SVD-IMoE: Pretrained Spectral Directions as Implicit Routers for Low-Rank Adaptation

Anonymous authors
Paper under double-blind review

Abstract

Standard LoRA activates every rank component for every token, whereas frozen random projections used in implicit rank-wise routing do not exploit pretrained weight geometry. We propose SVD-IMoE, a spectral routing framework with two complementary variants. SVD-IMoE-W constructs a calibration-free routing basis through stratified selection of pretrained right-singular directions and applies undamped inverse-square-root spectral scaling. SVD-IMoE-TA uses calibration activations to select task-aligned weight directions within the same spectral bins, together with a heuristic quality gate and damped adaptive normalization. Both variants freeze the routing basis, and only the LoRA up-projection receives gradients. At matched $r = 32$ and $k = 8$, SVD-IMoE-W improves GSM8K over FlyLoRA by 4.39 points on Llama 3.1-8B and 2.27 points on Qwen 2.5-7B. SVD-IMoE-TA improves Qwen HumanEval Pass@1, Pass@5, and Pass@10 over SVD-IMoE-W by 3.60, 3.27, and 2.21 points. For SVD-IMoE-W adapters, shared routing states enable exact $\boldsymbol{B}$-only fusion. Under four-adapter parameter averaging, SVD-IMoE-TA outperforms merged FlyLoRA on all six metrics for both backbones, with gains of 0.56 to 4.39 points. These results suggest that pretrained spectral geometry provides a useful routing prior, while activation geometry supports task-dependent specialization.

1 Introduction

Fine-tuning large language models (LLMs) for downstream tasks requires updating billions of parameters, which is computationally expensive and storage-intensive. Parameter-Efficient Fine-Tuning (PEFT) methods address this by updating only a small subset of the model while keeping the majority of the model frozen. Low-Rank Adaptation (LoRA) has become a dominant PEFT approach, decomposing weight updates as $\Delta \boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$, where $\boldsymbol{A}$ is randomly initialized and $\boldsymbol{B}$ is zero-initialized (Hu et al., 2022). Quantized training, adaptive rank allocation, and weight decomposition have further improved this basic recipe (Dettmers et al., 2023; Zhang et al., 2023; Liu et al., 2024). Nevertheless, standard LoRA remains dense in rank space: every token uses all r components, even when only a subset is relevant to its local computation.

Mixture-of-experts (MoE) methods provide a natural remedy through conditional computation, activating different parameter subsets for different tokens or inputs (Shazeer et al., 2017; Fedus et al., 2022). Recent LoRA-MoE hybrids explore explicit expert architectures (Cao et al., 2026; Zhuang et al., 2026), while spectral priors can further improve low-rank experts (Fan et al., 2025). FlyLoRA offers a complementary lightweight direction by interpreting individual LoRA rank components as implicit experts and routing tokens through a frozen random projection, with no learned router parameters (Zou et al., 2025). Yet spectral methods exploit the anisotropic structure of pretrained weights, whereas FlyLoRA's random projection is agnostic to that structure. This raises our central question: can an implicit rank-wise MoE retain FlyLoRA's zero learned-router-parameter overhead while grounding its router in the spectral geometry of pretrained weights? Random projections are inexpensive but agnostic to both the pretrained projection and the downstream task, whereas pretrained weights exhibit strong anisotropy. Figure 1 provides a conceptual comparison of three representative LoRA adaptation architectures. Dense LoRA applies the complete rank-r update to every token; explicit MoE-LoRA introduces a separate expert bank and an additional router; implicit rank-wise MoE-LoRA treats the rank components of a shared down-projection as implicit experts and activates only a subset for each token.

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.

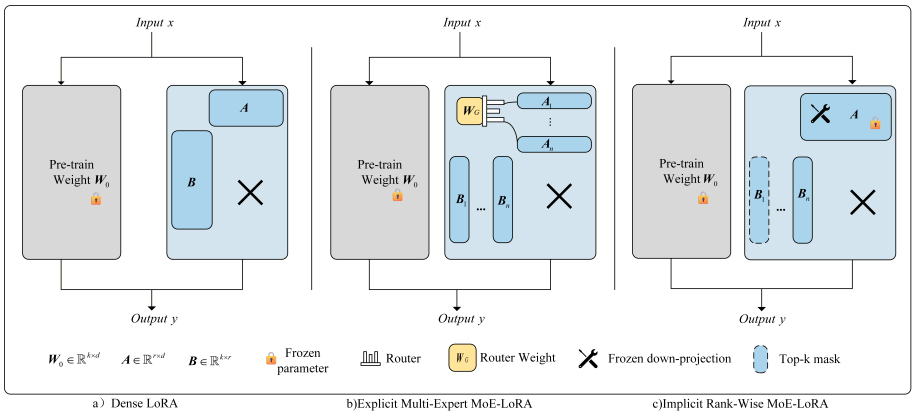

Figure 1: Conceptual comparison of dense LoRA, explicit MoE-LoRA, and implicit rank-wise MoE-LoRA. SVD-IMoE instantiates the implicit design with a frozen weight-derived spectral basis and token-wise top-k routing.

Figure 1(c) leaves a key design question open: how should the frozen rank-wise coordinate system be constructed? Although random directions provide a lightweight routing basis, they do not exploit the geometry already encoded in pretrained weights. We address this limitation by grounding implicit rank-wise routing in pretrained spectral structure.

We propose SVD-IMoE, a spectral implicit routing framework with two complementary algorithmic variants. SVD-IMoE-W constructs a calibration-free router from stratified right-singular directions of each pretrained projection and uses undamped inverse-square-root spectral scaling. The routing matrix $\boldsymbol{A}$ remains frozen, and only the LoRA up-projection $\boldsymbol{B}$ receives gradients; a small routing bias may be updated by an explicit non-gradient load-balancing rule. SVD-IMoE-TA uses a calibration eigenspace to select the most aligned pretrained direction within each spectral bin. A heuristic quality gate enables task-guided direction selection only when its mean alignment exceeds a random-subspace-derived threshold, and the task-adaptive implementation uses damped CV-adaptive normalization and task-informed bias initialization. If the gate fails, it reuses the stratified weight-only direction indices but retains the task-adaptive implementation's damped scale. Thus, SVD-IMoE-W and SVD-IMoE-TA are distinct complementary variants rather than a single-variable data-selection ablation. Our contributions are:

- We replace the frozen random projection of implicit rank-wise routing with a task-independent router derived from pretrained right-singular directions, without a learned gating network or explicit expert bank.
- We introduce stratified spectral selection, which divides the ordered singular spectrum into rank-dependent bins and selects one pretrained direction from each bin. We further develop variant-specific spectral scaling: SVD-IMoE-W uses undamped inverse-square-root normalization, whereas SVD-IMoE-TA employs a damped CV-adaptive rule.
- We develop a task-adaptive variant that combines activation-guided in-bin selection, a heuristic quality gate, task-informed routing-bias initialization, and damped adaptive normalization while keeping the routing directions anchored to pretrained weights.
- Across Llama 3.1-8B and Qwen 2.5-7B, we empirically characterize the complementary roles of the two variants: SVD-IMoE-W provides a strong calibration-free default, while SVD-IMoE-TA yields selective gains on code generation. Layer-wise diagnostics and four-adapter parameter-averaging experiments further examine task specialization and empirical cross-task composition.

