# LiST: Local-Simplex Test-Time LoRA Fusion

**Yihua Shao[2,3,*], Jia Li[4,*], Siyu Chen[3], Xinyu Luo[5], Yang Liu[6], Kecheng Chen[5], Xinwei Long[7], Lingyu Zhu[5], Fanhu Zeng[3], Maolin Wang[5], Ziyang Yan[8], Jingcai Guo[2], Hao Tang[9], Nicu Sebe[10], Zhenyi Wang[1]**

[1]SUSTech, [2]PolyU, [3]CASIA, [4]GDUT, [5]CityU, [6]BigAI, [7]THU, [8]TAU, [9]PKU, [10]UniTrento

[*]Equal contribution.

**Correspondence:** Zhenyi Wang, wangzy9@sustech.edu.cn and Jingcai Guo, jc-jingcai.guo@polyu.edu.hk

## Abstract

Task-specific LoRA adapters offer a modular way to specialize large language and vision-language models. However, existing adapter composition methods are mostly static and cannot adapt to individual test inputs. To address these issues, we propose **LiST**, a label-free test-time LoRA fusion framework that converts an existing LoRA bank into a target-conditioned local simplex and searches sample-specific fusion weights at inference time. LiST builds joint task representations from LoRA parameter anchors and prompt-level behavior vectors, retrieves neighboring adapters as a local search space, and performs branch-preserving fusion without updating the backbone or adapters. Candidate weights are selected by a prompt-level energy with prior, geometric, and stochastic-consistency constraints, and are deployed only when they pass a safe acceptance rule. Otherwise, LiST falls back to a target-conditioned prior. Experiments on multimodal and language benchmarks show that LiST outperforms static LoRA merging and conventional test-time adaptation baselines, while preserving task-specific adapter utility and improving robustness on unseen tasks. Code is available at https://github.com/YihuaJerry/LiST

## 1 Introduction

Multimodal Large Language Model (MLLMs) (Liu et al., 2024; Bai et al., 2025; Shao et al., 2026a) have become increasingly capable general-purpose multimodal systems that support visual tasks within a unified generative interface (Shao et al., 2025a). However, in practical deployments, a single pre-trained model is rarely sufficient for downstream tasks. A solution is to train LoRA adapters for tasks (Hu et al., 2022; Shao et al., 2025c), yielding a bank of adapters specialized for different domains (Huang et al., 2023; Shao et al., 2026b; Jiang et al., 2026; Yan et al., 2026, 2025). This adapter-bank paradigm is parameter-efficient, but it introduces an inference-time challenge: **Given a LoRA weight from a target task, how should the model adapt to the target task with less forgetting?**

Existing adapter usage strategies are typically static (Ilharco et al., 2022; Yu et al., 2024; Du et al., 2024; Yadav et al., 2023; Yang et al., 2024; Shao et al., 2026c; Zeng et al., 2025; Padkan et al., 2025). Activating only the target-task LoRA is safe but ignores transferable knowledge from related adapters, while fixed or similarity-based merging assigns the same adapter composition to all test inputs. Such task compositions cannot be adapted to instance-specific requirements, and direct parameter averaging may further distort the low-rank branch structure of LoRA adapters, causing unstable or harmful transfer (Shao et al., 2025b; Li et al., 2026). We therefore study label-free test-time LoRA fusion with a frozen backbone and a frozen bank of task-specific adapters. The goal is to construct a sample-specific fused adapter for each test input without updating either the backbone or the adapters. It is challenging because global fusion over the adapter bank can introduce negative transfer, and generative multimodal tasks do not share a unified objective for conventional test-time adaptation (Sun et al., 2020; Zhang et al., 2022; Niu et al., 2023; Yuan et al., 2023; Chen et al., 2026; Wang et al., 2026).

To address these challenges, we propose **LiST**, a Local-Simplex Prompt-Level Test-Time LoRA Fusion framework. LiST first builds joint task representations by combining LoRA parameter anchors with prompt-level behavior vectors. For each target task, these representations are used to retrieve neighboring adapters and construct a target-conditioned local simplex, which restricts test-time search to a plausible task neighborhood. At inference time, LiST optimizes only low-dimensional simplex weights around a task-conditioned prior. Fusion is performed at the LoRA branch-output



*Findings of the Association for Computational Linguistics: EMNLP 2026*, pages 19193–19205
October 24-29, 2026 ©2026 Association for Computational Linguistics

level rather than by averaging adapter parameters, preserving the structure of each task adapter. To select reliable weights without labels, LiST optimizes a prompt-level energy with prior, geometric, and stochastic-consistency constraints, followed by a safe acceptance rule that falls back to the task-conditioned prior when the searched solution is unreliable.

Empirical results show that LiST outperforms LoRA merging and conventional test-time adaptation baselines on multimodal and language tasks, while improving robustness on unseen tasks.

Our contributions are summarized as follows:

- We present **LiST**, a LoRA fusion method as label-free, sample-wise test-time fusion over a target-conditioned local simplex, optimizing only low-dimensional fusion weights while keeping the backbone frozen.
- We design a branch-preserving fusion and prompt-level energy framework that combines task uncertainty, stochastic consistency, target prior, and geometric constraints, together with a safe fallback rule for reliable adaptation.
- We demonstrate strong results on multimodal and language benchmarks, where LiST outperforms merging and conventional TTA baselines while improving unseen-task robustness.

