FLUXLORA: FLUX-LIMITED DYNAMICAL LOW-RANK ADAPTATION UNDER NOISY PSEUDO-LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pseudo-label errors can affect both the values and the subspaces of low-rank updates of vision foundation models in semi-supervised learning. To alleviate this, we propose **FluxLoRA**, which treats per-sample gradients as a matrix-valued field in the compact adaptation core. A feature graph and labeled gradient anchors guide nonlinear diffusion, while a separate reliability gate attenuates the residual directions that change the adaptation subspaces. The resulting update is evolved on a low-rank matrix manifold. We establish conditional properties of the diffusion solution and instantaneous residual control, without assuming recovery of clean gradients. We evaluate FluxLoRA on six VTAB tasks with two label budgets each, as well as CIFAR-10 and CIFAR-100 under limited-label settings. On VTAB, the reported V-PET integrations reach average accuracies of 67.1% with CLIP and 69.4% with DINOv3, exceeding the strongest baseline averages in our comparison by 6.1 and 4.8 percentage points, respectively. FluxLoRA also improves accuracy on standard image classification benchmarks, including CIFAR-10 and CIFAR-100.

1 INTRODUCTION

Vision foundation models (VFMs) provide transferable representations that are useful when labeled data are scarce [12, 28, 30]. In semi-supervised learning (SSL), parameter-efficient adaptation combines a small labeled set with pseudo-label supervision from a larger unlabeled set [13, 36, 37]. LoRA and other lightweight adaptation modules reduce the trainable parameter budget while retaining the pretrained backbone [8, 17–19, 23]. However, restricting the number of trainable parameters does not determine how the model should use gradients from incorrect pseudo-labels.

Pseudo-label errors remain a central difficulty. A wrong target produces a gradient that can reinforce the model's existing mistake through confirmation bias [1, 6]. High confidence is insufficient when predictions are miscalibrated [16, 26, 29]. Existing SSL methods mitigate pseudo-label errors through sample selection, confidence weighting, and target correction. However, erroneous pseudo-labels can still receive nonzero weights and influence the adaptation update (Figure 1A). This raises an optimization-level question: *at a fixed adaptation rank, how should gradients derived from potentially incorrect pseudo-labels modify the low-rank update?*

We examine this question through an orthogonal factorization $\Delta W = USV^\top$, where U and V have orthonormal columns. For a per-sample pseudo-label gradient $\widetilde{G}_i$, we define its core coordinates and outside-core residual as $\widetilde{C}_i = U^\top \widetilde{G}_i V$ and $R_i = \widetilde{G}_i - U\widetilde{C}_i V^\top$, respectively. The core component changes the update within the current input and output subspaces. The residual contains the tangent directions that rotate these subspaces, as well as a normal component removed by fixed-rank tangent projection (Figure 1B). This distinction exposes two ways in which pseudo-label errors can affect adaptation: *core contamination*, where erroneous gradients bias the update within the current subspaces, and *subspace drift*, where unreliable tangent directions rotate the adaptation subspaces (Figure 1C). Both effects can occur without changing the rank. A rank constraint alone therefore does not distinguish reliable from erroneous contributions to either component.

Motivated by this decomposition, we propose **FluxLoRA**, an optimization-level method for low-rank VFM adaptation under noisy pseudo-label supervision. Its central design combines two complementary operations: *labeled-anchor-guided core correction* and *reliability-gated residual updates*. The first treats per-sample core gradients as a compact matrix-valued field on a graph of frozen VFM

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.

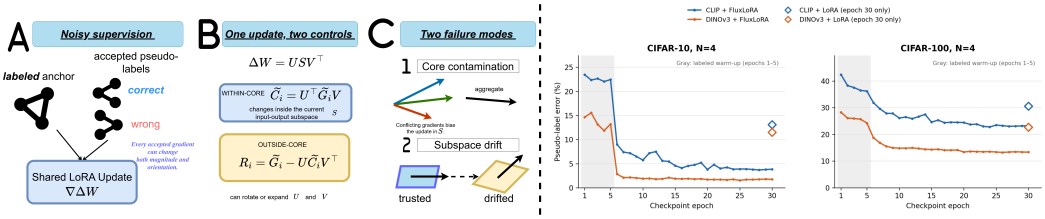


Figure 1: **Failure anatomy and training behavior of low-rank adaptation. Left:** (A) Labeled and pseudo-labeled samples contribute gradients to a shared LoRA update. (B) Each gradient is separated into a within-core component and an outside-core residual. (C) Incorrect pseudo-labels can cause core contamination and subspace drift, motivating separate controls. **Right:** Pseudo-label error on CIFAR-10 and CIFAR-100 with $N = 4$ labels per class. Solid curves show FluxLoRA with CLIP (blue) and DINOv3 (orange); hollow diamonds show standard LoRA at epoch 30. Gray regions mark the labeled-only warm-up. FluxLoRA achieves lower final error across both datasets and backbones.

features. Labeled gradients provide fixed anchors, while flux-limited nonlinear diffusion corrects the core field through bounded exchanges between neighboring nodes. This construction builds on graph-based SSL, manifold regularization, and nonlinear diffusion [2, 11, 27, 38, 39]. The second operation uses prediction reliability and consistency with the diffusion-adjusted core-gradient field to attenuate each sample's outside-core residual. Thus, FluxLoRA can modify a sample's within-subspace update while separately controlling its contribution to subspace rotation. Unlike assigning a single scalar weight to the entire pseudo-label gradient, this construction treats its two geometric effects differently. FluxLoRA can therefore complement different SSL methods by using their pseudo-label targets and sample weights to construct the low-rank update.

These operations define a complete update rule at a fixed rank. A short labeled-only warm-up initializes the adaptation subspaces. During subsequent SSL training, we combine the diffusion-adjusted core gradients with the reliability-gated residuals described above. The resulting update field is aggregated with the labeled gradients, projected onto the fixed-rank tangent space, and applied through a low-rank retraction [4, 5, 21, 25]. We additionally consider adaptive rank as an optional extension for adjusting capacity during training; neither core correction nor residual gating requires rank growth or pruning. Here, "correction" refers to the transformation induced by labeled-anchor-guided diffusion; it does not imply recovery of the gradient obtained from the unknown ground-truth label. The learned update can be merged into the frozen weights after training, introducing no additional inference operations [17].

We evaluate FluxLoRA across multiple VFM backbones, SSL methods, datasets, and label budgets. Figure 1 (right) illustrates the pseudo-label error achieved by the evaluated FluxLoRA configurations. To more precisely examine the update mechanism, we also compare directions from shared checkpoints at matched update norms and report fixed-rank comparisons with matched labeled-only warm-up and training schedules. These controlled comparisons address whether the gradient construction provides benefits beyond initialization, scalar reweighting, and adaptation capacity. Adaptive-rank results are considered separately to assess the tradeoff between accuracy and training cost.

Contributions. First, we distinguish within-core contamination from subspace-changing pseudo-label gradient contributions, identifying two effects of pseudo-label noise that remain present at a fixed adaptation rank. Second, we introduce FluxLoRA, which combines labeled-anchor-guided, flux-limited core correction with a separate reliability gate on outside-core residuals. We establish conditional energy-descent, finite-step stability, and instantaneous residual-control properties (Appendix A), without assuming recovery of clean gradients. Third, we evaluate the method through broad SSL comparisons and controlled update-direction and fixed-rank experiments, while examining adaptive rank as an optional capacity extension.

2 Related Work

Semi-supervised learning with vision foundation models. Modern SSL methods commonly combine pseudo-labeling with consistency training, often using weight-averaged teacher predictions [32].

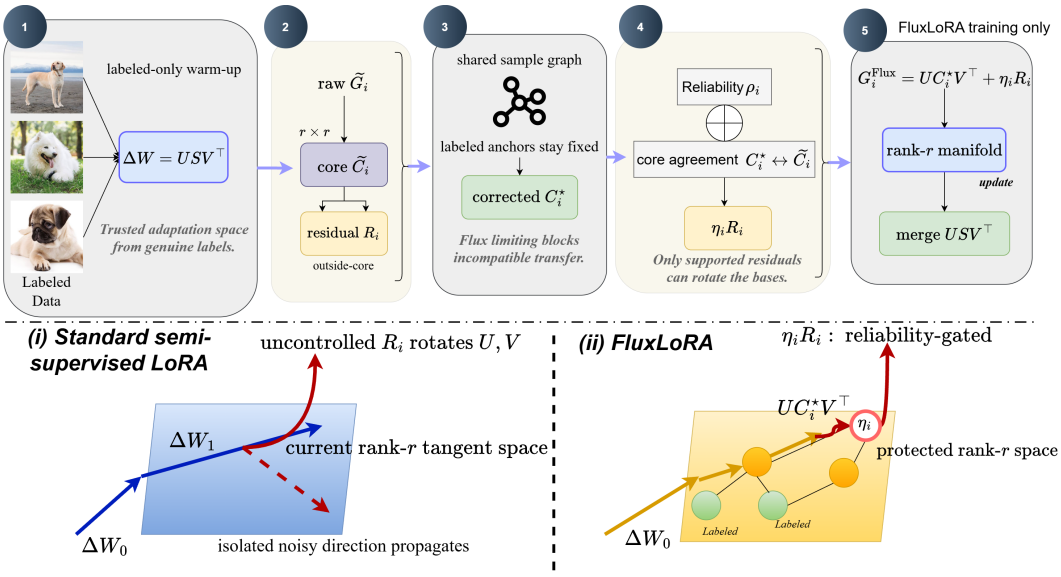


Figure 2: **Overview of FluxLoRA. Top:** After labeled-only warm-up, each pseudo-label gradient is split into a core and a residual. Anchor-guided, flux-limited diffusion corrects the core, while reliability gating scales the residual. The resulting gradient is projected onto the fixed-rank tangent space and applied through retraction. **Bottom:** Standard semi-supervised LoRA allows noisy gradients to alter both within-subspace updates and subspace directions. FluxLoRA limits core transfer and suppresses unsupported residual directions.

FixMatch retains confident predictions from weakly augmented views, while FlexMatch and SoftMatch adjust how pseudo-labels are selected or weighted [7, 31, 35]. S2Match further uses self-paced sampling to control the use of uncertain unlabeled examples [15]. However, incorrect pseudo-labels can be repeatedly reinforced through confirmation bias, and confidence may not reliably indicate correctness [1, 3, 6, 16, 29]. Recent VFM-based methods address related problems from the prediction side. FineSSL corrects bias in foundation-model predictions, V-PET ensembles complementary predictions from different VFM–PEFT models, and LADaS uses language information to debias and smooth pseudo-labels [13, 36, 37]. FluxLoRA operates at the gradient-update stage and is complementary to methods that select, weight, or refine pseudo-labels. We refer to the SSL method that supplies pseudo-label targets and sample weights as the base SSL method. Given these targets and weights, FluxLoRA modifies their contributions to the low-rank update through core-gradient diffusion and residual gating. It does not require a particular pseudo-label refinement method, but it does require a pseudo-labeling rule.

Parameter-efficient and structure-aware adaptation. LoRA, visual prompt tuning, AdaptFormer, FacT, and SSF reduce trainable parameters through low-rank updates, prompts, adapters, or feature modulation [8, 17–19, 23]. Graph-based SSL propagates information through feature neighborhoods [2, 38, 39], while nonlinear diffusion limits smoothing across boundaries [11, 27]. FluxLoRA diffuses an $r \times r$ core-gradient field with labeled anchors, gates outside-core residuals, and updates the low-rank matrix on a fixed-rank manifold [4, 5, 21, 25]. Appendix B provides further discussion.

3 Method

3.1 Overview

We use a labeled set $\mathcal{D}_l = \{(x_j^l, y_j)\}_{j=1}^{N_l}$ and an unlabeled set $\mathcal{D}_u = \{u_i\}_{i=1}^{N_u}$, with $C_{\text{cls}} \geq 2$ classes. The label $y_j \in \{1, \ldots, C_{\text{cls}}\}$ is a class index; $\text{onehot}(y_j)$ denotes its one-hot probability vector. Labeled and unlabeled samples have distinct node identifiers. FluxLoRA uses the pseudo-label targets and sample weights supplied by the base SSL method to construct low-rank updates. Algorithm 1 summarizes the LoRA and FluxLoRA training procedures.

Let $p_\theta(\cdot)$ and $p_{\bar{\theta}}(\cdot)$ denote the student and exponential-moving-average (EMA) teacher prediction functions. Each maps an input image to a probability vector over the C_{cls} classes. With weak and

strong augmentations a_w, a_s, the teacher's weak-view output is $q_i = p_{\bar{\theta}}(a_w(u_i))$, where $q_i(c)$ is the probability assigned to class c. The base SSL method constructs a target distribution $\widehat{q}_i$ and a nonnegative scalar weight w_i, including any acceptance mask. The target may be a soft distribution or a one-hot vector for a hard pseudo-label. Both $\widehat{q}_i$ and w_i are detached, meaning that no gradient is propagated through their construction during the student update.

For nonempty labeled and unlabeled index batches $\mathcal{B}_l, \mathcal{B}_u$, we define the student probability vectors $p_j^l = p_\theta(x_j^l)$ and $p_i^s = p_\theta(a_s(u_i))$, and losses $\ell_j^l = \mathrm{CE}(\mathrm{onehot}(y_j), p_j^l)$ and $\ell_i^u = \mathrm{CE}(\widehat{q}_i, p_i^s)$. Here $\mathrm{CE}(q,p) = -\sum_{c=1}^{C_{\mathrm{cls}}} q(c)\log p(c)$. The SSL objective is $\mathcal{L}_l + \lambda_u \mathcal{L}_u$, where

$$\mathcal{L}_l = \frac{1}{|\mathcal{B}_l|}\sum_{j\in\mathcal{B}_l} \ell_j^l, \qquad \mathcal{L}_u = \frac{1}{|\mathcal{B}_u|}\sum_{i\in\mathcal{B}_u} w_i \ell_i^u, \qquad \lambda_u \geq 0.$$

For an adapted layer, $W_0 \in \mathbb{R}^{d_{\mathrm{out}}\times d_{\mathrm{in}}}$ is frozen, and LoRA parameterizes its update by $\Delta W = BA$, with $B \in \mathbb{R}^{d_{\mathrm{out}}\times r}$ and $A \in \mathbb{R}^{r\times d_{\mathrm{in}}}$ [17]. For an input activation vector $x \in \mathbb{R}^{d_{\mathrm{in}}}$ to this linear layer, the output $z \in \mathbb{R}^{d_{\mathrm{out}}}$ is

$$z = W_0 x + \gamma \Delta W x = W_0 x + \gamma BAx, \qquad \gamma = \alpha / r_0. \tag{1}$$

Here x is a layer activation rather than a raw input image; for a tokenwise projection it is the activation of one token. The initial rank is r_0, $\alpha > 0$ is the LoRA scaling parameter, and γ remains fixed if the active rank r changes. After labeled-only warm-up, a compact SVD gives

$$\Delta W = USV^\top, \qquad U^\top U = V^\top V = I_r, \tag{2}$$

where $U \in \mathbb{R}^{d_{\mathrm{out}}\times r}$ and $V \in \mathbb{R}^{d_{\mathrm{in}}\times r}$ are orthonormal bases, $S \in \mathbb{R}^{r\times r}$ is the core, and I_r is the identity. The fixed-rank formulation below assumes $\mathrm{rank}(\Delta W) = r$, so that S is nonsingular. FluxLoRA diffuses per-sample core gradients and gates outside-core residuals before updating ΔW. The sample graph is shared across adapted layers; core states, flux thresholds, and matrix updates are layer-specific (Figure 2). We omit layer indices below.

3.2 Flux-Limited Core Gradient Field

We compute the unweighted pseudo-label gradient and its core gradient as

$$\widetilde{G}_i = \nabla_{\Delta W}\, \mathrm{CE}(\widehat{q}_i, p_i^s) \in \mathbb{R}^{d_{\mathrm{out}}\times d_{\mathrm{in}}}, \tag{3}$$

$$\widetilde{C}_i = U^\top \widetilde{G}_i V \in \mathbb{R}^{r\times r}. \tag{4}$$

We differentiate with respect to the update matrix ΔW and apply the sample weight w_i and coefficient λ_u when aggregating the per-sample update fields. For a linear layer with input activation x_i and output z_i, we define the scaled back-propagated derivative as $b_i = \gamma \nabla_{z_i} \ell_i^u$. Then, $\widetilde{G}_i = b_i x_i^\top$ and $\widetilde{C}_i = (U^\top b_i)(V^\top x_i)^\top$. For tokenwise projections, the corresponding outer products are summed over tokens. These expressions permit core-gradient computation without materializing each full per-sample matrix.

Graph and anchors. Let $h_i \in \mathbb{R}^{d_f}$ be a unit-normalized feature extracted from the frozen, unadapted VFM using deterministic evaluation transforms, where d_f is the feature dimension. For graph construction, we assign $\hat{y}_i = \arg\max_c q_i(c)$ to an unlabeled node, using the teacher's weak-view prediction, and $\hat{y}_j = y_j$ to a labeled node, using its observed label. These are class indices for graph construction and anchor retrieval, distinct from the loss targets $\widehat{q}_i$. For each labeled graph node j, we set $q_j = \mathrm{onehot}(y_j)$ using its observed label. We use $\mathcal{I}_l$ to index all labeled examples and retrieve a labeled anchor for each unlabeled sample $i \in \mathcal{B}_u$ as

$$j^\star(i) = \operatorname*{arg\,min}_{j\in\mathcal{I}_l:\, y_j = \hat{y}_i} \|h_i - h_j\|_2^2. \tag{5}$$

We assume that every class has a labeled example and resolve ties in class assignment and neighbor selection by index order. We collect the retrieved anchors in the set $\mathcal{A}_\mathcal{B} = \{j^\star(i) : i \in \mathcal{B}_u\}$. We then form the labeled node set $\mathcal{L}_\mathcal{B} = \mathcal{B}_l \cup \mathcal{A}_\mathcal{B}$ and the graph node set $\mathcal{V}_\mathcal{B} = \mathcal{B}_u \cup \mathcal{L}_\mathcal{B}$. We select up to k nearest neighbors per node by feature distance, excluding the node itself. We retain edges whose endpoints have the same assigned class and include both orientations of each retained edge.