2 Related Work

Parameter-efficient fine-tuning. LoRA (Hu et al., 2022) freezes the backbone and learns low-rank residual matrices. Its variants optimize quantization (Dettmers et al., 2023), rank allocation

(Zhang et al., 2023), and magnitude–direction decoupling (Liu et al., 2024). Recent work focuses on initialization and subspace design: PiSSA initializes updates via principal singular components (Meng et al., 2024), CorDA leverages contextual decomposition (Yang et al., 2024), LoRA-GA aligns initialization with full-tuning gradients (Wang et al., 2024), and MoRA expands effective update ranks with square parameterization (Jiang et al., 2024). Recent literature further exploits spectral properties of pretrained weights for principled adaptation. HiP-LoRA splits LoRA updates into dominant singular subspaces with stability regularization (Chen & Tan, 2026), while GOAT integrates singular-value priors into MoE architectures (Fan et al., 2025). Unlike these methods, SVD-IMoE builds a frozen token router from spectral decomposition, retaining standard rank-r LoRA as the trainable up-projection.

Mixture-of-experts adapters. Sparse MoE routes tokens to partial feed-forward experts (Shazeer et al., 2017; Fedus et al., 2022). Adapter-based MoELoRA equips multiple independent LoRA experts with learned routing, at the cost of extra parameters (Luo et al., 2024). MoRE assigns distinct LoRA ranks as multi-task experts (Zhang et al., 2025). Closest to our work, FlyLoRA (Zou et al., 2025) treats rank dimensions as implicit experts via a frozen random projection and top-k selection. Instead of random projection, we construct the routing basis deterministically from pretrained weight spectra, with optional task-guided component selection.

Spectral structure in adaptation. Pretrained weight spectra are highly imbalanced: pure top-r selection preserves energy but sacrifices diversity, while random selection maintains isotropy but ignores learned geometry (Golub & Van Loan, 1996). Concurrent spectral LoRA work explores routed experts (Nejatimanzari et al., 2026) and low-frequency dominance (Singh, 2026). SVD-IMoE-W addresses this tension through calibration-free stratification of pretrained right-singular directions; SVD-IMoE-TA preserves the same weight-derived candidate space but uses activation geometry to choose candidates within each spectral bin.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

Consider a frozen linear projection from input dimension d_{in} to output dimension d_{out}. LoRA adds a rank-r residual through a down-projection $\boldsymbol{A} \in \mathbb{R}^{r \times d_{\text{in}}}$ and an up-projection $\boldsymbol{B} \in \mathbb{R}^{d_{\text{out}} \times r}$, scaled by $\gamma = \alpha/r$:

$$\boldsymbol{y} = \boldsymbol{W}_0 \boldsymbol{x} + \gamma \boldsymbol{B}\boldsymbol{A}\boldsymbol{x}. \tag{1}$$

The implementation uses the standard LoRA scaling α/r; in the reported experiments this scale equals 2 for both $r = 8, \alpha = 16$ and $r = 32, \alpha = 64$.

In an implicit rank-wise MoE, applying $\boldsymbol{A}$ to the dropped input produces an r-dimensional vector of rank-wise expert activations $\boldsymbol{z} = \boldsymbol{A}\mathcal{D}(\boldsymbol{x})$. A binary token-dependent mask $\boldsymbol{m}(\boldsymbol{x}) \in \{0, 1\}^r$ selects k components. With $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{z} \odot \boldsymbol{m}(\boldsymbol{x})$, the routed update is

$$\boldsymbol{y} = \boldsymbol{W}_0 \boldsymbol{x} + \gamma \boldsymbol{B}\boldsymbol{u}(\boldsymbol{x}), \qquad \|\boldsymbol{m}(\boldsymbol{x})\|_0 = k. \tag{2}$$

Here $\mathcal{D}(\boldsymbol{x})$ denotes LoRA dropout. All reported experiments set its probability to zero, so the dropped input equals the original input. In every reported experiment the routing basis $\boldsymbol{A}$ is frozen (freeze_A=True), each row of $\boldsymbol{A}$ is a unit-norm right-singular direction, and only the up-projection $\boldsymbol{B}$ receives gradients from selected components. The routing bias is updated by the non-gradient rule in Equation 13 and does not contribute to the count of gradient-optimized parameters. The dense PyTorch implementation computes the full projected vector before masking; k describes conditional rank activity and gradient support, not a guaranteed wall-clock speedup. The variant-specific spectral scales and the common mask construction are specified in Equations 5 and 10–13.

3.2 SVD-IMOE-W: CALIBRATION-FREE SPECTRAL ROUTER

We compute a truncated singular value decomposition of each pretrained projection:

$$\boldsymbol{W}_0 \approx \boldsymbol{U}_q \boldsymbol{\Sigma}_q \boldsymbol{V}_q^{\mathsf{T}}, \qquad \boldsymbol{\Sigma}_q = \operatorname{diag}(\sigma_1, \ldots, \sigma_q), \tag{3}$$

The right-singular vectors serve as candidate routing directions, ordered by decreasing singular value $\sigma_1 \geq \cdots \geq \sigma_q$, and $\boldsymbol{v}_j$ denotes column j of $\boldsymbol{V}_q$. Directly selecting the first r vectors would overrepresent the head of the spectrum. We therefore partition the ordered candidate indices into r contiguous spectral bins $\mathcal{J}_1, \ldots, \mathcal{J}_r$, matching the split used in our implementation: writing $q = ar + m$ with $a = \lfloor q/r \rfloor$ and $m = q \bmod r$, bin b holds $a + 1$ candidates when $b \leq m$ and a candidates otherwise, so every bin is nonempty whenever $q \geq r$, as in all reported settings. We select the strongest direction within each bin:

$$j_b^{(\mathrm{W})} = \arg\max_{j \in \mathcal{J}_b} \sigma_j, \qquad \boldsymbol{A}_{b,:} = \boldsymbol{v}_{j_b^{(\mathrm{W})}}^{\mathsf{T}}. \tag{4}$$

