldsymbol{p}_{i,n}$ and $\boldsymbol{v}_{i,n} = W_v \boldsymbol{p}_{i,n}$ denote projected patch keys and values. The attention weights and attribute-specific representations are

$$\alpha_{i,a,n} = \frac{\exp(\boldsymbol{q}_a^\top \boldsymbol{k}_{i,n}/\sqrt{d_k})}{\sum_{m=1}^{N} \exp(\boldsymbol{q}_a^\top \boldsymbol{k}_{i,m}/\sqrt{d_k})}, \qquad \boldsymbol{o}_{i,a} = \sum_{n=1}^{N} \alpha_{i,a,n} \boldsymbol{v}_{i,n}, \tag{1}$$

where d_k is the key dimension. A learnable scalar readout r_a converts each query's output into one embedding coordinate. Denoting the complete auxiliary branch by g_a, the auxiliary and hash outputs are

$$\boldsymbol{z}_i = g_a(\boldsymbol{P}_i) = \left[r_1(\boldsymbol{o}_{i,1}), \ldots, r_{d_a}(\boldsymbol{o}_{i,d_a})\right]^\top, \qquad \boldsymbol{b}_i^q = \tanh(\phi(\boldsymbol{f}_i)) \,. \tag{2}$$

The notation r_a specifies the scalar associated with query a without prescribing whether readout parameters are shared. The resulting batch output remains in $\mathbb{R}^{B\times d_a}$. For each image, the attention matrix $[\alpha_{i,a,n}] \in \mathbb{R}^{d_a\times N}$ defines a soft association between attribute queries and patch locations, allowing different queries to aggregate different visual evidence. This explicit query-to-patch association is learned through the relational objective below, without region annotations. Because that objective supervises pairwise similarities rather than individual attribute coordinates, the attention maps alone do not establish that each query localizes its named semantic attribute. Accordingly, $\boldsymbol{z}_i$ is a relational embedding rather than a vector of calibrated attribute predictions.

For each sample, its label y_i selects the fixed class attribute vector $\boldsymbol{a}_{y_i} \in \mathbb{R}^{d_a}$, using binary attributes on AWA2 and the original continuous values on CUB and SUN. Stack the embeddings and their attribute targets as $\boldsymbol{Z}$ and $\boldsymbol{A}_{\mathcal{B}}$, whose i-th rows are $\boldsymbol{z}_i^\top$ and $\boldsymbol{a}_{y_i}^\top$, respectively, both in $\mathbb{R}^{B\times d_a}$, where $\top$ denotes transposition. An overbar denotes row-wise ℓ_2 normalization with numerical stabilization. The Gram matrices $\boldsymbol{R}^{\text{patch}} = \bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^\top$ and $\boldsymbol{R}^{\text{attr}} = \bar{\boldsymbol{A}}_{\mathcal{B}}\bar{\boldsymbol{A}}_{\mathcal{B}}^\top$ contain the pairwise cosine similarities shown in Figure 1(b). We match all entries through

$$\mathcal{L}_{\text{pat}} = \frac{1}{B^2}\left\|\boldsymbol{R}^{\text{patch}} - \boldsymbol{R}^{\text{attr}}\right\|_F^2 = \frac{1}{B^2}\left\|\bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^\top - \bar{\boldsymbol{A}}_{\mathcal{B}}\bar{\boldsymbol{A}}_{\mathcal{B}}^\top\right\|_F^2, \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm. Figure 1(b) groups rows and columns by seen (s) and unseen (u) samples for illustration. Both matrices contain seen–seen, seen–unseen, and unseen–unseen blocks; the loss includes all blocks, not only the outlined cross-group entries. In particular, each seen–unseen entry ties a real-image embedding to a pseudo-image embedding through their classes' attribute similarity, without requiring a one-to-one image pairing. For nonzero attribute vectors, same-class targets are one, while different-class targets retain graded semantic similarities.

The fixed targets constrain relations across images without requiring every attribute to be visible in each crop. Gradients from $\mathcal{L}_{\text{pat}}$ update g_a and the shared backbone parameters θ, but not ϕ directly. There is no gradient stop between the auxiliary branch and the encoder. The hash head reads CLS features from this same encoder, so patch supervision can affect the features used for hash learning even though the auxiliary branch is discarded at retrieval. This supplies semantic supervision to unseen pseudo-images in relation to seen real images, with the aim of learning representations that also transfer to unseen real queries. It does not directly align synthetic and real images of unseen classes or guarantee preservation of these relations after binarization.

3.4 Class-level Semantic Transfer via Attribute Geometry Alignment

The second module constrains continuous centers using class attributes. Seen centers receive hash supervision from real images and unseen centers from pseudo-images. Restricting the attribute constraint to pairs of unseen and seen classes connects these supervision sources without explicitly matching pairwise similarities within either group.

The center target uses attribute presence vectors $\boldsymbol{r}_c = \mathbb{I}[\boldsymbol{a}_c > 0]$, with similarity between classes given by $\boldsymbol{S}_{jl}^{\text{attr}} = \boldsymbol{r}_j^\top \boldsymbol{r}_l / (\|\boldsymbol{r}_j\|\|\boldsymbol{r}_l\|)$. This target measures shared attribute presence and discards attribute magnitudes, unlike the targets computed from the original attribute values in Eq. 3. The constraint therefore depends on the variation among these presence vectors. We align the unseen–seen block through

$$\mathcal{L}_{\text{geo}} = \frac{1}{|\mathcal{C}_u|\,|\mathcal{C}_s|}\sum_{j\in\mathcal{C}_u}\sum_{l\in\mathcal{C}_s}\left(\frac{\boldsymbol{c}_j^\top \boldsymbol{c}_l}{\|\boldsymbol{c}_j\|\,\|\boldsymbol{c}_l\|} - \boldsymbol{S}_{jl}^{\text{attr}}\right)^2. \tag{4}$$

This concentrates supervision on the relations linking unseen classes to seen classes. Gradients update both sets of centers, while the hash loss simultaneously ties each center to its image codes. The geometry loss has no direct gradient to the encoder or hash head; its effect on image representations is mediated by the learned centers and the hash loss.

The audited CUB attributes are on a 0–100 scale, while SUN attributes lie in $[0, 1]$. On both datasets, the zero threshold treats any positive value as presence, regardless of its magnitude. The target's

Algorithm 1 Final training of SST with the configuration selected using validation.