## 2 Related Work

### 2.1 Model Fusion

Low-rank Adaptation (LoRA) (Hu et al., 2022) is now a standard fine-tuning paradigm. In multi-task settings, early work viewed fine-tuned modules as parameter-space directions and combined them linearly (Ilharco et al., 2022; Liao et al., 2025), but naive merging can cause task interference, leading to conflict-aware or component-filtering methods (Yu et al., 2024; Yadav et al., 2023). Adaptive approaches such as AdaMerging (Yang et al., 2024) learn task- or layer-wise coefficients from unlabeled data, while PEFT-oriented methods further improve LoRA fusion, including PCB-merging (Du et al., 2024), LoraHub (Huang et al., 2023), RobustMerge (Zeng et al., 2025), and ICM-Fusion (Shao et al., 2026c). Recent methods make fusion input-dependent by predicting example-conditioned weights (Qorbani et al., 2025), including extensions to multimodal PEFT (Guo et al., 2025). Unlike these mostly static fusion approaches, our method treats LoRA composition as a continuous test-time search over an adapter manifold, yielding training-free, sample-specific compositions.

### 2.2 Test-Time Adaptation

Test-time adaptation improves robustness under distribution shift by refining models during inference. Early methods update model parameters using entropy minimization, self-supervision, or consistency objectives, such as Tent (Wang et al., 2020), Test-Time Training (Sun et al., 2020), and MEMO (Zhang et al., 2022); later work also considers constrained settings such as single-instance adaptation (Khurana et al., 2021). While effective, these approaches typically adapt model parameters or high-dimensional states. Recent methods reduce the adaptation space: ELaTTA (Luo et al., 2025) searches over low-dimensional latent coefficients, and distribution-regularized refinement (Chen et al., 2025) constrains optimization to training-supported regions. Building on this shift from parameter updates to controlled search, our method adapts the composition of frozen task-specific LoRA adapters instead of latent states or predictions. This yields parameter-free, lightweight test-time adaptation compatible with multi-task adapter reuse, while enabling sample-specific refinement.

## 3 Methodology

### 3.1 Problem and Overview

We study label-free test-time LoRA fusion with a frozen backbone and a frozen bank of task-specific adapters. Let $f_{\theta_0}$ denote the pretrained backbone and let

$$\mathcal{A} = \{\Delta\theta_1, \Delta\theta_2, \ldots, \Delta\theta_N\} \tag{1}$$

denote the LoRA bank, where $\Delta\theta_t$ is the adapter trained for task $t$. Given an unlabeled test input $x$ from target task $t$, LiST predicts a fused adapter parameterized by final simplex weights $\alpha^\dagger(x,t)$:

$$\hat{y} = f_{\theta_0,\alpha^\dagger(x,t)}(x). \tag{2}$$

All LoRA weights remain fixed. The only online variable is a low-dimensional simplex-logit perturbation $\delta$, which determines the fusion weights.

LiST separates offline task-level preparation from online sample-level adaptation. Offline, it builds task representations and stores four reusable objects for each target task: the local neighbor set

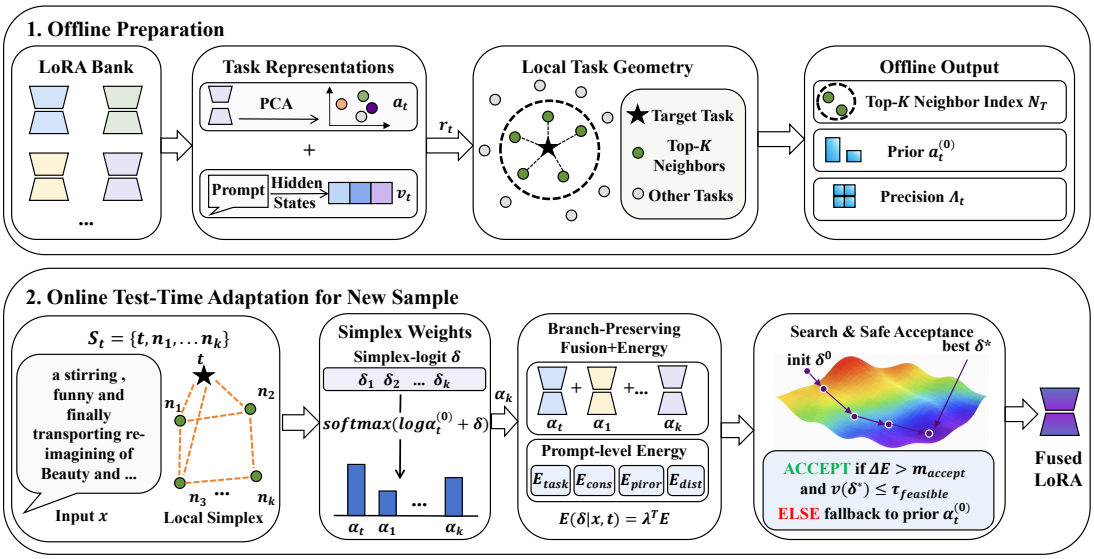



Figure 1: **Pipeline of LiST.** LiST first builds a local task simplex from an existing LoRA bank using task representations and neighbor retrieval. At test time, it searches simplex weights for each input, evaluates prompt-level energy under branch-preserving LoRA fusion, and accepts the searched weights only when they improve energy while satisfying feasibility constraints; otherwise, it falls back to the task-conditioned prior.

$\mathcal{S}_t$, the task-conditioned prior $\alpha_t^{(0)}$, the local precision matrix $\Lambda_t$, and the original LoRA branches. Online, these objects are fixed, and LiST searches only $\delta$ for each test input. The searched perturbation $\delta^\star$ and the final deployed weight $\alpha^\dagger$ are defined in Section 3.4.

