# STAGE: CONSTRAINED GRAPH-PHASE REFINEMENT FOR BINARY HASHING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Learning binary hash codes for image retrieval requires preserving semantic relationships while enabling independent image encoding. However, jointly refined targets can vary with batch composition or exceed the encoder's output range, and lower refinement energy does not necessarily yield better retrieval. We propose STAGE, a constrained graph-phase framework that learns an independent hash encoder from jointly refined targets. In its minibatch formulation, a graph-phase teacher combines graph smoothness, semantic supervision, phase separation, and bit decorrelation under a zero-mean constraint. For a fixed graph and energy, learnable per-bit positive-definite spectral operators adapt refinement dynamics while preserving the constrained stationary points. To facilitate target transfer, a fixed-reference extension produces bounded, zero-mean targets, caches them by image identity, and trains the teacher using the student's ranking loss after a virtual update. We establish mean conservation, conditional energy descent, and a bound relating binary imbalance to quantization error. Experiments on CIFAR-10, CIFAR-100, and VOC 2007 using frozen ResNet-18 features and 8-, 16-, and 32-bit codes show that both formulations achieve the highest mean mAP among the compared methods in all nine settings of their respective protocols. Under the fixed-reference protocol, STAGE improves mean mAP over matched scalar refinement by 0.15–3.91 percentage points. Graph computations are confined to training, retaining independent encoding and standard Hamming search at inference.

## 1 INTRODUCTION

Similarity hashing learns compact binary codes for efficient Hamming search while preserving semantic relationships and limiting quantization error, bit imbalance, and redundancy. Existing approaches include discrete optimization and continuation (Shen et al., 2015; Cao et al., 2017), semantic target learning (Chen et al., 2025; Yin et al., 2026), and cross-view alignment on pretrained features (Moummad et al., 2026a). Joint target refinement offers explicit control over code properties, but the resulting targets must be learned by an independent encoder. In minibatch refinement, targets may vary with batch composition or lie outside the encoder's output range. Moreover, **lower refinement energy need not improve student retrieval**. These issues motivate considering both constrained target construction and the effectiveness of target transfer to the encoder.

We propose **STAGE**, which separates constrained target refinement from amortized encoding (Figure 1). Building on spectral hashing and graph diffuse-interface models (Weiss et al., 2008; Bertozzi & Flenner, 2012; van Gennip & Budd, 2025), **STAGE-Batch** refines continuous codes using graph smoothness, semantic supervision, a double-well potential, and decorrelation under a zero-mean constraint. With a limited refinement budget, the resulting targets depend on the update trajectory. We therefore learn per-bit positive-definite graph-polynomial operators to adapt updates across graph frequencies. For a fixed graph and energy, these operators preserve the constrained stationary points while changing the refinement dynamics. The encoder learns to match the resulting detached targets.

To support target transfer, **STAGE-Ref** combines a fixed reference graph with a bounded, zero-mean readout and identity-based caching. The readout places targets within the encoder's output range, while caching keeps them unchanged between refreshes. The teacher learns from the student's ranking loss after a virtual update, linking target construction to its effect on student retrieval. Graph



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.

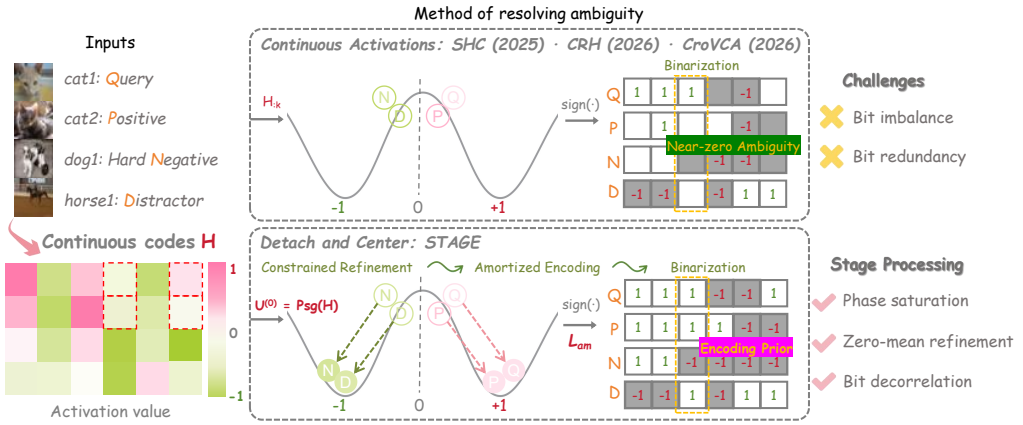



Figure 1: STAGE overview. Top: threshold sensitivity, bit imbalance, and redundancy. Bottom: a constrained graph-phase teacher generates training targets for an independent hash encoder. The ranking example is conceptual.

computations remain confined to training, preserving independent encoding and standard Hamming search at inference.

Experiments on CIFAR-10, CIFAR-100, and VOC 2007 use frozen ResNet-18 features and 8-, 16-, and 32-bit codes to evaluate retrieval and target transfer under separate batch and fixed-reference protocols. STAGE-Batch and STAGE-Ref achieve the highest mean mAP among the compared methods in all nine settings of their respective protocols. Controlled comparisons examine spectral updates, scalar gains, and direct regularization.

Our contributions are: ❶ Constrained graph-phase target refinement with learned per-bit spectral updates for independent hash encoding. ❷ A fixed-reference extension combining bounded, zero-mean targets, identity-based caching, and student feedback to support target transfer. ❸ Analysis establishing mean conservation, conditional energy descent, and a quantization–imbalance bound for teacher codes, supported by controlled evaluations of retrieval and target feasibility.

## 2 Related Work

**Hashing and target learning.** SHC learns semantic hash centers that preserve inter-class relationships (Chen et al., 2025), while CRH jointly learns the hash function and reassigns codebook centers (Yin et al., 2026). Broad Hashing combines Hadamard-based class targets with broad learning (Ng et al., 2025), and KALAHash incorporates class knowledge into low-resource adaptation and discrete optimization (Zhao et al., 2025). CroVCA aligns codes across semantic views on frozen embeddings (Moummad et al., 2026a), whereas Hashing-Baseline revisits PCA, random orthogonal projection, and thresholding on pretrained embeddings (Moummad et al., 2026b). Adaptive bit selection further addresses hashing scalability (Wang et al., 2025). Within the broader setting of code-target learning (Jeong & Song, 2018), STAGE jointly refines sample-level targets under explicit constraints and transfers them to an independent encoder that produces fixed-length binary codes.

**Distillation and student feedback.** Distillation transfers supervision between representations or models. Deep Hash Distillation encourages consistency across augmented views (Jang et al., 2022), while multiscale object-driven distillation transfers intermediate representations to compact hashing networks (Hussain et al., 2025). Meta-teacher learning and example reweighting further adapt supervision according to its effect on student performance (Pham et al., 2021; Ren et al., 2018). STAGE-Ref applies this feedback principle to constrained graph-phase targets: the student's ranking loss after a virtual update guides the teacher's refinement parameters, connecting target construction to the student's retrieval objective.

**Graph energies and spectral optimization.** Graph diffuse-interface methods combine graph smoothness with phase formation (Bertozzi & Flenner, 2012). Related developments include Allen–

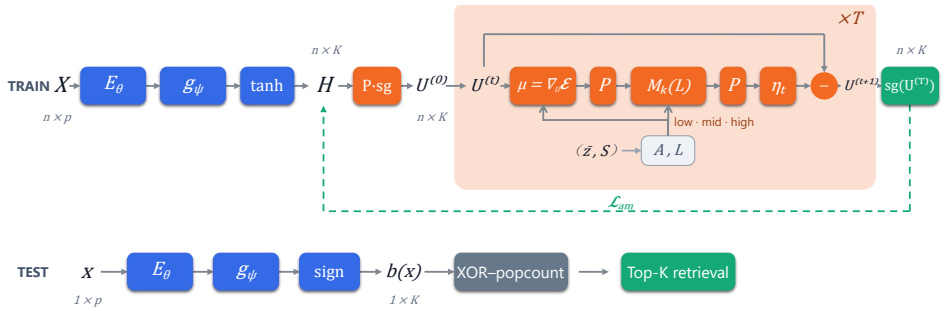



Figure 2: Batch graph-phase refinement backbone of STAGE. During training, the teacher refines student codes into targets for detached-target matching. At inference, images are independently encoded and binarized for Hamming search.

Cahn dynamics in ACMP (Wang et al., 2023b), graph-based image hashing in GHash and DGOH (Zhou et al., 2025; Cao et al., 2025), and predictive layers derived from a sampling-based energy in MuseGNN (Jiang et al., 2025). Our refinement operators also draw on polynomial spectral filtering (Defferrard et al., 2016) and connect to learned preconditioning. Graph Neural Preconditioners learn nonlinear inverse approximations for iterative solvers (Chen, 2025), while asymmetric learning addresses ill-conditioning in spectral GNNs (Liu & Wang, 2025). STAGE uses positive-definite graph-polynomial operators to control per-bit target refinement under a zero-mean constraint, with all graph computations confined to training.

## 3 Methodology

Given training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N_{\text{train}}}$, we learn $K$-bit binary codes $\mathbf{b}_{\theta,\psi}(x) = \operatorname{sign}(g_\psi(E_\theta(x))) \in \{-1, +1\}^K$, where $E_\theta$ is the feature extractor and $g_\psi$ is the hash head. Zero-valued entries are resolved using a fixed binary tie-breaking rule. For a minibatch $X$ of $n$ images, the continuous codes are

$$\mathbf{H} = \tanh(g_\psi(E_\theta(X))) \in (-1, 1)^{n\times K},$$

with rows corresponding to images and columns to hash bits. The similarity matrix $\mathbf{S} \in [0, 1]^{n\times n}$ encodes class agreement or multilabel overlap. For binary codes $\mathbf{b}_i$ and $\mathbf{b}_j$, the Hamming distance is $d_{\text{H}}(\mathbf{b}_i, \mathbf{b}_j) = (K - \mathbf{b}_i^\top \mathbf{b}_j)/2$. As illustrated in Figure 2, a graph-phase teacher refines detached student codes through constrained spectral updates, and the student learns to match the resulting targets. Section 3.4 extends this refinement backbone to a fixed-reference setting.

### 3.1 Constrained graph-phase teacher

For frozen image features $\bar{\mathbf{z}}_i$, we define affinities as $\rho S_{ij} + (1 - \rho)\exp((\cos(\bar{\mathbf{z}}_i, \bar{\mathbf{z}}_j) - 1)/\tau_g)$. Retaining the $r$ strongest neighbors per image, symmetrizing, and removing self-loops gives the adjacency matrix $\mathbf{A}$. Let $\mathbf{D}$ be diagonal with $D_{ii} = \sum_j A_{ij}$ and $d_{\max} = \max_i D_{ii}$. The scaled Laplacian $\mathbf{L} = (\mathbf{D} - \mathbf{A})/(2d_{\max} + \epsilon_0)$, $\epsilon_0 > 0$, satisfies $\mathbf{L}\mathbf{1}_n = \mathbf{0}$ and $\operatorname{spec}(\mathbf{L}) \subset [0, 1)$.