Require: Training images of seen classes $\mathcal{D}_s$ and pseudo-images of unseen classes $\mathcal{D}_u$; precomputed BLIP targets; class attributes $\{\boldsymbol{a}_c\}$; loss weights λ_{pat}, λ_{geo} and epoch count T selected using held-out seen classes (Sec. 4.1)

Ensure: A hash function $h : \mathcal{X} \to \{-1, +1\}^K$ and class centers $\boldsymbol{C} \in \mathbb{R}^{k \times K}$

Initialize the trainable BLIP encoder, hash head, and attribute branch; copy them to a momentum encoder
Initialize $\boldsymbol{C}$ by Gaussian sampling on AWA2 or Bernoulli sampling on CUB and SUN
for $t = 1, \ldots, T$ **do**
  for each mini-batch $\{(\boldsymbol{x}_i, y_i)\}$ from $\mathcal{D}_s \cup \mathcal{D}_u$ **do**
    Compute the CLS feature $\boldsymbol{f}_i$ and patch-token sequence $\boldsymbol{P}_i$ of $\boldsymbol{x}_i$
    $\boldsymbol{b}_i^q \leftarrow \tanh(\phi(\boldsymbol{f}_i))$; update the momentum copy by EMA ($m = 0.9$) and obtain $\boldsymbol{b}_i^m$ without gradients
    $\boldsymbol{z}_i \leftarrow g_a(\boldsymbol{P}_i)$: attend to patch keys/values with learnable attribute queries, then apply a scalar readout per query (Eqs. 1–2)
    $\mathcal{L}_{\text{align}} \leftarrow$ LogSoftmax contrastive alignment with fixed BLIP targets (Eq. 5)
    $\mathcal{L}_{\text{hash}} \leftarrow$ center MSE with momentum consistency (Eq. 6)
    $\mathcal{L}_{\text{pat}} \leftarrow$ pairwise attribute structure distillation (Eq. 3)
    $\mathcal{L}_{\text{geo}} \leftarrow$ unseen–seen center geometry alignment (Eq. 4)
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{hash}} + \lambda_{\text{pat}}\mathcal{L}_{\text{pat}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}}$ (Eq. 7)
    Update the query encoder, both heads, and $\boldsymbol{C}$ with one RMSprop step
  end for
end for
return The model after T epochs, using $h(\boldsymbol{x}) = \text{sign}(\phi(\boldsymbol{f}))$ or the adopted multi-view aggregation at retrieval

semantic resolution depends on variation in attribute presence. Identical nonzero presence vectors yield a target block whose entries are all one, favoring coincident center directions. Appendix A.8 examines this potential degeneracy in the audited attribute files.

Cosine similarities are unchanged by positive rescaling, so the geometry term alone does not determine center norms. The MSE term for class centers supplies a complementary constraint by fitting each center to its class's continuous image codes. Together, these terms allow attribute relations to guide the centers while retaining supervision from visual samples.

The patch term compares embeddings of images within a batch, whereas the geometry term compares all pairs of centers from unseen and seen classes, including classes absent from that batch. It constrains cosine similarities of continuous centers, without prescribing their absolute coordinates or guaranteeing that the same similarities hold after binarization. Centers use the initialization in Sec. 3.2 and are subsequently optimized by the hash and geometry losses.

3.5 Unified Training Objective

The patch and geometry terms supervise semantic relations. We combine them with visual alignment and hash supervision, which are shared by the full model and all ablation variants.

Visual alignment. The alignment loss compares normalized image features with fixed, precomputed BLIP targets through

$$\mathcal{L}_{\text{align}} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{B}\tilde{\boldsymbol{M}}_{ij}\log\frac{\exp(\bar{\boldsymbol{f}}_i^\top \hat{\bar{\boldsymbol{f}}}_j/\tau)}{\sum_{l=1}^{B}\exp(\bar{\boldsymbol{f}}_i^\top \hat{\bar{\boldsymbol{f}}}_l/\tau)}, \tag{5}$$

where B is the batch size, a bar denotes ℓ_2 normalization, $\hat{\boldsymbol{f}}_j$ is the target stored with sample j, and τ is the temperature. Targets include features of synthetic counterparts. For seen classes in SUN, the targets instead come from the real images themselves. The mask $\tilde{\boldsymbol{M}}$ identifies pairs from the same class, excludes the diagonal, and is normalized by row. Rows are zero for samples with no other example from the same class. The denominator includes all batch targets.

Hash supervision. The hash loss fits each image code to its class center and encourages consistency with the momentum copy,

$$\mathcal{L}_{\text{hash}} = \frac{1}{BK} \sum_{i=1}^{B} \left(\left\| \boldsymbol{b}_i^q - \boldsymbol{c}_{y_i} \right\|_2^2 + \alpha \left\| \bar{\boldsymbol{b}}_i^q - \text{sg}(\bar{\boldsymbol{b}}_i^m) \right\|_2^2 \right), \tag{6}$$

where y_i is the class label, $\boldsymbol{b}_i^q$ and $\boldsymbol{b}_i^m$ are codes from the query network and its momentum copy He et al. (2020) for the same augmented image, sg denotes stop-gradient, and $\alpha = 0.1$. The factor $1/(BK)$ implements an elementwise mean.

Combining visual alignment and hash supervision with the two transfer losses gives

$$\mathcal{L} = \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{hash}} + \lambda_{\text{pat}} \mathcal{L}_{\text{pat}} + \lambda_{\text{geo}} \mathcal{L}_{\text{geo}}, \tag{7}$$

where λ_{pat} and λ_{geo} balance the two transfer terms. Each iteration combines the terms computed within each batch with geometry alignment between all unseen and seen centers in a single backward pass. The overall training procedure is summarized in Algorithm 1.

For each dataset and code length, the loss weights and training duration are selected using validation on classes held out from the seen set, then frozen before final training and testing (Sec. 4.1). The retained loss keeps its selected weight in each removal variant.

3.6 Theoretical Analysis

We analyze when the center relations constrained by Eq. 4 carry over to binary image codes. The result is deterministic for a fixed model and does not assume convergence of the training algorithm.

Definition 3.1 (Transfer errors) *Let $\boldsymbol{v}(x)$ be the continuous vector immediately before retrieval binarization, including view aggregation when used, and let $\boldsymbol{h}(x) = \text{sign}(\boldsymbol{v}(x))$. For this analysis, fix the sign at zero to $+1$. Write $\boldsymbol{u}(x) = \boldsymbol{v}(x)/\|\boldsymbol{v}(x)\|_2$ and $\boldsymbol{t}(x) = \boldsymbol{h}(x)/\sqrt{K}$. For $g \in \{u, s\}$, $\mathbb{E}_g$ selects a class uniformly from $\mathcal{C}_g$ and then an image from a specified distribution for that class. Define*

$$E = \frac{1}{2} \sum_{g \in \{u,s\}} \mathbb{E}_g \|\boldsymbol{v}(x) - \boldsymbol{c}_y\|_2^2, \qquad Q = \frac{1}{2} \sum_{g \in \{u,s\}} \mathbb{E}_g \|\boldsymbol{t}(x) - \boldsymbol{u}(x)\|_2. \tag{8}$$

Here E measures fitting error to class centers and Q measures the change in direction caused by binarization.