The evaluated SVD-IMoE-W implementation associates each selected direction with an undamped inverse-square-root route scale:

$$g_b^{(\mathrm{W})} = \left[\max(\sigma_{j_b^{(\mathrm{W})}}, \varepsilon)\right]^{-1/2}. \tag{5}$$

3.3 SVD-IMoE-TA: Task-Adaptive Spectral Selection

SVD-IMoE-TA streams layer inputs from a small calibration set without storing raw activations. Let $\boldsymbol{X}^{(h)} \in \mathbb{R}^{n_h \times d_{\mathrm{in}}}$ contain the n_h valid token activations in calibration batch h, and let $n = \sum_h n_h$ denote the total number of valid calibration tokens. We measure the norm of each batch-mean activation and compute its coefficient of variation. In automatic mode, we use a centered covariance when the coefficient of variation is at least 0.3, and an uncentered second moment otherwise:

$$\begin{aligned} n &= \sum_h n_h, & \boldsymbol{\mu} &= \frac{1}{n} \sum_h (\boldsymbol{X}^{(h)})^{\mathsf{T}} \mathbf{1}_{n_h}, \\ \boldsymbol{M} &= \frac{1}{n} \sum_h (\boldsymbol{X}^{(h)})^{\mathsf{T}} \boldsymbol{X}^{(h)}, & \bar{\boldsymbol{x}}_h &= \frac{1}{n_h} (\boldsymbol{X}^{(h)})^{\mathsf{T}} \mathbf{1}_{n_h}, \\ \rho_h &= \|\bar{\boldsymbol{x}}_h\|_2, & \mathrm{CV} &= \frac{\mathrm{Std}_h(\rho_h)}{\max(\mathrm{Mean}_h(\rho_h), \varepsilon_{\mathrm{CV}})}. \end{aligned} \tag{6}$$

The covariance used by the implementation is selected from batch-mean norm variation:

$$\boldsymbol{C} = \begin{cases} \boldsymbol{M} - \boldsymbol{\mu}\boldsymbol{\mu}^{\mathsf{T}}, & \mathrm{CV} \geq 0.3, \\ \boldsymbol{M}, & \mathrm{CV} < 0.3. \end{cases} \tag{7}$$

Let $\boldsymbol{V}_t \in \mathbb{R}^{d_{\mathrm{in}} \times q_t}$ contain the q_t leading task eigendirections of $\boldsymbol{C}$, satisfying $\boldsymbol{C}\boldsymbol{V}_t = \boldsymbol{V}_t \boldsymbol{\Lambda}_t$, where $q_t = \min(4r, d_{\mathrm{in}})$. Within each weight-derived bin, TA selects:

$$j_b^{(\mathrm{TA})} = \arg\max_{j \in \mathcal{J}_b} \|\boldsymbol{V}_t^{\mathsf{T}} \boldsymbol{v}_j\|_2, \qquad \boldsymbol{A}_{b,:} = \boldsymbol{v}_{j_b^{(\mathrm{TA})}}^{\mathsf{T}}. \tag{8}$$

The columns of $\boldsymbol{V}_t$ are the q_t leading eigenvectors of $\boldsymbol{C}$. Candidate relevance is computed from this eigenspace. SVD-IMoE-TA replaces the SVD-IMoE-W choice inside each bin by the maximizer in Equation 8. Crucially, the task eigenspace is used only to select candidates: every row of $\boldsymbol{A}$ remains a right-singular direction of the pretrained weight. Task adaptation therefore redirects selected spectral coordinates toward task-aligned pretrained directions without placing activation eigenvectors directly in $\boldsymbol{A}$. Calibration can be scoped to the full sequence, the prompt, the question content, or the response. In the reported runs, GSM8K uses full-sequence activations because reasoning traces carry task signal; MMLU uses prompt-only calibration and text-only ScienceQA uses input-only calibration, so neither sees the supervised response; and CodeAlpaca uses output-only calibration. Task moments are computed once and cached.

Figure 2 summarizes the variant-specific router construction and the shared conditional-routing path of SVD-IMoE-W and SVD-IMoE-TA. SVD-IMoE-W constructs a calibration-free basis through stratified Weight-SVD selection, whereas SVD-IMoE-TA uses calibration activations and a heuristic gate for task-guided in-bin selection. The corresponding route scales are $\boldsymbol{g}^{(\mathrm{W})}$ and $\boldsymbol{g}^{(\mathrm{TA})}$. Both variants freeze their routing bases, apply token-wise top-k routing with a non-gradient bias, and train only the up-projection.

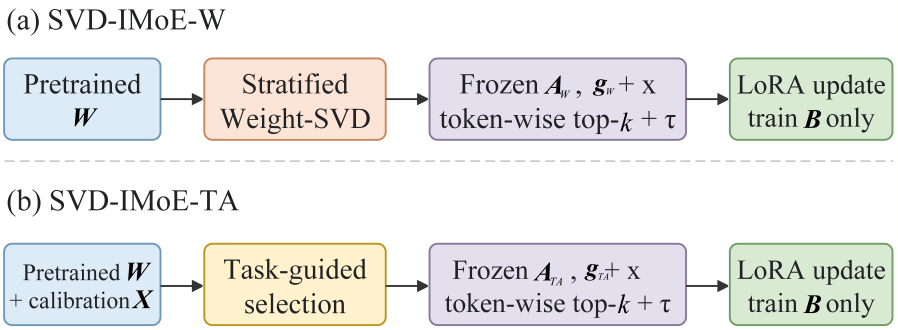

Figure 2: Overview of the SVD-IMoE-W and SVD-IMoE-TA variants. SVD-IMoE-W constructs a calibration-free basis through stratified Weight-SVD selection, whereas SVD-IMoE-TA uses calibration activations and a heuristic gate for task-guided in-bin selection. Both variants freeze their routing bases, apply token-wise top-k routing with a non-gradient bias, and train only $\boldsymbol{B}$.

3.4 Alignment Quality Gate

Task estimates can be noisy in layers where the calibration subspace is weakly aligned with the weight spectrum. We therefore compare the mean alignment across bins with an isotropic-random baseline. With $a_j = \|\boldsymbol{V}_t^\top \boldsymbol{v}_j\|_2$ for candidate j and $\bar{a} = r^{-1}\sum_{b=1}^{r} a_{j_b}$ denoting the mean alignment of the selected bins, the automatic threshold is $\theta_{\text{gate}} = 1.5\sqrt{q_t/d_{\text{in}}}$, where d_{in} is the input dimension and q_t the retained task-subspace dimension. Because the statistic uses a maximum within each bin, this threshold is a heuristic baseline and does not correct the corresponding order-statistic effect. Task-guided selection is accepted only when the mean alignment exceeds this threshold:

$$\theta_{\text{gate}} = 1.5\sqrt{\frac{q_t}{d_{\text{in}}}}, \qquad \text{accept task guidance if } \bar{a} > \theta_{\text{gate}}. \tag{9}$$

A positive configuration value overrides the automatic threshold, whereas a negative value disables the gate. The gate should be interpreted as a heuristic filter rather than a calibrated statistical significance test; sensitivity to its fixed multiplier is left for future ablation. If the test fails, TA restores the stratified weight indices and disables task-derived routing-bias initialization. It nevertheless retains the damped normalization rule defined for the task-adaptive variant. Consequently, this fallback is not equivalent to SVD-IMoE-W, which uses undamped inverse-square-root spectral scaling.