We refine codes on $\mathcal{C} = \{\mathbf{U} \in \mathbb{R}^{n\times K} : \mathbf{1}_n^\top \mathbf{U} = \mathbf{0}^\top\}$ using the projector $\mathbf{P} = \mathbf{I}_n - \mathbf{1}_n\mathbf{1}_n^\top/n$ and initialization $\mathbf{U}^{(0)} = \mathbf{P}\,\text{sg}(\mathbf{H})$, where sg stops gradients. The teacher energy is

$$\begin{aligned}\mathcal{E}(\mathbf{U}) = \frac{\lambda_g}{2nK}\operatorname{tr}(\mathbf{U}^\top\mathbf{L}\mathbf{U}) + \frac{\lambda_w}{4nK\varepsilon}\left\|\mathbf{U}\odot\mathbf{U} - \mathbf{1}_n\mathbf{1}_K^\top\right\|_F^2 \\ + \frac{\lambda_s}{|\Omega|}\sum_{(i,j)\in\Omega} w_{ij}\left[\operatorname{softplus}(\Theta_{ij}) - S_{ij}\Theta_{ij}\right] \\ + \frac{\lambda_o}{4K^2}\left\|\frac{\mathbf{U}^\top\mathbf{U}}{n} - \mathbf{I}_K\right\|_F^2,\end{aligned} \tag{1}$$

where $\Theta_{ij} = \mathbf{u}_i^\top\mathbf{u}_j/(K\tau_h)$, $\mathbf{u}_i^\top$ is row $i$ of $\mathbf{U}$, and $\Omega$ contains ordered sample pairs with weights $w_{ij}$. The terms encourage graph smoothness, values near $-1$ or $+1$, semantic agreement, and inter-bit

decorrelation. Only the zero-mean constraint is imposed exactly. With zero-mean columns, the orthogonality target $\mathbf{U}^\top\mathbf{U}/n = \mathbf{I}_K$ requires $K \le n-1$.

### 3.2 Spectral mobility and projected refinement

For each bit $k$, we learn a mobility operator that controls updates across graph frequencies. Let $\boldsymbol{\alpha}_k = \mathrm{softmax}(\mathbf{a}_k)$ for trainable logits $\mathbf{a}_k \in \mathbb{R}^3$, and define

$$\mathbf{M}_k(\mathbf{L}) = \delta\mathbf{I}_n + (1-\delta)\sum_{c\in\{\mathrm{low},\mathrm{mid},\mathrm{high}\}} \alpha_{kc} q_c(\mathbf{L}), \qquad 0 < \delta \le 1, \tag{2}$$

with $q_{\mathrm{low}}(\lambda) = 1-\lambda$, $q_{\mathrm{mid}}(\lambda) = 4\lambda(1-\lambda)$, and $q_{\mathrm{high}}(\lambda) = \lambda$. These responses give $\delta\mathbf{I}_n \preceq \mathbf{M}_k(\mathbf{L}) \preceq \mathbf{I}_n$ and $\mathbf{M}_k(\mathbf{L})\mathbf{P} = \mathbf{P}\mathbf{M}_k(\mathbf{L})$.

Writing $\boldsymbol{\mu}(\mathbf{U}) = \nabla_{\mathbf{U}}\mathcal{E}(\mathbf{U})$, the projected flow and discrete update are

$$\frac{\partial \mathbf{U}_{:k}}{\partial t} = -\mathbf{P}\mathbf{M}_k(\mathbf{L})\mathbf{P}\boldsymbol{\mu}_{:k}(\mathbf{U}), \tag{3a}$$

$$\mathbf{U}^{(t+1)}_{:k} = \mathbf{U}^{(t)}_{:k} - a_t\mathbf{P}\mathbf{M}_k(\mathbf{L})\mathbf{P}\boldsymbol{\mu}_{:k}(\mathbf{U}^{(t)}), \tag{3b}$$

where $a_t$ is the effective step after rescaling and backtracking. Armijo backtracking (Armijo, 1966) starts from $n\eta_t$, with $\eta_t = \eta_{\max}\sigma(s_t)$, trainable scalar $s_t$, and sigmoid function $\sigma$. If no trial is accepted, $a_t = 0$ and the state is retained.

Projection preserves zero bit means, while positive-definite mobility changes refinement trajectories without changing the constrained stationary points for a fixed graph and energy. Identity and learned per-bit scalar mobilities provide controls for frequency-dependent updates.

### 3.3 Amortized training

Let $w = (\theta, \psi)$ and $\phi = \{\mathbf{a}_k\}_{k=1}^K \cup \{s_t\}_{t=0}^{T-1}$ denote the encoder and teacher parameters. In all image experiments, $\theta$ is frozen; actual and virtual student updates optimize only $\psi$. The shared student objective $\mathcal{L}_{\mathrm{base}}(\mathbf{H}, \mathbf{S})$ combines pairwise semantic supervision with saturation, balance, and decorrelation penalties. The energy-driven base formulation uses

$$\mathcal{L}_{\mathrm{student}} = \mathcal{L}_{\mathrm{base}}(\mathbf{H}, \mathbf{S}) + \frac{\lambda_{\mathrm{am}}}{nK}\left\|\mathbf{H} - \mathrm{sg}(\mathbf{U}^{(T)})\right\|_F^2, \tag{4}$$

$$\mathcal{L}_{\mathrm{solver}} = \mathcal{E}(\mathbf{U}^{(T)}) + \lambda_{\mathrm{mono}}\sum_{t=0}^{T-1}\left[\mathcal{E}(\mathbf{U}^{(t+1)}) - \mathcal{E}(\mathbf{U}^{(t)})\right]_+^2, \tag{5}$$

where $[z]_+ = \max(z, 0)$.

**Algorithm 1** Energy-driven base training

**Require:** $\mathcal{D}$; $T$; initialized $w, \phi$

**for** each training batch $X$ **do**
  $\mathbf{H} \leftarrow \tanh(g_\psi(E_\theta(X)))$
  Construct $\mathbf{L}$ and define $\mathcal{E}$
  $\mathbf{U}^{(0)} \leftarrow \mathbf{P}\,\mathrm{sg}(\mathbf{H})$
  **for** $t = 0, \ldots, T-1$ **do**
    Select $a_t$ by backtracking
    Update $\mathbf{U}^{(t+1)}$ via (3b)
    $\mathbf{U}^{(t+1)} \leftarrow \mathbf{P}\mathbf{U}^{(t+1)}$
  **end for**
  $g_\phi \leftarrow \nabla_\phi\mathcal{L}_{\mathrm{solver}}$
  $g_w \leftarrow \nabla_w\mathcal{L}_{\mathrm{student}}$
  Update $\phi, w$ using $g_\phi, g_w$
**end for**
**return** $\mathbf{b}_{\theta,\psi}$

**Objectives and gradients.** The encoder matches detached targets, while the teacher minimizes terminal energy. The monotonicity penalty vanishes for nonincreasing steps and penalizes increases admitted by the backtracking tolerance. $\mathrm{sg}(\mathbf{H})$ blocks teacher-loss gradients to $w$; $\mathrm{sg}(\mathbf{U}^{(T)})$ blocks matching gradients to $\phi$. Both gradients are computed before either update.

**Numerical refinement.** The graph and energy remain fixed along each trajectory. Reprojection limits numerical drift in the zero-mean constraint.

**Computation.** Polynomial operators use products with $\mathbf{L}$ and $\mathbf{L}^2$, without eigendecomposition. Batch refinement uses dense matrices; only the encoder is retained for inference.

### 3.4 TRANSFER-AWARE REFINEMENT

Split the training subset into disjoint reference and outer-training sets $\mathcal{A}$ and $\mathcal{Q}$, excluding validation and final queries. Construct the reference graph $\mathbf{L}_{\mathcal{A}}$ and sampled semantic pairs once and keep them fixed. For $N = |\mathcal{A}|$, refine $\mathbf{U}^{(0)} = \mathbf{P}_{\mathcal{A}}\,\mathrm{sg}(\mathbf{H}_{\mathcal{A}})$ under the four-term reference energy $\mathcal{E}_{\mathcal{A}}$, where $\mathbf{P}_{\mathcal{A}} = \mathbf{I}_N - \mathbf{1}_N\mathbf{1}_N^\top/N$. Targets are cached by image identity and refreshed every $R = 8$ student updates.

**Bounded targets.** The readout maps refined codes into the encoder's output range while preserving zero column means:

$$Y_{ik} = a_y \tanh\left(\frac{U_{ik}^{(T)} - b_k}{\tau_y}\right), \qquad \sum_{i\in\mathcal{A}} Y_{ik} = 0, \qquad a_y = 0.99, \quad \tau_y = 0.5. \tag{6}$$

Bisection determines $b_k$ from the zero-sum condition. The resulting targets satisfy $|Y_{ik}| < a_y < 1$, although joint fitting by the shared encoder remains a learning task.

**Separating spectral shape from gain.** We normalize mobility on the mean-zero subspace:

$$c_k = \frac{\mathrm{tr}(\mathbf{P}_{\mathcal{A}}\mathbf{M}_k^0\mathbf{P}_{\mathcal{A}})}{N-1}, \qquad \widetilde{\mathbf{M}}_k = s_k\frac{\mathbf{M}_k^0}{c_k}, \qquad s_k = 0.05 + 0.95\sigma(\xi_k), \tag{7}$$

where $\mathbf{M}_k^0$ is (2) evaluated on $\mathbf{L}_{\mathcal{A}}$. The factor $\mathbf{M}_k^0/c_k$ has unit average response on the mean-zero subspace, separating spectral shape from gain $s_k$. The scalar control uses $s_k\mathbf{I}_N$, with matched step-size parameterization, readout, refresh schedule, and outer objective. The gain parameters $\{\xi_k\}_{k=1}^K$ are included in $\phi$; refinement uses initial trial steps $N\eta_t$ and the same backtracking rule.

**Student feedback.** Following student-feedback approaches (Pham et al., 2021; Ren et al., 2018), we replace $\mathcal{L}_{\mathrm{solver}}$ with $\mathcal{L}_{\mathrm{outer}}$ to learn $\phi$ through a virtual student update. At each refresh, retain the dependence of $\mathbf{Y}_\phi$ on $\phi$ and, for the next reference minibatch $\mathcal{B} \subseteq \mathcal{A}$, define

$$\ell_{\mathcal{B}}(w,\phi) = \mathcal{L}_{\mathrm{base}}(\mathbf{H}_{\mathcal{B}}, \mathbf{S}_{\mathcal{B}}) + \frac{\lambda_{\mathrm{am}}}{|\mathcal{B}|K}\,\|\mathbf{H}_{\mathcal{B}} - \mathbf{Y}_{\phi,\mathcal{B}}\|_F^2\,,$$

$$w^+(\phi) = \mathrm{AdamWStep}\big(w, \nabla_w\ell_{\mathcal{B}}; \mathcal{S}_w\big), \tag{8}$$

$$\mathcal{L}_{\mathrm{outer}}(\phi) = \mathcal{L}_{\mathrm{rank}}\big(w^+(\phi); \mathcal{Q}, \mathcal{A}\big). \tag{9}$$

Here $\mathcal{S}_w$ is the detached optimizer state, and the virtual step matches the actual AdamW update. The ranking loss uses queries $q \in \mathcal{Q}$ and relevant/irrelevant reference samples $p, n \in \mathcal{A}$. Only $\phi$ receives the outer gradient. The actual student uses detached targets computed *before* the teacher update and reuses them until the next refresh. Sampling, readout differentiation, the ranking loss, and meta-direct controls are detailed in Appendix A.2.

## 4 THEORETICAL PROPERTIES

We fix the graph, energy, and mobility along each refinement trajectory and write $\boldsymbol{\mu} = \nabla_{\mathbf{U}}\mathcal{E}(\mathbf{U})$ and $\mathbf{g}_k = \mathbf{P}\boldsymbol{\mu}_{:k}$.