Assumption 3.2 (Nonzero directions) *All presence vectors are nonzero, $\|\boldsymbol{c}_c\|_2 \geq \rho > 0$ for every class, and $\boldsymbol{v}(x) \neq \boldsymbol{0}$ almost surely under the specified image distributions. The error E is finite.*

Theorem 3.3 (Transfer to Hamming similarities) *Under Assumption 3.2, independently draw (x, y) using $\mathbb{E}_u$ and (x', y') using $\mathbb{E}_s$. If d_H counts differing bits, then*

$$\mathbb{E} \left| 1 - \frac{2 d_H(\boldsymbol{h}(x), \boldsymbol{h}(x'))}{K} - S_{yy'}^{\text{attr}} \right| \leq \sqrt{\mathcal{L}_{\text{geo}}} + \frac{4\sqrt{E}}{\rho} + 2Q. \tag{9}$$

The proof in Appendix A.1 combines stability of vector normalization with the identity between binary inner products and Hamming distance.

Remark 3.4 (Scope of the bound) *Geometry alignment controls one term; accurate fitting to centers and small binarization error are also needed for a tight bound. For real unseen images, E is not bounded by the training loss on pseudo-images. The result concerns unseen–seen relations, not unseen–unseen retrieval rankings or mAP. Patch distillation constrains a separate embedding space and can affect these errors through the shared backbone, but is not itself a bound on them.*

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on three zero-shot hashing benchmarks. AWA2 Xian et al. (2019) contains 37,322 images of 50 animal classes with 85-dimensional attributes, and we use

Table 1: Retrieval results on AWA2, CUB, and SUN. Literature scores retain their source protocols as contextual references. The PZSH and SST rows use our shared mAP@5000 protocol; bold identifies SST. See Sec. 4.1.

Method	AWA2				CUB				SUN			
	24	48	64	128	24	48	64	128	24	48	64	128
LSH	0.0106	0.0151	0.0204	0.0306	0.0055	0.0076	0.0095	0.0095	0.0598	0.0738	0.0764	0.0880
SH	0.1833	0.2729	0.2955	0.3441	0.0568	0.0810	0.0886	0.1191	0.0738	0.0811	0.0837	0.0880
ITQ	0.1999	0.2821	0.2964	0.3764	0.0533	0.0765	0.0892	0.1182	0.0725	0.0899	0.0918	0.1007
IMH	0.1282	0.1536	0.1613	0.1681	0.0330	0.0361	0.0364	0.0386	0.0572	0.0668	0.0638	0.0651
PCA	0.2165	0.2530	0.2701	0.2719	0.0547	0.0598	0.0632	0.0695	0.0808	0.0868	0.0891	0.0980
HashNet	0.2086	0.2386	0.2516	0.2749	0.0528	0.0566	0.0595	0.0633	0.0746	0.0825	0.0865	0.0944
GreedyHash	0.3420	0.4169	0.4240	0.4639	0.1132	0.1707	0.1841	0.2326	0.0873	0.1259	0.1398	0.1550
JMLH	0.3607	0.4364	0.4408	0.4711	0.1078	0.1555	0.1987	0.2310	0.0860	0.1168	0.1345	0.1500
ADSH	0.3360	0.4787	0.5105	0.5454	0.0858	0.1607	0.1827	0.2424	0.0886	0.1230	0.1341	0.1731
CSQ	0.3194	0.3988	0.3773	0.4072	0.0996	0.1588	0.1712	0.2201	0.0915	0.1211	0.1375	0.1669
DPN	0.1783	0.2086	0.2378	0.2565	0.0445	0.0728	0.0772	0.1003	0.0737	0.0825	0.0894	0.1126
BiHalf	0.3440	0.4036	0.4223	0.4577	0.0794	0.1280	0.1573	0.2142	0.0077	0.0490	0.0512	0.1307
OrthoCos	0.1709	0.2302	0.2312	0.2566	0.0451	0.0660	0.0736	0.0984	0.0728	0.0809	0.0868	0.1126
CIBHash	0.2113	0.2304	0.2481	0.2618	0.0351	0.0404	0.0411	0.0453	0.0690	0.0808	0.0809	0.0883
TBH	0.0941	0.1201	0.1073	0.1730	0.0157	0.0176	0.0226	0.0252	0.0169	0.0230	0.0248	0.0287
PIXEL	0.3819	0.4792	0.5133	0.5465	0.1136	0.1777	0.1994	0.2519	0.0923	0.1264	0.1415	0.1741
COMAE	0.3843	0.5073	0.5149	0.5801	0.1142	0.1803	0.2012	0.2532	0.0925	0.1278	0.1465	0.1764
PZSH	0.5132	0.5877	0.6149	0.6336	0.1397	0.2160	0.2281	0.2704	0.1055	0.1458	0.1771	0.1949
SST	**0.5769**	**0.6192**	**0.6288**	**0.6643**	**0.1581**	**0.2362**	**0.2547**	**0.3060**	**0.1273**	**0.1691**	**0.2097**	**0.2169**

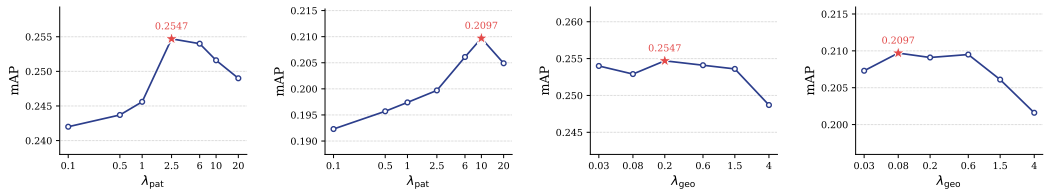
40 seen and 10 unseen classes. CUB Wah et al. (2011) contains 11,788 images of 200 bird species with 312-dimensional attributes, split into 150 seen and 50 unseen classes. SUN Xiao et al. (2010); Patterson & Hays (2012) contains 14,340 scene images of 717 classes with 102-dimensional attributes; we use 500 seen and 217 unseen classes. Together, these benchmarks cover animal categories, fine-grained bird species, and diverse scenes, enabling evaluation of semantic transfer across distinct visual domains and attribute vocabularies. Training combines real images from seen classes with Stable Diffusion pseudo-images of unseen classes. Real images of the final unseen classes are reserved for evaluation, testing transfer from synthetic training examples to real queries. Fixed BLIP targets provide the contrastive supervision in Sec. 3.5. The appendix gives the split sizes and corresponding training and evaluation counts.

Baseline methods. We compare with traditional hashing methods LSH Andoni & Indyk (2008), SH Weiss et al. (2008), ITQ Gong et al. (2013), PCA Maćkiewicz & Ratajczak (1993), and IMH Shen et al. (2015); deep hashing methods HashNet Cao et al. (2017), GreedyHash Su et al. (2018), JMLH Shen et al. (2019), ADSH Zhou et al. (2019), CSQ Yuan et al. (2020), DPN Fan et al. (2020), OrthoCos Hoe et al. (2021), BiHalf Li & van Gemert (2021), CIBHash Qiu et al. (2021), and TBH Shen et al. (2020); and zero-shot methods PIXEL Dong et al. (2024), COMAE Li et al. (2025), and PZSH Zhang et al. (2026). COMAE (ICMR 2025) and PZSH (AAAI 2026) provide recent strong zero-shot hashing baselines. On AWA2 and CUB, we follow the class splits, retrieval database construction, and AP computation used in the PZSH and PIXEL implementations. Our rerun comparisons use matched training and evaluation settings. Results taken from the literature retain their original protocols, as detailed in Appendix A.7.