### 3.2 Local Simplex Construction

LiST constructs a target-conditioned local simplex from two complementary task descriptors. First, for each task $t$, we flatten all trainable LoRA tensors into a task-level parameter vector $\ell_t$:

$$\ell_t = \text{concat}\left(\text{vec}(W_{t,1}), \ldots, \text{vec}(W_{t,M})\right) \in \mathbb{R}^D. \tag{3}$$

Then it is projected into a low-dimensional anchor space:

$$a_t = U^\top \ell_t \in \mathbb{R}^{d_a}, \qquad d_a \ll D, \tag{4}$$

where $U$ is obtained by PCA over the LoRA bank. The anchor $a_t$ is used only for neighbor retrieval and geometric regularization; the original LoRA parameters are never replaced.

Second, parameter proximity alone may not capture functional task similarity, so we extract a prompt-level behavior vector. Given a small support set $Q_t$, we run the backbone with adapter $\Delta\theta_t$ and average hidden representations:

$$v_t = \frac{1}{|Q_t|} \sum_{x \in Q_t} h_t(x). \tag{5}$$

We optionally project the behavior vector as $\tilde{v}_t = P(v_t)$ and then concatenate it with the anchor to obtain the final task representation $r_t = [a_t; \tilde{v}_t]$.

We compute task similarity by cosine similarity:

$$\text{sim}(t, i) = \frac{r_t^\top r_i}{\|r_t\|_2 \|r_i\|_2}. \tag{6}$$

For target task $t$, we retrieve top-$K$ neighboring adapters and define the local simplex as

$$\mathcal{S}_t = \{t\} \cup \text{TopK}_{i \neq t}\, \text{sim}(t, i). \tag{7}$$

Thus, online search is restricted to the target adapter and a small set of related adapters.

Inside $\mathcal{S}_t$, LiST defines a task-conditioned prior $\alpha_t^{(0)}$. The target LoRA receives mass $\rho$ as $\alpha_{t,t}^{(0)} = \rho$.

The remaining mass is distributed to neighbors according to similarity:

$$\alpha_{t,j}^{(0)} = (1-\rho) \frac{\exp(\text{sim}(t,j)/\tau)}{\sum_{k \in \mathcal{S}_t \setminus \{t\}} \exp(\text{sim}(t,k)/\tau)}, \tag{8}$$

where $j \neq t$. We also estimate the local covariance of anchors in $\mathcal{S}_t$:

$$\Sigma_t = \text{Cov}\{a_i \mid i \in \mathcal{S}_t\} + \epsilon I, \tag{9}$$

and denote precisely with $\Lambda_t = \Sigma_t^\dagger$.

The prior $\alpha_t^{(0)}$ is used both as the initialization of online search and as the fallback solution. The precision matrix $\Lambda_t$ defines a local trust region by measuring how far the fused anchor barycenter deviates from the target anchor.

**Unseen-domain inference.** For unseen-domain evaluation, no target-domain LoRA adapter is trained or used. LiST therefore forms the local simplex from the retrieved seen-domain adapters only, and replaces the task-conditioned prior in Eq. (10)–(11) with a similarity-normalized prior:

$$\alpha_j^{(0)} = \frac{\exp(\mathrm{sim}(u,j)/\tau)}{\sum_{k\in\mathcal{S}} \exp(\mathrm{sim}(u,k)/\tau)}, \quad j \in \mathcal{S}. \quad (10)$$

The same search objective and acceptance rule are applied, with fallback to this retrieval-based prior.

### 3.3 Energy Aware Branch-Preserving Fusion

LiST avoids direct LoRA parameter averaging. For a LoRA-augmented linear layer with frozen base weight $W_0$, each adapter $i \in \mathcal{S}_t$ has low-rank branch $(A_i, B_i)$ and scale $s_i$. Given simplex weights $\alpha$, LiST fuses branch outputs as

$$y = W_0 x + \sum_{i\in\mathcal{S}_t} \alpha_i s_i B_i(A_i x). \quad (11)$$

This preserves the structure of every LoRA branch and reduces online adaptation to simplex-weight optimization.

We parameterize simplex weights by an unconstrained perturbation $\delta$ around the task-conditioned prior:

$$\alpha(\delta) = \mathrm{softmax}\left(\log \alpha_t^{(0)} + \delta\right). \quad (12)$$

When $\delta = 0$, the fusion weight recovers the prior:

$$\alpha(0) = \alpha_t^{(0)}. \quad (13)$$

For each candidate $\delta$, LiST performs $M$ stochastic forward passes and obtains predictive objects in the model's native prediction space:

$$\{q_m(x,\delta)\}_{m=1}^{M}. \quad (14)$$

The main task energy is

$$E_{\mathrm{task}}(\delta|x) = \frac{1}{M}\sum_{m=1}^{M} U(q_m(x,\delta)), \quad (15)$$

where $U(\cdot)$ is a task-aware uncertainty functional.

To avoid unreliable low-uncertainty solutions, LiST adds three feasibility terms. The prior deviation is

$$E_{\mathrm{prior}}(\delta,t) = \mathrm{KL}\left(\alpha(\delta)\|\alpha_t^{(0)}\right). \quad (16)$$

The fused anchor barycenter is

$$z(\delta) = \sum_{i\in\mathcal{S}_t} \alpha_i(\delta) a_i. \quad (17)$$

The local geometric distance is

$$E_{\mathrm{dist}}(\delta,t) = (z(\delta) - a_t)^\top \Lambda_t (z(\delta) - a_t). \quad (18)$$

When predictive objects admit distributions, stochastic consistency is measured by

$$E_{\mathrm{cons}}(\delta|x) = \mathrm{JS}_{\mathrm{pairwise}}(q_1,\ldots,q_M). \quad (19)$$

where $JS$ indicates Jensen-Shannon Entropy.