**Mean conservation and energy dissipation.** For the base mobility (2), let $0 < \delta \le 1$ and $\mathbf{U}^{(0)} \in \mathcal{C}$. The flow (3a) and update (3b) preserve $\mathbf{1}_n^\top\mathbf{U} = \mathbf{0}^\top$ in exact arithmetic. Moreover,

$$\mathbf{P}\mathbf{M}_k(\mathbf{L})\mathbf{P}\boldsymbol{\mu}_{:k} = \mathbf{0} \quad\Longleftrightarrow\quad \mathbf{g}_k = \mathbf{0}, \tag{10}$$

$$\frac{d\mathcal{E}}{dt} = -\sum_{k=1}^K \mathbf{g}_k^\top\mathbf{M}_k(\mathbf{L})\mathbf{g}_k \le -\delta\|\mathbf{P}\boldsymbol{\mu}\|_F^2. \tag{11}$$

Thus, mobility preserves the constrained stationary points, and the continuous flow strictly decreases energy when the projected gradient is nonzero. Conservation and stationarity also hold for normalized mobility, which remains positive definite and commutes with the projector.

**Local discrete descent.** Assume that $\nabla_{\mathbf{U}}\mathcal{E}$ is $L_{\mathcal{E}}$-Lipschitz on the segment between successive iterates. Energy is nonincreasing for $0 < a_t < 2/L_{\mathcal{E}}$ under base mobility, or $0 < a_t < 2/(L_{\mathcal{E}} m_{\max})$ under normalized mobility, where $m_{\max} = \max_k s_k/c_k$. Here $a_t$ includes rescaling and backtracking; rejected trials give $a_t = 0$ and leave the state unchanged. The assumption is local because the double-well gradient is not globally Lipschitz on an unbounded domain.

**Binary imbalance.** For a mean-zero column $\mathbf{u} = \mathbf{U}_{:k}$, define $\mathbf{b} = \operatorname{sign}(\mathbf{u})$ using a fixed binary tie rule and $q_k = \|\mathbf{b} - \mathbf{u}\|_2^2/n$. Then

$$\frac{|\mathbf{1}_n^\top \mathbf{b}|}{n} \le \sqrt{q_k}. \tag{12}$$

Small quantization error and a zero continuous mean therefore imply small binary imbalance; a zero mean alone does not ensure balanced signs.

**Spectral interpretation.** At a constrained stationary point, neglect cross-bit coupling and assume that the fixed-bit Hessian restricted to the mean-zero subspace is positive definite and shares an orthonormal eigenbasis with $\mathbf{L}$. Each linearized error mode then satisfies

$$e_j(t) = \exp\big(-m_k(\lambda_j) h_j t\big) e_j(0),$$

where $\lambda_j$ and $h_j > 0$ are the corresponding Laplacian and Hessian eigenvalues, and $m_k(\lambda_j)$ is the mobility response. Mobility thus scales the modal decay rates under these assumptions. This interpretation motivates Figure 3 but does not establish global convergence of the coupled nonconvex dynamics.

**Target feasibility and transfer.** For each bit, the readout (6) admits a unique centering value $b_k$, ensuring $\mathbf{1}_N^\top \mathbf{Y} = \mathbf{0}^\top$ and $|Y_{ik}| < a_y < 1$. Energy descent applies to the internal state $\mathbf{U}$; the nonlinear readout may alter both its energy and signs. These properties alone do not guarantee improved student retrieval. Conditional transfer bounds relate ranking-surrogate improvement to update alignment and bound binary disagreement in terms of target margins and continuous matching error. Details and proofs are provided in Appendices A.1 and A.2.

## 5 Empirical Evaluation

We evaluate retrieval accuracy, target transfer, refinement dynamics, and training cost. Batch experiments evaluate retrieval with method-specific tuning; fixed-reference experiments examine target transfer and spectral shape under shared settings. The two protocols are analyzed separately. Details are provided in Appendices A.3–A.6.

### 5.1 Experimental Setup and Retrieval Results

**Setup.** We evaluate 8-, 16-, and 32-bit codes on CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and VOC 2007 (Everingham et al., 2007) using frozen ImageNet-1K ResNet-18 (He et al., 2016) features and a shared hash head. Training/database/final-query sizes are 5,000/50,000/1,000 for each CIFAR dataset and 3,511/5,011/4,952 for VOC; final queries are excluded from the databases. Methods are trained for 60 epochs with three seeds, with method-specific tuning and checkpoint selection using 500 validation queries per dataset. Matched controls include direct penalties $\mathcal{E}(\mathbf{H})$ and $\mathcal{E}(\mathbf{PH})$, isotropic refinement, and per-bit scalar refinement. CSQ (Yuan et al., 2020) is adapted to the same features and head. We report mean full-database Hamming mAP, with evaluation and implementation details in Appendix A.3. The final CIFAR queries were unused in the initial pilot, whereas VOC reuses the official test set accessed previously.

**Retrieval results.** STAGE (batch) achieves the highest mean mAP in all nine dataset–code-length settings (Table 1). It exceeds the strongest published baseline in each setting by 0.07–3.30 percentage points and outperforms both direct-energy controls throughout. Compared with per-bit scalar refinement, its mean gains range from 0.38 to 6.78 percentage points, with the largest improvement on 16-bit VOC 2007.

Table 1: Full-database Hamming mAP (%; three-seed means) under the batch protocol. CSQ* uses the shared frozen features and hash head. Bold and underlined values indicate the highest and second-highest means in each column, respectively.

| | CIFAR-10 | | | CIFAR-100 | | | VOC 2007 | | |
|---|---|---|---|---|---|---|---|---|---|
| Method / bits | 8 | 16 | 32 | 8 | 16 | 32 | 8 | 16 | 32 |
| *Published baselines* | | | | | | | | | |
| CSQ* (Yuan et al., 2020) | 70.71 | 73.28 | 74.75 | 7.05 | 13.78 | 21.42 | 65.39 | 75.02 | 78.07 |
| MDS (Wang et al., 2023a) | <u>71.25</u> | 74.37 | 76.08 | 9.01 | 15.11 | 23.42 | 65.29 | 75.66 | <u>79.07</u> |
| SHC (Chen et al., 2025) | 70.31 | <u>74.57</u> | <u>76.12</u> | 9.77 | <u>15.27</u> | 23.58 | 65.00 | 75.34 | 79.05 |
| CRH (Yin et al., 2026) | 71.22 | 74.30 | 75.92 | 8.52 | 14.03 | <u>24.08</u> | 66.09 | <u>76.31</u> | 78.87 |
| *Base model and direct regularization* | | | | | | | | | |
| Base model | 67.77 | 72.28 | 72.78 | 9.09 | 12.69 | 15.89 | 68.12 | 71.39 | 73.96 |
| Direct $\mathcal{E}(\mathbf{H})$ | 67.31 | 70.94 | 73.20 | 9.43 | 12.07 | 16.10 | <u>68.14</u> | 71.22 | 73.45 |
| Direct $\mathcal{E}(\mathbf{PH})$ | 69.69 | 70.21 | 73.61 | <u>9.78</u> | 12.51 | 16.14 | 67.45 | 69.92 | 74.36 |
| *Target refinement* | | | | | | | | | |
| Isotropic refinement | 69.29 | 72.12 | 73.76 | 9.61 | 12.07 | 19.45 | 65.29 | 70.47 | 77.03 |
| Per-bit scalar refinement | 67.65 | 72.33 | 73.52 | 9.46 | 12.99 | 18.85 | 65.39 | 69.77 | 77.29 |
| **STAGE** (batch) | **71.73** | **75.50** | **76.75** | **9.84** | **15.61** | **24.80** | **69.39** | **76.55** | **79.24** |

### 5.2 Transfer-Aware Refinement

Table 2 evaluates 8-, 16-, and 32-bit codes under a fixed-reference protocol, reserving 500 outer-training images per dataset. Reference, outer-training, validation, and final-query sets are disjoint. Methods share initialization, batch order, and checkpoint selection within each seed; meta methods additionally share outer-training data and the virtual-update procedure. Unlike the batch study, hyperparameters are fixed across methods. Training settings and additional controls are detailed in Appendices A.3.3 and A.3.4.

**Target feasibility.** Batch-refined targets exhibit context dependence, with 6.9%–56.1% of coordinates outside $[-1, 1]$. In contrast, the 8-bit fixed-reference study shows no range violations across 45 bounded-target probes, with a maximum absolute column mean of $2.53 \times 10^{-8}$. Identity-based caching keeps targets unchanged between refreshes, although fitting them remains the encoder's learning task. Detailed diagnostics appear in Appendix A.6.

**Retrieval results.** STAGE (full) achieves the highest mean mAP in all nine settings (Table 2), exceeding the strongest control in each setting by 0.12–0.71 percentage points. Its mean gains over meta-scalar refinement range from 0.15 to 3.91 percentage points. This scalar control shares the readout, refresh schedule, and outer supervision, while (7) separates spectral shape from gain. The largest differences occur on VOC 2007, where STAGE exceeds meta-scalar by 3.91, 2.94, and 2.14 percentage points at 8, 16, and 32 bits, respectively.

### 5.3 Spectral Dynamics

Across the four quadratic curvature profiles in Figure 3, learned mobility yields lower terminal energy than identity mobility and assigns stronger responses to low-curvature modes. These diagnostics illustrate the conditional spectral model in Section 4; they do not directly measure retrieval performance. Full settings appear in Appendix A.4.2.

### 5.4 Refinement Behavior and Training Cost

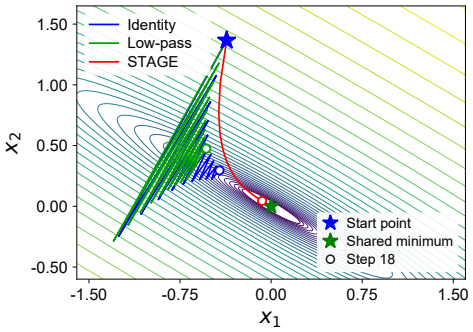
Figure 4 tracks four-step, 8-bit refinement on 192 fixed training images per dataset. Sign changes from $\mathbf{PH}$ to $\mathbf{U}^{(4)}$ generally become less frequent, while continuous energy still decreases late in

Table 2: Mean full-database Hamming mAP (%; three-seed means) under the fixed-reference protocol. Bold and underlined values indicate the highest and second-highest means in each column, respectively.