Evaluation and implementation. We report mAP@5000 over queries from unseen classes on all three datasets, ranking database images by Hamming distance between their respective binary hash codes. For each query, AP averages precision at relevant ranks within the top 5,000 results; queries with no relevant retrieval contribute zero. We use a trainable BLIP-base visual encoder, a two-layer hash head, and learnable centers initialized from $\mathcal{N}(0, 1)$ on AWA2 and Bernoulli $\{-1, +1\}$ sampling on CUB and SUN. RMSprop uses learning rates of 10^{-5} for the query network and 2×10^{-4} for the centers, with weight decay 10^{-5} and learning rates halved every 15 epochs. Run-level results and seed conventions are detailed in Appendix A.10. The EMA coefficient is 0.9 and the contrastive temperature is 0.07. We train for 25, 30, and 35 epochs on AWA2, CUB, and SUN. Appendix A.6 gives the shared batch sizes and augmentation settings; Appendix A.3 details retrieval views and aggregation. Hyperparameters and training duration are selected on held-out seen classes and fixed before final training and testing (Appendix A.5).

Table 2: Individual-run ablation results in mAP@5000; seed details and additional runs appear in Appendix A.10. The w/o both variant disables both transfer losses; each removal retains the other loss's weight. All rows share the protocol for each setting in Appendix A.6.

Variant	AWA2				CUB				SUN			
	24	48	64	128	24	48	64	128	24	48	64	128
w/o both	0.5132	0.5877	0.6149	0.6336	0.1397	0.2160	0.2281	0.2704	0.1055	0.1458	0.1771	0.1949
w/o Patch	0.5348	0.6037	0.6285	0.6508	0.1556	0.2226	0.2405	0.2881	0.1259	0.1581	0.1986	0.2140
w/o Geometry	0.5732	0.5975	0.6196	0.6483	0.1571	0.2312	0.2351	0.3035	0.1240	0.1605	0.1887	0.2088
SST	**0.5769**	**0.6192**	**0.6288**	**0.6643**	**0.1581**	**0.2362**	**0.2547**	**0.3060**	**0.1273**	**0.1691**	**0.2097**	**0.2169**



(a) CUB, λ_{pat} (b) SUN, λ_{pat} (c) CUB, λ_{geo} (d) SUN, λ_{geo}

Figure 2: Sensitivity of 64-bit retrieval mAP@5000 on CUB and SUN. Panels (a,b) vary λ_{pat}, and panels (c,d) vary λ_{geo}. Horizontal axes are logarithmic. Red stars mark the highest mAP in each displayed sweep.

4.2 Performance Comparison

Retrieval performance. Table 1 shows that SST exceeds PZSH at all twelve settings under the shared configurations. The gains vary with dataset and code length. On AWA2, the improvement is strongest at the shortest code length, whereas on CUB it grows as codes become longer. SUN benefits across all four code lengths, with the largest gain at 64 bits. These observations support the effectiveness of structured semantic transfer across animal, bird, and scene retrieval within the evaluated protocol. Literature scores retain the comparison limits in Sec. 4.1.

Code storage. Shorter SST codes can outperform longer baseline codes. On SUN, the 64-bit model exceeds the 128-bit baseline while using 50% fewer code bits; on CUB, the 48-bit model exceeds the 64-bit baseline. Each image retains one binary code after view aggregation. These comparisons concern code storage, not feature-extraction time or total index memory.

4.3 Ablation Study

Table 2 compares the full model with removal of either or both transfer losses under shared settings (Appendix A.6). Each single-loss variant improves on w/o both at all twelve settings, while the full model achieves the highest mAP, supporting complementary contributions. Removing the patch term causes the largest drop on AWA2 at 24 bits (0.0421). Geometry removal has its largest effect on AWA2 at 48 bits (0.0217), CUB at 64 bits (0.0196), and SUN at 64 bits (0.0210). The gains over the better single-loss variant are smaller on AWA2-64 and CUB-24 (0.0003 and 0.0010), indicating that the contributions vary with dataset and code length. Additional runs, design controls, and representation diagnostics appear in Appendices A.10, A.11, and A.9.

4.4 Sensitivity Analysis

Loss weights. Figure 2 examines the two transfer weights on CUB and SUN at 64 bits. The patch curves peak at $\lambda_{pat} = 2.5$ on CUB and 10.0 on SUN. CUB remains near its peak at 6.0, whereas SUN continues improving up to 10.0; both decline at the largest tested weight, 20.0. The geometry curves fluctuate less over the lower and intermediate tested weights, with displayed maxima at $\lambda_{geo} = 0.2$ on CUB and 0.08 on SUN, before declining at 4.0. The peak scores match the 64-bit table entries. The red stars identify maxima within each sweep, not a joint optimum or evidence that the sweep and main runs used identical configurations. The final 64-bit geometry weights selected using validation

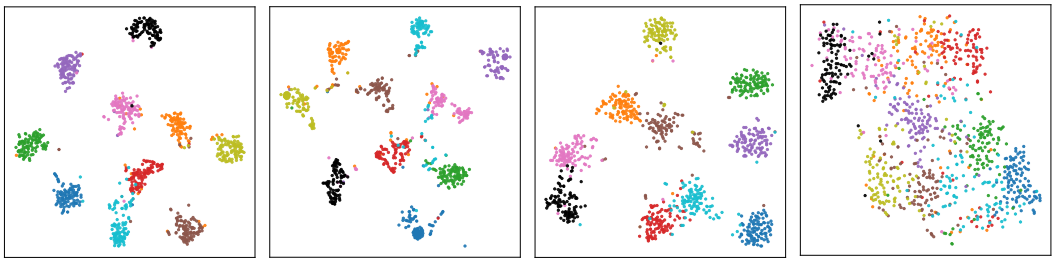


Figure 3: t-SNE of 48-bit AWA2 codes of queries from unseen classes. Colors denote classes; each method is embedded independently using the same t-SNE settings.



Figure 4: Top-10 retrievals for four 128-bit CUB queries, selected where SST outperforms PZSH. In each block, the top row shows SST and the bottom row shows PZSH. Green and red frames mark correct and incorrect matches, respectively.

are 0.2 on CUB and 0.08 on SUN, consistent with the displayed maxima. The numerical sizes of the weights do not directly indicate the modules' relative importance, since the losses act on different relations and parameter updates. Table 2 assesses their removal effects.