3.5 Variant-Specific Spectral Normalization

SVD-IMoE-TA uses the coefficient of variation of batch-mean activation norms to control a damped spectral correction. Let $\beta = \text{clip}(\text{CV}/0.3, 0, 1)$ when task guidance is accepted and $\beta = 1$ otherwise. Thus σ_{j_b} denotes the singular value of the final direction selected for component b. Let $\bar{\sigma} = r^{-1}\sum_{\ell=1}^{r}\sigma_{j_\ell}$. The adaptive exponent is determined by calibration statistics for accepted layers and is fixed to 1 for fallback layers. This differs from the SVD-IMoE-W scale in Equation 5, which uses an undamped inverse square root without CV modulation or output clamping. The TA route scale is:

$$g_b^{(\text{TA})} = \text{clip}\left([\max(\sigma_{j_b} + 0.1\bar{\sigma}, \varepsilon_\sigma)]^{-\beta/2}, 0.1, 10\right). \tag{10}$$

Here $\text{clip}(x, l, u) = \min(\max(x, l), u)$, and the clipping bounds avoid extreme route amplitudes. Equation 5 is the undamped scale used by SVD-IMoE-W, whereas Equation 10 defines the damped adaptive scale used by SVD-IMoE-TA. Neither rule is exact activation whitening. For either variant, routing scores and masks are:

$$s_b = |g_b z_b| + \tau_b, \qquad m_b = \mathbb{I}[b \in \text{TopK}(\boldsymbol{s}, k)]. \tag{11}$$

Accepted SVD-IMoE-TA layers additionally initialize the routing bias from normalized task alignment, where a_{j_b} is the alignment of the selected direction defined in Equation 8:

$$\tau_b^{(0)} = \begin{cases} 0.1\,\dfrac{a_{j_b}}{\max_j a_j + \varepsilon}, & \text{if task guidance is accepted,} \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

with $\varepsilon = 10^{-8}$ guarding against a vanishing denominator. The bias is then updated outside backpropagation:

$$\tau_b \leftarrow \tau_b + \eta\,\mathrm{sign}(\bar{c} - c_b), \qquad \bar{c} = \frac{1}{r}\sum_{b=1}^{r} c_b, \tag{13}$$

where c_b is the number of tokens assigned to component b and $\eta = 0.001$. In the implementation, c_b is accumulated for the current training forward (one micro-batch), the update is applied immediately after routing, and the usage counter is reset before the next forward; this update is performed only while training and is not part of backpropagation. Top-k index selection is detached from autograd. Only $\boldsymbol{B}$ receives gradients from selected numeric activations; $\boldsymbol{A}$ receives no gradient, and the routing bias changes only through the explicit non-gradient update in Equation 13.

3.6 Shared Spectral Coordinates for Fusion

SVD-IMoE-W adapters constructed from the same pretrained projection and SVD cache share the routing basis $\boldsymbol{A}$ and route scale $\boldsymbol{g}$. If their routing bias $\boldsymbol{\tau}$, top-k configuration, LoRA scaling, and dropout behavior are also shared, they produce the same token-wise mask. Define the shared routed activation and task-specific contribution as

$$\boldsymbol{u}(\boldsymbol{x}) = (\boldsymbol{A}\boldsymbol{x}) \odot \boldsymbol{m}(\boldsymbol{x}), \qquad \boldsymbol{f}_t(\boldsymbol{x}) = \gamma \boldsymbol{B}_t \boldsymbol{u}(\boldsymbol{x}). \tag{14}$$

Given fusion coefficients λ_t satisfying $\sum_{t=1}^{T} \lambda_t = 1$, the fused up-projection is

$$\boldsymbol{B}_{\mathrm{fuse}} = \sum_{t=1}^{T} \lambda_t \boldsymbol{B}_t. \tag{15}$$

Because the routed activation is shared, the fused adapter satisfies the exact token-wise identity

$$\gamma \boldsymbol{B}_{\mathrm{fuse}} \boldsymbol{u}(\boldsymbol{x}) = \sum_{t=1}^{T} \lambda_t \boldsymbol{f}_t(\boldsymbol{x}). \tag{16}$$

Equation 16 is exact only under shared routing states. For SVD-IMoE-W, this condition can be enforced by fixing $\boldsymbol{\tau} = \boldsymbol{0}$ and disabling the bias update. The reported checkpoints instead use $\eta = 0.001$, so Table 2 evaluates empirical parameter averaging rather than the strict identity; SVD-IMoE-TA may additionally differ in its selected directions and route scales.

3.7 Complexity Analysis

Both routed variants train the same number of parameters as FlyLoRA: rd_{out} elements in the up-projection, since $\boldsymbol{A}$ is frozen. The routing bias adds r scalars per adapter but is updated by the non-gradient rule in Equation 13 and therefore does not enter optimizer state. Concretely, all routed methods train 0.13% of the backbone on both Llama 3.1-8B and Qwen 2.5-7B, matching FlyLoRA and staying between LoRA-$r = 8$ and LoRA-$r = 32$. With the current dense kernels, forward computation is $O(d_{\mathrm{in}} r + d_{\mathrm{out}} r)$ per token: $\boldsymbol{A}$ is applied densely to obtain the r-dimensional routing vector, and the mask is applied before the dense $\boldsymbol{B}$ multiplication. Thus k restricts numerical and gradient activity but does not imply wall-clock sparse acceleration. Preprocessing is one-shot. Weight-SVD uses a randomized routine with rank 256 and four power iterations per pretrained projection, and is cached on disk together with the seed. For SVD-IMoE-TA, per-layer calibration accumulates streaming activations to form a second-moment matrix and a single deterministic eigendecomposition. Both are computed once per backbone or per task and reused across all training and evaluation runs.

Table 1: Adapter structure for a single linear projection. r is the LoRA rank and k the per-token top-k. The per-adapter routing bias $\boldsymbol{\tau}$ (r scalars) is updated by the non-gradient rule in Equation 13 and is not included in the trainable-parameter count. Because SVD-IMoE variants compute the complete down-projection before masking, activating $k < r$ components does not directly provide sparse-kernel acceleration.