We also include both orientations of every $(i, j^\star(i))$ edge, independently of the neighbor truncation. We count repeated nodes and edges once. We denote the resulting symmetric set of ordered edges by $\mathcal{E}_{\mathrm{ord}}$ and define the neighbor set as $\mathcal{N}(i) = \{j : (i,j) \in \mathcal{E}_{\mathrm{ord}}\}$. On this support, we define

$$\omega_{ij} = \mathbf{1}[\hat{y}_i = \hat{y}_j] \exp\left(-\frac{\|h_i - h_j\|_2^2}{\sigma_h^2} - \frac{\mathrm{JSD}(q_i, q_j)}{\sigma_q^2}\right) a_i a_j; \tag{6}$$

We set $\omega_{ij} = 0$ outside this support. The indicator $\mathbf{1}[\cdot]$ is one when its condition holds, $\sigma_h, \sigma_q > 0$ are bandwidths, and $\omega_{ij} = \omega_{ji}$. We set $a_j = 1$ for labeled nodes and compute agreement scores for unlabeled nodes as

$$a_i = a_i^{\mathrm{view}} a_i^{\mathrm{temp}}, \qquad a_i^{\mathrm{view}} = e^{-\mathrm{JSD}(q_i, q_i^s)/\tau_v}, \qquad a_i^{\mathrm{temp}} = e^{-\mathrm{JSD}(q_i, \bar{q}_i)/\tau_t}.$$

Here $q_i^s = p_{\bar{\theta}}(a_s(u_i))$ is the teacher's probability vector for the strongly augmented view, obtained through an additional forward pass; it is distinct from the student output p_i^s. The positive temperatures τ_v, τ_t control sensitivity to distributional differences. View agreement compares the current weak- and strong-view teacher outputs. Temporal agreement compares q_i with the cached weak-view teacher output $\bar{q}_i$ from the most recent previous iteration in which sample i was included in an unlabeled mini-batch. This is the sample's previous "visit," which need not be the preceding training iteration or the preceding epoch. The cache is keyed by sample identifier. On a sample's first visit, we set $\bar{q}_i = q_i$, giving $a_i^{\mathrm{temp}} = 1$; we overwrite its cache with the current q_i only after computing agreement. This cache stores a previous prediction, separately from the parameter EMA of the teacher. This cache stores the previous teacher probability vector for each sample. Separately, the EMA averages the teacher's effective layer weights and classification-head parameters

We use natural logarithms and $\mathrm{JSD}(p,q) = \frac{1}{2}\mathrm{KL}(p\|m) + \frac{1}{2}\mathrm{KL}(q\|m)$, where $m = (p+q)/2$ and $\mathrm{KL}(p\|m) = \sum_c p(c)\log[p(c)/m(c)]$, with zero-mass terms contributing zero. Identical distributions have agreement equal to one; larger JSD gives smaller agreement. The agreement scores compare the complete teacher probability vectors.

Boundary states and diffusion. For each $j \in \mathcal{L}_{\mathcal{B}}$, we compute the current labeled loss ℓ_j^l and its gradient $G_j^l = \nabla_{\Delta W} \ell_j^l$, then set $C_j^l = U^\top G_j^l V$. We use one labeled view per unique labeled node and reuse an existing batch gradient only for that same view. We clamp $C_j(\tau) = C_j^l$ during the solve, where τ denotes diffusion time. Cached features remain fixed, whereas anchor gradients are recomputed using the current student at each training iteration. Retrieved nodes outside $\mathcal{B}_l$ supply boundary states only; they do not add terms to $\mathcal{L}_l$ or its batch gradient. For an unlabeled node, we initialize $C_i(0) = \widetilde{C}_i$ and set

$$\rho_i = \exp\left(-\frac{\mathsf{H}(q_i)}{\log C_{\mathrm{cls}}}\right) a_i \exp\left(-\frac{d(h_i, \mathcal{F}_{\hat{y}_i})^2}{\sigma_a^2}\right). \tag{7}$$

Here $\mathsf{H}(q) = -\sum_c q(c)\log q(c)$ is the entropy, $\mathcal{F}_c = \{h_j : j \in \mathcal{I}_l, y_j = c\}$, $d(h, \mathcal{F}_c) = \min_{f \in \mathcal{F}_c} \|h - f\|_2$, and $\sigma_a > 0$ is the labeled-support bandwidth. The reliability score $\rho_i \in [0,1]$ combines predictive concentration, agreement, and proximity to labeled features of the assigned class. With $\kappa > 0$, $0 < \lambda_{\min} \le \lambda_{\max}$, and $\lambda_i = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min})\rho_i$, we evolve the core-gradient field according to

$$\frac{\partial C_i}{\partial \tau} = \kappa \sum_{j \in \mathcal{N}(i)} \omega_{ij} \psi_\delta(\|C_j - C_i\|_F)(C_j - C_i) - \lambda_i (C_i - \widetilde{C}_i), \quad i \in \mathcal{B}_u, \tag{8}$$

where $\psi_\delta(s) = [1 + (s/\delta)^2]^{-1/2}$ for $s \ge 0$ and $\delta > 0$ is the flux threshold. Here $\|\cdot\|_F$ is the Frobenius norm and $\|\cdot\|_2$ is the Euclidean norm for vectors. The fidelity term retains the raw core gradient more strongly when ρ_i is larger. The limiter bounds the magnitude of each pairwise exchange because $s\psi_\delta(s) \le \delta$. With the labeled states clamped and the symmetric weights fixed, this is the gradient flow with respect to the unlabeled states of

$$\mathcal{E}(\{C_i\}) = \frac{\kappa}{2} \sum_{(i,j) \in \mathcal{E}_{\mathrm{ord}}} \omega_{ij} \varphi_\delta(\|C_i - C_j\|_F) + \frac{1}{2} \sum_{i \in \mathcal{B}_u} \lambda_i \|C_i - \widetilde{C}_i\|_F^2, \tag{9}$$

where $\varphi_\delta(s) = \delta^2[\sqrt{1 + (s/\delta)^2} - 1]$. The factor $1/2$ in the edge term accounts for the two orientations.

During each solve, the graph, raw gradients, reliabilities, bases, labeled states, and diffusion parameters remain fixed. Starting from $C_i^{(0)} = \widetilde{C}_i$, we apply K synchronous explicit Euler steps, evaluating every right-hand side in Eq. 8 at the previous diffusion iterate. We denote the core gradient after K diffusion steps by $C_i^\star = C_i^{(K)}$. We define the weighted degree of node i as $d_i = \sum_{j \in \mathcal{N}(i)} \omega_{ij}$, the sum of its edge weights, and set $d_{\max} = \max_{i \in \mathcal{B}_u} d_i$. We choose the diffusion step size as

$$\Delta\tau = \min\left\{\Delta\tau_0, \frac{0.9}{\kappa d_{\max} + \lambda_{\max}}\right\},$$

which matches Eq. 104 in Appendix D.3. We compute these weighted degrees on the final graph, including reverse edges added during symmetrization and the retrieved-anchor edges. The final neighbor set can contain more than k nodes, so k does not bound d_i. Parameter values and bandwidth selection are given in Appendix D.3. Once the per-sample derivatives have been computed, they and the diffusion states are detached; there is no differentiation through the solve. We refer to this diffusion transformation as core correction. Conditional denoising assumptions are stated in Appendix A.

3.3 Core and Residual Updates

We decompose the raw pseudo-label gradient as

$$\widetilde{G}_i = U\widetilde{C}_i V^\top + R_i, \qquad R_i = \widetilde{G}_i - U\widetilde{C}_i V^\top. \tag{10}$$

The residual has zero core projection, $U^\top R_i V = 0$, but is not entirely tangent to the fixed-rank manifold. It contains two tangent blocks that can change the input and output subspaces, together with the normal block, $(I - UU^\top)\widetilde{G}_i(I - VV^\top)$. The latter is removed by fixed-rank tangent projection. The identity matrices, I, have the corresponding input or output dimensions. We define

$$\eta_i = \rho_i \exp\left(-\frac{\|C_i^\star - \widetilde{C}_i\|_F^2}{\tau_g(\|\widetilde{C}_i\|_F^2 + \epsilon)}\right), \tag{11}$$

where $\tau_g > 0$ is a gate temperature and $\epsilon > 0$ stabilizes denominators. The gate satisfies $0 \le \eta_i \le \rho_i \le 1$ and decreases as the relative discrepancy between the raw and diffusion-adjusted core gradients increases. We construct the per-sample update field and batch update field as

$$G_i^{\mathrm{Flux}} = UC_i^\star V^\top + \eta_i R_i, \tag{12}$$

$$G_{\mathrm{rob}} = G_l + \frac{\lambda_u}{|\mathcal{B}_u|}\sum_{i \in \mathcal{B}_u} w_i G_i^{\mathrm{Flux}}, \qquad G_l = \frac{1}{|\mathcal{B}_l|}\sum_{j \in \mathcal{B}_l} G_j^l. \tag{13}$$

FluxLoRA uses G_{rob} to update the low-rank matrix. The sample weight w_i supplied by the base SSL method scales the entire per-sample update field, whereas the residual gate η_i acts only on the outside-core residual. The graph uses the full unlabeled batch: $w_i = 0$ removes a sample's direct aggregate contribution but not its possible influence through diffusion on neighboring nodes. The pseudo-label targets and loss weights are preserved, while the low-rank update direction is changed.

For $\mathcal{M}_r = \{X \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}} : \mathrm{rank}(X) = r\}$, the projected flow is

$$\frac{d\Delta W}{ds} = -P_{T_{\Delta W}\mathcal{M}_r}(G_{\mathrm{rob}}), \tag{14}$$

where s is optimization time, distinct from diffusion time τ. At a rank-r matrix ΔW, the tangent-space orthogonal projection of an ambient matrix G is

$$P_{T_{\Delta W}\mathcal{M}_r}(G) = UU^\top G + GVV^\top - UU^\top GVV^\top. \tag{15}$$

In particular, the projected per-sample field is

$$\begin{aligned} P_{T_{\Delta W}\mathcal{M}_r}(G_i^{\mathrm{Flux}}) = {} & UC_i^\star V^\top \\ & + \eta_i\big[(I - UU^\top)\widetilde{G}_i VV^\top + UU^\top \widetilde{G}_i (I - VV^\top)\big]. \end{aligned}$$

This makes explicit how core correction and residual gating act on different tangent components. The residual gate attenuates each sample's subspace-changing contribution.

Table 1: Average top-1 accuracy (%). The left block (VTAB) reports averages over 12 dataset–budget settings; full per-setting means and standard deviations over five runs are reported in Table 10 in Appendix D. The right block (CIFAR10/CIFAR100) reports five-seed average results.

		VTAB (12-setting average)			CIFAR10 N=1		CIFAR10 N=4		CIFAR100 N=4		CIFAR100 N=100	
Backbone	SSL method	LoRA	AdaptFormer	FluxLoRA	LoRA	FluxLoRA	LoRA	FluxLoRA	LoRA	FluxLoRA	LoRA	FluxLoRA
CLIP	Labeled only	55.8	55.5	56.7	83.2	84.0	91.7	92.3	68.4	69.2	83.8	84.3
	FixMatch	53.7	53.7	61.5	85.6	90.3	94.8	97.2	72.5	77.8	86.1	87.4
	FlexMatch	56.2	53.9	63.5	88.4	92.6	95.7	97.8	75.8	79.4	86.9	88.0
	SoftMatch	59.7	56.3	65.5	91.3	94.1	96.8	98.2	79.6	82.1	87.8	88.6
	FineSSL	53.9	51.6	64.5	92.4	95.6	97.4	98.8	81.1	82.6	88.4	89.1
	ST	54.7	56.7	62.4	87.8	91.6	95.2	97.4	74.3	78.7	85.9	87.2
	PET	59.3	59.7	65.8	90.7	94.0	96.5	98.1	78.8	81.9	87.5	88.5
	V-PET	60.6	61.0	**67.1**	93.1	95.9	97.6	98.7	81.6	83.5	88.6	89.4
DINOv3	Labeled only	60.1	58.3	61.6	87.6	88.5	94.3	94.8	75.7	76.6	88.4	88.8
	FixMatch	56.0	56.4	64.9	89.1	93.5	96.6	97.9	79.3	83.4	89.6	90.5
	FlexMatch	57.9	55.7	66.9	91.7	95.0	97.2	98.3	82.1	85.5	90.2	91.0
	SoftMatch	61.5	57.5	68.4	94.2	96.3	98.0	98.6	85.6	87.3	90.9	91.5
	FineSSL	56.1	53.2	67.7	95.3	96.6	98.7	98.5	87.9	88.4	91.6	92.0
	ST	59.2	59.9	65.9	90.8	94.2	96.8	98.0	81.2	84.6	89.8	90.7
	PET	62.7	63.1	68.5	93.7	95.8	97.8	98.5	85.2	87.1	90.8	91.4
	V-PET	64.2	64.6	**69.4**	95.6	96.8	98.4	98.8	87.7	88.8	91.5	92.1

At training iteration t, with learning rate $\zeta_t > 0$, we apply

$$\Delta W_{t+1} = \mathrm{Retr}_r\left[\Delta W_t - \zeta_t P_{T_{\Delta W_t}\mathcal{M}_r}(G_{\mathrm{rob},t})\right]. \tag{16}$$

Here, Retr_r denotes a truncated SVD retaining the leading r singular components of its argument. It gives a rank-at-most-r matrix and is a smooth rank-r local retraction, where the retained singular values are nonzero and separated from the discarded ones. We use this local retraction for the fixed-rank update. The step can be evaluated in augmented bases of dimension at most $2r$ followed by a small-core SVD. The factors U, S, V are refreshed from the retracted update before the next iteration.

No momentum, adaptive preconditioning, or weight decay is applied to ΔW after warm-up. AdamW updates A, B and the classification head during labeled-only warm-up. At the SVD transition, factor optimizer states are discarded and the head's moment states are retained. Thereafter, the head continues with AdamW on the SSL objective, while the low-rank matrix follows Eq. 16. The EMA teacher is updated after the student update and any optional rank change, using effective layer weights rather than the basis-dependent factors, as specified in Appendix D.3.

Optional rank changes. Rank changes are separate from Eq. 16. Let $G_{\mathrm{rel}} = G_l + \lambda_u |\mathcal{B}_u|^{-1} \sum_{i\in\mathcal{B}_u} w_i \eta_i R_i$. The normal-gradient ratio is

$$e_{\mathrm{out}} = \frac{\|(I - UU^\top) G_{\mathrm{rel}} (I - VV^\top)\|_F}{\|G_{\mathrm{rel}}\|_F + \epsilon}. \tag{17}$$

The numerator measures the normal component that is discarded by a fixed-rank tangent update and can be used to identify candidate growth directions. We use this ratio as the growth diagnostic with the optional settings described in Appendix D.3. Core correction and residual gating require neither growth nor pruning. At inference, we merge $W = W_0 + \gamma USV^\top$.

4 Experimental Setup

We evaluate FluxLoRA on six VTAB tasks with two label budgets each, and on CIFAR-10/100, using identical data splits across methods. We use frozen CLIP and DINOv3 ViT-B/16 backbones. FluxLoRA starts at rank $r_0 = 4$, with five labeled-only warm-up epochs and $K = 8$ diffusion steps. We report top-1 accuracy as mean $\pm$ standard deviation over five runs and average VTAB results equally across the 12 settings. Appendix D details the datasets, training protocols, and evaluation metrics; Appendix D.3 specifies the FluxLoRA configuration.

5 Main Result

Table 1 shows that FluxLoRA improves the reported accuracy across both backbones and all evaluated SSL methods. On VTAB, FluxLoRA + V-PET achieves 67.1% accuracy with CLIP and 69.4%

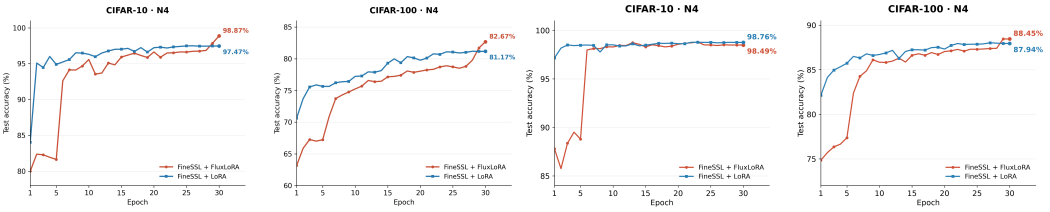


Figure 3: **Training dynamics of FineSSL with standard LoRA and FluxLoRA under the CIFAR-10 and CIFAR-100 N4 settings.** From left to right, the first two panels report results with CLIP ViT-B/16, while the last two panels report results with DINOv3 ViT-B/16. Each curve shows the test accuracy over 30 training epochs, with the final accuracy marked at the endpoint.

Table 2: **Component ablation on VTAB-SSL and CIFAR10.** We report average top-1 accuracy (%). Each variant removes or replaces one mechanism in FluxLoRA. Avg. Δ denotes the mean accuracy change relative to the complete configuration across CLIP and DINOv3. The CIFAR10 column reports the single-seed result obtained with DINOv3 under the $N = 4$ setting using FineSSL as the base SSL method.