Since these terms have different scales, LiST uses a relative normalized augmented-Lagrangian objective. We normalize the task energy against the prior baseline:

$$\widetilde{E}_{\mathrm{task}} = \frac{E_{\mathrm{task}}(\delta|x,t) - E_{\mathrm{task}}(0|x,t)}{s_{\mathrm{task}} + \epsilon}. \quad (20)$$

Each constraint is converted into a normalized violation:

$$c_i = \frac{E_i - \tau_i}{s_i + \epsilon}, \qquad i \in \{\mathrm{prior}, \mathrm{dist}, \mathrm{cons}\}. \quad (21)$$

The final energy is

$$\mathcal{E}(\delta|x,t) = \widetilde{E}_{\mathrm{task}} + \sum_i \lambda_i [c_i]_+ + \frac{\rho_{\mathrm{AL}}}{2}\sum_i [c_i]_+^2, \quad (22)$$

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

If the consistency term is undefined for a task-specific predictive object, it is omitted.

Because each energy component is normalized relative to the prior baseline or converted into a normalized violation, LiST uses fixed coefficients across tasks. The thresholds and normalization scales are estimated from offline task statistics, and the global hyperparameters are kept fixed throughout evaluation. Sensitivity analyses for $K$, $M$, and $m_{\mathrm{accept}}$ are reported in Section A.2.

### 3.4 Search and Safe Prediction

The objective $\mathcal{E}(\delta|x,t)$ is stochastic and derivative-free, but the search dimension is only $|\mathcal{S}_t|$. We therefore use a standard two-stage CMA-ES optimizer (Luo et al., 2025) over the simplex-logit variable $\delta$, initialized at $\delta_0 = 0$

The best searched perturbation is denoted by

$$\delta^\star = \arg\min_{\delta} \mathcal{E}(\delta|x,t). \quad (23)$$

| Method | Seen Tasks | | | | | | | | | Unseen Tasks | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SciQA | Image | VQA | REC | OCR | VizWiz | Flickr | IconQA | **Avg.** | AVQA | Image-R | S2W | TabMWP | **Avg.** |
| **Results on Qwen3-VL-8B** | | | | | | | | | | | | | | |
| Original Model | 85.97 | 35.01 | 63.37 | 48.13 | 62.91 | 27.26 | 28.63 | 67.50 | 52.35 | 81.77 | 81.47 | 10.82 | 58.30 | 58.09 |
| LoRA | 97.71 | 97.74 | **75.22** | 89.21 | 72.83 | **74.94** | 61.69 | **99.16** | 83.56 | – | – | – | – | – |
| Task-Arithmetic (Ilharco et al., 2022) | 78.97 | 73.44 | 60.53 | 78.05 | 62.85 | 45.34 | 38.17 | 88.79 | 65.77 | 69.30 | 80.72 | 10.86 | 61.23 | 55.52 |
| TIES-Merging (Yadav et al., 2023) | 75.05 | 81.17 | 64.75 | 63.92 | 61.66 | 41.84 | 39.06 | 84.25 | 63.96 | 67.86 | 82.89 | 9.85 | 65.11 | 56.43 |
| DARE (Yu et al., 2024) | 87.36 | 81.70 | 60.33 | 73.32 | 54.59 | 48.86 | 38.79 | 88.06 | 66.63 | 68.29 | 83.01 | 9.38 | 65.35 | 56.50 |
| PCB-Merging (Du et al., 2024) | 83.61 | 47.25 | 53.65 | 56.21 | 59.50 | 33.76 | 28.03 | 73.09 | 54.39 | 71.73 | 85.12 | 9.64 | 64.87 | 57.84 |
| ICM-Fusion (Shao et al., 2026c) | 91.75 | 86.98 | 67.48 | 77.81 | 64.10 | 53.23 | 43.48 | 89.69 | 71.81 | 75.72 | 84.38 | 8.33 | 68.76 | 59.29 |
| RobustMerge (Zeng et al., 2025) | 87.31 | 81.56 | 64.16 | 77.20 | 71.45 | 48.99 | 41.72 | 91.97 | 70.55 | 68.29 | 84.91 | 11.10 | 65.89 | 57.54 |
| AdaMerging (Yang et al., 2024) | 83.35 | 40.28 | 63.31 | 74.72 | 58.46 | 41.74 | 34.82 | 71.40 | 58.51 | 73.91 | 85.47 | 8.64 | 66.29 | 58.58 |
| TENT (Wang et al., 2020) | 96.67 | 95.82 | 72.07 | 87.97 | 69.28 | 72.13 | 56.92 | 98.09 | 81.12 | 81.67 | 87.17 | 12.46 | 71.15 | 63.11 |
| MEMO (Zhang et al., 2022) | 96.64 | 96.82 | 71.94 | 86.03 | 71.91 | 71.91 | 59.84 | 96.86 | 81.49 | 83.85 | 88.45 | 13.19 | 71.74 | 64.31 |
| **LiST (Ours)** | **97.75** | **97.78** | 75.16 | **90.03** | **73.12** | 74.88 | **62.88** | 99.09 | **83.84** | **87.68** | **89.94** | **16.66** | **78.42** | **68.18** |
| **Results on LLaVA-V1.5-7B** | | | | | | | | | | | | | | |
| Original Model | 71.73 | 40.87 | 42.88 | 36.10 | 31.23 | 41.03 | 39.07 | 38.09 | 42.62 | 78.45 | 70.36 | 10.05 | 23.10 | 45.49 |
| LoRA | 90.73 | 97.40 | **68.12** | 75.70 | **54.55** | 70.12 | 57.28 | **73.33** | 73.40 | – | – | – | – | – |
| Task-Arithmetic (Ilharco et al., 2022) | 70.98 | 57.43 | 53.89 | 35.66 | 35.31 | 46.82 | 42.56 | 36.68 | 47.42 | 67.03 | 68.82 | 8.46 | 33.34 | 44.41 |
| TIES-Merging (Yadav et al., 2023) | 68.94 | 58.92 | 49.18 | 36.19 | 31.28 | 41.98 | 42.41 | 31.53 | 45.05 | 66.94 | 67.57 | 9.72 | 34.14 | 44.59 |
| DARE (Yu et al., 2024) | 68.76 | 57.63 | 52.84 | 32.84 | 36.18 | 39.18 | 45.77 | 37.67 | 46.36 | 68.25 | 66.94 | 8.39 | 33.78 | 44.34 |
| PCB-Merging (Du et al., 2024) | 70.63 | 61.89 | 51.22 | 36.73 | 35.99 | 42.39 | 43.12 | 31.98 | 46.74 | 69.30 | 66.17 | 9.46 | 33.21 | 44.53 |
| ICM-Fusion (Shao et al., 2026c) | 78.94 | 75.32 | 62.41 | 45.68 | 41.96 | 57.15 | 44.17 | 59.34 | 58.12 | 76.52 | 69.54 | 9.84 | 35.68 | 47.90 |
| RobustMerge (Zeng et al., 2025) | 73.43 | 65.54 | 56.81 | 42.84 | 38.77 | 46.61 | 33.81 | 45.82 | 50.45 | 68.77 | 67.98 | 8.84 | 32.82 | 44.60 |
| AdaMerging (Yang et al., 2024) | 71.23 | 71.37 | 47.21 | 38.19 | 30.41 | 47.85 | 41.92 | 42.43 | 48.83 | 69.38 | 66.43 | 7.29 | 32.20 | 43.83 |
| TENT (Wang et al., 2020) | 82.65 | 95.87 | 63.07 | 63.68 | 43.67 | 58.84 | 46.97 | 60.23 | 64.37 | 78.41 | 78.92 | 12.26 | 42.18 | 52.94 |
| MEMO (Zhang et al., 2022) | 84.01 | 96.35 | 65.91 | 65.93 | 42.58 | 61.88 | 47.19 | 59.18 | 65.38 | 79.57 | 77.23 | 13.48 | 43.23 | 53.38 |
| **LiST (Ours)** | **91.65** | **97.59** | 67.99 | **75.73** | 54.52 | **70.16** | **57.37** | 73.26 | **73.53** | **83.68** | **81.94** | **15.34** | **46.64** | **56.15** |