4.5 Visualization and Case Study

Hash space visualization. Figure 3 projects the 48-bit codes of AWA2 queries from unseen classes, with relatively compact class groupings for SST. Because each method is embedded independently, distances across panels and cluster sizes do not measure geometry in Hamming space. This view complements the retrieval and ablation results without isolating either loss's contribution. The class labels help reveal separation and overlap among unseen categories, providing a visual check of class organization in the projected codes.

Retrieval examples. Figure 4 illustrates selected successes and errors on unseen CUB queries. The paired rows enable direct comparison of relevant and irrelevant images retrieved for the same query, highlighting differences in the composition of the top-ranked results. The colored borders also expose remaining retrieval errors despite the improvements in these selected examples. Both correct and incorrect matches can share similar colors; these cases do not establish local attribute correspondence as the source of improvement.

5 Conclusion

We presented SST, a zero-shot hashing method transferring semantic relations to visual representations and class centers. The patch branch distills class-attribute relations into the shared visual backbone through batchwise similarities of embeddings formed by attribute-query cross-attention. The geometry loss aligns unseen–seen cosine similarities of continuous class centers with attribute similarities. Experiments on AWA2, CUB, and SUN across four hash lengths show retrieval gains from the proposed transfer losses at all twelve settings. Experiments across more random seeds support consistent gains over removing either or both transfer losses; a subset is reported in the appendix. At retrieval, each image uses a binary code; attribute relations enter through training and require no additional semantic inputs for a query. Future work will examine spatial grounding of attribute queries and attribute geometry at multiple semantic granularities.

AI USE STATEMENT

The authors developed the research ideas, implemented the methods, conducted the experiments, analyzed the results, and wrote the original manuscript without assistance from generative AI tools. We only used AI tools for polishing when writing research papers. The authors reviewed that feedback, verified the final statement, and take full responsibility for the content of the submission.

ETHICS STATEMENT

Our work uses publicly available benchmark datasets (AWA2, CUB, and SUN) and pre-trained models, and does not involve human subjects, private data, or deployment concerns. Synthetic pseudo-images are generated from class-level semantic prompts with a pre-trained stable diffusion model; these are used solely for training zero-shot retrieval models and do not target any individual. We foresee no direct harmful applications of this work.

REPRODUCIBILITY STATEMENT

The appendix describes optimization, dataset splits, retrieval implementation, and shared comparison settings. It reports individual results and seed conventions, including means, population standard deviations, and paired gains for the additional repeated runs. Code is available anonymously at https://anonymous.4open.science/r/SST-B163.

REFERENCES

Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.

Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE international conference on computer vision*, pp. 5608–5617, 2017.

Li Chen, Rui Liu, Yuxiang Zhou, Xudong Ma, Yong Chen, and Dell Zhang. Deep hashing with semantic hash centers for image retrieval. *ACM Transactions on Information Systems*, 43(6): 160:1–160:38, 2025. doi: 10.1145/3749983.

Peitao Cheng, Hanwen Cui, Xi Zhang, and Xiumei Wang. Unsupervised multi-semantic similarity reconstruction contrastive hashing for multi-label image retrieval. *Expert Systems with Applications*, 301:130414, 2026. doi: 10.1016/j.eswa.2025.130414.

Zeyu Dong, Qingqing Long, Yihang Zhou, Pengfei Wang, Zhihong Zhu, Xiao Luo, Yidong Wang, Pengyang Wang, and Yuanchun Zhou. Pixel: Prompt-based zero-shot hashing via visual and textual semantic alignment. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 487–496, 2024.

Lixin Fan, Kam Woh Ng, Ce Ju, Tianyu Zhang, and Chee Seng Chan. Deep polarized network for supervised learning of accurate binary hashing codes. In *IJCAI*, pp. 825–831, 2020.

Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2916–2929, 2013.

Yuchen Guo, Guiguang Ding, Jungong Han, and Yue Gao. Sitnet: Discrete similarity transfer network for zero-shot hashing. In *IJCAI*, pp. 1767–1773, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Jiun Tian Hoe, Kam Woh Ng, Tianyu Zhang, Chee Seng Chan, Yi-Zhe Song, and Tao Xiang. One loss for all: Deep hashing with a single cosine similarity based learning objective. *Advances in neural information processing systems*, 34:24286–24298, 2021.

Yan Jiang, Zhongmiao Qi, Jianhao Li, Jiangbo Qian, Chong Wang, and Yu Xin. Zero-shot hashing based on reconstruction with part alignment. *arXiv preprint arXiv:2503.07037*, 2025.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Yunqiang Li and Jan van Gemert. Deep unsupervised image hashing by maximizing bit entropy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 2002–2010, 2021.

Yuqi Li, Qingqing Long, Yihang Zhou, Ran Zhang, Zhiyuan Ning, Zhihong Zhu, Yuanchun Zhou, Xuezhi Wang, and Meng Xiao. Comae: Comprehensive attribute exploration for zero-shot hashing. In *Proceedings of the 2025 International Conference on Multimedia Retrieval*, pp. 733–742, 2025.

Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993.

Genevieve Patterson and James Hays. SUN attribute database: Discovering, annotating, and recognizing scene attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Zexuan Qiu, Qinliang Su, Zijing Ou, Jianxing Yu, and Changyou Chen. Unsupervised hashing with contrastive information bottleneck. *arXiv preprint arXiv:2105.06138*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Fumin Shen, Chunhua Shen, Qinfeng Shi, Anton Van den Hengel, Zhenmin Tang, and Heng Tao Shen. Hashing on nonlinear manifolds. *IEEE Transactions on Image Processing*, 24(6):1839–1851, 2015.

Yuming Shen, Jie Qin, Jiaxin Chen, Li Liu, Fan Zhu, and Ziyi Shen. Embarrassingly simple binary representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

Yuming Shen, Jie Qin, Jiaxin Chen, Mengyang Yu, Li Liu, Fan Zhu, Fumin Shen, and Ling Shao. Auto-encoding twin-bottleneck hashing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2818–2827, 2020.

Yang Shi, Xiushan Nie, Xingbo Liu, Lu Yang, and Yilong Yin. Zero-shot hashing via asymmetric ratio similarity matrix. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):5426–5437, 2023.

Shupeng Su, Chao Zhang, Kai Han, and Yonghong Tian. Greedy hash: Towards fast optimization for accurate hash coding in cnn. *Advances in neural information processing systems*, 31, 2018.

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

Hong Wang, Yongyue Fu, Qibing Qin, Wenfang Zhang, and Lei Huang. Deep attribute-aware hashing for zero-shot image retrieval. *Expert Systems with Applications*, 325:132545, 2026. doi: 10.1016/j.eswa.2026.132545.

Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, volume 21, pp. 1753–1760, 2008.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2019.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Yang Yang, Yadan Luo, Weilun Chen, Fumin Shen, Jie Shao, and Heng Tao Shen. Zero-shot hashing via transferring supervised knowledge. In *Proceedings of the 24th ACM international conference on Multimedia*, pp. 1286–1295, 2016.

Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. Central similarity quantization for efficient image and video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3083–3092, 2020.