Method	Frozen $\boldsymbol{A}$	Trainable parameters	Active ranks/token
LoRA	No	$r(d_{\text{in}} + d_{\text{out}})$	r
FlyLoRA	random	rd_{out}	k
SVD-IMoE-W	Stratified Weight SVD	rd_{out}	k
SVD-IMoE-TA	Task-selected Weight SVD	rd_{out}	k

4 Experiments

Our experiments address three questions. First, does replacing a frozen random router with pre-trained spectral directions improve rank-wise sparse adaptation? Second, when does activation-guided direction selection provide additional specialization beyond the calibration-free weight prior? Third, how compatible are independently trained adapters under parameter averaging?

4.1 Experimental Setup

Models and tasks. We evaluate Llama 3.1-8B (Grattafiori et al., 2024) and Qwen 2.5-7B (Team, 2024) on GSM8K, MMLU, the text-only subset of ScienceQA, and HumanEval (Cobbe et al., 2021; Hendrycks et al., 2020; Lu et al., 2022; Chen et al., 2021). ScienceQA removes image-dependent examples and contains 6,508 training and 2,224 test instances; GSM8K contains 7,473 training and 1,319 test instances; and MMLU evaluation covers 14,042 questions. Code-generation adapters are fine-tuned on CodeAlpaca-20K (Chaudhary, 2023), which contains 20,022 instruction-response pairs, and are evaluated on the 164 HumanEval problems using Pass@1, Pass@5, and Pass@10. HumanEval examples are used only for evaluation.

Baselines and training. We compare dense LoRA at $r = 8, \alpha = 16$ and $r = 32, \alpha = 64$ with FlyLoRA, SVD-IMoE-W, and SVD-IMoE-TA. All routed methods use $r = 32$, $k = 8$, $\alpha = 64$, zero LoRA dropout, and adapt q_proj, k_proj, v_proj, and o_proj. Dense LoRA trains both low-rank factors, whereas FlyLoRA and both SVD-IMoE variants freeze the routing projection $\boldsymbol{A}$ and optimize only $\boldsymbol{B}$. FlyLoRA initializes $\boldsymbol{A}$ randomly. SVD-IMoE-W constructs $\boldsymbol{A}$ from calibration-free stratified Weight-SVD selection, whereas SVD-IMoE-TA uses calibration activations to refine direction selection within the same spectral bins. Models are trained with 4-bit NF4 quantization. For routed methods, the routing bias is updated outside gradient descent with $\eta = 0.001$. Routed methods are compared at matched $r = 32$ and $k = 8$: LoRA-$r = 8$ trains 0.08% (Llama) and 0.07% (Qwen), LoRA-$r = 32$ trains 0.34% and 0.26%, while FlyLoRA, SVD-IMoE-W, and SVD-IMoE-TA each train 0.13% of backbone parameters.

4.2 Single-Task Results

Table 2 first reports task-specific adaptation results. SVD-IMoE-W provides the clearest evidence for the proposed spectral-routing principle. It achieves the highest GSM8K mean on both backbones, reaching 55.57% on Llama and 79.68% on Qwen. These results exceed FlyLoRA by 4.39 and 2.27 percentage points, respectively, under matched $r = 32$ and $k = 8$. On Llama, SVD-IMoE-W also approaches the best ScienceQA result while training substantially fewer parameters than LoRA-$r = 32$. Dense LoRA remains stronger on several knowledge and code-generation metrics, demonstrating that sparse rank-wise routing and dense adaptation have complementary strengths rather than one method uniformly dominating the other.

SVD-IMoE-TA is an optional specialization mechanism rather than a uniformly stronger replacement for SVD-IMoE-W. On MMLU, ScienceQA, and GSM8K, the two variants differ by at most 1.36 points, and SVD-IMoE-W remains stronger on most reasoning comparisons. In contrast, TA improves all three HumanEval metrics over W on both backbones. The gains are 1.01, 2.73, and 3.06 points on Llama and 3.60, 3.27, and 2.21 points on Qwen for Pass@1, Pass@5, and Pass@10,

Table 2: Single-task and four-adapter parameter-averaging results (%). Dense LoRA uses the indicated rank; routed methods use $r = 32$ and $k = 8$. Single and Merged denote task-specific checkpoints and their uniform four-adapter parameter average, respectively. Code adapters are trained on CodeAlpaca-20K and evaluated on HumanEval. Bold marks the highest value within each backbone, metric, and status block (Single or Merged) separately; subscripts report the uncertainty of each estimate.