Table 1: **Multimodal results.** Across seen and unseen tasks, LiST delivers the best average performance across multimodal backbones, preserving task-specific LoRA utility while improving generalization to unseen distributions.

LiST does not always deploy the lowest-energy candidate. Let the energy improvement over the prior be

$$\Delta E = \mathcal{E}(0|x,t) - \mathcal{E}(\delta^\star|x,t), \tag{24}$$

and let the maximum normalized violation be

$$\nu(\delta^\star) = \max\{c_{\text{prior}}(\delta^\star), c_{\text{dist}}(\delta^\star), c_{\text{cons}}(\delta^\star)\}. \tag{25}$$

LiST accepts the searched candidate only if

$$\Delta E > m_{\text{accept}} \quad \text{and} \quad \nu(\delta^\star) \leq \tau_{\text{feasible}}. \tag{26}$$

The final deployed fusion weight is

$$\alpha^\dagger(x,t) = \begin{cases} \alpha(\delta^\star), & \text{if Eq. 26 holds,} \\ \alpha_t^{(0)}, & \text{otherwise.} \end{cases} \tag{27}$$

The final prediction is generated by the frozen backbone with $\alpha^\dagger$. This safe rule prevents high-violation low-energy candidates from being deployed and connects online search to the offline task-conditioned prior.

# 4 Experiment

## 4.1 Experimental Setup

**Training Details.** For multimodal tasks, we perform experiments on Qwen3-VL-8B (Bai et al., 2025) and LLaVA-V1.5-7B (Liu et al., 2024). For language tasks, we perform experiments on DeepSeek-7B (Bi et al., 2024) and Qwen3-8B (Yang et al., 2025). We perform all experiments on four RTX 4090 GPUs.

**Benchmarks.** We widely evaluate multimodal tasks on MM-MergeBench (Zeng et al., 2025) and language tasks on GLUE benchmark (Wang et al., 2018). Details can be found in Sec. A.1.

**Compared Baselines.** We extensively compare our method against model fusion and test-time adaptation (TTA) methods to demonstrate its effectiveness. Specifically, the model fusion methods include Task-Arithmetic (Ilharco et al., 2022), TIES-Merging (Yadav et al., 2023), DARE (Yu et al., 2024), PCB-Merging (Du et al., 2024), ICM-Fusion (Shao et al., 2026c) and RobustMerge (Zeng et al., 2025). The TTA method includes AdaMerging (Yang et al., 2024), MEMO (Zhang et al., 2022) and TENT (Wang et al., 2020).