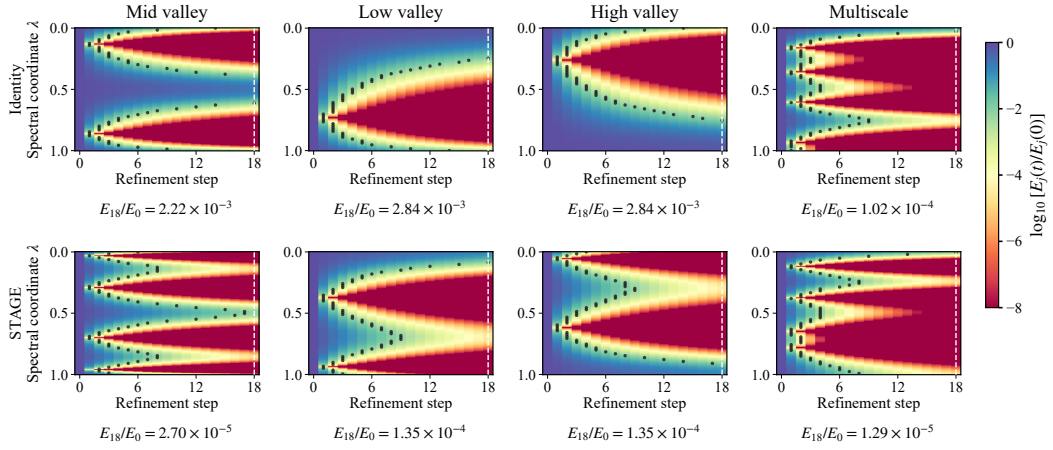
| | CIFAR-10 | | | CIFAR-100 | | | VOC 2007 | | |
|---|---|---|---|---|---|---|---|---|---|
| Method / bits | 8 | 16 | 32 | 8 | 16 | 32 | 8 | 16 | 32 |
| *Reference-set supervision* | | | | | | | | | |
| Reference base | 66.94 | 71.20 | 72.60 | 9.13 | 12.50 | 14.30 | <u>67.81</u> | <u>71.30</u> | <u>72.30</u> |
| Bounded targets ($T = 0$) | 69.78 | 72.90 | 73.74 | <u>9.70</u> | <u>12.93</u> | <u>14.59</u> | 64.97 | 69.17 | 70.60 |
| *Additional outer-training supervision* | | | | | | | | | |
| Meta-direct (raw) | 68.62 | 72.21 | 73.27 | 8.35 | 11.92 | 13.91 | 63.76 | 68.26 | 69.87 |
| Meta-direct (centered) | 66.29 | 70.81 | 72.34 | 7.96 | 11.62 | 13.72 | 65.05 | 69.23 | 70.64 |
| Meta-weighted targets | 69.20 | 72.56 | 73.50 | 9.30 | 12.63 | 14.39 | 65.20 | 69.34 | 70.73 |
| Spectral refinement + $\mathcal{Q}$ | <u>69.90</u> | <u>72.98</u> | <u>73.78</u> | 9.20 | 12.55 | 14.34 | 65.00 | 69.19 | 70.61 |
| Meta-scalar | 69.89 | 72.97 | <u>73.78</u> | 9.18 | 12.54 | 14.33 | 64.44 | 68.77 | 70.28 |
| **STAGE** (full) | **70.27** | **73.20** | **73.93** | **9.98** | **13.05** | **15.30** | **68.35** | **71.71** | **72.42** |

**Experimental setup.** We study four diagonal quadratic objectives with 256 spectral modes, unit initial amplitudes, and $\delta = 0.05$. Each method selects its scalar step size to minimize energy after 18 updates.

**Optimization trajectories.** From the same initial state, learned mobility approaches the shared minimum with less oscillation in the displayed modes. For the mid-valley profile, it reduces the effective condition number from 50.96 to 13.65 and the energy ratio $E_{18}/E_0$ from $2.22 \times 10^{-3}$ to $2.70 \times 10^{-5}$ relative to identity mobility.




**(a)** Mid-valley trajectories




**(b)** Spectral-mode decay across four curvature profiles

Figure 3: Controlled spectral dynamics: (a) two-mode trajectory projections; (b) modal energy $\log_{10}[E_j(t)/E_j(0)]$, clipped to $[-8, 0]$, for identity (top) and learned (bottom) mobility. Black dots mark the first step reaching 1% of initial modal energy for sampled modes; dashed lines mark step 18.

training under the shared Armijo safeguard; method ordering varies across datasets and epochs. This

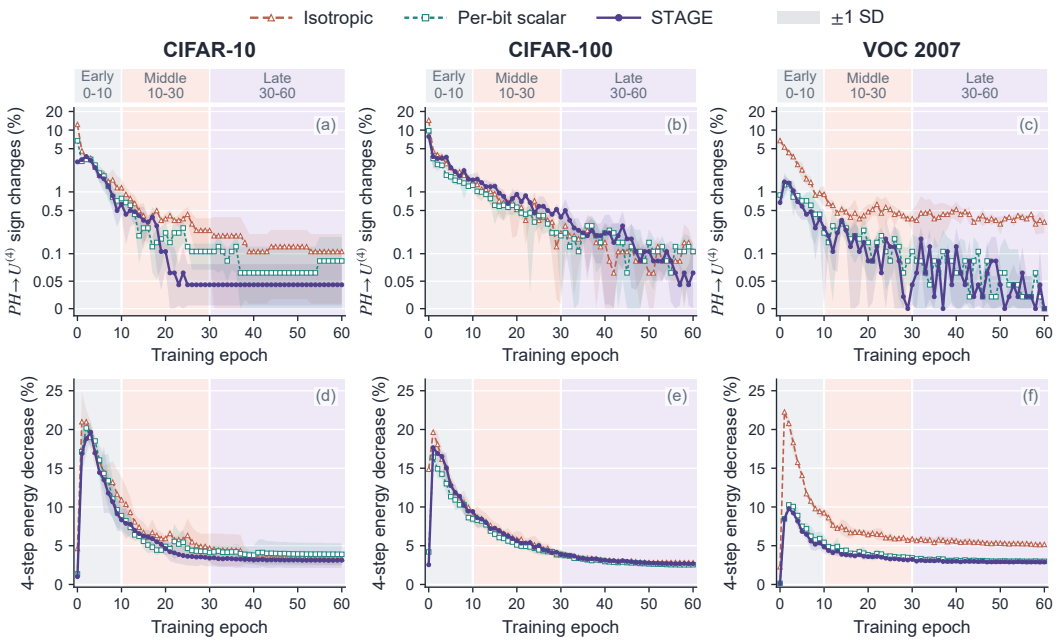



Figure 4: Four-step refinement with 8-bit codes on 192 fixed training images per dataset: (a–c) sign changes from $\mathbf{PH}$ to $\mathbf{U}^{(4)}$ (%); (d–f) relative energy decrease (%). The sign-change y-axes are linear up to 0.1% and logarithmic above. Curves show three-seed means; bands indicate $\pm 1$ sample SD, with lower bounds clipped at zero.

refinement incurs additional training cost. In 16-bit probes, STAGE (batch) takes 7.3–7.6 times as long per update as the base model. Under a nominal 30-second budget for 8-bit codes, batch STAGE achieves 72.06% and 70.34% mAP on CIFAR-10 and VOC 2007, respectively, compared with 67.82% and 69.16% for the base model. Over 60 epochs, fixed-reference STAGE takes 2.31–2.39 times as long as the reference base and approximately 1.10 times as long as meta-scalar refinement; equal-time retrieval has not been evaluated for this protocol. Selected-checkpoint and centering diagnostics appear in Appendix A.6, with timing details in Appendix A.6.3.

## 6 Discussion and Conclusion

We presented STAGE, a constrained graph-phase framework that learns an independent binary hash encoder from jointly refined targets. Learnable per-bit spectral operators adapt refinement dynamics while preserving the constrained stationary points for a fixed graph and energy. Our analysis establishes mean conservation and conditional energy descent, and bounds binary imbalance in teacher codes through quantization error. The fixed-reference extension addresses target range and consistency through a bounded, zero-mean readout and identity-based caching, while using the student's ranking loss after a virtual update to guide teacher learning.

Experiments on CIFAR-10, CIFAR-100, and VOC 2007 with frozen ResNet-18 features show that STAGE-Batch achieves the highest mean mAP among the compared methods across all nine dataset–code-length settings. These results support constrained target refinement as a useful source of supervision for binary hashing, with graph computations confined to training and independent encoding retained for standard Hamming search. Future work will investigate more efficient refinement and evaluate the framework with end-to-end feature learning, stronger baselines, and larger retrieval databases.

### AI use statement

Generative AI was used solely to polish the language in selected parts of the manuscript, and the authors take full responsibility for the final content.

## REFERENCES


S. M. Allen and J. W. Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27(6):1085–1095, 1979.

Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966. doi: 10.2140/pjm.1966.16.1.

A. L. Bertozzi and A. Flenner. Diffuse interface models on graphs for classification of high-dimensional data. *Multiscale Modeling and Simulation*, 10(3):1090–1118, 2012.

Yuan Cao, Xiangru Chen, Zifan Liu, Wenzhe Jia, Fanlei Meng, and Jie Gui. Deep graph online hashing for multi-label image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1953–1961, 2025. doi: 10.1609/aaai.v39i2.32191.

Z. Cao, M. Long, J. Wang, and P. S. Yu. HashNet: Deep learning to hash by continuation. In *ICCV*, 2017.

J. Chen. Graph neural preconditioners for iterative solutions of sparse linear systems. In *ICLR*, 2025.

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*, 2025. doi: 10.1145/3749983.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html.

D. Dua and C. Graff. UCI Machine Learning Repository. University of California, Irvine, 2019.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes challenge 2007 (VOC2007) results. PASCAL VOC Challenge, 2007. URL https://www.robots.ox.ac.uk/~vgg/projects/pascal/VOC/voc2007/.

Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, 2011.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Abid Hussain, Heng-Chao Li, Muqadar Ali, Mehboob Hussain, Ali Choo, Danish Ali, and Amir Rehman. Deep supervised hashing with multiscale object-driven knowledge distillation for image retrieval. *International Journal of Multimedia Information Retrieval*, 14:37, 2025. doi: 10.1007/s13735-025-00380-w.

Young Kyun Jang, Geonmo Gu, Byungsoo Ko, Isaac Kang, and Nam Ik Cho. Deep hash distillation for image retrieval. In *Computer Vision – ECCV 2022*, pp. 354–371. Springer, 2022. doi: 10.1007/978-3-031-19781-9_21.

Y. Jeong and H. O. Song. Efficient end-to-end learning for quantizable representations. In *ICML*, 2018.

H. Jiang, R. Liu, Z. Huang, Y. Wang, X. Yan, Z. Cai, M. Wang, and D. Wipf. MuseGNN: Forming scalable, convergent GNN layers that minimize a sampling-based energy. In *ICLR*, 2025.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Fangbing Liu and Qing Wang. Asymmetric learning for spectral graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18798–18806, 2025. doi: 10.1609/aaai.v39i18.34069.

Ilyass Moummad, Kawtar Zaher, Hervé Goëau, and Alexis Joly. Image hashing via cross-view code alignment in the age of foundation models. In *CVPR Workshops*, pp. 3589–3598, 2026a.

Ilyass Moummad, Kawtar Zaher, Lukas Rauch, and Alexis Joly. Hashing-Baseline: Rethinking hashing in the age of pretrained models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2026b. URL https://www.cmsworkshops.com/ICASSP2026/view_paper.php?PaperNum=14268&bare=1.

Wing W. Y. Ng, Xuyu Liu, Xing Tian, Ting Wang, Jianjun Zhang, and C. L. Philip Chen. Broad hashing for image retrieval. *Neurocomputing*, 636:130031, 2025. doi: 10.1016/j.neucom.2025.130031.

H. Pham, Z. Dai, Q. Xie, and Q. V. Le. Meta Pseudo Labels. In *CVPR*, pp. 11557–11568, 2021.

M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to Reweight Examples for Robust Deep Learning. In *ICML*, volume 80 of *PMLR*, pp. 4334–4343, 2018.

Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. Supervised discrete hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 37–45, 2015.

Yves van Gennip and Jeremy Budd. *A Prolegomenon to Differential Equations and Variational Methods on Graphs*. Elements in Non-local Data Interactions: Foundations and Applications. Cambridge University Press, 2025. doi: 10.1017/9781009346641. URL https://doi.org/10.1017/9781009346641.