Model	Method	Status	MMLU	ScienceQA	GSM8K	Pass@1	Pass@5	Pass@10
Llama-3.1-8B	LoRA-$r = 8$	Single	$53.65_{\pm 0.89}$	$87.41_{\pm 0.53}$	$45.64_{\pm 0.20}$	$40.18_{\pm 0.20}$	$65.45_{\pm 1.71}$	$73.96_{\pm 0.73}$
	LoRA-$r = 8$	Merged	$51.13_{\pm 0.72}$	$35.16_{\pm 2.39}$	$43.90_{\pm 2.37}$	$37.93_{\pm 2.81}$	$64.03_{\pm 2.16}$	$72.96_{\pm 1.84}$
	LoRA-$r = 32$	Single	$53.73_{\pm 1.39}$	**91.68**$_{\pm 0.16}$	$53.82_{\pm 0.24}$	**41.52**$_{\pm 1.46}$	**66.35**$_{\pm 2.23}$	**74.96**$_{\pm 1.26}$
	LoRA-$r = 32$	Merged	$52.93_{\pm 0.59}$	$34.67_{\pm 1.52}$	**48.98**$_{\pm 0.92}$	**38.17**$_{\pm 1.74}$	$64.78_{\pm 1.43}$	$73.59_{\pm 1.40}$
	FlyLoRA	Single	$55.85_{\pm 1.15}$	$90.06_{\pm 0.34}$	$51.18_{\pm 0.61}$	$38.57_{\pm 1.43}$	$65.60_{\pm 2.14}$	$74.91_{\pm 0.95}$
	FlyLoRA	Merged	$54.61_{\pm 1.03}$	$81.74_{\pm 0.97}$	$39.88_{\pm 1.04}$	$33.69_{\pm 1.12}$	$64.02_{\pm 1.92}$	$74.34_{\pm 1.02}$
	SVD-IMoE-W	Single	$56.69_{\pm 1.34}$	$91.10_{\pm 0.25}$	**55.57**$_{\pm 0.43}$	$38.59_{\pm 1.31}$	$62.13_{\pm 2.07}$	$70.88_{\pm 0.58}$
	SVD-IMoE-W	Merged	$54.35_{\pm 1.34}$	**83.68**$_{\pm 1.13}$	$46.10_{\pm 1.22}$	$35.95_{\pm 1.43}$	$64.10_{\pm 2.01}$	$72.85_{\pm 0.71}$
	SVD-IMoE-TA	Single	**56.72**$_{\pm 0.87}$	$90.69_{\pm 0.17}$	$54.21_{\pm 0.37}$	$39.60_{\pm 1.53}$	$64.86_{\pm 2.24}$	$73.94_{\pm 1.24}$
	SVD-IMoE-TA	Merged	**55.52**$_{\pm 0.92}$	$82.87_{\pm 0.88}$	$43.75_{\pm 1.17}$	$35.67_{\pm 1.38}$	**65.09**$_{\pm 1.75}$	**74.90**$_{\pm 1.33}$
Qwen-2.5-7B	LoRA-$r = 8$	Single	$63.75_{\pm 0.72}$	$86.83_{\pm 0.12}$	$49.28_{\pm 0.18}$	$62.44_{\pm 1.54}$	$84.49_{\pm 0.38}$	$87.95_{\pm 0.65}$
	LoRA-$r = 8$	Merged	$60.24_{\pm 0.69}$	$35.30_{\pm 2.07}$	**80.06**$_{\pm 1.52}$	$65.85_{\pm 1.27}$	$86.09_{\pm 0.70}$	$89.28_{\pm 0.91}$
	LoRA-$r = 32$	Single	**66.93**$_{\pm 1.57}$	**91.64**$_{\pm 0.20}$	$54.35_{\pm 0.15}$	$63.35_{\pm 1.83}$	$85.05_{\pm 1.22}$	$89.41_{\pm 0.79}$
	LoRA-$r = 32$	Merged	**64.49**$_{\pm 1.26}$	$35.12_{\pm 1.63}$	$78.47_{\pm 1.36}$	$65.43_{\pm 1.66}$	**86.49**$_{\pm 1.05}$	**89.96**$_{\pm 0.76}$
	FlyLoRA	Single	$66.02_{\pm 1.13}$	$90.47_{\pm 0.17}$	$77.41_{\pm 0.45}$	$58.02_{\pm 1.91}$	$83.91_{\pm 1.84}$	$89.89_{\pm 0.42}$
	FlyLoRA	Merged	$49.00_{\pm 1.83}$	$81.07_{\pm 1.09}$	$75.51_{\pm 1.20}$	$61.86_{\pm 1.90}$	$84.62_{\pm 1.24}$	$88.23_{\pm 0.58}$
	SVD-IMoE-W	Single	$65.40_{\pm 1.17}$	$90.83_{\pm 0.21}$	**79.68**$_{\pm 0.43}$	$61.86_{\pm 2.53}$	$83.11_{\pm 2.60}$	$87.80_{\pm 2.44}$
	SVD-IMoE-W	Merged	$53.80_{\pm 1.12}$	**82.55**$_{\pm 0.94}$	$78.62_{\pm 1.07}$	$64.51_{\pm 1.51}$	$85.68_{\pm 0.98}$	$89.63_{\pm 0.82}$
	SVD-IMoE-TA	Single	$65.35_{\pm 1.14}$	$90.02_{\pm 0.12}$	$79.53_{\pm 0.48}$	**65.46**$_{\pm 1.44}$	**86.38**$_{\pm 1.56}$	**90.01**$_{\pm 0.63}$
	SVD-IMoE-TA	Merged	$53.39_{\pm 1.48}$	$82.06_{\pm 1.18}$	$78.24_{\pm 1.14}$	**66.25**$_{\pm 1.39}$	$86.15_{\pm 0.85}$	$89.43_{\pm 0.73}$

respectively. On Qwen, SVD-IMoE-TA also exceeds LoRA-$r = 32$ by 2.11, 1.33, and 0.60 points. These results identify activation geometry as a selective code-specialization signal, while calibration-free Weight-SVD routing remains the more robust default.

Ablation-style reading. At matched r and k, GSM8K single-task accuracy moves from FlyLoRA to SVD-IMoE-W to SVD-IMoE-TA as $51.18 \rightarrow 55.57 \rightarrow 54.21$ on Llama 3.1-8B and $77.41 \rightarrow 79.68 \rightarrow 79.53$ on Qwen 2.5-7B. Spectral replacement therefore provides the main GSM8K gains ($+4.39$/$+2.27$ points), whereas task guidance mainly benefits code generation. Figure 3 is consistent with the geometric effect of task-adaptive selection but does not establish a direct relationship with downstream performance.

4.3 Fusion Results

To examine cross-task compatibility, we uniformly average the corresponding parameters of four independently trained adapters for MMLU, ScienceQA, GSM8K, and CodeAlpaca, using coefficients $\lambda_t = 1/4$. The resulting adapter is evaluated on all four benchmarks without additional training. This experiment is an empirical parameter-averaging diagnostic, not an evaluation of the exact $\boldsymbol{B}$-only fusion identity: independently trained checkpoints do not necessarily share identical routing biases, route scales, or token-wise masks.

The merged results in Table 2 reveal a different trade-off. Dense LoRA loses about 57 ScienceQA points on both backbones under uniform four-adapter averaging (91.68 to 34.67 on Llama; 91.64 to 35.12 on Qwen), whereas merged SVD-IMoE-W and SVD-IMoE-TA retain 82–84 points and merged FlyLoRA retains 81 points, indicating that frozen rank-wise routing is far less sensitive to destructive factor averaging on this task. Among the routed methods, SVD-IMoE-W outperforms merged FlyLoRA on ten of twelve backbone–metric comparisons, and SVD-IMoE-TA outperforms it on all twelve with gains up to 4.39 points, which is consistent with Weight-SVD providing a useful basis for parameter averaging. W and TA remain complementary: W is stronger on several reasoning metrics while TA retains code-oriented specialization; these findings support improved cross-task compatibility but do not establish uniform preservation of every capability under parameter averaging.