## 4.2 Main Results

**Outperforming Across Multimodal Tasks.** As shown in Table 1, on both models, LiST achieves the best overall performance across multimodal benchmarks. It consistently improves over the original model and remains close to, or even surpasses, single-task LoRA on several datasets, indicating effective use of transferable knowledge while pre-

| Method | SST-2 | MRPC | RTE | QNLI | QQP | CoLA | MNLI-m | MNLI-mm | STS-B | WNLI | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Results on DeepSeek-7B** | | | | | | | | | | | |
| Original Model | 60.89 | 58.03 | 47.65 | 52.11 | 40.13 | 7.85 | 33.70 | 34.02 | 13.87 | 47.88 | 39.61 |
| LoRA | 97.24 | **88.72** | **89.16** | 94.96 | 90.47 | 67.49 | 91.12 | 91.02 | 90.16 | 77.46 | 87.78 |
| Task-Arithmetic (Ilharco et al., 2022) | 58.37 | 35.47 | 58.48 | 40.14 | 36.83 | 8.09 | 58.75 | 60.15 | 24.13 | 57.74 | 43.81 |
| TIES-Merging (Yadav et al., 2023) | 68.92 | 48.69 | 68.92 | 51.05 | 47.77 | 43.65 | 46.86 | 49.73 | 34.01 | 54.36 | 51.40 |
| DARE (Yu et al., 2024) | 49.08 | 15.80 | 47.29 | 52.26 | 39.41 | 4.69 | 32.89 | 33.45 | 21.03 | 56.33 | 35.22 |
| PCB-Merging (Du et al., 2024) | 75.57 | 49.56 | 46.21 | 51.19 | 47.71 | 37.09 | 41.79 | 44.61 | 24.46 | 47.88 | 46.61 |
| ICM-Fusion (Shao et al., 2026c) | 82.19 | 78.76 | 72.53 | 69.21 | 63.68 | 52.84 | 76.69 | 78.13 | 63.18 | 61.17 | 69.84 |
| RobustMerge (Zeng et al., 2025) | 72.82 | 41.96 | 63.17 | 51.27 | 33.86 | 28.14 | 79.93 | 81.20 | 46.85 | 56.33 | 55.55 |
| AdaMerging (Yang et al., 2024) | 60.43 | 50.82 | 61.73 | 50.85 | 35.35 | 24.74 | 72.26 | 72.79 | 59.72 | 45.07 | 53.38 |
| TENT (Wang et al., 2020) | 93.91 | 85.29 | 86.08 | 89.87 | 89.16 | 62.91 | 87.17 | 88.21 | 86.09 | 74.78 | 84.35 |
| MEMO (Zhang et al., 2022) | 94.76 | 85.74 | 85.32 | 91.89 | 87.97 | 65.38 | 89.08 | 88.93 | 89.73 | 75.12 | 85.39 |
| **LiST (Ours)** | **97.25** | 88.63 | 88.95 | **94.98** | **90.49** | **68.24** | **91.16** | **91.07** | **90.18** | **77.84** | **87.88** |
| **Results on Qwen3-8B** | | | | | | | | | | | |
| Original Model | 74.03 | 64.61 | 66.64 | 70.48 | 51.83 | 42.89 | 51.02 | 49.96 | 64.45 | 39.43 | 57.53 |
| LoRA | 96.67 | 89.97 | 89.89 | 95.78 | 90.70 | 65.94 | 91.63 | 91.62 | 90.08 | 70.42 | 87.27 |
| Task-Arithmetic (Ilharco et al., 2022) | 86.90 | 68.35 | 79.13 | 75.69 | 60.80 | 54.85 | 74.12 | 74.44 | 78.77 | 56.33 | 70.94 |
| TIES-Merging (Yadav et al., 2023) | 65.36 | 74.91 | 48.01 | 73.67 | 80.01 | 52.64 | 57.20 | 64.91 | 51.96 | 57.74 | 62.64 |
| DARE (Yu et al., 2024) | 76.33 | 69.25 | 69.89 | 72.71 | 57.21 | 42.37 | 81.47 | 81.26 | 70.10 | 47.60 | 66.82 |
| PCB-Merging (Du et al., 2024) | 50.57 | 74.21 | 48.73 | 51.25 | 53.55 | 36.08 | 41.85 | 43.31 | 76.91 | 57.74 | 53.42 |
| ICM-Fusion (Shao et al., 2026c) | 82.23 | 78.46 | 73.53 | 86.80 | 81.13 | 53.67 | 76.70 | 79.44 | 82.56 | 60.56 | 75.51 |
| RobustMerge (Zeng et al., 2025) | 80.64 | 78.50 | 79.16 | 84.32 | 82.39 | 59.70 | 81.12 | 81.27 | 80.15 | 57.74 | 76.50 |
| AdaMerging (Yang et al., 2024) | 53.89 | 75.04 | 48.73 | 51.61 | 54.03 | 32.13 | 46.82 | 48.83 | 68.43 | 53.52 | 53.30 |
| TENT (Wang et al., 2020) | 85.73 | 81.76 | 80.48 | 91.59 | 84.82 | 60.23 | 81.28 | 81.47 | 84.45 | 63.38 | 79.52 |
| MEMO (Zhang et al., 2022) | 91.58 | 82.22 | 83.17 | 90.26 | 85.58 | 61.10 | 83.55 | 84.16 | 86.49 | 60.56 | 80.87 |
| **LiST (Ours)** | **96.79** | **90.27** | **90.25** | **95.84** | **90.82** | **67.35** | **91.73** | **91.80** | **90.31** | **71.83** | **87.70** |

Table 2: **Language results.** Across diverse tasks and LLM backbones, LiST achieves the best average performance, preserving task-specific understanding while leveraging transferable knowledge across related tasks.

| **Method** | **Avg** |
|---|---|
| Target LoRA | 87.35 |
| Static Prior Fusion | 78.83 |
| + Energy Search | 82.75 |
| + Prior Constraint | 83.32 |
| + Prior + Geo | 85.91 |
| + Prior + Geo + Stability | 87.55 |
| **Full LiST** | **87.70** |