Liangdao Wang, Yan Pan, Cong Liu, Hanjiang Lai, Jian Yin, and Ye Liu. Deep hashing with minimal-distance-separated hash centers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23455–23464, 2023a.

Min Wang, Wengang Zhou, Xin Yao, and Houqiang Li. Adaptive bit selection for scalable deep hashing. *IEEE Transactions on Image Processing*, 34:1048–1059, 2025. doi: 10.1109/TIP.2025.3533215.

Y. Wang, K. Yi, X. Liu, Y. G. Wang, and S. Jin. ACMP: Allen-Cahn message passing with attractive and repulsive forces for graph neural networks. In *ICLR*, 2023b.

Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NeurIPS*, 2008.

Shuo Yin, Zhiyuan Yin, Yuqing Hou, Rui Liu, Yong Chen, and Dell Zhang. Codebook-centric deep hashing: End-to-end joint learning of semantic hash centers and neural hash function. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(14):12018–12026, 2026. doi: 10.1609/aaai.v40i14.38190.

L. Yuan, T. Wang, X. Zhang, F. E. H. Tay, Z. Jie, W. Liu, and J. Feng. Central similarity quantization for efficient image and video retrieval. In *CVPR*, 2020.

Shu Zhao, Tan Yu, Xiaoshuai Hao, Wenchao Ma, and Vijaykrishnan Narayanan. KALAHash: Knowledge-anchored low-resource adaptation for deep hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10465–10473, 2025. doi: 10.1609/aaai.v39i10.33136.

Xudong Zhou, Jun Tang, Ke Wang, Nian Wang, and Han Chen. Graph hashing network for image retrieval. *Image and Vision Computing*, 162:105677, 2025. doi: 10.1016/j.imavis.2025.105677.

# A Appendix

## A.1 Proofs and Energy Gradients

We establish the properties in Section 4 for a fixed graph, energy, and mobility along each teacher trajectory.

### A.1.1 Conservation, stationarity, and dissipation

The symmetric Laplacian satisfies $L\mathbf{1} = 0$ and $\sigma(L) \subset [0, 1)$. Hence $\delta I \preceq M_k(L) \preceq I$ and $M_k P = P M_k$. For $g_k = P\mu_{:k}$, positive definiteness gives $P M_k P \mu_{:k} = M_k g_k = 0$ if and only if $g_k = 0$, proving the constrained stationary-point equivalence stated in Section 4. Moreover,

$$\frac{d}{dt}(\mathbf{1}^\top U_{:k}) = -\mathbf{1}^\top P M_k P \mu_{:k} = 0, \tag{13}$$

$$\frac{d\mathcal{E}}{dt} = -\sum_k g_k^\top M_k g_k \le -\delta \|P\mu\|_F^2. \tag{14}$$

The discrete update has the same zero-sum increment. Reprojection limits numerical drift; continuous dissipation is strict away from the constrained stationary set.

### A.1.2 Discrete descent and binary balance

Let $D_{:k} = M_k g_k$. Using $\|M_k g_k\|_2^2 \le g_k^\top M_k g_k$, the local descent lemma gives

$$\mathcal{E}(U - aD) - \mathcal{E}(U) \le -a(1 - aL_{\mathcal{E}}/2) \sum_k g_k^\top M_k g_k \le 0 \quad \text{for } 0 < a < 2/L_{\mathcal{E}}. \tag{15}$$

Here $a$ includes time rescaling and backtracking. Normalized mobility instead requires $a < 2/(L_{\mathcal{E}} m_{\max})$, with $m_{\max} = \max_k s_k/c_k$. The Lipschitz assumption is local because the quartic well has an unbounded Hessian on an unbounded domain. This is an exact-arithmetic descent statement under the stated step-size condition; the numerical acceptance tolerance is specified in Appendix A.3.2.

To prove the binary-imbalance bound in Section 4, $\mathbf{1}^\top u = 0$ implies $\mathbf{1}^\top b = \mathbf{1}^\top (b - u)$. Cauchy–Schwarz yields $|\mathbf{1}^\top b|/n \le \|b - u\|_2/\sqrt{n} = \sqrt{q_k}$. The bound holds under any fixed binary tie rule, with tied coordinates included in $q_k$.

### A.1.3 Conditional spectral rates

**Conditional spectral rate control.** Consider a fixed-bit linearization at a stationary point, neglecting cross-bit Hessian blocks. On the mean-zero subspace, assume a positive-definite Hessian block sharing an orthonormal eigenbasis with $L$. Then

$$e_j(t) = e^{-m_k(\lambda_j)h_j t} e_j(0), \qquad \kappa_{\text{eff},k} = \frac{\max_j m_k(\lambda_j)h_j}{\min_j m_k(\lambda_j)h_j}, \tag{16}$$

where $\lambda_j$, $h_j$, and $m_k(\lambda_j)$ are the Laplacian eigenvalue, Hessian curvature, and mobility response. Explicit Euler contracts every mode when $0 < a < 2/\max_j m_k(\lambda_j)h_j$. *Proof.* On the constrained subspace, $P$ is the identity and the linearization is $\dot{e} = -M_k H_* e$. Expansion in the shared eigenbasis gives $\dot{e}_j = -m_k(\lambda_j)h_j e_j$ and the Euler multiplier $1 - a m_k(\lambda_j)h_j$, proving both statements. These assumptions specify the quadratic interpretation used in Figure 3.

Global projection conserves total bit mass, not mass within each connected component. The latter would require $I - C(C^\top C)^{-1}C^\top$ for component indicators $C$.

### A.1.4 Energy gradients

Let $C = U^\top U/n - I_K$ and $R_{ij} = w_{ij}[\sigma(\Theta_{ij}) - S_{ij}]$ on ordered pairs in $\Omega$, with zero entries elsewhere. The gradients of the four unweighted terms are

$$G_g = \frac{LU}{nK}, \qquad G_w = \frac{U \odot U \odot U - U}{nK\varepsilon}, \tag{17}$$

$$G_s = \frac{(R + R^\top)U}{|\Omega| K \tau_h}, \tag{18}$$

$$G_o = \frac{UC}{nK^2}. \tag{19}$$

Each step uses $\mu(U) = \lambda_g G_g + \lambda_w G_w + \lambda_s G_s + \lambda_o G_o$. The quartic term follows Allen & Cahn (1979). For sampling with replacement, repeated occurrences contribute additively to $R$. In Equation (18), $w_{ij}$ is the coefficient before the explicit division by $|\Omega|$; the effective coefficient of each pair is $w_{ij}/|\Omega|$. The reference energy below instead absorbs pair normalization into its weights and does not apply another division by the number of sampled pairs.

**Numerical check.** In float64 ($n = 17$, $K = 7$, seed 2027), the relative gradient error is $1.45 \times 10^{-16}$ and projected column-sum residual is $1.39 \times 10^{-17}$. The measured mobility spectrum $[0.1492, 0.9153]$ lies within the prescribed interval $[0.05, 1]$.

## A.2 Fixed-Reference Refinement and Transfer Bounds

We give implementation details and proofs for Section 3.4.

### A.2.1 Reference energy and cached targets

For the disjoint training sets $A, Q$, let $N = |A|$ and $P_A = I - \mathbf{1}\mathbf{1}^\top/N$. Construct $L_A$ once with the original affinity, 12-neighbor sparsification, symmetrization, and scaling. The reference energy uses the same graph, well, and decorrelation terms as the batch energy, replacing $n, L$ by $N, L_A$. Its semantic term is

$$\mathcal{E}_{A,s}(U) = \lambda_s \sum_{(i,j)\in\Omega_A} w_{ij}[\mathrm{softplus}(\Theta_{ij}) - S_{ij}\Theta_{ij}], \qquad \Theta_{ij} = u_i^\top u_j/(K\tau_h). \tag{20}$$

The fixed sample contains 16,384 positive and 16,384 negative ordered pairs drawn with replacement. Each occurrence has weight $1/32768$; duplicates retain multiplicity. These weights already normalize the term: there is no further division by $|\Omega_A|$. Its unweighted gradient is $(R_A + R_A^\top)U/(K\tau_h)$, where $(R_A)_{ij}$ sums $w_{ij}[\sigma(\Theta_{ij}) - S_{ij}]$ over occurrences of $(i, j)$. Controls share the graph and pairs. Initialize $U^{(0)} = P_A \,\mathrm{sg}(H_A)$ and cache targets by image identity for $R = 8$ student updates; they can change at the next refresh.

### A.2.2 Gain-normalized spectral mobility

For the mobility in Section 3.4, $c_k = \mathrm{tr}(P_A M_k^0 P_A)/(N-1)$ and $\widetilde{M}_k = s_k M_k^0/c_k$, with $s_k = .05 + .95\sigma(\xi_k)$. The scalar control uses $s_k I$ with identical gain initialization, learned steps, readout, refresh schedule, and outer objective; the isotropic control uses $I$ with learned steps. The required traces are

$$\begin{aligned}
\mathrm{tr}(P_A q_{\mathrm{low}}(L_A) P_A) &= N - 1 - \mathrm{tr}(L_A), \\
\mathrm{tr}(P_A q_{\mathrm{mid}}(L_A) P_A) &= 4[\mathrm{tr}(L_A) - \mathrm{tr}(L_A^2)], \\
\mathrm{tr}(P_A q_{\mathrm{high}}(L_A) P_A) &= \mathrm{tr}(L_A).
\end{aligned} \tag{21}$$

Graph traces are computed once. The constant shape belongs to the softmax family's closure: $(\alpha_l, \alpha_m, \alpha_h) = (1/2, 0, 1/2)$ gives $M_k^0/c_k = I$. Since $\delta I \preceq M_k^0 \preceq I$, we have $\delta \le c_k \le 1$ and $(s_k\delta/c_k)I \preceq \widetilde{M}_k \preceq (s_k/c_k)I$. Positive definiteness and commutation with $P_A$ preserve mass and constrained stationary points. The descent lemma with $\|\widetilde{M}_k g_k\|^2 \le m_{\max} g_k^\top \widetilde{M}_k g_k$ gives $0 < a < 2/(L_{\mathcal{E}} m_{\max})$, where $m_{\max} = \max_k s_k/c_k$. Effective steps include the factor $N$ and Armijo backtracking.

### A.2.3 BOUNDED READOUT AND ITS DERIVATIVE

Write $\mathcal{R}(U)_{ik} = a_y \tanh((U_{ik} - b_k)/\tau_y)$, with $a_y = .99$, $\tau_y = .5$, and $\sum_i \mathcal{R}(U)_{ik} = 0$. The column sum is continuous and strictly decreasing in $b_k$, with limits $Na_y$ and $-Na_y$; hence the root is unique. Thus $Y = \mathcal{R}(U^{(T)})$ has $|Y_{ik}| < a_y < 1$ and $\mathbf{1}^\top Y = 0$ in exact arithmetic. Coordinate feasibility does not ensure that a shared head fits all targets. For $v_i = (a_y/\tau_y)\operatorname{sech}^2((U_i - b)/\tau_y)$, implicit differentiation gives

$$\frac{\partial Y}{\partial U} = \operatorname{diag}(v) - \frac{vv^\top}{\mathbf{1}^\top v}. \tag{22}$$

The root uses 40 bisection iterations on $[\min_i U_i - 12, \max_i U_i + 12]$; outer differentiation uses this Jacobian. Energy descent concerns internal $U$; readout can change both energy and signs.