Component	Variant	FluxLoRA + V-PET			FluxLoRA + FineSSL			CIFAR10 $N = 4$
		CLIP	DINOv3	Avg. Δ	CLIP	DINOv3	Avg. Δ	DINOv3
Full method	**Complete configuration**	**67.1**	**69.4**		**64.5**	**67.7**		**98.5**
Adaptive rank	Fixed rank	66.4	68.8	-0.65	63.9	66.9	-0.70	98.4
Manifold evolution	Euclidean factor update	65.8	67.9	-1.40	63.3	66.1	-1.40	98.1
Initialization	No labeled-only warm-up	65.0	67.6	-1.95	62.6	65.4	-2.10	97.8
Residual gating	No residual gate ($\eta_i = 1$)	64.6	66.7	-2.60	61.7	65.1	-2.70	97.4
Flux limiting	Linear diffusion ($\psi_\delta = 1$)	63.9	66.5	-3.05	61.5	64.1	-3.30	97.2
Core diffusion	No core diffusion ($K = 0$)	63.1	65.7	-3.85	60.2	63.9	-4.05	96.9
Labeled anchoring	No labeled anchors	62.0	64.9	-4.80	59.1	62.5	-5.30	96.3

accuracy with DINOv3, exceeding the strongest standard PEFT baselines by 6.1 and 4.8 percentage points, respectively. The CIFAR results show larger gains under smaller label budgets, suggesting that FluxLoRA is particularly useful when labeled supervision is scarce. Figure 3 further shows that FluxLoRA starts from a lower accuracy but achieves a higher final accuracy in three of the four plotted comparisons; DINOv3 on CIFAR-10 is the exception.

6 Ablation Study

Ablation protocol. We remove or replace one component at a time, keeping the other settings unchanged. Table 2 reports average accuracy over 12 VTAB-SSL settings with CLIP and DINOv3, using V-PET and FineSSL as base SSL methods. Accuracy changes average over both backbones. We also report single-seed CIFAR-10 results with DINOv3 and $N = 4$.

Component contributions. Removing labeled anchors causes the largest accuracy drops: 4.80 and 5.30 percentage points with V-PET and FineSSL, respectively. Disabling core-gradient diffusion reduces accuracy by 3.85 and 4.05 points, while replacing flux-limited diffusion with linear diffusion reduces it by 3.05 and 3.30 points. Removing residual gating or labeled-only warm-up also lowers accuracy, whereas Euclidean factor updates and fixed-rank adaptation produce smaller drops. The single-seed CIFAR-10 results show the same ordering.

Mechanism-level diagnostics. Figure 4 shows that adaptive variants reach similar final mean ranks near 16 but follow different rank trajectories. After warm-up, diffusion adjusts core-gradient magnitudes, and residual gates below one attenuate outside-core contributions.

Controlled comparisons. Table 3(a) compares one-step updates from matched checkpoints at matched update norms. FluxLoRA achieves a mean ΔCE of -0.012 and a descent rate of 73%, compared with -0.010 and 68% for scalar reliability weighting.

Table 3(b) controls labeled-only warm-up and the training schedule. At rank 16, FluxLoRA reaches 98.4% on CIFAR-10 and 88.0% on CIFAR-100, compared with 98.0% and 87.9% for LoRA and 98.3% and 87.7% for scalar reliability weighting. Adaptive rank reaches 98.5% and 88.4%, while reducing training time from 1.84/1.88 to 1.68/1.72 hours, respectively.

Table 3: Direction quality and controlled comparisons of FluxLoRA. (a) Matched-checkpoint, norm-matched one-step interventions. Paired Δ denotes the mean cross-entropy change relative to scalar reliability weighting (ρ); negative values indicate a larger loss reduction. (b) End-to-end comparisons on CIFAR-10 and CIFAR-100 with $N = 4$. All configurations use DINOv3, the same labeled-only warm-up, data splits, and training schedule. Fixed-rank configurations use $r = 16$.

(a) Local update quality on CIFAR-100 with N = 4 (protocol and metric definitions are in Appendix D.5)

Direction construction	Mean ΔCE $\downarrow$	Descent rate (%)$\uparrow$	Paired Δ vs. ρ $\downarrow$
Raw aggregate	-0.007	58.0	$+0.003$
Scalar reliability (ρ)	-0.010	68.0	0.000
Scalar consistency gate (η)	-0.009	66.0	$+0.001$
FluxLoRA gradient	$\mathbf{-0.012}$	**73.0**	$\mathbf{-0.002}$

(b) End-to-end comparisons with controlled capacity and initialization

Configuration	Rank	CIFAR-10 Accuracy (%)$\uparrow$	CIFAR-100 Accuracy (%)$\uparrow$	Adapter parameters (M)	Training time C10 / C100 (h)
LoRA + labeled warm-up	Fixed 16	98.0	87.9	0.590	1.30 / 1.33
Manifold update + scalar reliability only	Fixed 16	98.3	87.7	0.602	1.44 / 1.48
FluxLoRA + constant residual gate	Fixed 16	98.1	87.5	0.602	1.80 / 1.84
FluxLoRA	Fixed 16	98.4	88.0	0.602	1.84 / 1.88
FluxLoRA	Adaptive 4–16	98.5	88.4	0.148–0.602	1.68 / 1.72

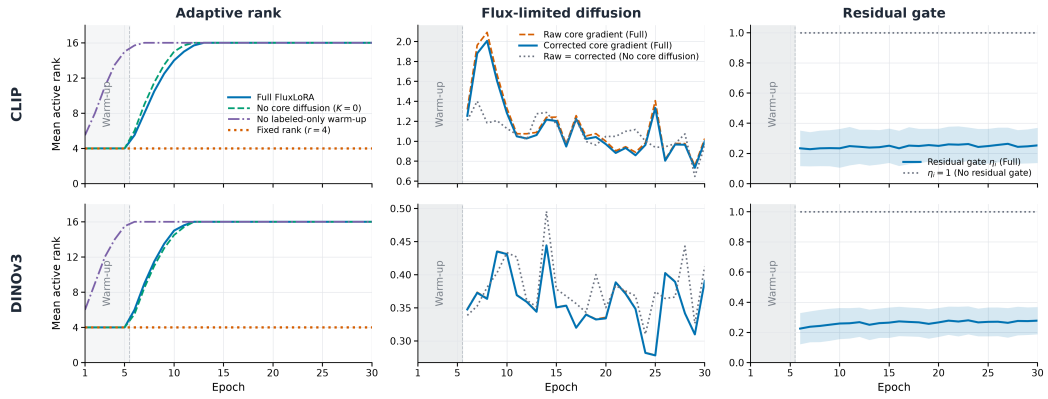


Figure 4: **Mechanism-level diagnostics of FluxLoRA with CLIP (top) and DINOv3 (bottom).** Left: evolution of the mean active rank. Middle: magnitudes of the raw and diffusion-adjusted LoRA core gradients. Right: evolution of the residual gate η_i.

Training and deployment efficiency. Table 4 reports training times of $1.17\times$ and $1.34\times$ those of LoRA for CLIP and DINOv3, respectively. Core-gradient diffusion costs $O(K|\mathcal{E}_u|r^2)$ per layer, where $\mathcal{E}_u$ contains ordered edges originating at unlabeled nodes. We merge the learned update into the frozen weights for inference.

Table 4: Training efficiency on CIFAR-10 N=4. Wall-clock time on a single RTX 6000 Ada GPU.

Backbone	LoRA	FluxLoRA	Ratio	Diffusion cost
CLIP	1h27m21s	1h41m55s	$1.17\times$	$\mathcal{O}(K\|\mathcal{E}_u\|r^2)$
DINOv3	1h15m18s	1h40m45s	$1.34\times$	$\mathcal{O}(K\|\mathcal{E}_u\|r^2)$

7 Conclusion

We introduce FluxLoRA to address core contamination and subspace drift during low-rank VFM adaptation under noisy pseudo-labels. Our key innovation combines labeled-anchor-guided core diffusion with separate reliability-gated residual updates on a low-rank matrix manifold. Theoretically, we characterize the geometric effects of pseudo-label gradients and establish conditional energy-descent, finite-step stability, and instantaneous residual-control properties. Experiments across datasets, label budgets, SSL methods, and backbones show improved accuracy, with no additional inference operations after weight merging.

REFERENCES

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2020.

[2] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(11), 2006.

[3] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 6912–6920, 2021.

[4] Gianluca Ceruti, Jonas Kusch, and Christian Lubich. A rank-adaptive robust integrator for dynamical low-rank approximation. *BIT Numerical Mathematics*, 62(4):1149–1174, 2022.

[5] Gianluca Ceruti, Jonas Kusch, and Christian Lubich. A parallel rank-adaptive integrator for dynamical low-rank approximation. *SIAM Journal on Scientific Computing*, 46(3):B205–B228, 2024.

[6] Baixu Chen, Junguang Jiang, Ximei Wang, Pengfei Wan, Jianmin Wang, and Mingsheng Long. Debiased self-training for semi-supervised learning. *Advances in Neural Information Processing Systems*, 35:32424–32437, 2022.

[7] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv preprint arXiv:2301.10921*, 2023.

[8] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in neural information processing systems*, 35:16664–16678, 2022.

[9] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[10] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

[11] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Kai Gan and Tong Wei. Erasing the bias: Fine-tuning foundation models for semi-supervised learning. *arXiv preprint arXiv:2405.11756*, 2024.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361. IEEE, 2012.

[15] Dayan Guan, Yun Xing, Jiaxing Huang, Aoran Xiao, Abdulmotaleb El Saddik, and Shijian Lu. S2match: Self-paced sampling for data-limited semi-supervised learning. *Pattern Recognition*, 159:111121, 2025.

[16] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[18] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022.

[19] Shibo Jie and Zhi-Hong Deng. Fact: Factor-tuning for lightweight adaptation on vision transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 1060–1068, 2023.

[20] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

[21] Othmar Koch and Christian Lubich. Dynamical low-rank approximation. *SIAM Journal on Matrix Analysis and Applications*, 29(2):434–454, 2007.

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[23] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[25] Christian Lubich and Ivan V Oseledets. A projector-splitting integrator for dynamical low-rank approximation. *BIT Numerical Mathematics*, 54(1):171–188, 2014.

[26] Shambhavi Mishra, Balamurali Murugesan, Ismail Ben Ayed, Marco Pedersoli, and Jose Dolz. Do not trust what you trust: Miscalibration in semi-supervised learning. *arXiv preprint arXiv:2403.15567*, 2024.

[27] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

[29] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021.

[30] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[31] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[32] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[33] 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.

[34] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

[35] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in neural information processing systems*, 34:18408–18419, 2021.

[36] Ping Zhang, Zheda Mai, Quang-Huy Percy Nguyen, and Wei-Lun Harry Chao. Revisiting semi-supervised learning in the era of foundation models. *Advances in Neural Information Processing Systems*, 38:59295–59324, 2026.

[37] Na Zheng, Xuemeng Song, Xue Dong, Aashish Nikhil Ghosh, Liqiang Nie, and Roger Zimmermann. Language-assisted debiasing and smoothing for foundation model-based semi-supervised learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25708–25717. IEEE, 2025.

[38] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.

[39] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pp. 912–919, 2003.

APPENDIX

A THEORETICAL ANALYSIS

A.1 MATRIX-VALUED GRAPH SIGNAL DENOISING

Throughout this appendix, H denotes the matrix of labeled boundary sources, whereas H denotes prediction entropy and CE denotes cross-entropy. The spectral-gap lower bound is $\underline{\sigma}_t$, distinct from the fixed LoRA scale γ. For a matrix, $\|\cdot\|_2$ denotes the spectral norm; $\langle A, B\rangle_F = \mathrm{tr}(A^\top B)$ is the Frobenius inner product. Diffusion is analyzed on the full graph independently of sample weights; sample weights enter the aggregate-update and trajectory bounds.

We first analyze the continuous core-gradient dynamics underlying Eq. 8. Throughout this analysis, the graph weights in Eq. 6, the reliabilities in Eq. 7, the labeled boundary states, and the low-rank bases in Eq. 2 are held fixed during a single diffusion solve. The following result establishes that the flux limiter gives uniformly bounded edge influence and that the resulting dynamics admit a strongly convex variational formulation with a unique globally attracting equilibrium.

Lemma A.1 (Bounded flux and well-posedness of the core-gradient diffusion)**.** *Assume that the graph in Eq. 6 has symmetric weights $\omega_{ij} = \omega_{ji} \geq 0$, that the labeled states are fixed throughout the diffusion solve, and that $0 < \lambda_{\min} \leq \lambda_i \leq \lambda_{\max}$. Let $X = (C_i)_{i\in\mathcal{B}_u}$ with $\|X\|^2 = \sum_{i\in\mathcal{B}_u} \|C_i\|_F^2$. Then, the flux induced by Eq. 8 is uniformly bounded, Eq. 8 is then the negative gradient flow of Eq. 9, and $\mathcal{E}$ is $\lambda_{\min}$-strongly convex with a unique minimizer X^∞. Moreover,*

$$\frac{d}{d\tau}\mathcal{E}(X(\tau)) = -\|\nabla\mathcal{E}(X(\tau))\|^2 \leq 0, \tag{18}$$

and

$$\|X(\tau) - X^\infty\| \leq e^{-\lambda_{\min}\tau}\|X(0) - X^\infty\|. \tag{19}$$

Proof. We define the flux associated with the limiter in Eq. 8 by

$$F_\delta(D) = \psi_\delta(\|D\|_F)D = \frac{D}{\sqrt{1 + \|D\|_F^2/\delta^2}}. \tag{20}$$

Its norm satisfies

$$\begin{aligned}\|F_\delta(D)\|_F &= \frac{\|D\|_F}{\sqrt{1 + \|D\|_F^2/\delta^2}} \\ &= \frac{\delta\|D\|_F}{\sqrt{\delta^2 + \|D\|_F^2}} \\ &\leq \delta.\end{aligned} \tag{21}$$

For $D = C_i - C_j$, the edge potential φ_δ in Eq. 9 satisfies

$$\begin{aligned}\nabla_D\varphi_\delta(\|D\|_F) &= \nabla_D\left[\delta^2\left(\sqrt{1 + \frac{\|D\|_F^2}{\delta^2}} - 1\right)\right] \\ &= \delta^2\frac{1}{2}\left(1 + \frac{\|D\|_F^2}{\delta^2}\right)^{-1/2}\frac{2D}{\delta^2} \\ &= \frac{D}{\sqrt{1 + \|D\|_F^2/\delta^2}} \\ &= F_\delta(D).\end{aligned} \tag{22}$$

Differentiating again gives

$$\begin{aligned}\nabla_D^2\varphi_\delta(\|D\|_F) &= \nabla_D\left[\left(1 + \frac{\|D\|_F^2}{\delta^2}\right)^{-1/2}D\right] \\ &= \left(1 + \frac{\|D\|_F^2}{\delta^2}\right)^{-1/2}I - \frac{1}{\delta^2}\left(1 + \frac{\|D\|_F^2}{\delta^2}\right)^{-3/2}(D\otimes D).\end{aligned} \tag{23}$$

For $\langle D, H\rangle_F = 0$,

$$\begin{aligned}\nabla_D^2 \varphi_\delta(\|D\|_F)[H] &= \left(1+\frac{\|D\|_F^2}{\delta^2}\right)^{-1/2} H - \frac{1}{\delta^2}\left(1+\frac{\|D\|_F^2}{\delta^2}\right)^{-3/2}\langle D, H\rangle_F D \\ &= \left(1+\frac{\|D\|_F^2}{\delta^2}\right)^{-1/2} H,\end{aligned} \tag{24}$$

while in the radial direction,

$$\begin{aligned}\nabla_D^2 \varphi_\delta(\|D\|_F)[D] &= \left(1+\frac{\|D\|_F^2}{\delta^2}\right)^{-1/2} D - \frac{\|D\|_F^2}{\delta^2}\left(1+\frac{\|D\|_F^2}{\delta^2}\right)^{-3/2} D \\ &= \left(1+\frac{\|D\|_F^2}{\delta^2}\right)^{-3/2} D.\end{aligned} \tag{25}$$

Therefore,

$$0 < \left(1+\frac{\|D\|_F^2}{\delta^2}\right)^{-3/2} \le \left(1+\frac{\|D\|_F^2}{\delta^2}\right)^{-1/2} \le 1, \tag{26}$$

and hence

$$0 \preceq \nabla_D^2 \varphi_\delta(\|D\|_F) \preceq I. \tag{27}$$

Using Eq. 9, the symmetry of ω_{ij}, and Eq. 22,

$$\begin{aligned}\nabla_{C_i}\mathcal{E}(X) &= \kappa \sum_{j\in\mathcal{N}(i)} \omega_{ij} F_\delta(C_i - C_j) + \lambda_i(C_i - \widetilde{C}_i) \\ &= -\kappa \sum_{j\in\mathcal{N}(i)} \omega_{ij} F_\delta(C_j - C_i) + \lambda_i(C_i - \widetilde{C}_i),\end{aligned} \tag{28}$$

so Eq. 8 gives

$$\begin{aligned}-\nabla_{C_i}\mathcal{E}(X) &= \kappa \sum_{j\in\mathcal{N}(i)} \omega_{ij} F_\delta(C_j - C_i) - \lambda_i(C_i - \widetilde{C}_i) \\ &= \kappa \sum_{j\in\mathcal{N}(i)} \omega_{ij} \psi_\delta(\|C_j - C_i\|_F)(C_j - C_i) - \lambda_i(C_i - \widetilde{C}_i) \\ &= \frac{\partial C_i}{\partial \tau}.\end{aligned} \tag{29}$$