Table 3: **Component ablation results.**

| **Variant** | **Avg** | **Entropy ↓** | **JS ↓** | **KL ↓** | **Geo. Dist. ↓** | **Wrong Confident ↓** |
|---|---|---|---|---|---|---|
| Entropy Only | 79.36 | 7.89 | 0.239 | 3.98 | 3.91 | 9.616 |
| Entropy + JS | 81.14 | 7.85 | 0.216 | 3.70 | 3.58 | 9.509 |
| Entropy + Prior | 83.97 | 7.73 | 0.174 | 3.118 | 2.953 | 9.505 |
| Entropy + Geo | 86.12 | 7.43 | 0.163 | 2.311 | 1.011 | 9.478 |
| Prior + Geo | 86.69 | 7.35 | 0.101 | 1.684 | 0.859 | 9.450 |
| **Full Energy** | **87.70** | **7.13** | **0.078** | **1.178** | **0.772** | **9.227** |

Table 4: **Energy and constraint ablation results.**

| **Neighbor Strategy** | Avg | **Static Fusion** | **Neg. Transfer ↓** | **Geo. Dist. ↓** |
|---|---|---|---|---|
| Target Only | 87.35 | 84.18 | **0.00** | **0.00** |
| Random K | 82.95 | 80.99 | 0.13 | 2.87 |
| Anchor Only | 84.97 | 85.01 | 0.11 | 0.90 |
| Behavior Only | 84.80 | 85.86 | 0.09 | 0.83 |
| Joint Representation | 86.22 | 86.12 | 0.07 | 0.74 |
| Global Simplex | 86.87 | 86.43 | 0.11 | 1.02 |
| **Full Local Simplex** | **87.70** | **87.68** | 0.02 | 0.65 |

Table 5: **Ablation on local simplex construction.**

serving target-task capability. Compared with static merging methods and test-time adaptation baselines, LiST better balances fusion and adaptation. These results show that sample-adaptive local-simplex fusion can more effectively exploit complementary information among related LoRA adapters while preserving task-specific knowledge.

**Consistent Effectiveness for Language Modality.** As shown in Table 2, LiST also achieves strong performance on both models. It performs better than single-task LoRA on most tasks, showing that LiST can generalize to text-only LLM adaptation while selectively reusing useful cross-task knowledge. It indicates LiST provides more stable and effective fusion. These results indicate that sample-wise test-time search better preserves target-task capability while exploiting the task-conditioned geometry of related LoRA adapters.

**Out-of-Distribution Generalization.** As shown in Table 1, we further evaluates LiST on unseen domains in MM-MergeBench. These domains contain only test samples and provide no train-

| Variant | Avg. | Accepted Subset | Rejected Subset | Neg. Transfer ↓ | Fallback Rate |
|---|---|---|---|---|---|
| Prior Only | 80.84 | 76.09 | **88.41** | 0.17 | – |
| Always Accept | 82.83 | 78.16 | 82.38 | 0.13 | 0.00 |
| Energy-only Accept | 86.60 | 79.38 | 84.57 | 0.09 | 29.71 |
| Constraint-only Accept | 86.92 | 80.79 | 86.06 | 0.08 | 36.38 |
| **Full Accept Rule** | **87.70** | **82.76** | **88.41** | **0.02** | **59.92** |

Table 6: **Safe fallback ablation results.**

ing samples for constructing target-domain LoRA adapters, so no unseen-domain LoRA is available during evaluation. Under this stricter protocol, LiST reuses only the adapters trained on seen domains and constructs the local simplex from retrieved seen-domain adapters. Despite the absence of target-domain adapters, LiST achieves the best unseen-domain average on both Qwen3-VL-8B and LLaVA-V1.5-7B, consistently improving over the original model as well as static merging and test-time adaptation baselines. These results suggest that sample-wise search over a local adapter simplex can transfer useful capabilities from related seen domains, while the task-conditioned prior and feasibility-aware acceptance rule help reduce negative transfer from irrelevant adapters.

### 4.3 Ablation Studies

All ablation experiments are conducted on the GLUE benchmark using Qwen3-8B.

#### 4.3.1 Effect of Core Components

As shown in Table 3, we ablate the major components of LiST. The target-task LoRA provides a strong and safe baseline but cannot exploit knowledge from related adapters. Static prior fusion performs worse, suggesting that fixed-weight adapter mixing can disrupt task-specific behavior. Test-time energy search improves performance through sample-wise adapter selection, while the prior constraint stabilizes the search by limiting deviation from the target task. The geometric constraint yields a larger gain, highlighting the importance of staying close to the local task manifold. Stochastic stability further improves robustness, and Full LiST achieves the best average result. Overall, these results show that LiST benefits from combining sample-wise search, target-aware prior regularization, local geometry, and stochastic consistency.