### A.2.4 STUDENT FEEDBACK AND UPDATE ORDER

The matching loss and virtual AdamW step are defined in Section 3.4. Teacher initialization is detached from $w$, while $Y_\phi$ retains derivatives to $\phi$. The detached optimizer state supplies the moments; the virtual step matches the actual learning rate, bias correction, weight decay, and clipping. A $10^{-30}$ square-root stabilizer permits differentiation at zero. For 128 fixed label-valid triplets per refresh, $q \in Q$ and $p, n \in A$, define $\widehat{H} = \tanh(\operatorname{atanh}(H)/\tau_b)$ and $d(q,p) = [1 - \widehat{H}(q)^\top \widehat{H}(p)/K]/2$. Then

$$\mathcal{L}_{\text{outer}}(\phi) = \frac{1}{|\mathcal{T}|} \sum_{(q,p,n)\in\mathcal{T}} \operatorname{softplus}\left(\frac{d_{w^+(\phi)}(q,p) - d_{w^+(\phi)}(q,n) + m}{\tau_r}\right), \tag{23}$$

with $\tau_b = .5$, $m = .1$, $\tau_r = .1$, and inverse-tanh inputs clamped to $[-1 + 10^{-6}, 1 - 10^{-6}]$. Only $\phi$ receives the outer gradient. The actual student uses detached targets computed before the teacher update for the current and seven further updates; updated teacher parameters enter the next refresh. Meta-direct controls instead use $\gamma\mathcal{E}_A(H_A)$ or $\gamma\mathcal{E}_A(P_A H_A)$, learning $\gamma = 4\sigma(\zeta)$, initialized at .5, through the same outer data and virtual optimizer. Their actual direct energy is evaluated every student update. The feedback construction follows Pham et al. (2021); Ren et al. (2018).

### A.2.5 CONDITIONAL TRANSFER BOUNDS

**Student utility.** Let $F$ be a differentiable ranking surrogate with $L_F$-Lipschitz gradient on the relevant segment, rather than discrete Hamming mAP. From the same parameters and detached AdamW state, compare $w_b = \text{AdamWStep}(w, g_0; \mathcal{S}_w)$ and $w_+ = \text{AdamWStep}(w, g_0 + \lambda g_T; \mathcal{S}_w)$, where $g_0, g_T$ are the base and matching gradients evaluated at $w$. Both steps use the same clipping, moment update, bias correction, weight decay, and learning rate. For $\Delta = w_+ - w_b$, the descent lemma yields

$$F(w_+) - F(w_b) \le \langle \nabla F(w_b), \Delta\rangle + \frac{L_F}{2}\|\Delta\|^2. \tag{24}$$

For SGD with learning rate $\beta$, $\Delta = -\beta\lambda g_T$. Utility therefore depends on update alignment, which teacher energy alone does not impose.

**Binary agreement.** For $H, Y \in \mathbb{R}^{N\times K}$, a common binary tie rule, and $\gamma > 0$, the fraction of disagreeing signs satisfies

$$e_{\text{bin}} \le \frac{1}{NK}\sum_{i,k} \mathbf{1}\{|Y_{ik}| < \gamma\} + \frac{\|H - Y\|_F^2}{NK\gamma^2}. \tag{25}$$

Indeed, a disagreement with $|Y_{ik}| \ge \gamma$ requires $|H_{ik} - Y_{ik}| \ge \gamma$; count small-margin coordinates separately and sum squared errors. Neither bound guarantees discrete retrieval accuracy or generalization.

## A.3 EXPERIMENTAL SETTINGS

### A.3.1 DATA, FEATURES, AND EVALUATION

Image split sizes are given in Section 5.1. With split seed 20270905, each CIFAR training set is stratified from the official 50,000-image database. Per-class test permutations reserve 500 validation

queries and 1,000 final queries in a disjoint block unused by the initial pilot; validation uses 10,000 fixed database items. VOC trainval contains 3,511 training images, 500 validation queries, and 1,000 database-only images. Validation excludes its own queries (4,511 database images); all 4,952 official test images are final queries. Training images may occur in the database; validation queries do not train the head and final queries never occur in the database. The initial pilot also accessed the VOC test set.

**Features and head.** All image methods use frozen, evaluation-mode torchvision ResNet-18 IMAGENET1K_V1 features: bilinear resize to short side 256, center crop 224, ImageNet normalization, and 512-dimensional pre-classifier outputs. Training-only standardization uses an SD floor of .01. The head is Linear(512,128)–LayerNorm–GELU–Linear(128,128)–GELU–Linear(128,$K$)–tanh.

**Retrieval and uncertainty.** Images are encoded independently, thresholded once, and ranked by Hamming distance with fixed database-index tie breaking. Relevance is class equality on CIFAR and any shared VOC object class, including XML objects marked difficult. Full-database mAP averages precision at relevant ranks and then over queries. Means and sample SDs use seeds 1, 2, 3 on one split. A seed-paired 95% interval would require the individual seed differences and $t_{.975,2} = 4.30265$; marginal means and SDs alone do not determine it. Such an interval would describe seed variation, not independent query replicates. The fixed-reference comparisons in Appendix A.5.2 report descriptive differences of displayed means, without paired confidence intervals.

### A.3.2 Batch-teacher training and validation selection

**Shared losses and graph.** Matched controls share balanced pairwise BCE and penalties $.2\,\mathrm{mean}[(H^2-1)^2]$, $.05\,\mathrm{mean}[(H^\top H/n - I)^2]$, and $.05\,\mathrm{mean}[(\mathbf{1}^\top H/n)^2]$. Positive and negative ordered non-self pairs each receive total BCE weight $1/2$ when present. Teacher-energy normalization follows Appendix A.1.4. Direct graph adds graph energy; projection and refinement use detached-target MSE. Affinity mixing is .5, both temperatures are .2, and 12 directed neighbors are symmetrized without self-loops. The Laplacian is $L = (D - A)/(2d_{\max} + 10^{-6})$. Teacher weights are $(.2, .4, 1, .05)$ for graph, well, semantic, and decorrelation terms, with $\varepsilon = .25$, $\delta = .05$, and $T = 4$. Batch experiments use dense matrices and all ordered pairs.

**Safeguard and optimizers.** For $d_{:k} = nPM_kP\mu_{:k}$, a finite $U' = P(U - \eta d)$ is accepted if

$$\mathcal{E}(U') \le \mathcal{E}(U) - 10^{-4}\eta\langle\mu, d\rangle + 10^{-7}\max(1, |\mathcal{E}(U)|).$$

Otherwise halve the step up to 12 times, retaining $U$ if no trial is accepted. Differentiation follows the accepted branch. The additive tolerance can admit a small energy increase; rejection need not mean stationarity. Student AdamW uses weight decay $10^{-5}$ and cosine decay with a 5% floor; solver Adam uses .03, clipping is 5, and the monotonicity coefficient is 10. Batches contain 192 images; remainders below 96 merge with the preceding batch, giving 26/26/18 updates per epoch and $K \le n - 1$. Student and solver have separate gradient paths.

**Selection and controls.** Calibration uses seed 900 and 30 epochs: learning rates $\{.0005, .002\}$, graph weights $\{.2, 2, 20\}$, matching weights $\{.5, 2\}$, and CSQ quantization weights $\{.001, .05\}$. Teachers select rate/weight at cap 2, then compare caps .5 and 8 at that pair. This gives six candidates for each graph/teacher method, four for projection/CSQ, and two for the base: 306 validation-only runs. Table 3 gives selected configurations. Final runs last 60 epochs; validation at epochs 10,20,. . .,60 selects one checkpoint, preferring earlier ties, before final-query evaluation. Tuning never computes test metrics.

**Adapted baselines.** CSQ uses center BCE on $(H + 1)/2$ and absolute-value quantization. Centers start from Hadamard rows and their negatives, with distinct Bernoulli centers if there are more than $2K$ classes. Multilabel centers use majority signs and a fixed random tie vector per seed. Short-code centers and the frozen-feature MLP are adaptations, not replications of published end-to-end CSQ scores. ITQ (Gong & Lazebnik, 2011) uses training-feature full-SVD PCA and 50 Procrustes rotations.

Table 3: Selected batch settings. Graph/Proj. report auxiliary weights; Iso./Scalar/Batch report matching weight / step cap. Their student learning rate is .002 and the base uses no auxiliary term. CSQ reports learning rate ($\times 1000$) / quantization weight. Initial steps are .2 at cap .5 and .8 otherwise. Final seeds share these settings.

| Data / bits | Graph | Proj. | Iso. | Scalar | Batch | CSQ |
|---|---|---|---|---|---|---|
| C10 / 8 | 0.2 | 2 | 2/8 | 2/8 | 2/8 | 0.5/0.001 |
| C10 / 16 | 0.2 | 0.5 | 0.5/0.5 | 0.5/0.5 | 0.5/0.5 | 0.5/0.05 |
| C10 / 32 | 0.2 | 0.5 | 0.5/0.5 | 0.5/0.5 | 0.5/2 | 0.5/0.001 |
| C100 / 8 | 0.2 | 2 | 2/8 | 2/2 | 2/8 | 2/0.05 |
| C100 / 16 | 0.2 | 0.5 | 0.5/0.5 | 0.5/2 | 0.5/0.5 | 2/0.001 |
| C100 / 32 | 0.2 | 0.5 | 0.5/2 | 0.5/0.5 | 0.5/0.5 | 2/0.001 |
| VOC / 8 | 2 | 2 | 0.5/8 | 2/0.5 | 2/0.5 | 0.5/0.001 |
| VOC / 16 | 0.2 | 0.5 | 0.5/0.5 | 0.5/0.5 | 2/2 | 2/0.05 |
| VOC / 32 | 2 | 0.5 | 0.5/0.5 | 2/2 | 0.5/0.5 | 2/0.001 |

**Complete-energy controls.** Direct controls add $\gamma\mathcal{E}(H)$ or $\gamma\mathcal{E}(PH)$ to the base loss, using the same graph, batches, four energy terms, and weights. Centered energy differentiates through $P$; its well, semantic, and decorrelation terms intentionally overlap related base penalties. Selecting rate $\{.0005, .002\}$ and $\gamma \in \{.5, 2\}$ gives 72 tuning and 54 efficacy runs. Teachers additionally search step caps. Selected direct settings and epochs are in `additions/experiments/selection_direct_*.json` and the run records.

### A.3.3 Fixed-reference student-feedback protocol

Reserve 500 outer-training images using seed 20260910, stratified by class for CIFAR and randomly permuted for VOC. Reference sizes are 4,500/4,500/3,011. Base supervision uses only $A$; meta methods also use $Q$ through the virtual update. Energy-only comparisons therefore change access to outer supervision. Standardization uses the original training subset, including $Q$, but excludes validation and final queries.

All methods share student initialization and reference-batch order within a seed, the preceding head, batch/remainder rule, student optimizer, clipping, schedule, and checkpoint selection. Student learning rate is .002 for 60 epochs. Matching and fixed direct-energy weights are .5. Teachers use four safeguarded steps, cap 8, initial step .8, initial gains .525, and uniform spectral mixtures. Solver and meta-direct Adam use .003. Settings are shared across methods, without method-specific tuning. The ten-method 8-bit study uses three datasets and three seeds ($10 \times 3 \times 3 = 90$ runs). The same settings, with the reported $K$, apply to Table 2. Its two additional controls are defined below. The 90-run count describes the original ten-method, 8-bit study, not all methods and code lengths in Table 2. Target diagnostics and timing retain their original 8-bit scope; the retrieval summaries below use the main-table means wherever the results overlap.