Thus, collecting all unlabeled states,

$$\frac{dX}{d\tau} = -\nabla\mathcal{E}(X). \tag{30}$$

For any perturbation $Z = (Z_i)_{i\in\mathcal{B}_u}$, with perturbations of the clamped labeled states set to zero, Eq. 27 gives

$$\begin{aligned}\langle Z, \nabla^2\mathcal{E}(X)Z\rangle &= \frac{\kappa}{2}\sum_{(i,j)\in\mathcal{E}_{\text{ord}}} \omega_{ij}\left\langle Z_i - Z_j, \nabla_D^2\varphi_\delta(\|C_i - C_j\|_F)[Z_i - Z_j]\right\rangle_F + \sum_{i\in\mathcal{B}_u}\lambda_i\|Z_i\|_F^2 \\ &\ge \sum_{i\in\mathcal{B}_u}\lambda_i\|Z_i\|_F^2 \\ &\ge \lambda_{\min}\sum_{i\in\mathcal{B}_u}\|Z_i\|_F^2 \\ &= \lambda_{\min}\|Z\|^2.\end{aligned} \tag{31}$$

Thus, $\mathcal{E}$ is $\lambda_{\min}$-strongly convex. Since $\varphi_\delta \ge 0$,

$$\begin{aligned}\mathcal{E}(X) &\ge \frac{1}{2}\sum_{i\in\mathcal{B}_u}\lambda_i\|C_i - \widetilde{C}_i\|_F^2 \\ &\ge \frac{\lambda_{\min}}{2}\sum_{i\in\mathcal{B}_u}\|C_i - \widetilde{C}_i\|_F^2,\end{aligned} \tag{32}$$

so $\mathcal{E}$ is coercive and therefore admits a unique minimizer X^∞ satisfying

$$\nabla\mathcal{E}(X^\infty) = 0. \tag{33}$$

Equation 27, the finiteness of the graph, and $\lambda_i \le \lambda_{\max}$ imply that $\nabla\mathcal{E}$ is globally Lipschitz. Hence, Eq. 30 admits a unique global solution. Along this solution,

$$\begin{aligned}\frac{d}{d\tau}\mathcal{E}(X(\tau)) &= \left\langle \nabla\mathcal{E}(X(\tau)), \frac{dX(\tau)}{d\tau} \right\rangle \\ &= -\left\langle \nabla\mathcal{E}(X(\tau)), \nabla\mathcal{E}(X(\tau)) \right\rangle \\ &= -\|\nabla\mathcal{E}(X(\tau))\|^2 \\ &\le 0.\end{aligned} \tag{34}$$

Finally, by Eq. 33 and the $\lambda_{\min}$-strong convexity,

$$\begin{aligned}\frac{1}{2}\frac{d}{d\tau}\|X(\tau) - X^\infty\|^2 &= \left\langle X(\tau) - X^\infty, \frac{dX(\tau)}{d\tau} \right\rangle \\ &= -\left\langle X(\tau) - X^\infty, \nabla\mathcal{E}(X(\tau)) - \nabla\mathcal{E}(X^\infty) \right\rangle \\ &\le -\lambda_{\min}\|X(\tau) - X^\infty\|^2.\end{aligned} \tag{35}$$

Applying Grönwall's inequality,

$$\begin{aligned}\|X(\tau) - X^\infty\|^2 &\le e^{-2\lambda_{\min}\tau}\|X(0) - X^\infty\|^2, \\ \|X(\tau) - X^\infty\| &\le e^{-\lambda_{\min}\tau}\|X(0) - X^\infty\|.\end{aligned} \tag{36}$$

□

Lemma A.1 establishes that, for fixed graph weights, reliabilities, labeled states, and low-rank bases, Eq. 8 defines a well-posed continuous gradient flow with a unique equilibrium and monotonically decreasing energy. Corollary A.2 further shows that the nonlinear flux limits the instantaneous effect of corrupted neighbors by their total graph conductance, rather than by the magnitude of their gradient discrepancies.

Corollary A.2 (Bounded influence of corrupted neighbors)**.** *Under Lemma A.1, let $\mathcal{N}_b(i) \subseteq \mathcal{N}(i)$ denote any set of corrupted neighbors of node i. Then, their instantaneous contribution to the diffusion update satisfies*

$$\left\| \kappa \sum_{j\in\mathcal{N}_b(i)} \omega_{ij} F_\delta(C_j - C_i) \right\|_F \le \kappa\delta \sum_{j\in\mathcal{N}_b(i)} \omega_{ij}. \tag{37}$$

Hence, the influence of arbitrarily large corrupted gradient discrepancies is controlled by the total conductance of the corrupted neighborhood rather than by their magnitudes.

Proof. By the triangle inequality and Eq. 21,

$$\begin{aligned}\left\| \kappa \sum_{j\in\mathcal{N}_b(i)} \omega_{ij} F_\delta(C_j - C_i) \right\|_F &\le \kappa \sum_{j\in\mathcal{N}_b(i)} \omega_{ij}\|F_\delta(C_j - C_i)\|_F \\ &\le \kappa\delta \sum_{j\in\mathcal{N}_b(i)} \omega_{ij}.\end{aligned} \tag{38}$$

□

The implementation in Sec. 3.2, however, does not evolve Eq. 8 to equilibrium, but applies only K explicit steps with step size $\Delta\tau$. We therefore analyze this finite-step discretization and derive sufficient step-size conditions under which it preserves the descent structure of Eq. 9.

Assumption A.3 (Explicit diffusion step size)**.** *Let*

$$d_i = \sum_{j \in \mathcal{N}(i)} \omega_{ij}, \qquad d_{\max} = \max_{i \in \mathcal{B}_u} d_i, \qquad L_\star = 2\kappa d_{\max} + \lambda_{\max}. \tag{39}$$

The explicit diffusion step size satisfies

$$0 < \Delta\tau < \frac{2}{L_\star}. \tag{40}$$

Lemma A.4 (Descent and stability of the explicit diffusion step)**.** *Under the hypotheses of Lemma A.1, define*

$$d_i = \sum_{j \in \mathcal{N}(i)} \omega_{ij}, \qquad d_{\max} = \max_{i \in \mathcal{B}_u} d_i, \qquad L_\star = 2\kappa d_{\max} + \lambda_{\max}.$$

Here, each weighted degree sums the incident edge weights in the symmetric graph used for diffusion, including edges to clamped labeled nodes. Then, $\nabla\mathcal{E}$ is $L_\star$-Lipschitz.

(i) Energy descent. *Under Assumption A.3, namely $0 < \Delta\tau < 2/L_\star$, the explicit Euler update*

$$X^{(k+1)} = X^{(k)} - \Delta\tau \nabla\mathcal{E}(X^{(k)}) \tag{41}$$

satisfies

$$\begin{aligned} \mathcal{E}(X^{(k+1)}) &\le \mathcal{E}(X^{(k)}) - \Delta\tau \left(1 - \frac{L_\star \Delta\tau}{2}\right) \|\nabla\mathcal{E}(X^{(k)})\|^2 \\ &\le \mathcal{E}(X^{(k)}). \end{aligned} \tag{42}$$

The decrease is strict whenever $\nabla\mathcal{E}(X^{(k)}) \neq 0$.

(ii) Convex-combination stability. *If, in addition,*

$$\Delta\tau(\kappa d_i + \lambda_i) \le 1 \qquad \text{for every } i \in \mathcal{B}_u, \tag{43}$$

each unlabeled-node update is a convex combination of its current state, its neighbors' current states, and its raw core gradient $\widetilde{C}_i$. A sufficient uniform condition is

$$0 < \Delta\tau \le \frac{1}{\kappa d_{\max} + \lambda_{\max}}. \tag{44}$$

With $C_i^{(0)} = \widetilde{C}_i$ and fixed labeled states C_j^ℓ, this implies, for every finite k,

$$\max_{i \in \mathcal{B}_u} \|C_i^{(k)}\|_F \le M_0, \qquad M_0 := \max\left\{ \max_{i \in \mathcal{B}_u} \|\widetilde{C}_i\|_F,\ \max_{j \in \mathcal{L}_\mathcal{B}} \|C_j^\ell\|_F \right\}, \tag{45}$$

where $\mathcal{L}_\mathcal{B}$ denotes the clamped labeled nodes in the diffusion graph.

Proof. We first establish the Lipschitz bound and energy descent, and then prove the convex-combination property under its additional step-size condition.

By Eq. 23,

$$0 \preceq \nabla_D^2 \varphi_\delta(\|D\|_F) \preceq I. \tag{46}$$

For a perturbation $Z = (Z_i)_{i \in \mathcal{B}_u}$, extend $Z_j = 0$ to the clamped labeled nodes. Since $\mathcal{E}_{\text{ord}}$ contains both orientations of each edge and $\omega_{ij} = \omega_{ji}$,

$$\begin{aligned} \langle Z, \nabla^2\mathcal{E}(X)Z \rangle &= \frac{\kappa}{2} \sum_{(i,j) \in \mathcal{E}_{\text{ord}}} \omega_{ij} \left\langle Z_i - Z_j,\ \nabla_D^2 \varphi_\delta(\|C_i - C_j\|_F)[Z_i - Z_j] \right\rangle_F \\ &\quad + \sum_{i \in \mathcal{B}_u} \lambda_i \|Z_i\|_F^2 \\ &\le \frac{\kappa}{2} \sum_{(i,j) \in \mathcal{E}_{\text{ord}}} \omega_{ij} \|Z_i - Z_j\|_F^2 + \lambda_{\max} \|Z\|^2 \\ &\le \kappa \sum_{(i,j) \in \mathcal{E}_{\text{ord}}} \omega_{ij} \left(\|Z_i\|_F^2 + \|Z_j\|_F^2 \right) + \lambda_{\max} \|Z\|^2 \\ &= 2\kappa \sum_{i \in \mathcal{B}_u} d_i \|Z_i\|_F^2 + \lambda_{\max} \|Z\|^2 \\ &\le (2\kappa d_{\max} + \lambda_{\max}) \|Z\|^2 = L_\star \|Z\|^2. \end{aligned} \tag{47}$$

Together with convexity of $\mathcal{E}$, this gives

$$0 \preceq \nabla^2 \mathcal{E}(X) \preceq L_\star I, \tag{48}$$

and therefore

$$\|\nabla \mathcal{E}(Y) - \nabla \mathcal{E}(X)\| \leq L_\star \|Y - X\|. \tag{49}$$

Applying the smoothness inequality to Eq. 41 yields

$$\begin{aligned} \mathcal{E}(X^{(k+1)}) &\leq \mathcal{E}(X^{(k)}) + \left\langle \nabla \mathcal{E}(X^{(k)}), X^{(k+1)} - X^{(k)} \right\rangle \\ &\quad + \frac{L_\star}{2} \|X^{(k+1)} - X^{(k)}\|^2 \\ &= \mathcal{E}(X^{(k)}) - \Delta\tau \left(1 - \frac{L_\star \Delta\tau}{2}\right) \|\nabla \mathcal{E}(X^{(k)})\|^2. \end{aligned} \tag{50}$$

Since $0 < \Delta\tau < 2/L_\star$, the coefficient of the squared gradient norm is positive. Thus

$$\mathcal{E}(X^{(k+1)}) < \mathcal{E}(X^{(k)}) \quad \text{whenever} \quad \nabla \mathcal{E}(X^{(k)}) \neq 0. \tag{51}$$

This proves part (i).

For part (ii), we define

$$b_{ij}^{(k)} = \omega_{ij} \psi_\delta \left(\|C_j^{(k)} - C_i^{(k)}\|_F \right). \tag{52}$$

The flux limiter satisfies $0 < \psi_\delta(s) \leq 1$, whereas graph weights may vanish. Hence,

$$0 \leq b_{ij}^{(k)} \leq \omega_{ij}. \tag{53}$$

The nodewise explicit update can be written as

$$\begin{aligned} C_i^{(k+1)} &= \left[1 - \Delta\tau \left(\kappa \sum_{j \in \mathcal{N}(i)} b_{ij}^{(k)} + \lambda_i \right) \right] C_i^{(k)} \\ &\quad + \Delta\tau\kappa \sum_{j \in \mathcal{N}(i)} b_{ij}^{(k)} C_j^{(k)} + \Delta\tau \lambda_i \widetilde{C}_i. \end{aligned} \tag{54}$$

Using Eq. 53 and the additional condition in Eq. 43,

$$\begin{aligned} 1 - \Delta\tau \left(\kappa \sum_{j \in \mathcal{N}(i)} b_{ij}^{(k)} + \lambda_i \right) &\geq 1 - \Delta\tau(\kappa d_i + \lambda_i) \\ &\geq 0. \end{aligned} \tag{55}$$

All other coefficients are nonnegative, and

$$\left[1 - \Delta\tau \left(\kappa \sum_j b_{ij}^{(k)} + \lambda_i \right) \right] + \Delta\tau\kappa \sum_j b_{ij}^{(k)} + \Delta\tau\lambda_i = 1. \tag{56}$$

Thus, Eq. 54 is a convex combination. The condition in Eq. 44 implies Eq. 43 because $d_i \leq d_{\max}$ and $\lambda_i \leq \lambda_{\max}$.

Finally, the initial unlabeled states and the fixed labeled states have Frobenius norms bounded by M_0. Convexity of the Frobenius norm and induction on k therefore establish Eq. 45. □

Lemma A.4 separates two properties of the finite-K diffusion solver. Assumption A.3 guarantees energy descent, whereas the additional condition in Eq. 43 guarantees nonnegative mixing coefficients and the uniform state bound. The latter does not follow from the energy-descent condition alone. Under the standing assumption $\lambda_{\max} > 0$, the uniform bound in Eq. 44 is sufficient for both properties, since

$$\frac{1}{\kappa d_{\max} + \lambda_{\max}} < \frac{2}{2\kappa d_{\max} + \lambda_{\max}}.$$

These guarantees concern numerical stability of the diffusion solve; they do not by themselves establish recovery of clean core gradients or improvement of the training objective. We next examine gradient estimation under an explicit graph-smooth signal model.

Assumption A.5 (Graph-consistent clean field and conditional noise)**.** *Let $\mathcal{U} = \mathcal{B}_u$ denote the n unlabeled nodes and $\mathcal{L} = \mathcal{L}_\mathcal{B}$ the clamped labeled nodes, including retrieved anchors. Fix symmetric nonnegative conductances $b_{ij} = b_{ji} \geq 0$, with $b_{ii} = 0$, and labeled core-gradient states $\{C_a^\ell\}_{a\in\mathcal{L}}$.*

We define the Dirichlet graph Laplacian $L_b \in \mathbb{R}^{n\times n}$ and the boundary source $H = (H_i)_{i\in\mathcal{U}}$ by

$$\begin{aligned}(L_b)_{ii} &= \sum_{\substack{j\in\mathcal{U}\\ j\neq i}} b_{ij} + \sum_{a\in\mathcal{L}} b_{ia},\\ (L_b)_{ij} &= -b_{ij}, \qquad i \neq j,\\ H_i &= \sum_{a\in\mathcal{L}} b_{ia} C_a^\ell.\end{aligned} \tag{57}$$

Let $\Lambda = \mathrm{diag}(\lambda_i)$ with $\lambda_i > 0$, and let $\kappa > 0$.

Let $X^\dagger$ denote the core gradients obtained from correct labels. We assume that the clean core-gradient field $X^\dagger = (C_i^\dagger)_{i\in\mathcal{U}}$ is consistent with the labeled boundary states:

$$L_b X^\dagger = H. \tag{58}$$

The raw field satisfies

$$\begin{aligned}\widetilde{X} &= X^\dagger + \Xi,\\ \mathbb{E}[\mathrm{vec}(\Xi) \mid \mathcal{F}] &= 0,\\ \mathbb{E}[\mathrm{vec}(\Xi)\,\mathrm{vec}(\Xi)^\top \mid \mathcal{F}] &= \sigma^2 I_{nr^2}, \qquad \sigma^2 > 0.\end{aligned} \tag{59}$$

Here, $\mathcal{F}$ collects the fixed graph, fidelity coefficients, labeled boundary states, and clean field. The conditional moment assumptions are modeling assumptions; they do not follow merely from freezing the conductances during a diffusion solve. The uncentered consistency condition is $L_b X^\dagger = H$, not $L_b X^\dagger = 0$. If X_h is a harmonic extension satisfying $L_b X_h = H$, then and only under this consistency assumption, the centered field $Y^\dagger = X^\dagger - X_h$ satisfies $L_b Y^\dagger = 0$. Neither anchor connectivity nor freezing the graph guarantees clean-field consistency or the conditional noise model. These assumptions describe the fixed-conductance model, not a guarantee for the state-dependent, finite-step nonlinear solver used during training.

Lemma A.6 (Mean-square denoising of the core-gradient field)**.** *Under Assumption A.5, consider the fixed-conductance diffusion*

$$\frac{dX}{d\tau} = -\kappa(L_b X - H) - \Lambda(X - \widetilde{X}). \tag{60}$$

Its unique equilibrium satisfies

$$X^\infty - X^\dagger = (\kappa L_b + \Lambda)^{-1}\Lambda\Xi. \tag{61}$$

Moreover,

$$\begin{aligned}\mathbb{E}[\|X^\infty - X^\dagger\|^2 \mid \mathcal{F}] &= \sigma^2 r^2\,\mathrm{tr}\left[(\kappa L_b + \Lambda)^{-1}\Lambda^2(\kappa L_b + \Lambda)^{-1}\right]\\ &\leq \sigma^2 r^2 n\\ &= \mathbb{E}[\|\widetilde{X} - X^\dagger\|^2 \mid \mathcal{F}].\end{aligned} \tag{62}$$

If $L_b \neq 0$, the inequality is strict:

$$\mathbb{E}[\|X^\infty - X^\dagger\|^2 \mid \mathcal{F}] < \mathbb{E}[\|\widetilde{X} - X^\dagger\|^2 \mid \mathcal{F}]. \tag{63}$$

Here, $\|X\|^2 = \sum_{i\in\mathcal{U}} \|C_i\|_F^2$.