Haofeng Zhang, Yang Long, and Ling Shao. Zero-shot hashing with orthogonal projection for image retrieval. *Pattern Recognition Letters*, 117:201–209, 2019.

Hui Zhang, Weikang Gao, Tao Yang, and Yuan Cao. Proxy zero-shot hashing with multimodal fusion via stable diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12529–12537, 2026.

Jianyang Zhang, Qianli Luo, Guowu Yang, Wenjing Yang, Weide Liu, Guosheng Lin, and Fengmao Lv. Attribute-formed class-specific concept space: Endowing language bottleneck model with better interpretability and scalability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30291–30300, 2025.

Chang Zhou, Lai-Man Po, Wilson YF Yuen, Kwok Wai Cheung, Xuyuan Xu, Kin Wai Lau, Yuzhi Zhao, Mengyang Liu, and Peter HW Wong. Angular deep supervised hashing for image retrieval. *IEEE Access*, 7:127521–127532, 2019.

A Appendix

A.1 Proof of Theorem 3.3

For nonzero vectors $\boldsymbol{a}, \boldsymbol{b}$, the triangle inequality and the reverse triangle inequality give

$$\left\| \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_2} - \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|_2} \right\|_2 \le \frac{\|\boldsymbol{a}-\boldsymbol{b}\|_2}{\|\boldsymbol{b}\|_2} + \frac{|\|\boldsymbol{b}\|_2 - \|\boldsymbol{a}\|_2|}{\|\boldsymbol{b}\|_2} \le \frac{2\|\boldsymbol{a}-\boldsymbol{b}\|_2}{\|\boldsymbol{b}\|_2}. \tag{10}$$

Let $\widehat{\boldsymbol{c}}_y = \boldsymbol{c}_y/\|\boldsymbol{c}_y\|_2$, $e(x) = \|\boldsymbol{v}(x) - \boldsymbol{c}_y\|_2$, and $q(x) = \|\boldsymbol{t}(x) - \boldsymbol{u}(x)\|_2$. Assumption 3.2 and Eq. 10 imply

$$\|\boldsymbol{t}(x) - \widehat{\boldsymbol{c}}_y\|_2 \le q(x) + \|\boldsymbol{u}(x) - \widehat{\boldsymbol{c}}_y\|_2 \le q(x) + 2e(x)/\rho.$$

All normalized vectors have unit norm. Applying Cauchy–Schwarz to the difference of two inner products therefore yields

$$\begin{aligned} |\boldsymbol{t}(x)^\top \boldsymbol{t}(x') - S^{\text{attr}}_{yy'}| &\le |\widehat{\boldsymbol{c}}_y^\top \widehat{\boldsymbol{c}}_{y'} - S^{\text{attr}}_{yy'}| + \|\boldsymbol{t}(x) - \widehat{\boldsymbol{c}}_y\|_2 + \|\boldsymbol{t}(x') - \widehat{\boldsymbol{c}}_{y'}\|_2 \\ &\le |\widehat{\boldsymbol{c}}_y^\top \widehat{\boldsymbol{c}}_{y'} - S^{\text{attr}}_{yy'}| + q(x) + q(x') + \frac{2}{\rho}\big(e(x) + e(x')\big). \end{aligned}$$

Independent uniform sampling of the unseen and seen classes makes the expected squared center residual exactly $\mathcal{L}_{\text{geo}}$. Its expected absolute value is at most $\sqrt{\mathcal{L}_{\text{geo}}}$. Moreover,

$$\mathbb{E}_u e + \mathbb{E}_s e \le \sqrt{\mathbb{E}_u e^2} + \sqrt{\mathbb{E}_s e^2} \le 2\sqrt{E}, \qquad \mathbb{E}_u q + \mathbb{E}_s q = 2Q.$$

Finally, $\boldsymbol{t}(x)^\top \boldsymbol{t}(x') = 1 - 2d_H(\boldsymbol{h}(x), \boldsymbol{h}(x'))/K$. Taking expectations proves Eq. 9. □

The quantities in the theorem are evaluated on the specified image distributions for a fixed model. In particular, E uses a uniform distribution over classes, has no $1/K$ normalization, and includes any retrieval view aggregation. It is therefore not identical to the mini-batch hash loss in Eq. 6. No assumption that training attains small E or Q on unseen real images is made.

A.2 Training and Retrieval Settings

All settings use the MSE hash loss for class centers with momentum consistency ($\alpha = 0.1$), contrastive temperature $\tau = 0.07$, and EMA coefficient 0.9. The main retrieval and ablation tables use seed 42; the representation diagnostics identify each run's seed separately. Centers are initialized from $\mathcal{N}(0, 1)$ on AWA2 and Bernoulli $\{-1, +1\}$ sampling on CUB and SUN. RMSprop uses learning rates of 10^{-5} for the query network and 2×10^{-4} for the centers, with weight decay 10^{-5}; both learning rates are halved every 15 epochs.

A.3 Retrieval Implementation

At inference we optionally aggregate multiple views of each image before binarization. Writing $\boldsymbol{b}^{(v)} = \tanh(\phi(\boldsymbol{f}^{(v)}))$, the current implementation applies an additional bounded transform $q(\boldsymbol{b}) = \tanh(\boldsymbol{b})$ before averaging, giving

$$\boldsymbol{b} = \operatorname{sign}\Big(\tfrac{1}{V}\sum_{v=1}^{V} q(\boldsymbol{b}^{(v)})\Big), \tag{11}$$

where the $V = 10$ views consist of five 224×224 crops and their horizontal flips. The image is resized to a short side of 256 pixels, and the crop origins are $(0, 0)$, $(0, 32)$, $(32, 0)$, $(32, 32)$, and $(16, 16)$; these are the four corners and center only for a square resized image. Both the signs and magnitudes of the transformed codes from each view contribute to the final binary code. We use this aggregation on AWA2 and CUB, and a single view on SUN.

A.4 Dataset Split Details

AWA2 uses 40 seen and 10 unseen classes. Its 4,400 training images comprise 100 real images per seen class and 40 pseudo-images per unseen class. CUB uses 150 seen and 50 unseen classes, with 30 real images per seen class and 40 pseudo-images per unseen class, giving 6,500 training images.

Evaluation uses 1,500 queries from unseen classes and a database of 5,788 images. SUN uses 500 seen and 217 unseen classes. Training uses 8,000 real images from seen classes and 4,557 pseudo-images, or 21 per unseen class; evaluation uses 4,340 queries from unseen classes and a database of 14,340 images. All pseudo-images are generated by Stable Diffusion from prompts based on class names.