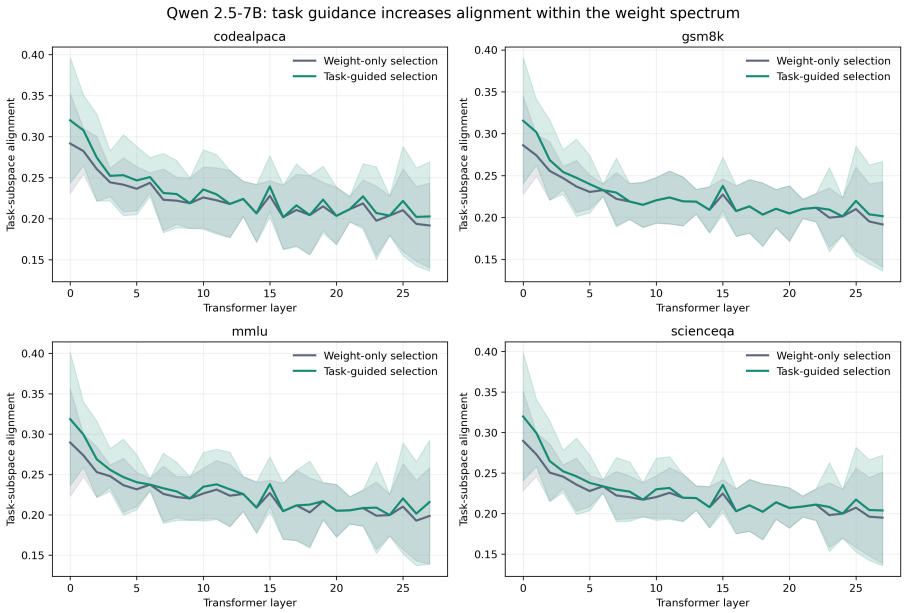

Figure 3: Layer-wise task-alignment diagnostics on Qwen 2.5-7B. Curves report the alignment between the selected weight directions and the calibration subspace; quality-gate acceptance rates are 15–21% across tasks. SVD-IMoE-TA increases this alignment. The diagnostics are consistent with TA's geometric effect but do not directly predict downstream accuracy.

5 Conclusion

We presented SVD-IMoE, a spectral implicit-MoE framework that grounds rank-wise routing in the right-singular directions of pretrained weights, with two complementary variants: SVD-IMoE-W provides a calibration-free stratified router with undamped inverse-square-root scaling, while SVD-IMoE-TA refines in-bin selection with activation geometry under a heuristic quality gate and damped adaptive normalization. Under matched rank and top-k, SVD-IMoE-W improves GSM8K over FlyLoRA by 4.39 points on Llama 3.1-8B and 2.27 points on Qwen 2.5-7B, while SVD-IMoE-TA delivers selective code-generation gains of up to 3.60 HumanEval Pass@1 points on Qwen 2.5-7B relative to SVD-IMoE-W. In our four-adapter averaging experiment, SVD-IMoE-TA outperforms merged FlyLoRA on all twelve evaluated backbone-metric combinations, suggesting that pretrained spectral geometry can provide a useful routing prior while activation geometry supports task-dependent specialization. The current implementation uses dense PyTorch kernels, so these results do not claim wall-clock or memory savings from top-k routing; specialized sparse kernels and hardware profiling are needed to establish such benefits. Several directions remain for future work: the sensitivity of the quality-gate multiplier and the design of non-uniform fusion weights informed by adapter- or task-level statistics.

AI Use Statement

In this work, generative-AI tools were **not used** for any tasks requiring mandatory disclosure, including synthetic-data generation, theoretical-model development, hypothesis formulation, experiment design, method implementation, dataset processing, result interpretation, mathematical proofwriting, and translation.

Additionally, we used generative-AI tools for recommended-disclosure tasks: document-format conversion, polishing manuscript wording, improving readability, and grammatical revision. We used ChatGPT for these purposes.

All AI-assisted text was fully reviewed, revised, and verified by the authors. All experiment designs, raw experimental data, analysis conclusions, and scientific claims are original and manually

produced by the authors. We take full responsibility for the final content of this work, including all text, claims, and experimental results.

Reproducibility Statement

The paper specifies the algorithm, routing rules, data splits, preprocessing, calibration scopes, training configuration, and evaluation protocol. Anonymous source code and configuration files are submitted as supplementary material.

References

Jie Cao, Tianwei Lin, Bo Yuan, et al. Moa: Heterogeneous mixture of adapters for parameter-efficient fine-tuning of large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 21056–21073, 2026.

Sahil Chaudhary. Code alpaca: An instruction-following llama model for code generation. *GitHub Repository*, 2023.

Lixian Chen and Jianhong Tan. Hip-lora: Budgeted spectral plasticity for robust low-rank adaptation. *arXiv preprint arXiv:2604.17751*, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115, 2023.

Chenghao Fan, Zhenyi Lu, Sichen Liu, et al. Make lora great again: Boosting lora with adaptive singular values and mixture-of-experts optimization alignment. In *International Conference on Machine Learning (ICML)*, 2025.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23:1–39, 2022.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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. In *International Conference on Learning Representations (ICLR)*, 2022.

Ting Jiang, Shaohan Huang, and Shengyue Luo. Mora: High-rank updating for parameter-efficient fine-tuning. *arXiv preprint arXiv:2405.12130*, 2024.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Tongxu Luo, Jiahe Lei, Fangyu Lei, et al. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models. *arXiv preprint arXiv:2402.12851*, 2024.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *arXiv preprint arXiv:2404.02948*, 2024.

Omid Nejatimanzari, Hojat Asgariandehkordi, Taha Koleilat, Yiming Xiao, and Hassan Rivaz. Sparse spectral lora: Routed experts for medical vlms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Rajveer Singh. Spectrallora: Is low-frequency structure sufficient for lora adaptation? a spectral analysis of weight updates. *arXiv preprint arXiv:2604.10649*, 2026.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. *arXiv preprint arXiv:2407.05000*, 2024.

Yibo Yang, Xiaojie Li, and Zhongzhu Zhou. Corda: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. *arXiv preprint arXiv:2406.05223*, 2024.

Dacao Zhang, Kun Zhang, Shimao Chu, Le Wu, Xin Li, and Si Wei. More: A mixture of low-rank experts for adaptive multi-task learning. *arXiv preprint arXiv:2505.22694*, 2025.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, et al. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2023.

Yuan Zhuang, Yi Shen, and Yuexin Bian. Ld-mole: Learnable dynamic routing for mixture of lora experts. In *International Conference on Learning Representations (ICLR)*, 2026.

Heming Zou, Yunliang Zang, Wutong Xu, Yao Zhu, and Xiangyang Ji. Flylora: Boosting task decoupling and parameter efficiency via implicit rank-wise mixture-of-experts. In *Advances in Neural Information Processing Systems*, 2025.

A Data Details

All supervised datasets are converted into an instruction–input–response format. GSM8K contains 7,473 training examples and 1,319 test examples. Each question is used as the instruction, and the original chain-of-thought solution, including the final "####" answer marker, is used as the response. Evaluation applies greedy decoding and exact numerical matching after extracting the final marked answer.

MMLU adapters are trained on the auxiliary training split and evaluated on 14,042 questions across its subjects. Each example contains a question, multiple-choice candidates, and the correct option index. ScienceQA is restricted to text-only examples by removing questions associated with images, resulting in 6,508 training and 2,224 test instances. The question is used as the instruction; the optional hint and answer choices form the input context. Lecture and solution fields are excluded to avoid answer leakage, and the target is the zero-based answer index.