### A.3.4 Additional fixed-reference controls

Both controls inherit Appendix A.3.3; let $F_Q(w)$ be the ranking loss underlying Equation (23), evaluated at $w$ using the same 128 triplets per refresh. STAGE-Ref is the meta-spectral method denoted STAGE (full) in the main results.

**Meta-weighted targets.** Set $Y^{(0)} = \mathcal{R}(P_A \operatorname{sg}(H_A))$ and replace the virtual matching coefficient by a single global $\gamma(\zeta) = 4\sigma(\zeta)$, initialized at .5 ($\zeta = \log(1/7)$):

$$\ell_B^{\mathrm{MW}} = \mathcal{L}_{\mathrm{base}}(H_B) + \frac{\gamma(\zeta)}{|B|K}\|H_B - Y_B^{(0)}\|_F^2.$$

Only $\zeta$ is updated through $\zeta \to \gamma \to w^+ \to F_Q$, using the same virtual AdamW state and coefficient Adam rate .003. There are no learned refinement steps, gains, or shapes. The actual student uses the detached pre-update coefficient and target, cached for $R = 8$ updates; the new coefficient enters the next refresh.

Table 4: Fixed-reference controls. All target methods use bounded readout and identity-based caching. "Outer" is the ranking loss after a virtual student update. The final two rows are additional controls, outside the original ten-method 8-bit diagnostic study.

| Control | Auxiliary student signal | Auxiliary learning / use of $Q$ |
|---|---|---|
| Reference base | None | None |
| Direct raw / centered | Fixed-weight energy | No $Q$ |
| Meta-direct raw / centered | Learned-weight energy | Outer learns $\gamma$ |
| Bounded $T = 0$ | Unrefined target MSE | No $Q$ |
| Energy-spectral | Refined target MSE | Teacher minimizes terminal energy; no $Q$ |
| Meta-isotropic | Refined target MSE | Outer learns steps |
| Meta-scalar | Refined target MSE | Outer learns steps and gains |
| STAGE-Ref | Refined target MSE | Outer learns steps, gains, shape |
| Meta-weighted targets | Unrefined target MSE | Outer learns one matching weight |
| Spectral refinement $+Q$ | Refined target MSE and ranking | Energy-trained teacher; $Q$ directly updates student |

**Spectral refinement $+Q$.** Keep the four-step spectral teacher, normalized gains, readout, and cache, but learn $\phi$ from $\mathcal{E}_A(U_\phi^{(T)})$ with detached student initialization. The actual student minimizes

$$\ell_B^{\mathrm{SQ}} = \mathcal{L}_{\mathrm{base}}(H_B) + \frac{.5}{|B|K}\|H_B - \mathrm{sg}(Y_{\phi,B})\|_F^2 + \mathbf{1}_{\{s \bmod 8=0\}} F_Q(w),$$

where $s$ is the zero-based update index. The coefficient-one ranking term updates only the current student once per refresh; no virtual update or teacher meta-gradient is used. Pre-update targets are cached for eight updates. Thus both the teacher objective and the route of $Q$ supervision differ from STAGE-Ref.

## A.4 Mechanism and Spectral Diagnostics

### A.4.1 Phase saturation and conservation

The supplementary Digits pilot uses 1,797 samples (64 features, 10 classes) (Dua & Graff, 2019), split 1,078/359/360 for training/query/database, and seeds 11, 29, 47. It uses unsafeguarded updates and different auxiliary-loss matching from the final image protocol. Removing the well increases quantization error from .0212 to .0608; removing conservation increases mean maximum column-sum drift from $1.53 \times 10^{-5}$ to 35.76. The full teacher exceeds isotropic mAP by .0014, with overlapping seed variation. These pilot observations do not replace matched image-level component ablations; full pilot settings and values are retained in the supplementary records.

### A.4.2 Spectral shape under controlled curvature

Figure 3 uses 256 uniformly spaced $\lambda_j \in [0, 1]$, unit initial amplitudes, $\delta = .05$, and

$$h_{\mathrm{mid}}(\lambda) = .01 + 2(\lambda - .5)^2, \qquad h_{\mathrm{low}}(\lambda) = .01 + 2\lambda^2,$$
$$h_{\mathrm{high}}(\lambda) = .01 + 2(1 - \lambda)^2, \qquad h_{\mathrm{multi}}(\lambda) = .03 + .35\lambda + .45\sin^2(2\pi\lambda). \tag{26}$$

The optimizer selects simplex weights and a stable scalar step by 18-step terminal energy; fixed baselines independently select steps under the same criterion. Learned $(\alpha_l, \alpha_m, \alpha_h)$ are $(0, 1, 0)$, $(1, 0, 0)$, $(0, 0, 1)$, and $(.885, 0, .115)$ in the displayed profile order. At a simplex vertex, the operator equals that fixed band when floor and step agree. Replay uses archived full-precision parameters and

$$e_j^{(t+1)} = (1 - \eta m_j h_j) e_j^{(t)}, \qquad \frac{E_t}{E_0} = \frac{\sum_j h_j (1 - \eta m_j h_j)^{2t}}{\sum_j h_j}.$$

Heatmaps show $\log_{10}[E_j(t)/E_j(0)]$, clipped to $[-8, 0]$, with $E_j(t) = h_j[e_j^{(t)}]^2/2$; black dots mark first 99% reduction for every fourth sampled mode. Terminal annotations report total energy. The

contour panel projects the modes $\lambda_s = 127/255$ and $\lambda_f = 0$ onto $x_1 = -\sqrt{3}e_s/2 + e_f/2$ and $x_2 = e_s/2 + \sqrt{3}e_f/2$; contours set other coordinates to zero. Identity, low-pass, and learned steps are approximately 3.607411, 3.721472, and 12.708492. Low-pass uses $m_j = .05 + .95(1 - \lambda_j)$ and reaches $E_{18}/E_0 = 9.82 \times 10^{-3}$. These constructed problems illustrate conditional rates, not measured image-graph curvature.

## A.5 Supplementary Retrieval Comparisons

### A.5.1 Batch controls

Direct graph tests smoothness without a teacher; isotropic and scalar controls separate refinement and learned gain from spectral shape. Projection-only matching reduces to a mean penalty: $\|H - PH\|_F^2/(nK) = \|\mathbf{1}^\top H/n\|_2^2/K$. From the displayed means in Table 1, STAGE-Batch has the highest mean mAP in all nine dataset–code-length settings. It exceeds the strongest published baseline in each setting by .07–3.30 percentage points and exceeds both complete-energy controls throughout. Differences from per-bit scalar refinement range from +.38 to +6.78 percentage points, with the largest at 16-bit VOC 2007. These are descriptive differences, not paired significance tests. Marginal SDs do not determine paired uncertainty. Direct-graph means and SDs are retained in the supplementary records.

### A.5.2 Eight-bit fixed-reference results

Table 5 reports the ten-method supplementary comparison at 8 bits. Means shared with Table 2 follow that table; direct raw/centered, energy-spectral, and meta-isotropic results remain as reported in the supplementary study. Sample SDs are retained for unchanged entries. The updated STAGE-Ref means on CIFAR-100 and VOC 2007 are 9.98% and 68.35%, respectively; their corresponding SDs require the underlying seed-level scores and are not reported here. The two controls in Appendix A.3.4 appear in the main table but remain outside this ten-method supplementary display. Meta-direct and meta-teacher methods share outer-training information; energy-spectral also differs in its access to that information.

Table 5: Eight-bit fixed-reference mAP (%): three-seed means, with sample SD where available. Shared means follow Table 2. STAGE-Ref is STAGE (full) in the main table; bold marks the best mean. A dagger marks an updated mean whose corresponding SD is unavailable.

| Method | CIFAR-10 | CIFAR-100 | VOC 2007 |
|---|---|---|---|
| Reference base | $66.94_{\pm 4.45}$ | $9.13_{\pm .76}$ | $67.81_{\pm .13}$ |
| Direct raw | $68.54_{\pm .11}$ | $8.37_{\pm .54}$ | $63.74_{\pm 1.22}$ |
| Direct centered | $66.19_{\pm 3.73}$ | $7.99_{\pm .16}$ | $65.07_{\pm .28}$ |
| Meta-direct raw | $68.62_{\pm .20}$ | $8.35_{\pm .56}$ | $63.76_{\pm 1.21}$ |
| Meta-direct centered | $66.29_{\pm 3.43}$ | $7.96_{\pm .19}$ | $65.05_{\pm .32}$ |
| Bounded $T = 0$ | $69.78_{\pm .43}$ | $9.70_{\pm .83}$ | $64.97_{\pm .29}$ |
| Energy-spectral | $70.04_{\pm .56}$ | $9.02_{\pm .44}$ | $64.26_{\pm 1.31}$ |
| Meta-isotropic | $67.72_{\pm 3.61}$ | $8.94_{\pm .70}$ | $63.39_{\pm 1.44}$ |
| Meta-scalar | $69.89_{\pm .49}$ | $9.18_{\pm .32}$ | $64.44_{\pm .75}$ |
| **STAGE-Ref** | $\mathbf{70.27}_{\pm .22}$ | $\mathbf{9.98}^{\dagger}$ | $\mathbf{68.35}^{\dagger}$ |

Across the nine settings in Table 2, STAGE-Ref has the highest mean mAP, with gains of .12–.71 percentage points over the strongest control in each column. Its gains over meta-scalar range from .15 to 3.91 percentage points. At 8 bits, those gains are .38/.80/3.91 percentage points on CIFAR-10/CIFAR-100/VOC, while the corresponding gains over bounded $T = 0$ are .49/.28/3.38 percentage points.

Table 6 gives descriptive differences computed directly from the displayed 8-bit means in Table 5. The previous seed-paired intervals cannot be transferred to the updated CIFAR-100 and VOC means without the corresponding seed-level results. No paired intervals are reported in this revised comparison, and positive mean differences alone do not establish statistical significance or isolate the causal benefit of spectral shape.

Table 6: Eight-bit fixed-reference mean differences: STAGE-Ref minus each control, in percentage points. Differences are computed from the displayed means in Table 5; they are descriptive comparisons, not seed-paired significance tests.

| Comparator | CIFAR-10 | CIFAR-100 | VOC 2007 |
|---|---|---|---|
| Reference base | +3.33 | +0.85 | +0.54 |
| Meta-direct raw | +1.65 | +1.63 | +4.59 |
| Meta-direct centered | +3.98 | +2.02 | +3.30 |
| Bounded $T = 0$ | +0.49 | +0.28 | +3.38 |
| Energy-spectral | +0.23 | +0.96 | +4.09 |
| Meta-isotropic | +2.55 | +1.04 | +4.96 |
| Meta-scalar | +0.38 | +0.80 | +3.91 |

### A.6 Target Transfer and Computational Cost

The diagnostic and timing measurements in this section retain their reported experimental scope. They are not inferred from the retrieval means and do not supply the missing seed-level retrieval statistics for the updated fixed-reference comparisons.

#### A.6.1 Batch targets and context dependence

Table 7(a) reloads selected student/solver checkpoints on a fixed random 192-image training batch per dataset. Across code lengths, 6.901%–56.109% of target coordinates lie outside $[-1, 1]$. Figure 4 separately follows 27 runs (three datasets, three methods, three seeds) over epochs 0–60. Relative energy decrease is $100[\mathcal{E}(PH) - \mathcal{E}(U^{(4)})]/\mathcal{E}(PH)$, computed per seed before averaging. STAGE-Batch epoch-60 sign changes are .043%/.065%/0% and energy decreases are 3.13%/2.68%/2.90% on CIFAR-10/CIFAR-100/VOC; these differ from selected checkpoints. The largest selected-probe mean residual is $3.85 \times 10^{-8}$; the largest recorded one-step energy change is $-9.41 \times 10^{-5}$. No primary teacher step exhausted backtracking.