A.5 Separation of Validation and Final Testing

The counts of seen and unseen classes in the dataset description refer to the final evaluation split. Model selection introduces a separate partition of the seen classes into disjoint subsets, with

$$\mathcal{C}_s = \mathcal{C}_{\text{meta-seen}} \dot{\cup} \mathcal{C}_{\text{meta-unseen}}, \qquad \mathcal{C}_{\text{meta-unseen}} \cap \mathcal{C}_u = \varnothing.$$

Only real images from the meta-seen classes enter validation training. Meta-unseen classes are represented during this stage by their semantic information and generated pseudo-images; their held-out real images are used for validation retrieval. We select hyperparameters and the epoch count using validation mAP, then freeze the configuration and train the final model with the original seen classes and the pseudo-images of unseen classes. The final checkpoint is determined by this fixed epoch count, with no selection based on mAP on the test queries. The same validation partition is used for all variants. Removal variants inherit the selected full-model configuration without additional tuning, as detailed in Appendix A.6.

A.6 Shared Baseline and Ablation Settings

For each dataset and code length, the baseline and both removal variants share the full model's training and retrieval settings. The comparison shares the real and pseudo training samples, fixed BLIP targets, class attributes, lists of queries and database images, relevance definition, AP implementation, backbone initialization, optimizer, schedule, seed, augmentation, batch size, retrieval views, and selected configuration. The table below summarizes the transfer-loss weights. A zero disables the corresponding loss; it is neither a retrieval score nor a missing value.

Variant	Patch weight	Geometry weight
PZSH baseline (w/o both)	0	0
w/o Patch	0	selected λ_{geo}
w/o Geometry	selected λ_{pat}	0
SST	selected λ_{pat}	selected λ_{geo}

The retained loss is not retuned after removing the other loss. Thus these are removals from a fixed configuration of the full model. PZSH baseline results in Tables 1 and 2 are the same runs with both transfer terms disabled. The original main and ablation tables use seed 42; additional runs are reported separately in Appendix A.10.

Where enabled, augmentation uses random resized crop and horizontal flip. Batch size is 64 on CUB at 24/48 bits and SUN at 24 bits, and 32 elsewhere. AWA2 at 64 bits and SUN at 128 bits disable augmentation in all four variants. Retrieval uses ten-view aggregation on AWA2 and CUB and a single view on SUN for all corresponding variants. View aggregation increases the cost of feature extraction while retaining one K-bit code per image. These shared choices hold augmentation, batch size, and view count fixed for each ablation comparison.

A.7 Additional Baselines and References

For AWA2 and CUB, results of methods preceding COMAE and PZSH are taken from the comparison reported by PZSH Zhang et al. (2026); COMAE results come from its paper Li et al. (2025). The PZSH baseline rows in Tables 1 and 2 are our runs using the visual alignment and hash objectives in Sec. 3.5 and the shared settings above; they are not copied from the original paper's result table. We report our results as mAP@5000 on all three datasets. PZSH did not report SUN results, so its SUN row is our extension of the baseline; the other SUN baseline values come from COMAE's comparison Li et al. (2025).

For AWA2 and CUB, our class splits, retrieval database construction, and AP computation follow the PZSH and PIXEL implementations. Both our model and the rerun PZSH comparison use these

settings with a cutoff of 5,000 and the shared training and retrieval conditions in Appendix A.6. This implementation-level protocol inheritance does not establish that every imported score in PZSH's published comparison was obtained under the same conditions. On SUN, the 500/217 class split is adopted from prior deep hashing experiments, rather than from PZSH, which does not report this dataset.

Literature scores in Table 1 retain their source protocols. PZSH labels its published comparison as mAP@all, whereas COMAE reports mAP@5000. COMAE describes 645 seen and 72 unseen SUN classes; our SUN experiments use 500 and 217, respectively. Thus the imported SUN scores are contextual references across different class partitions, rather than a controlled ranking. Identical dataset names and code lengths do not establish identical class identities, database membership, AP normalization, or retrieval views. The quantitative attribution to the proposed losses is based on the shared baseline and removal configurations above.

A.8 Attribute Presence Analysis

Table 3: Presence similarity statistics over unordered pairs of distinct classes. Active denotes the mean fraction of positive attributes per class; r is Pearson correlation between presence and continuous-attribute cosine similarities over the same pairs. SD measures variation across class pairs, not training seeds.

Dataset	Pairs	Active (%)	Mean ± SD	Min–Max	r
AWA2	1,225	36.75	0.5186 ± 0.1379	0.1404–0.9286	1.000
CUB	19,900	57.39	0.7319 ± 0.0573	0.4092–0.9048	0.536
SUN	256,686	53.53	0.6243 ± 0.1246	0.1614–0.9283	0.837

Target distribution. We examine cosine similarities of the presence vectors $\boldsymbol{r}_c = \mathbb{I}[\boldsymbol{a}_c > 0]$ in the audited attribute files. Table 3 summarizes all unordered pairs of distinct classes, including both seen and unseen classes. Every class has a nonzero, distinct presence vector, and no pair has similarity above 0.95. Thus, these files do not exhibit identical presence vectors or a constant similarity target. CUB nevertheless has a more concentrated distribution than AWA2 or SUN. Thresholding also reduces the standard deviation of pairwise similarities relative to continuous attributes, from 0.1390 to 0.0573 on CUB and from 0.2279 to 0.1246 on SUN. The correlations in Table 3 show that presence and continuous attributes encode different relations, particularly on CUB. These statistics exclude the constant-target failure mode for the audited files; they do not establish that thresholding improves retrieval or that the learned binary codes preserve these relations.

A.9 Patch Representation Diagnostics

Table 4: SUN 128-bit representation diagnostics across matched seeds. Each row is one run; w/o Patch disables the patch relation loss but retains an auxiliary embedding for evaluation.

		Auxiliary embedding		Binary code	
Variant	Seed	All	Different	All	Different
`SST`	42	0.9176	0.9069	0.6735	0.6617
`SST`	43	0.9192	0.9110	0.6581	0.6480
`SST`	44	0.9046	0.9039	0.6610	0.6543
w/o Geometry	42	0.8993	0.8985	0.5450	0.5345
w/o Geometry	43	0.9038	0.9031	0.6070	0.5985
w/o Geometry	44	0.8744	0.9002	0.5539	0.5413
w/o Patch	42	0.9014	0.8875	0.4755	0.4642
w/o Patch	43	0.8910	0.8912	0.4361	0.4236
w/o Patch	44	0.8983	0.9001	0.5306	0.5206

Table 4 reports Pearson correlations between pairwise representation similarities and pairwise cosine similarities of the original continuous class attributes for SUN at 128 bits. All runs use 4,340 unseen

queries. "All" includes all unordered pairs of distinct query images, while "Different" additionally excludes pairs whose labels are identical. Similarities are computed after normalizing auxiliary embeddings; for K-bit binary codes the normalized inner product equals $1 - 2d_H/K$. These are image-pair correlations, not correlations between learned class centers and the presence targets of the geometry loss.