Code-generation adapters are trained on all 20,022 CodeAlpaca instruction–response pairs. CodeAlpaca examples are shuffled with seed 23 during preprocessing, but no held-out CodeAlpaca subset

is used for reporting. Generalization is evaluated exclusively on the 164 HumanEval problems. For each problem, we generate 20 completions with temperature 0.8 and top-$p = 0.95$, and report the official execution-based Pass@1, Pass@5, and Pass@10 estimates.

SVD-IMoE-TA constructs its calibration statistics using only training data. Four scopes are available: full-sequence, prompt-only, content-only (question and choices, without the prompt template), and output-only. GSM8K uses full-sequence calibration; MMLU uses prompt-only calibration; text-only ScienceQA uses input-only calibration, implemented as content-only for Llama 3.1-8B and prompt-only for Qwen 2.5-7B; and CodeAlpaca uses output-only calibration. MMLU and ScienceQA therefore calibrate on inputs only, so no response is seen for those two tasks, whereas GSM8K and CodeAlpaca deliberately calibrate on response-bearing text because the reasoning trace and the code completion carry the task signal. Calibration examples are shuffled with seed 23 and processed in micro-batches of eight, with at most 500 calibration batches. No validation or test examples are used to construct the task eigenspaces.

B Implementation Details

We implement SVD-IMoE by extending the PEFT LoRA layers in PyTorch. The pretrained right-singular directions are computed once for the query, key, value, and output projections of every attention layer and cached on disk. Randomized SVD is configured with a requested rank of 256, uses four power iterations, and is initialized with seed 23. The SVD preprocessing stage loads the pretrained weights in float16 without quantization and performs the decomposition in float32.

SVD-IMoE-W divides the ordered singular spectrum into r contiguous bins and selects the strongest singular direction from each bin. Its route scales use the undamped inverse square root of the selected singular values. SVD-IMoE-TA retains the same weight-derived candidate space but uses a calibration eigenspace to select the most aligned candidate within each bin. It retains up to $4r$ task eigenvectors, uses the heuristic random-subspace gate described in Equation 9, and applies damped CV-adaptive spectral normalization. If the gate rejects task guidance, TA restores the stratified weight indices and zero task-informed bias initialization while retaining its damped normalization rule.

For both variants, the routing projection $\boldsymbol{A}$ is frozen and only the up-projection $\boldsymbol{B}$ is gradient-optimized. Token-wise routing selects k components according to the absolute scaled projection scores plus the non-gradient routing bias. The bias is updated with step size $\eta = 0.001$. The current implementation uses dense PyTorch kernels for both projections; top-k masking restricts numerical and gradient contributions but does not provide sparse-kernel acceleration or reduce optimizer-state allocation.

For the fusion diagnostic, we uniformly average the corresponding parameters of independently trained MMLU, ScienceQA, GSM8K, and CodeAlpaca adapters using $\lambda_t = 1/4$, without additional training. Because the checkpoints independently update their routing biases, this experiment evaluates empirical parameter-averaging compatibility rather than the strict shared-routing $\boldsymbol{B}$-only identity in Equation 16.

C Training and Hyperparameter Details

Hardware environment. All experiments were conducted on a single AutoDL vGPU instance with 32 GB of GPU memory. Training and evaluation used one visible GPU without distributed data parallelism.

Optimization. All experiments use seed 23. We use the 8-bit AdamW optimizer with a learning rate of 3×10^{-4}, a cosine learning-rate schedule, a warmup ratio of 0.01, zero weight decay, and gradient checkpointing. The effective batch size is 128, consisting of a per-device micro-batch size of 8 and 16 gradient-accumulation steps on a single GPU. Backbone models are loaded using 4-bit NF4 quantization with double quantization and float16 computation during training. No validation subset is removed, and the training loss is applied to both prompt and response tokens.

FlyLoRA, SVD-IMoE-W, and SVD-IMoE-TA use $r = 32$, $k = 8$, $\alpha = 64$, zero LoRA dropout, and target q_proj, k_proj, v_proj, and o_proj. The routing-bias update step is $\eta = 0.001$. Dense LoRA

Table 3: Task-specific training, calibration, and generation settings. Calibration scope applies only to SVD-IMoE-TA; SVD-IMoE-W, FlyLoRA, and dense LoRA do not use task calibration. "Input only" means that no response tokens enter the calibration stream: ScienceQA is calibrated on question and answer choices (content-only) for Llama 3.1-8B and on the rendered prompt without the response (prompt-only) for Qwen 2.5-7B.

Task	Epochs	Cutoff	Learning rate	Calibration	Eval batch	Max new
GSM8K	1	200	3×10^{-4}	Full sequence	4	512
MMLU	1	128	3×10^{-4}	Prompt only	8	128
ScienceQA	20	150	3×10^{-4}	Input only	8	256
CodeAlpaca/HumanEval	2	200	3×10^{-4}	Output only	4	512

uses the same target modules and dropout setting: LoRA-$r = 8$ uses $\alpha = 16$, whereas LoRA-$r = 32$ uses $\alpha = 64$.

D LIMITATIONS

Our evaluation is limited to two dense 7–8B language-model backbones and four downstream adaptation datasets. All reported training runs use seed 23; consequently, the results characterize observed model–task trade-offs rather than statistical significance across training seeds. Any uncertainty values reported in the tables should therefore be interpreted according to their stated estimation procedure and not as repeated-training standard deviations unless multiple training seeds were explicitly executed.

SVD-IMoE-W and SVD-IMoE-TA are complete algorithmic variants rather than a single-variable ablation. Besides task-guided direction selection, they differ in spectral normalization and routing-bias initialization. Their performance difference therefore cannot be attributed exclusively to activation-based selection. Moreover, the task-adaptive quality gate uses a fixed heuristic multiplier of 1.5 relative to the random-subspace alignment scale. We do not yet provide a sensitivity analysis for this multiplier, the calibration-set size, or the calibration scope.

The current implementation applies dense kernels to both LoRA projections. Token-wise top-k masking restricts numerical activity and gradient contributions but does not reduce optimizer-state allocation or guarantee wall-clock acceleration. Specialized sparse kernels are required to realize computational savings from conditional rank activity.

Finally, the fusion experiments consider only uniform four-adapter parameter averaging. Independently trained checkpoints may have different routing biases, route scales, and token-wise masks; therefore, these experiments do not verify the exact shared-routing $\boldsymbol{B}$-only identity in Equation 16. A controlled comparison with fixed shared routing states, non-uniform fusion coefficients, larger backbones, MoE backbones, and additional task combinations remains future work.