At each of 54 selected checkpoints, eight fixed training anchors are placed in 32 batches with different companions. For independent prediction $h$, context-specific target $U_c$, and empirical mean $\bar{U}$,

$$\mathbb{E}_c(h - U_c)^2 = (h - \bar{U})^2 + \mathrm{Var}_c(U_c),$$
$$\min_{h \in [-1,1]} \mathbb{E}_c(h - U_c)^2 = \mathrm{Var}_c(U_c) + (|\bar{U}| - 1)_+^2.$$

Table 7(b) reports the 8-bit subset. The independent LayerNorm head has zero seed-mean sign disagreement and maximum seed-mean absolute output difference $6.56 \times 10^{-7}$ across these contexts. Initial mobility/identity direction cosines for STAGE-Batch are .9983/.9976/.9840, versus 1 for identity. These observations concern the sampled anchors and companions.

#### A.6.2 Reference-target feasibility and post-hoc matching

Targets are regenerated at each selected checkpoint in the 8-bit study. Across 45 probes (five methods, three datasets, three seeds), the out-of-range fraction is zero, maximum absolute target is .988395035, and maximum absolute column mean is $2.53 \times 10^{-8}$. Within-cache variance is zero by indexed construction. Table 8 measures checkpoint consistency, not fitting to previously cached training targets. At $\gamma = .2$, the clipped disagreement bound ranges from .080 to 1.000; measured sign error is reported separately. Float64 checks give readout directional finite-difference error $1.38 \times 10^{-6}$, mass residual $7.22 \times 10^{-12}$, virtual/actual AdamW difference at most $7.45 \times 10^{-9}$, and sparse energy-gradient difference from autograd at most $1.09 \times 10^{-11}$.

#### A.6.3 Training cost and evaluation budgets

Batch timing uses one RTX 3080 Laptop GPU (16 GB), PyTorch 2.7.1+cu128, float32, and disabled TF32. Shared feature extraction costs 75.40/74.66/29.30 seconds on CIFAR-10/CIFAR-100/VOC and is separate.

Table 7: Batch-target diagnostics (three-seed means). (a) Target MSE / percentage outside $[-1, 1]$ at selected checkpoints; each cell reports the indicated code length. (b) Eight-bit context variance, exact empirical minimum bounded MSE, and agreement with the majority teacher sign.

| *(a) STAGE-Batch: MSE / outside (%)* | | | |
|---|---|---|---|
| Dataset | 8 bits | 16 bits | 32 bits |
| CIFAR-10 | $1.84\times10^{-2}$ / 45.616 | $5.59\times10^{-3}$ / 49.034 | $6.40\times10^{-3}$ / 56.109 |
| CIFAR-100 | $1.95\times10^{-2}$ / 6.901 | $5.88\times10^{-3}$ / 25.987 | $5.90\times10^{-3}$ / 34.375 |
| VOC 2007 | $1.31\times10^{-2}$ / 31.576 | $1.75\times10^{-2}$ / 38.965 | $3.00\times10^{-2}$ / 39.757 |

| *(b) Context dependence* | | | |
|---|---|---|---|
| Data / method | Var$(U)$ | Best bounded MSE | Sign consistency (%) |
| C10 / Iso | $5.47\times10^{-3}$ | $8.19\times10^{-3}$ | 100.00 |
| C10 / Batch | $6.21\times10^{-3}$ | $8.60\times10^{-3}$ | 99.98 |
| C100 / Iso | $1.03\times10^{-2}$ | $1.03\times10^{-2}$ | 99.76 |
| C100 / Batch | $9.44\times10^{-3}$ | $9.44\times10^{-3}$ | 99.80 |
| VOC / Iso | $8.27\times10^{-3}$ | $1.41\times10^{-2}$ | 99.77 |
| VOC / Batch | $5.13\times10^{-3}$ | $8.05\times10^{-3}$ | 100.00 |

Table 8: Eight-bit fixed-reference target diagnostics, three-seed means, using targets regenerated at the selected checkpoints. $E_T/E_0$ measures internal energy; sign error compares student $H_A$ with bounded $Y$.

| Teacher | $E_T/E_0$ | MSE$(H, Y)$ | Sign error (%) |
|---|---|---|---|
| *CIFAR-10* | | | |
| Bounded $T = 0$ | 1.0000 | 0.00395 | 0.034 |
| Energy-spectral | 0.7738 | 0.00711 | 0.131 |
| Meta-isotropic | 0.7473 | 0.01202 | 0.370 |
| Meta-scalar | 0.7809 | 0.00702 | 0.147 |
| STAGE-Ref | 0.7854 | 0.00602 | 0.119 |
| *CIFAR-100* | | | |
| Bounded $T = 0$ | 1.0000 | 0.00755 | 0.008 |
| Energy-spectral | 0.7782 | 0.01452 | 0.194 |
| Meta-isotropic | 0.7394 | 0.02305 | 0.525 |
| Meta-scalar | 0.7992 | 0.01262 | 0.151 |
| STAGE-Ref | 0.8020 | 0.01255 | 0.133 |
| *VOC 2007* | | | |
| Bounded $T = 0$ | 1.0000 | 0.03560 | 0.195 |
| Energy-spectral | 0.8357 | 0.04069 | 0.255 |
| Meta-isotropic | 0.8075 | 0.04376 | 0.486 |
| Meta-scalar | 0.8459 | 0.04084 | 0.477 |
| STAGE-Ref | 0.8503 | 0.03910 | 0.288 |

**Equal-time retrieval.** Table 9 uses 8-bit codes and a nominal 30-second budget. A time-indexed cosine schedule and common validation opportunities at epoch boundaries after .5, 1, 2, 5, 15, and 30 seconds select one final checkpoint. Actual durations are 30.004–30.686 seconds because stopping is at epoch boundaries. Synchronized training time includes graph construction and teacher/student updates, excluding feature extraction, validation, and retrieval. Runs reuse fixed-epoch selected settings without a separate time-budget search.

Table 9: Equal-time 8-bit Hamming mAP (%) under a nominal 30-second budget, mean ± sample SD over three seeds. Bold marks the highest mean. Separate 16-bit update costs are in Table 10(a).

| Method | CIFAR-10 | VOC 2007 |
|---|---|---|
| Matched head | $67.82 \pm 3.99$ | $69.16 \pm 0.54$ |
| Direct graph | $66.97 \pm 4.73$ | $69.46 \pm 0.80$ |
| Isotropic | $67.34 \pm 3.91$ | $65.32 \pm 1.44$ |
| STAGE-Batch | $\mathbf{72.06} \pm 4.14$ | $\mathbf{70.34} \pm 2.08$ |
| CSQ (adapted) | $70.98 \pm 1.02$ | $65.31 \pm 0.78$ |

**Update and full-training costs.** Table 10(a) gives 16-bit four-epoch update probes (three repeats). STAGE-Batch peak allocated memory, including cached features and labels, is 154.8/178.3/60.8 MB versus 153.1/176.6/58.1 MB for the base. Panel (b) measures synchronized 8-bit, 60-epoch fixed-reference training (three seeds), including reference-head passes, sparse energy, refreshes, backtracking, readout, outer losses, higher-order derivatives, and optimizer updates. It excludes feature extraction, graph/pair/triplet setup (.230/.088/.052 seconds), validation, and retrieval. Direct energy is evaluated every update; targets refresh every eight. These panels use different protocols, code lengths, and time units. Equal-time retrieval has not been evaluated for STAGE-Ref.

Table 10: Training cost, mean ± sample SD. (a) Batch-protocol 16-bit update time in milliseconds (three repeats). (b) Fixed-reference 8-bit training time for 60 epochs in seconds (three seeds), excluding setup and evaluation. Bold identifies STAGE within each protocol.

| Method | CIFAR-10 | CIFAR-100 | VOC 2007 |
|---|---|---|---|
| *(a) Batch protocol: ms/update, 16 bits* | | | |
| Base model | $4.34 \pm .62$ | $4.47 \pm .31$ | $4.07 \pm .11$ |
| Isotropic | $27.90 \pm 3.66$ | $30.26 \pm .58$ | $30.06 \pm .46$ |
| Per-bit scalar | $25.74 \pm 3.02$ | $31.16 \pm .20$ | $30.47 \pm .29$ |
| **STAGE-Batch** | $33.00 \pm .32$ | $32.72 \pm .43$ | $30.64 \pm 3.05$ |
| CSQ (adapted) | $3.24 \pm .18$ | $2.77 \pm .10$ | $2.05 \pm .01$ |
| *(b) Fixed-reference protocol: seconds/60 epochs, 8 bits* | | | |
| Reference base | $7.50 \pm .17$ | $7.48 \pm .09$ | $5.37 \pm .21$ |
| Direct raw | $13.27 \pm .22$ | $13.38 \pm .13$ | $9.18 \pm .15$ |
| Direct centered | $13.25 \pm .26$ | $13.45 \pm .10$ | $9.30 \pm .25$ |
| Meta-direct raw | $16.89 \pm .04$ | $16.96 \pm .42$ | $12.18 \pm .27$ |
| Meta-direct centered | $17.19 \pm .22$ | $17.23 \pm .16$ | $12.07 \pm .13$ |
| Bounded $T = 0$ | $9.82 \pm .41$ | $9.95 \pm .56$ | $6.39 \pm .44$ |
| Energy-spectral | $15.18 \pm 1.03$ | $14.48 \pm .40$ | $10.12 \pm .41$ |
| Meta-isotropic | $16.02 \pm .47$ | $15.58 \pm .10$ | $10.96 \pm .29$ |
| Meta-scalar | $15.91 \pm .52$ | $16.29 \pm .86$ | $11.27 \pm .23$ |
| **STAGE-Ref** | $17.44 \pm .08$ | $17.90 \pm .38$ | $12.42 \pm .63$ |

For reference edges $\mathcal{E}_A$ and sampled pairs $\Omega_A$, energy gradients cost $O(|\mathcal{E}_A|K + |\Omega_A|K + NK^2)$ and the cache uses $O(NK)$ memory. Spectral mobility adds sparse Laplacian products. Graph

construction temporarily forms dense $N \times N$ affinities; larger-scale construction remains untested. Inference uses independent encoding and Hamming search; packed-search latency was not measured.

### A.7 Reproducibility and Scope

Implementation and run records are under `additions/experiments/`: `run_additions.py`, selected settings, splits, `summary.json`, `all_new_test_runs.csv`, and `original_records`. Plot data are in `figure3_data` and `figure4_data`; feature provenance, cache hashes, environment, and reproduction instructions are in `feature_provenance` and `REPRODUCE_zh.md`. The accompanying `stage_supplementary_records.md` retains the pilot settings, direct-graph scores, and ancillary diagnostics removed from the condensed display. Raw images, weights, and feature caches are external; fixed seeds do not ensure bitwise agreement across devices.

Evidence is limited to frozen ResNet-18 features, three seeds, fixed splits, and the disclosed VOC test-set reuse. The earlier unsafeguarded CIFAR-100 isotropic run with nonfinite loss is excluded. Matched image-level phase ablations, stronger baselines, other backbones, end-to-end learning, and larger databases remain to be evaluated.