Proof. We set $A = \kappa L_b + \Lambda$. The Dirichlet graph Laplacian is positive semidefinite: for any $z \in \mathbb{R}^n$,

$$z^\top L_b z = \frac{1}{2}\sum_{i,j\in\mathcal{U}} b_{ij}(z_i - z_j)^2 + \sum_{i\in\mathcal{U}}\sum_{a\in\mathcal{L}} b_{ia} z_i^2 \geq 0. \tag{64}$$

Since $\Lambda \succ 0$, we have $A \succ 0$. Thus, the affine flow has a unique equilibrium satisfying

$$AX^\infty = \kappa H + \Lambda\widetilde{X}. \tag{65}$$

By Eq. 58,

$$AX^\dagger = \kappa H + \Lambda X^\dagger. \tag{66}$$

Subtracting these identities gives

$$\begin{aligned} A(X^\infty - X^\dagger) &= \Lambda \Xi, \\ X^\infty - X^\dagger &= S\Xi, \qquad S := A^{-1}\Lambda. \end{aligned} \tag{67}$$

We next establish that S is entrywise nonnegative and has rows that sum up to one, at most. First, A^{-1} is entrywise nonnegative. To see this, let $Az = y$ with $y \geq 0$ entrywise. If $z_i < 0$ is a minimum entry of z, then

$$\begin{aligned} (Az)_i &= \lambda_i z_i + \kappa \sum_{\substack{j \in \mathcal{U} \\ j \neq i}} b_{ij}(z_i - z_j) + \kappa \sum_{a \in \mathcal{L}} b_{ia} z_i \\ &< 0, \end{aligned} \tag{68}$$

contradicting $y_i \geq 0$. Hence $z \geq 0$, proving the claim.

Let

$$h_i = \sum_{a \in \mathcal{L}} b_{ia}.$$

Then $h = L_b \mathbf{1} \geq 0$, and $A\mathbf{1} = \Lambda \mathbf{1} + \kappa h$. Consequently,

$$S \geq 0 \quad \text{entrywise}, \qquad S\mathbf{1} = \mathbf{1} - \kappa A^{-1} h \leq \mathbf{1}. \tag{69}$$

Furthermore, $S_{ii} = \lambda_i (A^{-1})_{ii} > 0$ because $A^{-1} \succ 0$.

Using the conditional noise covariance,

$$\begin{aligned} \mathbb{E}[\|X^\infty - X^\dagger\|^2 \mid \mathcal{F}] &= \mathbb{E}[\|S\Xi\|^2 \mid \mathcal{F}] \\ &= \sigma^2 r^2 \operatorname{tr}(S^\top S) \\ &= \sigma^2 r^2 \operatorname{tr}\left[A^{-1}\Lambda^2 A^{-1}\right]. \end{aligned} \tag{70}$$

For each row of S, nonnegativity and the row-sum bound give

$$\sum_{j=1}^n S_{ij}^2 \leq \left(\sum_{j=1}^n S_{ij} \right)^2 \leq 1. \tag{71}$$

Summing over rows,

$$\operatorname{tr}(S^\top S) = \sum_{i,j=1}^n S_{ij}^2 \leq n. \tag{72}$$

Meanwhile,

$$\mathbb{E}[\|\widetilde{X} - X^\dagger\|^2 \mid \mathcal{F}] = \mathbb{E}[\|\Xi\|^2 \mid \mathcal{F}] = \sigma^2 r^2 n. \tag{73}$$

This proves the non-strict bound.

For strictness, equality in $\operatorname{tr}(S^\top S) \leq n$ would require equality in both rowwise inequalities for every row. Each row would therefore contain exactly one nonzero entry, equal to one. Since $S_{ii} > 0$, that entry must be the diagonal entry, so $S = I$. But $A^{-1}\Lambda = I$ implies $A = \Lambda$, hence $\kappa L_b = 0$. Because $\kappa > 0$, this contradicts $L_b \neq 0$. Thus,

$$L_b \neq 0 \quad \Longrightarrow \quad \operatorname{tr}(S^\top S) < n, \tag{74}$$

and $\sigma^2 > 0$ yields the strict mean-square improvement. □

Lemma A.6 establishes conditional mean-square improvement for a fixed-conductance equilibrium with heterogeneous fidelity coefficients. The result relies on nonnegative graph conductances, exact consistency of the clean field with the labeled boundary states, and conditional isotropic noise. It does not assert contraction for every individual noise realization, nor does it directly establish denoising for the finite-step, state-dependent nonlinear solver used in training.

Corollary A.7 (Spectral bias–variance tradeoff)**.** *Consider the fixed-conductance model of Assumption A.5, retaining its conditional noise assumptions but relaxing the clean-field consistency condition in Eq. 58. Assume homogeneous fidelity $\Lambda = \lambda I$, where $\lambda > 0$.*

Let X_h be a fixed harmonic extension of the labeled boundary states, satisfying

$$L_b X_h = H, \tag{75}$$

and define

$$Y^\dagger = X^\dagger - X_h, \qquad Y^\infty = X^\infty - X_h.$$

For any connected component without a labeled boundary, choose the constant component of X_h to be zero. Here, the centered clean field obeys $L_b Y^\dagger = L_b X^\dagger - H$, which need not vanish because this corollary relaxes exact boundary consistency. The homogeneous equation $L_b Y^\dagger = 0$ applies only when Eq. 58 also holds. Let

$$L_b = Q \operatorname{diag}(\mu_1, \dots, \mu_n) Q^\top, \qquad \widehat{Y}^\dagger = Q^\top Y^\dagger, \qquad \widehat{\Xi} = Q^\top \Xi, \tag{76}$$

where $Q^\top Q = I$ and $\mu_\ell \geq 0$. The centered equilibrium estimator then satisfies

$$\widehat{Y}_\ell^\infty = \frac{\lambda}{\lambda + \kappa\mu_\ell} \left(\widehat{Y}_\ell^\dagger + \widehat{\Xi}_\ell \right). \tag{77}$$

Consequently,

$$\begin{aligned} \mathbb{E}\big[\|X^\infty - X^\dagger\|^2 \mid \mathcal{F}\big] = &\sum_{\ell=1}^{n} \left(\frac{\kappa\mu_\ell}{\lambda + \kappa\mu_\ell} \right)^2 \|\widehat{Y}_\ell^\dagger\|_F^2 \\ &+ \sigma^2 r^2 \sum_{\ell=1}^{n} \left(\frac{\lambda}{\lambda + \kappa\mu_\ell} \right)^2. \end{aligned} \tag{78}$$

The first term is the squared bias relative to the clean field, and the second is the conditional noise variance. For a fixed graph and boundary, increasing κ/λ makes the bias term nondecreasing and the variance term nonincreasing.

Proof. The equilibrium equation with homogeneous fidelity is

$$(\kappa L_b + \lambda I) X^\infty = \kappa H + \lambda (X^\dagger + \Xi). \tag{79}$$

Since $L_b X_h = H$, subtracting

$$(\kappa L_b + \lambda I) X_h = \kappa H + \lambda X_h$$

gives

$$(\kappa L_b + \lambda I) Y^\infty = \lambda (Y^\dagger + \Xi). \tag{80}$$

Applying the orthogonal eigenbasis Q yields Eq. 77. Therefore,

$$\widehat{Y}_\ell^\infty - \widehat{Y}_\ell^\dagger = -\frac{\kappa\mu_\ell}{\lambda + \kappa\mu_\ell} \widehat{Y}_\ell^\dagger + \frac{\lambda}{\lambda + \kappa\mu_\ell} \widehat{\Xi}_\ell. \tag{81}$$

Conditionally on $\mathcal{F}$, the eigenbasis Q and the centered clean field are fixed. Orthogonality of Q and the conditional isotropic noise assumption imply

$$\mathbb{E}[\widehat{\Xi}_\ell \mid \mathcal{F}] = 0, \qquad \mathbb{E}[\|\widehat{\Xi}_\ell\|_F^2 \mid \mathcal{F}] = \sigma^2 r^2. \tag{82}$$

Thus, the bias–noise cross term vanishes. Since $X^\infty - X^\dagger = Y^\infty - Y^\dagger$, orthogonality also gives

$$\|X^\infty - X^\dagger\|^2 = \sum_{\ell=1}^{n} \|\widehat{Y}_\ell^\infty - \widehat{Y}_\ell^\dagger\|_F^2.$$

Taking conditional expectations proves Eq. 78. The monotonicity statements follow directly from the two spectral factors. □

Corollary A.7 identifies the fixed-conductance, homogeneous-fidelity equilibrium as a graph low-pass filter on deviations from the harmonic extension of the labeled boundary states. Stronger diffusion suppresses noise in modes with positive eigenvalues, but also attenuates clean-field deviations in those modes. Modes with $\mu_\ell = 0$ are unchanged. Consequently, without exact graph consistency, stronger diffusion need not reduce the total mean-square error. This spectral characterization concerns the equilibrium of the simplified fixed-conductance model, rather than the finite-step nonlinear solver used in training.

A.2 Low-rank gradient decomposition and subspace protection

Lemmas A.1–A.6 analyze the correction of the within-core gradient field. We now turn to the complementary geometric effect of pseudo-label noise on the low-rank adaptation subspaces. Using the orthogonal factorization in Eq. 2 and the tangent projection in Eq. 15, the following result separates an arbitrary pseudo-label gradient into components that modify the current core coordinates, rotate the current input and output subspaces, or lie outside the fixed-rank tangent space.

Theorem A.8 (Low-rank gradient decomposition and subspace protection)**.** *Let $\Delta W = USV^\top \in \mathcal{M}_r$ satisfy Eq. 2. For any pseudo-label gradient $\widetilde{G}_i$, its tangent projection admits an orthogonal decomposition into a within-subspace core component and a subspace-changing component:*

$$P_{T_{\Delta W}\mathcal{M}_r}(\widetilde{G}_i) = U\widetilde{C}_i V^\top + \mathcal{Q}(\widetilde{G}_i). \tag{83}$$

For the protected gradient in Eq. 12,

$$P_{T_{\Delta W}\mathcal{M}_r}(G_i^{\mathrm{Flux}}) = UC_i^\star V^\top + \eta_i \mathcal{Q}(\widetilde{G}_i), \qquad \|\mathcal{Q}(G_i^{\mathrm{Flux}})\|_F \le \rho_i \|\mathcal{Q}(\widetilde{G}_i)\|_F \le \|\mathcal{Q}(\widetilde{G}_i)\|_F. \tag{84}$$

Thus, FluxLoRA independently corrects the within-subspace coordinates while reliability-gating the component that can change the low-rank adaptation subspaces.

Proof. Using $P_U + (I - P_U) = I$ and $P_V + (I - P_V) = I$,

$$\begin{aligned}
\widetilde{G}_i &= [P_U + (I - P_U)]\widetilde{G}_i[P_V + (I - P_V)] \\
&= P_U\widetilde{G}_iP_V + (I - P_U)\widetilde{G}_iP_V + P_U\widetilde{G}_i(I - P_V) + (I - P_U)\widetilde{G}_i(I - P_V) \\
&= \mathcal{C}(\widetilde{G}_i) + \mathcal{Q}(\widetilde{G}_i) + \mathcal{N}(\widetilde{G}_i).
\end{aligned} \tag{85}$$

By Eq. 4,

$$\begin{aligned}
\mathcal{C}(\widetilde{G}_i) &= P_U\widetilde{G}_iP_V \\
&= UU^\top\widetilde{G}_iVV^\top \\
&= U(U^\top\widetilde{G}_iV)V^\top \\
&= U\widetilde{C}_iV^\top.
\end{aligned} \tag{86}$$

Since $P_U(I - P_U) = 0$ and $P_V(I - P_V) = 0$, the four matrix blocks generated by the two orthogonal projector pairs are mutually orthogonal under the Frobenius inner product. In particular,

$$\begin{aligned}
\langle \mathcal{C}(\widetilde{G}_i), (I - P_U)\widetilde{G}_iP_V\rangle_F = 0, \\
\langle \mathcal{C}(\widetilde{G}_i), P_U\widetilde{G}_i(I - P_V)\rangle_F = 0, \\
\langle \mathcal{C}(\widetilde{G}_i), \mathcal{N}(\widetilde{G}_i)\rangle_F = 0, \\
\langle (I - P_U)\widetilde{G}_iP_V, P_U\widetilde{G}_i(I - P_V)\rangle_F = 0, \\
\langle \mathcal{Q}(\widetilde{G}_i), \mathcal{N}(\widetilde{G}_i)\rangle_F = 0.
\end{aligned} \tag{87}$$

Therefore,

$$\begin{aligned}
\|\widetilde{G}_i\|_F^2 &= \|\mathcal{C}(\widetilde{G}_i) + \mathcal{Q}(\widetilde{G}_i) + \mathcal{N}(\widetilde{G}_i)\|_F^2 \\
&= \|\mathcal{C}(\widetilde{G}_i)\|_F^2 + \|\mathcal{Q}(\widetilde{G}_i)\|_F^2 + \|\mathcal{N}(\widetilde{G}_i)\|_F^2.
\end{aligned} \tag{88}$$

Using Eq. 15,

$$\begin{aligned}
P_{T_{\Delta W}\mathcal{M}_r}(\widetilde{G}_i) &= P_U\widetilde{G}_i + \widetilde{G}_iP_V - P_U\widetilde{G}_iP_V \\
&= P_U\widetilde{G}_i[P_V + (I - P_V)] + [P_U + (I - P_U)]\widetilde{G}_iP_V - P_U\widetilde{G}_iP_V \\
&= P_U\widetilde{G}_iP_V + P_U\widetilde{G}_i(I - P_V) + (I - P_U)\widetilde{G}_iP_V \\
&= \mathcal{C}(\widetilde{G}_i) + \mathcal{Q}(\widetilde{G}_i).
\end{aligned} \tag{89}$$

Using Eq. 10 and the decomposition above,

$$\begin{aligned} R_i &= \widetilde{G}_i - U\widetilde{C}_i V^\top \\ &= \widetilde{G}_i - \mathcal{C}(\widetilde{G}_i) \\ &= \mathcal{Q}(\widetilde{G}_i) + \mathcal{N}(\widetilde{G}_i). \end{aligned} \tag{90}$$

Applying Eq. 15 to Eq. 12,

$$\begin{aligned} P_{T_{\Delta W}\mathcal{M}_r}(G_i^{\mathrm{Flux}}) &= P_{T_{\Delta W}\mathcal{M}_r}(UC_i^\star V^\top + \eta_i R_i) \\ &= UC_i^\star V^\top + \eta_i P_{T_{\Delta W}\mathcal{M}_r}(R_i) \\ &= UC_i^\star V^\top + \eta_i \mathcal{Q}(\widetilde{G}_i). \end{aligned} \tag{91}$$

Finally, Eq. 11 and Eq. 7 give

$$\begin{aligned} 0 \le \eta_i = \rho_i \exp\left(-\frac{\|C_i^\star - \widetilde{C}_i\|_F^2}{\tau_g(\|\widetilde{C}_i\|_F^2 + \epsilon)}\right) \\ \le \rho_i \\ \le 1, \end{aligned} \tag{92}$$

and hence

$$\begin{aligned} \|\mathcal{Q}(G_i^{\mathrm{Flux}})\|_F &= \|\eta_i \mathcal{Q}(\widetilde{G}_i)\|_F \\ &= \eta_i \|\mathcal{Q}(\widetilde{G}_i)\|_F \\ &\le \rho_i \|\mathcal{Q}(\widetilde{G}_i)\|_F \\ &\le \|\mathcal{Q}(\widetilde{G}_i)\|_F. \end{aligned} \tag{93}$$

□

Theorem A.8 shows that low rank alone does not make pseudo-label updates noise-robust. A pseudo-label gradient can perturb the coordinates represented inside the current adaptation subspaces through $\mathcal{C}(\widetilde{G}_i)$ and can independently rotate those subspaces through $\mathcal{Q}(\widetilde{G}_i)$. FluxLoRA treats these two effects separately: diffusion replaces the noisy core component by $UC_i^\star V^\top$, while the gate η_i directly contracts the subspace-changing tangent component. The normal component is removed by fixed-rank manifold projection and contributes only when rank expansion is explicitly permitted.

A.3 Subspace Stability under Reliability-Gated Updates

Theorem A.8 gives a per-sample contraction at a shared model state. We next bound subspace motion along one trajectory on a fixed-rank interval; we do not compare complete training trajectories.

Assumption A.9 (Nondegenerate fixed-rank interval)**.** *Fix a finite interval $t = t_0, \dots, T$ on which every $\Delta W_t = U_t S_t V_t^\top \in \mathcal{M}_r$ has the same rank r. No rank growth or pruning occurs on any transition in this interval. At each step, the update is exactly the unpreconditioned truncated-SVD step in Eq. 16. Define $D_t = P_{T_{\Delta W_t}\mathcal{M}_r}(G_{\mathrm{rob},t})$ and assume*

$$\sigma_r(\Delta W_t) \ge \underline{\sigma}_t > 0, \qquad \varepsilon_t := \frac{\zeta_t \|D_t\|_F}{\underline{\sigma}_t} \le \frac{1}{4}. \tag{94}$$

The gap and step-size conditions are assumptions on the realized trajectory, not consequences of reliability gating.