Across seeds 42, 43, and 44, the full model has the highest auxiliary and binary-code correlations in both pair groups. Its auxiliary correlations over different-class pairs are 0.9039–0.9110, versus 0.8875–0.9001 for w/o Patch. The high correlations without patch supervision show that auxiliary attribute-related structure cannot be attributed solely to the patch loss. The full model's binary-code correlations over different-class pairs exceed w/o Patch by 0.1975, 0.2244, and 0.1337, respectively, and exceed w/o Geometry in every matched seed. Thus, stronger attribute agreement is also observed in the retrieval codes, although it remains below auxiliary-embedding agreement. These diagnostics neither establish local attribute correspondence nor a causal relationship between correlation and retrieval mAP.

Pairs share query images and are not independent experimental repetitions; their number cannot substitute for the number of training seeds.

A.10 Retrieval Results Across Random Seeds

Table 5: Additional retrieval results in mAP@5000. SD uses ddof=0; all rows have $n = 3$.

Setting	Variant	Mean	SD	Seed 42	Seed 43	Seed 44
AWA2-64	SST	0.6291	0.0005	0.6288	0.6298	0.6287
AWA2-64	w/o Patch	0.6209	0.0060	0.6285	0.6203	0.6138
AWA2-64	w/o Geometry	0.6177	0.0057	0.6196	0.6235	0.6099
AWA2-64	w/o both	0.6074	0.0054	0.6149	0.6050	0.6023
CUB-24	SST	0.1577	0.0003	0.1581	0.1578	0.1573
CUB-24	w/o Patch	0.1518	0.0033	0.1556	0.1521	0.1476
CUB-24	w/o Geometry	0.1521	0.0035	0.1571	0.1492	0.1501
CUB-24	w/o both	0.1381	0.0026	0.1397	0.1401	0.1345
CUB-64	SST	0.2530	0.0058	0.2547	0.2453	0.2591
CUB-64	w/o Patch	0.2413	0.0014	0.2405	0.2401	0.2433
CUB-64	w/o Geometry	0.2400	0.0037	0.2351	0.2408	0.2441
CUB-64	w/o both	0.2238	0.0034	0.2281	0.2198	0.2234
CUB-128	SST	0.3143	0.0065	0.3060	0.3150	0.3219
CUB-128	w/o Patch	0.2989	0.0086	0.2881	0.3091	0.2996
CUB-128	w/o Geometry	0.3094	0.0042	0.3035	0.3120	0.3127
CUB-128	w/o both	0.2761	0.0079	0.2704	0.2872	0.2706
SUN-64	SST	0.2092	0.0019	0.2097	0.2067	0.2112
SUN-64	w/o Patch	0.1936	0.0040	0.1986	0.1889	0.1932
SUN-64	w/o Geometry	0.1972	0.0070	0.1887	0.1971	0.2058
SUN-64	w/o both	0.1858	0.0062	0.1771	0.1911	0.1893
SUN-128	SST	0.2162	0.0006	0.2169	0.2163	0.2155
SUN-128	w/o Patch	0.2092	0.0068	0.2140	0.2141	0.1996
SUN-128	w/o Geometry	0.2037	0.0067	0.2088	0.2081	0.1942
SUN-128	w/o both	0.1931	0.0023	0.1949	0.1945	0.1898

For compactness, Table 5 reports six settings with seeds 42, 43, and 44 from broader experiments that consistently favor the full model over either or both loss removals. Each row summarizes three runs using population SD, $\sigma = \sqrt{\frac{1}{3}\sum_s (m_s - \bar{m})^2}$ (ddof=0), not a confidence interval. Seed 42 reproduces Tables 1 and 2; seeds 43 and 44 extend those comparisons.

Table 6: Full-model gain over the better single-loss variant within each seed. SD uses ddof=0.

Setting	Δ_{42}	Δ_{43}	Δ_{44}	Mean	SD
AWA2-64	+0.0003	+0.0063	+0.0149	+0.0072	0.0060
CUB-24	+0.0010	+0.0057	+0.0072	+0.0046	0.0026
CUB-64	+0.0142	+0.0045	+0.0150	+0.0112	0.0048
CUB-128	+0.0025	+0.0030	+0.0092	+0.0049	0.0030
SUN-64	+0.0111	+0.0096	+0.0054	+0.0087	0.0024
SUN-128	+0.0029	+0.0022	+0.0159	+0.0070	0.0063

For each seed, define $\Delta_s = m_s^{\text{full}} - \max(m_s^{\text{w/o Patch}}, m_s^{\text{w/o Geometry}})$. Table 6 summarizes these paired differences. The comparison selects the stronger single-loss variant separately within each seed, rather than subtracting independently averaged scores.

Table 7: Full-model gain over w/o both within each seed. SD uses ddof=0; each setting has three paired runs.

Setting	G_{42}	G_{43}	G_{44}	Mean	SD
AWA2-64	+0.0139	+0.0248	+0.0264	+0.0217	0.0056
CUB-24	+0.0184	+0.0177	+0.0228	+0.0196	0.0023
CUB-64	+0.0266	+0.0255	+0.0357	+0.0293	0.0046
CUB-128	+0.0356	+0.0278	+0.0513	+0.0382	0.0098
SUN-64	+0.0326	+0.0156	+0.0219	+0.0234	0.0070
SUN-128	+0.0220	+0.0218	+0.0257	+0.0232	0.0018

For comparison with removal of both losses, let $G_s = m_s^{\text{full}} - m_s^{\text{w/o both}}$.

The full model outperforms both single-loss variants and w/o both in all eighteen setting–seed comparisons, with the highest mean in all six settings. Together with the broader experiments, these results support complementary benefits across the evaluated settings and seeds, without establishing statistical significance.

A.11 Alternative Branch Inputs and Geometry Targets

Table 8: CUB and SUN design controls in mAP@5000 under matched settings at seed 42.

Variant	CUB-64	CUB-128	SUN-64	SUN-128
`SST`	0.2547	0.3060	0.2097	0.2169
CLS auxiliary input	0.2383	0.2925	0.1856	0.2011
Continuous geometry target	0.2351	0.2982	0.1902	0.1875

The CLS control replaces the patch-token sequence with one CLS key/value token, retaining the attribute-query branch's query/readout structure, d_a-dimensional output, and relation loss. This removes spatial selection and tests access to patch-level information. The continuous-target control retains patch cross-attention but replaces presence similarities with continuous-attribute cosine similarities in the same unseen–seen geometry MSE objective.

Table 8 compares CUB and SUN controls under matched settings at seed 42. Transfer weights $(\lambda_{\text{pat}}, \lambda_{\text{geo}})$ are $(2.5, 0.2)$ and $(6.0, 0.3)$ for CUB at 64 and 128 bits, and $(10.0, 0.08)$ and $(2.5, 0.3)$ for SUN, respectively. Only the indicated input or target changes; full-model scores reproduce the main table.

At 64 and 128 bits, the full model exceeds the CLS control by 0.0164 and 0.0135 on CUB, and 0.0241 and 0.0158 on SUN. Gains over continuous targets are 0.0196 and 0.0078 on CUB, and 0.0195 and 0.0294 on SUN. These runs support patch-token cross-attention and presence targets in all four settings, but do not establish local attribute correspondence or generalization across seeds or datasets.