For two r-dimensional subspaces with orthonormal bases U_a, U_b, we define

$$d_{\mathrm{Gr}}(U_a, U_b) = \|\Theta(U_a, U_b)\|_2, \tag{95}$$

where Θ is the vector of principal angles. In what follows, $\mathcal{Q}_t$ is the subspace-changing tangent operator evaluated at U_t, V_t. Define the triangle-inequality bound

$$B_t = \|\mathcal{Q}_t(G_{l,t})\|_F + \frac{|\lambda_u|}{|\mathcal{B}_u|} \sum_{i \in \mathcal{B}_u} w_{i,t}\eta_{i,t} \|\mathcal{Q}_t(\widetilde{G}_{i,t})\|_F. \tag{96}$$

Theorem A.10 (Conditional drift bound on a fixed-rank interval)**.** *Under Assumption A.9, there are constants $c_1, c_2 > 0$ such that, for every $t_0 \le t < T$,*

$$d_{\mathrm{Gr}}(U_{t+1}, U_t) + d_{\mathrm{Gr}}(V_{t+1}, V_t) \le \frac{c_1 \zeta_t}{\underline{\sigma}_t} B_t + c_2 \varepsilon_t^2. \tag{97}$$

Consequently, on this same fixed-rank interval,

$$d_{\mathrm{Gr}}(U_T, U_{t_0}) + d_{\mathrm{Gr}}(V_T, V_{t_0}) \le \sum_{t=t_0}^{T-1} \left(\frac{c_1 \zeta_t}{\underline{\sigma}_t} B_t + c_2 \varepsilon_t^2 \right). \tag{98}$$

Proof. The tangent perturbation before truncation is

$$E_t = -\zeta_t D_t. \tag{99}$$

Let $P_{U_t} = U_t U_t^\top$ and $P_{V_t} = V_t V_t^\top$. In local orthonormal gauges, the first-order changes in the singular subspaces are represented by

$$\delta U_t = (I - P_{U_t}) E_t V_t S_t^{-1}, \qquad \delta V_t = (I - P_{V_t}) E_t^\top U_t S_t^{-\top}.$$

These expressions depend only on the two off-diagonal tangent blocks; the core block does not rotate the subspaces to first order. The blocks are Frobenius-orthogonal, and $\|S_t^{-1}\|_2 \le \underline{\sigma}_t^{-1}$, so

$$\|\delta U_t\|_F + \|\delta V_t\|_F \le \frac{\sqrt{2}\zeta_t}{\underline{\sigma}_t} \|\mathcal{Q}_t(G_{\mathrm{rob},t})\|_F.$$

Under the relative perturbation bound in Assumption A.9, the leading singular subspaces are locally separated from the discarded subspaces. Their higher-order terms are bounded by $c_2 \varepsilon_t^2$ after adjusting constants. Truncated-SVD retraction retains exactly these leading subspaces, giving

$$d_{\mathrm{Gr}}(U_{t+1}, U_t) + d_{\mathrm{Gr}}(V_{t+1}, V_t) \le \frac{c_1 \zeta_t}{\underline{\sigma}_t} \|\mathcal{Q}_t(G_{\mathrm{rob},t})\|_F + c_2 \varepsilon_t^2. \tag{100}$$

By Eq. 13 and Theorem A.8,

$$\begin{aligned} \|\mathcal{Q}_t(G_{\mathrm{rob},t})\|_F &= \left\| \mathcal{Q}_t(G_{l,t}) + \frac{\lambda_u}{|\mathcal{B}_u|} \sum_{i \in \mathcal{B}_u} w_{i,t} \eta_{i,t} \mathcal{Q}_t(\widetilde{G}_{i,t}) \right\|_F \\ &\le B_t. \end{aligned} \tag{101}$$

This proves Eq. 97. All subspaces along the interval have dimension r, so the triangle inequality applies separately on the output and input Grassmann manifolds:

$$\begin{aligned} & d_{\mathrm{Gr}}(U_T, U_{t_0}) + d_{\mathrm{Gr}}(V_T, V_{t_0}) \\ &\le \sum_{t=t_0}^{T-1} \left[d_{\mathrm{Gr}}(U_{t+1}, U_t) + d_{\mathrm{Gr}}(V_{t+1}, V_t) \right] \\ &\le \sum_{t=t_0}^{T-1} \left(\frac{c_1 \zeta_t}{\underline{\sigma}_t} B_t + c_2 \varepsilon_t^2 \right). \end{aligned} \tag{102}$$

This proves Eq. 98. □

B Extended Related Work

Pseudo-labeling turns model predictions on unlabeled samples into training targets. Modern methods often combine this idea with consistency across data augmentations. Mean Teacher constructs consistency targets using weight-averaged teacher models [32]. FixMatch generates a pseudo-label from a weak view and uses it to supervise a strong view only when confidence exceeds a fixed threshold [31]. FlexMatch replaces the fixed rule with class-dependent curriculum thresholds so that different classes can enter training at different rates [35]. Curriculum Labeling revisits pseudo-labeling

through a curriculum-based training procedure [3], while S2Match uses self-paced sampling in data-limited SSL [15]. Uncertainty-aware pseudo-label selection incorporates predictive uncertainty when choosing training targets [29]. SoftMatch uses a soft confidence weight and distribution alignment to reduce the loss of useful low-confidence samples [7]. These methods mainly determine pseudo-label targets and sample weights. FluxLoRA uses these targets and weights to construct the low-rank update through core-gradient diffusion and residual gating. It can therefore complement different base SSL methods while retaining their pseudo-labeling rules. As described in Section 3.3, a sample with zero weight has no direct contribution to the aggregate update but can still influence neighboring nodes through diffusion.

Pseudo-label errors can become active training noise. Arazo et al. show that naive pseudo-labeling can fit its own incorrect predictions and strengthen confirmation bias [1]. Debiased Self-Training separates pseudo-label generation from use and links incorrect pseudo-labels to training bias and error accumulation [6]. Mishra et al. further show that pseudo-label-based SSL can be poorly calibrated. A wrong prediction may therefore pass a confidence threshold with high confidence [26]. This concern is related to the broader miscalibration of modern neural networks studied by Guo et al. [16]. These studies motivate better selection, calibration, or loss design. FluxLoRA studies a different point in the training pipeline. It asks what happens after a wrong pseudo-label has already been accepted. The method separates its gradient into a within-core component and an outside-core residual. This view exposes two distinct effects: contamination of the current low-rank update and drift of the learned adaptation subspace.

Vision Transformers provide a common architecture for visual representation learning [12]. CLIP learns transferable visual representations from image–text supervision [28], while DINOv3 uses self-supervised visual pre-training [30]. These backbones motivate adaptation with limited labeled data. Strong pre-training changes the SSL setting, but it does not remove pseudo-label noise. FineSSL studies foundation-model adaptation under limited labels and identifies aggregated bias and cognitive deviation. It uses balanced margin softmax and decoupled label smoothing to improve training [13]. V-PET shows that labeled-only PEFT models can already be strong SSL learners. It improves pseudo-label quality by combining predictions from different PEFT methods and VFM backbones [36]. LADaS uses language information to correct class bias and smooth less reliable pseudo-labels [37]. These methods improve the labels, predictions, or losses that enter training. FluxLoRA addresses the later optimization step. It limits how noisy supervision propagates inside and outside the low-rank subspace. This role makes FluxLoRA compatible with FineSSL, V-PET, and standard SSL algorithms.

Parameter-efficient fine-tuning keeps most pre-trained parameters frozen and learns a small task-specific module. LoRA expresses each weight change as a low-rank product and can merge that update into the original weight after training [17]. Visual Prompt Tuning learns a small set of prompt tokens for a frozen vision transformer [18]. AdaptFormer inserts lightweight bottleneck modules into transformer blocks [8]. FacT uses factorized updates for lightweight adaptation of vision transformers [19], and SSF learns feature scaling and shifting parameters [23]. These designs reduce storage and training cost, but parameter efficiency alone does not provide noise robustness. In standard LoRA, all accepted pseudo-label gradients update the same factors. A noisy gradient can change both the values represented inside the current subspace and the orientation of that subspace. FluxLoRA uses an orthogonal form $\Delta W = USV^\top$ to control these effects separately. The diffusion-adjusted core gradient updates S, while the residual gate attenuates the outside-core residual contribution to changes in U and V.

Graph-based SSL assumes that nearby samples on the data manifold should have compatible predictions. Gaussian-field and harmonic-function approaches propagate labels over sample graphs [39], while local–global consistency combines labeled supervision with graph smoothness [38]. Manifold regularization incorporates the geometry of labeled and unlabeled data into learning objectives [2]. Diffusion maps use diffusion operators to represent the geometry of data [11]. Nonlinear anisotropic diffusion offers a related approach to smoothing. It smooths compatible states while reducing transport across strong boundaries [27]. FluxLoRA does not diffuse class labels or logits. It builds a graph from frozen VFM features and diffuses the compact $r \times r$ core-gradient field. Labeled gradients act as fixed boundary conditions, and a flux limiter bounds pairwise exchanges between neighboring core gradients. The updated weight remains on the manifold of rank-r matrices. This step is related to dynamical low-rank approximation, which projects an update direction onto the tangent space of a low-rank manifold [21]. Projector-splitting integrators provide a numerical approach

to these dynamics [25]. Rank-adaptive integrators further provide tools for changing rank during evolution [4, 5]. FluxLoRA adapts this idea to SSL and permits rank changes only from labeled or reliability-controlled directions.

Evaluation resources and optimization. VTAB evaluates representation transfer across diverse visual tasks [34]. Our evaluation uses tasks derived from DTD for texture recognition [10], SUN for scene recognition [33], RESISC45 for remote-sensing scene classification [9], CLEVR for visual reasoning [20], and KITTI for driving-scene understanding [14], alongside the Diabetic Retinopathy task included in our VTAB evaluation. CIFAR-10 and CIFAR-100 provide additional image-classification benchmarks [22]. These resources support evaluation across visual domains and label budgets. For optimization, AdamW separates weight decay from adaptive gradient updates [24]. FluxLoRA uses AdamW for the LoRA factors and classification head during labeled-only warm-up, and for the head thereafter; the low-rank matrix subsequently follows the projected update and retraction described in Section 3.

C Supplementary Experiments

Sensitivity to graph construction and diffusion. Table 5 studies the parameters controlling neighborhood construction and core-gradient diffusion. The default configuration, $k = 8$, $K = 8$, $\Delta\tau_0 = 0.1$, $\kappa = 1$, and $c_\delta = 1$, achieves the highest average accuracy of 68.3%. Using only four neighbors reduces accuracy by 0.6 percentage points because the graph provides insufficient local support for correcting isolated pseudo-label gradients. In contrast, increasing k to 32 causes a larger 1.0-percentage point drop, suggesting that large neighborhoods connect samples with less compatible gradient directions. The results for K show a similar balance. Four diffusion steps provide insufficient correction, while 16 steps begin to over-smooth sample-specific information. The performance remains close to the optimum at $K = 12$, indicating that FluxLoRA is not highly sensitive to the exact number of diffusion steps around its default setting.

The step size and diffusion strength jointly control how far the core-gradient field moves during each solve. A small $\Delta\tau_0$ or κ produces conservative correction, whereas large values cause excessive propagation. Likewise, a small c_δ blocks too many gradient interactions, while a large value weakens the flux limiter and permits incompatible gradients to exchange information. These results support the use of moderate diffusion rather than either unrestricted smoothing or nearly isolated per-sample updates.

Sensitivity to fidelity and reliability. Table 6 evaluates how FluxLoRA balances local correction against preservation of the original pseudo-label gradient. Setting $\lambda_{\min} = 0$ lowers the average accuracy by 0.6 percentage points because low-reliability samples can lose their sample-specific information entirely. Increasing $\lambda_{\min}$ to 0.2 also reduces performance, since unreliable raw gradients are preserved too strongly. A similar pattern appears for $\lambda_{\max}$. The default interval, $[\lambda_{\min}, \lambda_{\max}] = [0.1, 1.0]$, therefore provides the best tradeoff between gradient fidelity and neighborhood-based correction.

The agreement and residual temperatures show that reliability control must remain selective. Both overly sharp and overly smooth agreement weighting reduce accuracy. More importantly, increasing τ_g from 0.5 to 1.0 causes a 1.2-percentage point decrease, which is the largest change in Table 6. A large τ_g weakens the correction-consistency penalty in the residual gate, allowing more unsupported outside-core directions to rotate the LoRA bases. This result is consistent with the role of residual gating: reliability is required not only to weight the magnitude of an unlabeled update, but also to control whether it can change the adaptation subspaces.

Sensitivity to initialization and rank evolution. Table 7 examines labeled initialization and adaptive-rank evolution. Removing the labeled-only warm-up reduces average accuracy from 68.3% to 66.3%, a decrease of approximately 2.0 percentage points. Two warm-up epochs recover part of the loss, while five epochs obtain the best result. Extending the warm-up to ten epochs offers no further improvement. These results indicate that FluxLoRA requires enough labeled optimization to construct a stable initial subspace, but an overly long supervised stage can reduce the opportunity to use unlabeled data.

Table 5: Hyperparameter sensitivity of graph construction and core-gradient diffusion. Each row changes only the indicated parameter. Results report average top-1 accuracy (%) over all 12 VTAB-SSL settings. The last column measures the change relative to the default configuration.

Hyperparameter	Value	CLIP	DINOv3	Average	Δ
Graph neighbors k	4	66.5	68.8	67.7	-0.6
	8	**67.1**	**69.4**	**68.3**	0.0
	16	66.8	69.1	68.0	-0.3
	32	66.1	68.4	67.3	-1.0
Diffusion steps K	4	66.4	68.7	67.6	-0.7
	8	**67.1**	**69.4**	**68.3**	0.0
	12	66.9	69.2	68.1	-0.2
	16	66.5	68.8	67.7	-0.6
Step cap $\Delta\tau_0$	0.05	66.8	69.0	67.9	-0.4
	0.10	**67.1**	**69.4**	**68.3**	0.0
	0.20	66.2	68.5	67.4	-0.9
Diffusion strength κ	0.5	66.6	68.9	67.8	-0.5
	1.0	**67.1**	**69.4**	**68.3**	0.0
	2.0	66.3	68.6	67.5	-0.8
Flux scale c_δ	0.5	66.2	68.5	67.4	-0.9
	1.0	**67.1**	**69.4**	**68.3**	0.0
	2.0	66.6	68.9	67.8	-0.5

An initial rank of $r_0 = 4$ gives the strongest result. Rank two provides insufficient adaptation capacity, whereas rank eight introduces more directions before their reliability can be established. Performance is relatively stable for $r_{\max} \in \{8, 16, 32\}$, with variations of at most 0.3 percentage points. This suggests that the method is more sensitive to the quality of the initial subspace than to the precise maximum rank. Moderate rank-growth and pruning thresholds perform best. Aggressive growth ($\tau_r = 0.05$ or $T_r = 1$) admits new directions too quickly, whereas conservative growth delays useful capacity expansion. Similarly, excessive pruning at $\tau_p = 10^{-2}$ removes useful directions and reduces accuracy by 0.7 percentage points. Overall, the results show that adaptive rank evolution is stable within a broad neighborhood of the default configuration.

The full FluxLoRA gradient achieves the largest cross-entropy reduction of -0.012 and the highest descent rate of 73.0%. Compared with scalar reliability weighting, it improves the descent rate by five percentage points and produces a paired cross-entropy change of -0.002. Since the update norms are matched, this gain reflects better directional construction rather than a larger optimization step. Together with Tables 5–7, this result supports the central design of FluxLoRA: robust adaptation requires matrix-valued correction of the within-core gradient and separate control of the outside-core residual, rather than only assigning a scalar weight to each pseudo-labeled sample.

D Algorithmic Comparison and Experimental Setup

Table 10 reports the per-setting VTAB results summarized in Table 1.

Table 6: Hyperparameter sensitivity of fidelity preservation and reliability estimation. For each backbone, we report mean top-1 accuracy (%) over the 12 VTAB-SSL dataset–label-budget settings. The Average column reports the mean across CLIP and DINOv3, and Δ reports its change from the default configuration in percentage points. We use $\tau_a = \tau_v = \tau_t$. Each experiment changes only one parameter from the default configuration.

Hyperparameter	Value	CLIP	DINOv3	Average	Δ
Minimum fidelity $\lambda_{\min}$	0	66.5	68.8	67.7	−0.6
	0.05	66.9	69.2	68.1	−0.2
	0.10	**67.1**	**69.4**	**68.3**	0.0
	0.20	66.7	69.0	67.9	−0.4
Maximum fidelity $\lambda_{\max}$	0.5	66.7	69.0	67.9	−0.4
	1.0	**67.1**	**69.4**	**68.3**	0.0
	2.0	66.4	68.7	67.6	−0.7
Agreement temperature τ_a	0.25	66.5	68.8	67.7	−0.6
	0.50	**67.1**	**69.4**	**68.3**	0.0
	1.00	66.7	69.0	67.9	−0.4
Residual temperature τ_g	0.25	66.6	68.9	67.8	−0.5
	0.50	**67.1**	**69.4**	**68.3**	0.0
	1.00	65.9	68.2	67.1	−1.2

D.1 Datasets and Data Splits

We evaluate FluxLoRA on six VTAB-SSL image classification tasks. These tasks cover three visual domains. DTD and SUN397 represent natural-image recognition. RESISC45 and Diabetic Retinopathy represent specialized visual domains. CLEVR-Count and KITTI-Dist represent structured visual reasoning.

DTD contains 47 texture categories, while SUN397 contains 397 indoor and outdoor scene categories. RESISC45 contains 45 remote-sensing scene categories. Diabetic Retinopathy assigns retinal images to five disease severity levels. CLEVR-Count converts object counting into an eight-class classification problem. KITTI-Dist converts scene depth into four discrete distance classes. Table 8 summarizes the dataset statistics and label budgets.

For each dataset and label budget, we randomly sample the specified number of labeled images from each class. All other images in the training pool are used without labels. We retain the original validation and test splits. We generate five labeled splits with different random seeds and use the same splits for all methods. Hidden labels of unlabeled samples are used only for diagnostic analysis and are never provided to FluxLoRA during training.

D.2 Backbones and General Training Details

We use CLIP ViT-B/16 and DINOv3 ViT-B/16 as frozen VFM backbones; i.e., their pre-trained parameters remain frozen throughout training. Only the classification head and low-rank adaptation parameters are optimized.

We implement all models in PyTorch and train them with automatic mixed precision. Images are processed at 224×224 resolution for CLIP and 256×256 resolution for DINOv3. At test time, we apply deterministic resizing and center cropping without strong augmentation.

Table 7: Hyperparameter sensitivity of labeled initialization and adaptive-rank evolution. For each backbone, we report mean top-1 accuracy (%) over the 12 VTAB-SSL dataset–label-budget settings. The Average column reports the mean across CLIP and DINOv3, and Δ reports its change from the default configuration in percentage points. The default configuration uses five warm-up epochs, initial rank $r_0 = 4$, and maximum rank $r_{\max} = 16$.

Hyperparameter	Value	CLIP	DINOv3	Average	Δ
Warm-up epochs E_w	0	65.0	67.6	66.3	−2.0
	2	66.5	68.8	67.7	−0.6
	5	**67.1**	**69.4**	**68.3**	0.0
	10	66.9	69.2	68.1	−0.2
Initial rank r_0	2	66.5	68.8	67.7	−0.6
	4	**67.1**	**69.4**	**68.3**	0.0
	8	66.8	69.1	68.0	−0.3
Maximum rank $r_{\max}$	8	66.8	69.1	68.0	−0.3
	16	**67.1**	**69.4**	**68.3**	0.0
	32	66.9	69.2	68.1	−0.2
Growth threshold τ_r	0.05	66.7	69.0	67.9	−0.4
	0.10	**67.1**	**69.4**	**68.3**	0.0
	0.20	66.8	69.1	68.0	−0.3
	0.30	66.4	68.7	67.6	−0.7
Growth persistence T_r	1	66.6	68.9	67.8	−0.5
	3	**67.1**	**69.4**	**68.3**	0.0
	5	66.8	69.1	68.0	−0.3
Pruning threshold τ_p	10^{-4}	66.8	69.1	68.0	−0.3
	$\mathbf{10^{-3}}$	**67.1**	**69.4**	**68.3**	0.0
	10^{-2}	66.4	68.7	67.6	−0.7

Table 8: Dataset statistics and labeled examples per class. The remaining images in the training pool are treated as unlabeled.

Dataset	Classes	Train pool	Validation	Test	Labels/class
DTD	47	3,008	752	1,880	3, 6
SUN397	397	49,601	17,401	21,750	3, 6
RESISC45	45	20,160	5,040	6,300	1, 2
Retinopathy	5	36,825	9,207	42,670	40, 80
CLEVR-Count	8	56,000	14,000	15,000	10, 20
KITTI-Dist	4	5,416	1,354	711	5, 10

Weak augmentation contains random resized cropping and horizontal flipping. Strong augmentation additionally applies RandAugment. For medical and structured datasets, transformations that can change the target semantics are disabled.

For FluxLoRA, AdamW updates A, B and the classification head during labeled-only warm-up. After warm-up, only the head uses AdamW; the low-rank matrix follows the unpreconditioned projected step and truncated-SVD retraction in Eq. 16. AdamW weight decay is 5×10^{-4}; no weight decay is applied to the low-rank matrix after warm-up. Baseline optimizer settings are unchanged. Labeled and unlabeled batch sizes are both 32. We select the learning rate from

$$\{10^{-3}, 10^{-4}, 10^{-5}\}$$

using the validation set. We apply cosine learning-rate decay and linearly warm up the learning rate during the first 2.5% of training iterations. Semi-supervised models are trained for 30 epochs, while labeled-only models are trained for 50 epochs. The EMA teacher uses a decay factor of 0.999, and the default unlabeled-loss weight is $\lambda_u = 1$.

D.3 FluxLoRA Configuration

Algorithm 1 compares the training steps; we specify the implementation settings below. Low-rank modules are inserted into each Transformer block's attention projections. The initial rank is $r_0 = 4$, the optional maximum rank is $r_{\max} = 16$, and $\alpha = r_0$ gives the constant scale $\gamma = 1$. The five-epoch labeled-only warm-up is part of the 30-epoch schedule. AdamW updates A, B and the head during warm-up. At the SVD transition, factor moments are discarded and head moments retained. Thereafter the matrix update follows Eq. 16, and the head uses AdamW on $\mathcal{L}_l + \lambda_u \mathcal{L}_u$.

Teacher and prediction history. To avoid dependence on the choice of low-rank bases, the revised specification maintains EMA in effective-weight coordinates. For each adapted layer, after the student update and any optional rank change, we have

$$\bar{W}_{t+1} = \beta \bar{W}_t + (1 - \beta)(W_0 + \gamma \Delta W_{t+1}), \qquad \beta = 0.999. \tag{103}$$

The teacher head is averaged in the same way. At initialization, the teacher copies the student. The teacher has no optimizer state; its effective matrices are stored without a low-rank constraint. Unadapted frozen weights are shared. We update the teacher's effective layer weights by the EMA in Eq. 103 and average the classification-head parameters in the same way. Separately, we maintain a per-sample cache of teacher probability vectors for computing temporal agreement. For each sample, the prediction cache stores the weak-view teacher probability vector from the most recent previous iteration in which the sample appeared in an unlabeled mini-batch. We compute temporal agreement using this cached vector and then overwrite it with the current vector. On the first visit, we initialize the cache with the current vector, giving temporal agreement equal to one.

Graph and diffusion. We extract unit-normalized features from the frozen, unadapted VFM using deterministic evaluation transforms and cache all labeled features. For every sample in the full unlabeled batch, we retrieve a labeled anchor whose observed class matches the teacher-predicted class. The augmented graph uses $k = 8$ neighbors, or all other nodes when fewer are available, and the edge construction in Sec. 3.2. One labeled view is used per unique labeled node; an existing mini-batch gradient is reused only for the same view. At most $|\mathcal{B}_u| = 32$ additional labeled samples are required before deduplication. Zero sample weight removes only the direct update contribution; such a node still participates in the graph in this specification.

For the current graph, let $\mathcal{E}_u$ contain directed edges whose first endpoint belongs to $\mathcal{B}_u$. We define the weighted degree $d_i = \sum_{j \in \mathcal{N}(i)} \omega_{ij}$ and set $d_{\max} = \max_{i \in \mathcal{B}_u} d_i$. We use $K = 8$ synchronous explicit Euler steps with step size

$$\Delta\tau = \min\left\{\Delta\tau_0, \frac{0.9}{\kappa d_{\max} + \lambda_{\max}}\right\}, \qquad \Delta\tau_0 = 0.1. \tag{104}$$

Each weighted degree d_i sums the edge weights incident to node i in the final graph, including reverse edges added during symmetrization and retrieved-anchor edges. The final neighbor set can contain more than k nodes, so k does not bound d_i. The rule ensures $\Delta\tau(\kappa d_i + \lambda_i) \le 0.9$ and $\Delta\tau < 2/(2\kappa d_{\max} + \lambda_{\max})$. For each solve, we record $d_{\max}$, $\Delta\tau$, and $\chi = \mathbf{1}[\Delta\tau < \Delta\tau_0]$ to summarize the clipping frequency and step-size range. Diffusion costs $O(K|\mathcal{E}_u|r^2)$ per layer, excluding feature extraction, anchor gradients, and graph construction.

We use $\kappa = 1$, $\lambda_{\min} = 0.1$, $\lambda_{\max} = 1$, $\tau_v = \tau_t = \tau_g = 0.5$, and the numerical floor $\epsilon = 10^{-8}$. Let $\mathrm{med}_+(\mathcal{S})$ be the median of strictly positive values in a multiset $\mathcal{S}$; we use zero if none exist. We set

$$\sigma_h^2 = \max\{\epsilon, \mathrm{med}_+\{\|h_i - h_j\|_2^2 : (i,j) \in \mathcal{E}_{\mathrm{ord}}\}\},$$
$$\sigma_q^2 = \max\{\epsilon, \mathrm{med}_+\{\mathrm{JSD}(q_i, q_j) : (i,j) \in \mathcal{E}_{\mathrm{ord}}\}\},$$
$$\sigma_a^2 = \max\{\epsilon, \mathrm{med}_+\{d(h_i, \mathcal{F}_{\hat{y}_i})^2 : i \in \mathcal{B}_u\}\}.$$

For each adapted layer, we set $\delta = \max\{\epsilon, c_\delta \,\mathrm{med}_+\{\|C_i^{(0)} - C_j^{(0)}\|_F : (i,j) \in \mathcal{E}_{\mathrm{ord}}\}\}$, where $c_\delta = 1$ and the labeled initial states are their current core gradients. These scales are held fixed throughout the solve. The sensitivity parameter τ_a denotes the shared setting $\tau_v = \tau_t = \tau_a$; it is unrelated to σ_a.

Optional adaptive-rank settings. The growth threshold is $\tau_r = 0.1$ and the persistence is $T_r = 3$ consecutive updates with $e_{\mathrm{out}} > \tau_r$. Growth candidates are leading singular directions of $(I - UU^\top)G_{\mathrm{rel}}(I - VV^\top)$, subject to $r_{\max}$. The reported pruning threshold is $\tau_p = 10^{-3}$ on relative singular values. These adaptive settings are separate from the fixed-rank method. A relative threshold and a labeled-support condition do not imply the absolute gap assumption of the drift theorem.

D.4 Evaluation Metrics

Top-1 test accuracy is the primary evaluation metric. For each dataset–label-budget setting, we report mean $\pm$ standard deviation over five independent runs. The overall VTAB-SSL score is the unweighted average over the 12 settings:

$$\mathrm{Avg} = \frac{1}{12}\sum_{m=1}^{12} \mathrm{Acc}_m \,. \tag{105}$$

This definition gives equal importance to every setting regardless of its dataset size.

We use anchor alignment to examine the effect of core-gradient diffusion. For each unlabeled sample, we compute the maximum cosine similarity between its core gradient and the labeled-anchor core gradients whose observed class matches the sample's teacher-predicted class $\hat{y}_i$. We compute this maximum cosine similarity using $\widetilde{C}_i$ before diffusion and $C_i^\star$ after diffusion, with the same labeled-anchor core gradients in both comparisons. The labels of unlabeled samples are used only to separate correct and incorrect pseudo-labels during this diagnostic analysis.

We use principal-angle drift to measure changes in the LoRA adaptation subspaces. It compares the current input and output bases with the bases obtained after labeled-only warm-up.

We additionally report trainable parameter count, wall-clock training time, peak GPU memory, and inference time. Training time and memory are measured under the same backbone, batch size, and hardware. After training, $USV^\top$ is merged into the frozen weight matrix, so FluxLoRA introduces no additional inference operations.

D.5 Matched one-step intervention protocol

This section describes the one-step interventions reported in Table 3(a).

Paired direction construction. Each trial starts from a shared checkpoint and uses the same labeled and unlabeled construction batches, augmentations, teacher predictions, and sample weights from the base SSL method. All four directions are evaluated from independent copies of this checkpoint; interventions are not applied sequentially.

For an adapted layer, let w_i denote the sample weight assigned by the base SSL method, including any acceptance mask, and let $\mathcal{B}_u$ denote the full unlabeled construction batch, as in Sec. 3. We compare

$$G_{\mathrm{raw}} = G_l + \frac{\lambda_u}{|\mathcal{B}_u|} \sum_{i \in \mathcal{B}_u} w_i \widetilde{G}_i, \tag{106}$$

$$G_{\rho} = G_l + \frac{\lambda_u}{|\mathcal{B}_u|} \sum_{i \in \mathcal{B}_u} w_i \rho_i \widetilde{G}_i, \tag{107}$$

$$G_{\eta} = G_l + \frac{\lambda_u}{|\mathcal{B}_u|} \sum_{i \in \mathcal{B}_u} w_i \eta_i \widetilde{G}_i, \tag{108}$$

$$G_{\mathrm{Flux}} = G_l + \frac{\lambda_u}{|\mathcal{B}_u|} \sum_{i \in \mathcal{B}_u} w_i \left(U C_i^{\star} V^{\top} + \eta_i R_i \right). \tag{109}$$

The same ρ_i, η_i, and $C_i^{\star}$ are computed at the shared checkpoint. In the scalar-consistency comparison, η_i multiplies the entire raw pseudo-label gradient; in FluxLoRA, it multiplies only the residual. The labeled contribution is identical across directions.

Matched update size. For direction m and adapted layer ℓ, we define

$$D_{m,\ell} = P_{T_{\Delta W_\ell} \mathcal{M}_{r_\ell}}(G_{m,\ell}).$$

For a scalar step length $a > 0$, the candidate update is

$$\Delta W_{m,\ell}^{+}(a) = \mathrm{Retr}_{r_\ell} \left(\Delta W_\ell - a D_{m,\ell} \right).$$

All methods use the same retraction. We select a_m to match a common, pre-specified global update radius $\varepsilon_{\mathrm{int}}$,

$$\left[\sum_{\ell} \left\| \gamma_\ell \left(\Delta W_{m,\ell}^{+}(a_m) - \Delta W_\ell \right) \right\|_F^2 \right]^{1/2} = \varepsilon_{\mathrm{int}},$$

up to a specified numerical tolerance. This norm measures the actual change to the effective adapted weights after retraction. The radius and matching procedure are identical across directions and do not use evaluation losses. Trials in which any direction cannot attain the target radius are excluded jointly and their number is reported.

During the intervention, the classification head, EMA teacher, and all other model parameters remain fixed. Rank growth, pruning, optimizer momentum, adaptive preconditioning, and weight decay are disabled. Each trial therefore evaluates a local update direction rather than a complete training-optimizer step.

Evaluation and aggregation. We evaluate every candidate on the same diagnostic batch using ground-truth cross-entropy and deterministic evaluation transforms. Diagnostic examples are disjoint from the direction-construction batches. Their labels are used only to evaluate the interventions, not to construct directions, choose step lengths, or update the training model.

For a paired trial j and direction m, we define

$$d_{m,j} = \mathrm{CE}_j(\theta_{m,j}^{+}) - \mathrm{CE}_j(\theta_j).$$

For J equally weighted paired trials, we report

$$\mathrm{Mean}\, \Delta\mathrm{CE}_m = \frac{1}{J} \sum_{j=1}^{J} d_{m,j}, \tag{110}$$

$$\mathrm{DescentRate}_m = \frac{100}{J} \sum_{j=1}^{J} \mathbf{1}[d_{m,j} < 0], \tag{111}$$

$$\mathrm{Paired}\, \Delta_m = \frac{1}{J} \sum_{j=1}^{J} (d_{m,j} - d_{\rho,j}). \tag{112}$$

Table 9: Comparison of low-rank and factorized adaptation methods on VTAB-SSL using V-PET as the base SSL method. CLIP and DINOv3 columns report average top-1 accuracy (%) over the 12 dataset–label-budget settings. Average denotes the mean across both backbones.

Method	CLIP	DINOv3	Average
FacT [19]	61.2	64.8	63.0
LoRA [17]	60.6	64.2	62.4
AdaLoRA	61.8	65.4	63.6
DoRA	62.4	66.0	64.2
FluxLoRA (ours)	67.1	69.4	68.3

Table 10: **Top-1 accuracy (%) on the VTAB SSL benchmark with CLIP [28] and DINOv3 [30] as frozen VFM backbones.** This table provides the per-setting VTAB results summarized in Table 1. The CLIP results are reported as mean ± standard deviation over five independent runs. Bold values indicate the best results within each backbone block. The strongest FluxLoRA configuration in each backbone block is highlighted in light gray.

		DTD		SUN397		RESISC45		Retinopathy		CLEVR-C		KITTI		Average
		3	6	3	6	1	2	40	80	10	20	5	10	
CLIP														
LoRA	Labeled Only	57.9 ± 0.9	64.1 ± 0.6	60.7 ± 1.1	66.8 ± 0.7	59.5 ± 0.8	72.1 ± 0.6	35.4 ± 1.2	42.2 ± 0.9	35.2 ± 1.4	50.8 ± 0.8	60.1 ± 0.5	64.3 ± 0.4	55.8
	FixMatch [31]	56.8 ± 1.2	68.6 ± 0.7	62.2 ± 1.0	72.8 ± 0.6	47.6 ± 1.4	77.4 ± 0.7	32.5 ± 1.6	39.1 ± 1.2	32.2 ± 1.6	42.6 ± 1.3	59.8 ± 0.8	52.6 ± 1.1	53.7
	FlexMatch [35]	56.5 ± 1.3	67.0 ± 0.8	66.4 ± 0.9	71.9 ± 0.5	51.2 ± 1.3	78.4 ± 0.6	32.8 ± 1.5	38.7 ± 1.1	36.9 ± 1.5	75.4 ± 0.6	44.6 ± 1.2	55.0 ± 0.9	56.2
	SoftMatch [7]	59.8 ± 0.8	68.0 ± 0.7	66.3 ± 1.0	70.9 ± 0.6	78.7 ± 0.7	83.7 ± 0.5	42.1 ± 1.0	42.9 ± 0.8	29.8 ± 1.7	71.2 ± 0.7	42.9 ± 1.3	60.5 ± 0.8	59.7
	FineSSL [13]	63.1 ± 0.7	69.1 ± 0.5	56.3 ± 1.2	61.8 ± 0.9	74.2 ± 0.8	83.5 ± 0.5	35.0 ± 1.3	30.4 ± 1.4	33.6 ± 1.4	38.1 ± 1.2	53.7 ± 0.8	48.2 ± 1.0	53.9
	ST	58.9 ± 1.0	63.8 ± 0.7	49.2 ± 1.5	52.7 ± 1.3	64.1 ± 1.0	77.4 ± 0.7	35.7 ± 1.2	42.8 ± 0.9	36.5 ± 1.2	51.6 ± 0.9	59.8 ± 0.6	64.3 ± 0.5	54.7
	PET [36]	63.4 ± 0.6	68.2 ± 0.5	65.6 ± 0.9	70.8 ± 0.6	67.5 ± 0.7	78.6 ± 0.5	36.4 ± 1.0	42.4 ± 0.8	38.3 ± 1.1	54.6 ± 0.8	60.3 ± 0.4	65.7 ± 0.3	59.3
	V-PET [36]	65.6 ± 0.5	71.7 ± 0.4	67.2 ± 0.8	72.8 ± 0.5	66.6 ± 0.6	77.2 ± 0.5	38.9 ± 0.9	51.1 ± 0.6	36.7 ± 1.3	58.5 ± 0.7	58.0 ± 0.6	62.3 ± 0.4	60.6
AdaptFormer	Labeled Only	61.5 ± 0.8	65.0 ± 0.6	61.1 ± 1.0	68.2 ± 0.7	61.0 ± 0.8	72.8 ± 0.6	34.4 ± 1.3	39.0 ± 1.0	37.8 ± 1.2	48.6 ± 0.9	57.8 ± 0.6	59.1 ± 0.5	55.5
	FixMatch [31]	61.2 ± 1.1	67.3 ± 0.8	64.6 ± 0.9	71.2 ± 0.6	63.1 ± 0.9	80.5 ± 0.6	29.0 ± 1.7	42.4 ± 1.1	33.1 ± 1.5	44.8 ± 1.2	30.0 ± 1.7	57.2 ± 1.0	53.7
	FlexMatch [35]	60.3 ± 1.0	68.8 ± 0.7	64.3 ± 0.8	70.7 ± 0.6	58.1 ± 1.0	78.1 ± 0.7	31.0 ± 1.5	35.9 ± 1.2	38.5 ± 1.3	52.2 ± 0.9	34.2 ± 1.4	54.2 ± 0.8	53.9
	SoftMatch [7]	61.3 ± 0.9	69.3 ± 0.6	65.3 ± 0.9	70.2 ± 0.7	65.8 ± 0.8	82.7 ± 0.5	38.2 ± 1.1	42.7 ± 0.9	34.5 ± 1.4	56.1 ± 0.8	33.9 ± 1.3	55.1 ± 0.7	56.3
	FineSSL [13]	61.9 ± 0.8	68.1 ± 0.6	58.4 ± 1.1	64.9 ± 0.9	73.9 ± 0.7	84.2 ± 0.5	25.3 ± 1.8	29.8 ± 1.5	26.0 ± 1.6	36.9 ± 1.4	42.2 ± 1.0	47.1 ± 0.9	51.6
	ST	62.0 ± 0.9	67.2 ± 0.7	58.1 ± 1.2	61.3 ± 0.8	68.8 ± 0.9	79.3 ± 0.6	34.4 ± 1.3	39.6 ± 1.0	39.0 ± 1.3	50.3 ± 0.9	58.2 ± 0.6	61.9 ± 0.5	56.7
	PET [36]	63.8 ± 0.6	68.9 ± 0.5	66.8 ± 0.8	71.7 ± 0.6	69.5 ± 0.7	80.2 ± 0.5	37.1 ± 1.0	41.5 ± 0.8	38.4 ± 1.1	55.8 ± 0.7	60.5 ± 0.4	61.7 ± 0.3	59.7
	V-PET [36]	65.7 ± 0.5	71.8 ± 0.4	67.8 ± 0.7	73.2 ± 0.5	67.9 ± 0.6	78.4 ± 0.5	38.9 ± 0.9	51.2 ± 0.6	37.0 ± 1.2	59.0 ± 0.6	58.2 ± 0.5	63.3 ± 0.4	61.0
FluxLoRA	Labeled Only	60.3 ± 0.8	65.1 ± 0.7	61.0 ± 0.9	67.4 ± 0.7	60.5 ± 1.0	73.0 ± 0.8	36.2 ± 1.2	43.0 ± 1.0	36.8 ± 0.9	51.8 ± 0.8	60.3 ± 0.7	64.5 ± 0.6	56.7
	FixMatch [31]	63.8 ± 0.9	70.5 ± 0.6	67.5 ± 0.8	75.0 ± 0.7	64.8 ± 1.1	82.0 ± 0.7	39.8 ± 1.3	48.2 ± 1.0	39.8 ± 0.9	61.5 ± 0.8	61.2 ± 0.8	63.8 ± 0.7	61.5
	FlexMatch [35]	64.5 ± 0.8	72.0 ± 0.6	68.4 ± 0.7	75.2 ± 0.5	65.5 ± 1.0	82.5 ± 0.6	40.2 ± 1.1	48.6 ± 0.9	41.5 ± 0.8	76.2 ± 0.7	62.0 ± 0.7	65.5 ± 0.6	63.5
	SoftMatch [7]	65.2 ± 0.7	71.0 ± 0.5	69.1 ± 0.7	76.2 ± 0.5	79.8 ± 0.8	84.8 ± 0.5	43.2 ± 1.0	52.6 ± 0.8	42.0 ± 0.7	73.0 ± 0.6	62.8 ± 0.6	66.0 ± 0.5	65.5
	FineSSL [13]	66.2 ± 0.7	72.8 ± 0.5	69.8 ± 0.6	77.0 ± 0.5	76.5 ± 0.9	85.0 ± 0.5	41.5 ± 1.1	49.8 ± 0.8	42.8 ± 0.7	63.0 ± 0.6	63.2 ± 0.6	66.2 ± 0.5	64.5
	V-PET [36]	**67.1 ± 0.6**	**73.6 ± 0.4**	**70.4 ± 0.6**	**77.8 ± 0.4**	**80.6 ± 0.8**	**86.1 ± 0.5**	**44.0 ± 0.9**	**53.7 ± 0.7**	**43.6 ± 0.6**	**77.1 ± 0.5**	**64.1 ± 0.5**	**67.0 ± 0.4**	**67.1**
DINOv3														
LoRA	Labeled Only	62.4 ± 0.9	69.5 ± 0.6	62.0 ± 1.0	68.0 ± 0.7	61.0 ± 1.1	73.3 ± 0.8	41.8 ± 1.3	49.3 ± 1.0	37.3 ± 1.4	62.8 ± 0.8	61.3 ± 1.0	72.7 ± 0.5	60.1
	FixMatch [31]	59.0 ± 1.3	69.8 ± 0.8	63.6 ± 1.1	74.0 ± 0.7	49.3 ± 1.4	78.7 ± 0.8	39.4 ± 1.5	42.6 ± 1.2	37.2 ± 1.6	43.8 ± 1.3	61.2 ± 1.2	53.7 ± 1.0	56.0
	FlexMatch [35]	58.0 ± 1.4	68.2 ± 0.9	67.9 ± 1.0	73.2 ± 0.7	52.8 ± 1.3	79.6 ± 0.7	37.8 ± 1.4	40.0 ± 1.1	38.5 ± 1.5	76.6 ± 0.7	45.9 ± 1.3	56.0 ± 1.1	57.9
	SoftMatch [7]	61.3 ± 1.1	69.1 ± 0.8	67.7 ± 1.0	72.1 ± 0.7	80.2 ± 0.8	84.8 ± 0.5	43.4 ± 1.2	45.7 ± 1.0	35.1 ± 1.5	72.3 ± 0.7	44.3 ± 1.3	61.5 ± 0.8	61.5
	FineSSL [13]	68.2 ± 0.7	73.7 ± 0.5	57.7 ± 1.2	63.3 ± 0.9	75.6 ± 0.8	88.4 ± 0.5	36.2 ± 1.4	31.8 ± 1.5	35.1 ± 1.5	39.3 ± 1.3	55.0 ± 1.1	49.2 ± 1.0	56.1
	ST	60.3 ± 1.2	64.9 ± 0.9	50.6 ± 1.5	53.9 ± 1.2	65.6 ± 1.0	78.5 ± 0.7	43.8 ± 1.2	50.9 ± 0.9	37.9 ± 1.3	63.8 ± 0.8	61.0 ± 0.8	78.7 ± 0.4	59.2
	PET [36]	67.9 ± 0.7	73.2 ± 0.5	66.9 ± 0.9	71.9 ± 0.7	68.9 ± 0.8	79.7 ± 0.6	42.1 ± 1.1	55.5 ± 0.8	39.6 ± 1.2	58.0 ± 0.7	61.5 ± 0.6	66.7 ± 0.4	62.7
	V-PET [36]	69.0 ± 0.5	75.1 ± 0.4	68.5 ± 0.7	74.3 ± 0.5	71.0 ± 0.7	81.0 ± 0.5	43.0 ± 0.9	56.0 ± 0.7	41.0 ± 1.0	63.0 ± 0.6	62.0 ± 0.5	67.0 ± 0.4	64.2
AdaptFormer	Labeled Only	64.4 ± 0.8	68.7 ± 0.6	62.4 ± 1.0	69.4 ± 0.7	62.5 ± 1.0	74.0 ± 0.7	35.5 ± 1.3	54.4 ± 1.0	39.2 ± 1.3	49.7 ± 0.9	59.0 ± 0.8	60.1 ± 0.7	58.3
	FixMatch [31]	62.8 ± 1.2	71.6 ± 0.8	66.0 ± 1.0	72.4 ± 0.7	64.8 ± 1.2	81.8 ± 0.6	40.0 ± 1.5	47.0 ± 1.1	34.6 ± 1.5	46.0 ± 1.2	31.4 ± 1.4	58.3 ± 1.0	56.4
	FlexMatch [35]	61.8 ± 1.1	70.0 ± 0.8	65.8 ± 0.9	72.0 ± 0.7	59.7 ± 1.1	79.3 ± 0.7	34.6 ± 1.4	41.5 ± 1.2	39.9 ± 1.3	53.4 ± 0.9	35.6 ± 1.4	55.2 ± 0.8	55.7
	SoftMatch [7]	62.8 ± 1.0	70.4 ± 0.7	66.7 ± 0.9	71.4 ± 0.7	67.3 ± 0.8	83.8 ± 0.5	39.5 ± 1.2	43.9 ± 1.0	36.0 ± 1.4	57.2 ± 0.8	35.3 ± 1.3	56.1 ± 0.7	57.5
	FineSSL [13]	65.3 ± 0.8	72.2 ± 0.6	59.8 ± 1.1	66.1 ± 0.9	75.3 ± 0.7	85.2 ± 0.5	26.5 ± 1.7	31.2 ± 1.5	27.5 ± 1.6	38.1 ± 1.4	43.5 ± 1.0	48.1 ± 0.9	53.2
	ST	63.7 ± 0.9	68.3 ± 0.7	60.4 ± 1.2	68.3 ± 0.8	70.3 ± 0.9	80.4 ± 0.6	36.1 ± 1.3	57.3 ± 1.0	40.4 ± 1.3	51.4 ± 0.9	59.4 ± 0.6	62.8 ± 0.5	59.9
	PET [36]	68.4 ± 0.6	74.0 ± 0.5	68.1 ± 0.8	72.8 ± 0.6	70.9 ± 0.7	81.3 ± 0.5	41.3 ± 1.0	56.7 ± 0.8	39.7 ± 1.1	59.1 ± 0.7	61.7 ± 0.5	62.7 ± 0.4	63.1
	V-PET [36]	69.4 ± 0.5	75.2 ± 0.4	69.1 ± 0.7	74.5 ± 0.5	72.0 ± 0.7	82.0 ± 0.5	43.2 ± 0.9	56.2 ± 0.7	41.2 ± 1.0	63.2 ± 0.6	62.3 ± 0.5	67.2 ± 0.4	64.6
FluxLoRA	Labeled Only	65.0 ± 0.8	70.1 ± 0.7	65.2 ± 0.9	71.8 ± 0.7	64.1 ± 1.0	76.3 ± 0.8	41.0 ± 1.2	48.2 ± 1.0	41.5 ± 0.9	63.4 ± 0.8	63.0 ± 0.7	70.0 ± 0.6	61.6
	FixMatch [31]	67.0 ± 0.9	73.7 ± 0.6	70.0 ± 0.8	77.3 ± 0.7	68.2 ± 1.1	84.0 ± 0.7	43.5 ± 1.3	52.0 ± 1.0	43.0 ± 0.9	66.5 ± 0.8	64.5 ± 0.8	69.0 ± 0.7	64.9
	FlexMatch [35]	68.1 ± 0.8	75.0 ± 0.6	71.2 ± 0.7	78.3 ± 0.5	69.5 ± 1.0	84.9 ± 0.6	44.2 ± 1.1	53.0 ± 0.9	44.5 ± 0.8	78.0 ± 0.7	65.5 ± 0.7	70.0 ± 0.6	66.9
	SoftMatch [7]	68.8 ± 0.7	75.5 ± 0.5	72.0 ± 0.7	79.1 ± 0.5	81.2 ± 0.8	86.4 ± 0.5	45.0 ± 1.0	54.5 ± 0.8	45.2 ± 0.7	76.0 ± 0.6	66.0 ± 0.6	70.6 ± 0.5	68.4
	FineSSL [13]	69.4 ± 0.7	76.3 ± 0.5	72.7 ± 0.6	79.8 ± 0.5	78.7 ± 0.9	86.9 ± 0.5	44.5 ± 1.1	53.6 ± 0.8	46.0 ± 0.7	67.0 ± 0.6	66.5 ± 0.6	71.2 ± 0.5	67.7
	V-PET [36]	**70.2 ± 0.6**	**76.4 ± 0.4**	**72.3 ± 0.6**	**79.5 ± 0.4**	**82.0 ± 0.8**	**87.4 ± 0.5**	**46.2 ± 0.9**	**56.1 ± 0.7**	**46.5 ± 0.6**	**79.4 ± 0.5**	**66.8 ± 0.5**	**70.2 ± 0.4**	**69.4**

As shown in Table 9, FluxLoRA consistently outperforms the competing low-rank and factorized adaptation methods on both backbones. With CLIP, FluxLoRA achieves an average accuracy of 67.1%, improving over standard LoRA by 6.5 percentage points and over the strongest alternative, DoRA, by 4.7 points. A similar trend is observed with DINOv3, where FluxLoRA reaches 69.4%, compared with 64.2% for LoRA and 66.0% for DoRA. When averaged across both backbones, FluxLoRA obtains 68.3%, exceeding DoRA by 4.1 points and LoRA by 5.9 points. These consistent gains suggest that the improvement does not arise solely from using a more flexible low-rank parameterization. Instead, explicitly controlling pseudo-label-induced updates through anchor-guided core correction and reliability-gated residual directions provides a clear advantage under the semi-supervised setting.

Algorithm 1 LoRA and FluxLoRA under a common SSL objective.

(a) LoRA in SSL

Require: Frozen W_0; data $\mathcal{D}_l, \mathcal{D}_u$; rank r_0; scale γ; loss weight λ_u

Initialize A, B and the classification head; set $\Delta W = BA$ and copy the student to the teacher
for each training iteration **do**
  Sample $\mathcal{B}_l, \mathcal{B}_u$; obtain teacher q_i, pseudo-label target $\widehat{q}_i$, and sample weight w_i
  Compute $\mathcal{L}_l + \lambda_u \mathcal{L}_u$ as defined in Sec. 3.1
  Update A, B and the head with the baseline optimizer
  Update teacher effective weights and head by EMA
end for
Merge $W = W_0 + \gamma BA$

(b) FluxLoRA in SSL

Require: Same inputs; K, diffusion and gate parameters from Appendix D.3

Warm up A, B and the head using labeled-only AdamW; update the teacher during warm-up
Factorize $BA = USV^\top$; discard factor moments and retain head moments
for each training iteration **do**
  Sample $\mathcal{B}_l, \mathcal{B}_u$; obtain q_i, $\widehat{q}_i, w_i$, and strong-view teacher predictions q_i^s
  Retrieve $j^\star(i)$; construct $\mathcal{V}_\mathcal{B}$ and symmetric $\mathcal{E}_{\mathrm{ord}}$
  Compute current $\widetilde{G}_i, \widetilde{C}_i$ and labeled states C_j^l; extra anchors do not enter G_l
  Compute ω_{ij}, ρ_i using the prediction cache, then overwrite the cache with q_i
  Initialize $C_i^{(0)} = \widetilde{C}_i$; clamp $C_j^{(n)} = C_j^l$ for $j \in \mathcal{L}_\mathcal{B}$
  Choose $\Delta\tau$ by Eq. equation 104; record $d_{\max}, \Delta\tau, \chi$
  for $n = 0, \ldots, K-1$ **do**
    Take one explicit Euler step of Eq. equation 8 for all unlabeled nodes
  end for
  Set $C_i^\star = C_i^{(K)}$; compute R_i, η_i and G_{rob} by Eqs. equation 10–equation 13
  Apply the projected truncated-SVD step in Eq. equation 16
  Update the head with AdamW on the SSL objective; optionally apply the separately specified rank-change rule
  Update teacher effective weights and head by Eq. equation 103
end for
Merge $W = W_0 + \gamma USV^\top$

AI USE STATEMENT

We used OpenAI's ChatGPT as a research-support tool for feedback on high-level research framing and methodological presentation, literature discovery and bibliographic checking, and drafting, revising, and polishing limited portions of the manuscript. It was also used for final consistency and minor-error checks. Generative AI was not used to generate source code, run experiments, or produce experimental data or numerical results. All scientific decisions, experimental protocols, analyses, and conclusions were made by the authors. All cited sources were selected by the authors and independently verified against primary publisher or project records, and all AI-assisted text and suggestions were reviewed, verified, and revised by the authors.The authors take full responsibility for the technical content, claims, citations, and final presentation of this work.

REPRODUCIBILITY STATEMENT

Section 3 describes FluxLoRA's core-gradient diffusion, residual gating, and low-rank update rule. Appendix A provides the theoretical assumptions and derivations. Appendix D describes the datasets, label budgets, training settings, and evaluation metrics, while Appendix D.3 specifies the graph construction, diffusion parameters, teacher updates, and prediction cache. Algorithm 1 summarizes

the training procedures. Appendix D.5 details the matched-checkpoint, norm-matched intervention protocol. These descriptions provide the methodological and experimental information needed to understand and reproduce the reported comparisons. Upon acceptance, we will immediately release the complete code and trained model checkpoints.