#### 4.3.2 Ablation on Energy Constraints

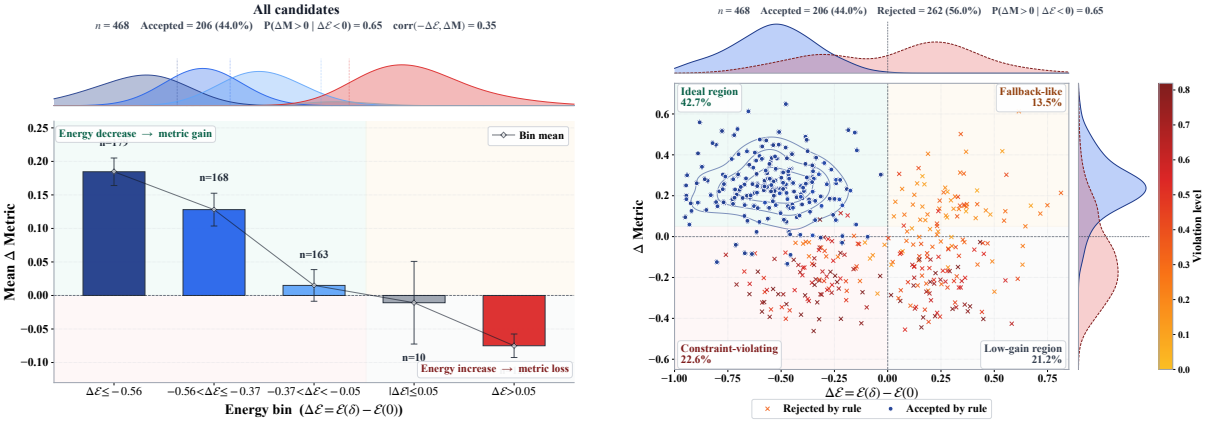
Table 4 ablates the energy components. Entropy-only optimization gives the weakest performance and the highest wrong-confident score, confirming that uncertainty minimization alone can produce over-confident unreliable predictions. Adding stochastic consistency improves performance and JS stability by encouraging agreement under LoRA branch perturbations. Prior and geometric constraints further restrict the search to task-consistent and locally plausible regions, with the geometric term giving a particularly strong gain. The full energy achieves the best average performance and the lowest entropy, JS divergence, KL deviation, geometric distance, and wrong-confident score. These results show that the proposed energy is a feasibility-aware surrogate that balances confidence, stability, task identity, and local manifold plausibility.

#### 4.3.3 Ablation on Local Simplex Construction

Table 5 compares different simplex construction strategies. The target-only baseline is safe but lacks cross-task transfer, while random neighbors cause strong negative transfer. Anchor-only and behavior-only retrieval improve over random selection but remain suboptimal, suggesting that parameter geometry and model behavior provide complementary task-similarity signals. Their joint representation further improves performance and reduces geometric displacement. Global simplex search underperforms the full local simplex, indicating that irrelevant adapters weaken the local manifold assumption. Full local simplex achieves the best average performance, low negative transfer, and small geometric distance, confirming the importance of target-conditioned local retrieval for reliable adapter reuse.

#### 4.3.4 Safe acceptance and fallback.

Table 6 compares acceptance rules for deploying searched fusion weights. The prior-only baseline is stable but cannot fully exploit transferable adapters, while always accepting searched candidates improves performance but increases negative transfer. Energy-only and constraint-only rules help, but each misses part of the reliability criterion. The full acceptance rule achieves the best average perfor-




(a) Binned energy change vs. metric gain.

(b) Candidate-level alignment under constraint violations.

Figure 2: **Energy–performance alignment under feasibility constraints.** Candidate perturbations are grouped by relative energy change, $\Delta\mathcal{E} = \mathcal{E}(\delta) - \mathcal{E}(0)$. Larger energy decreases generally correspond to larger downstream gains. However, low-energy candidates with high normalized constraint violations can still hurt performance, indicating that energy minimization must be coupled with feasibility-aware selection.

mance and the lowest negative transfer by requiring both sufficient energy improvement and low constraint violation. Its higher fallback rate shows that LiST rejects risky candidates and returns to the task-conditioned prior when search is unreliable.

## 5 Analyses

Since LiST performs label-free test-time search, an important question is whether the optimized energy is aligned with the downstream task metric. We analyze this by comparing the energy change of candidate simplex perturbations with their corresponding metric change, both measured relative to the task-conditioned prior. Specifically, for a candidate perturbation $\delta$, we define

$$\Delta\mathcal{E} = \mathcal{E}(\delta) - \mathcal{E}(0) \tag{28}$$

where smaller value indicates larger energy decrease, and measure the corresponding task-metric change $\Delta\mathrm{Metric}$.

Figure 2(a) bins candidate perturbations by $\Delta\mathcal{E}$. Candidates with larger energy decreases yield higher average metric gains, while energy-increasing candidates lead to negative changes. This shows that the proposed prompt-level energy is an effective surrogate for test-time selection.

However, energy reduction alone does not ensure reliable adaptation. Figure 2(b) shows a candidate-level scatter plot, with each point representing a simplex perturbation and color indicating the maximum normalized constraint violation,

$$\nu(\delta) = \max\{c_{\text{prior}}, c_{\text{dist}}, c_{\text{cons}}\} \tag{29}$$

Although many low-energy candidates improve the task metric, some low-energy candidates with large constraint violations still degrade performance. These harmful candidates usually deviate from the target prior, move outside the local task geometry, or produce unstable stochastic predictions. This observation explains why LiST does not simply deploy the lowest-energy candidate. Instead, the safe acceptance rule accepts a searched candidate only when it achieves sufficient energy improvement while remaining feasible; otherwise, the model falls back to the task-conditioned prior.

Overall, the analysis shows that energy decrease is useful but must be interpreted under feasibility constraints. The prior, geometry, and consistency constraints make the energy signal more reliable for final prediction. More analyses is in Section B.

## 6 Conclusion

In this paper, we introduced LiST, a local-simplex framework for label-free test-time LoRA fusion. LiST converts a frozen adapter bank into a target-conditioned local search space, performs branch-preserving fusion, and optimizes only low-dimensional simplex weights using a prompt-level energy with prior, geometry, and consistency constraints. A safe acceptance rule further prevents unreliable low-energy solutions by falling back to the task-conditioned prior. Experiments on multimodal and language benchmarks show that LiST improves over static merging and conventional TTA baselines while preserving task-specific adapter utility.

## Limitations

LiST assumes access to a bank of related task-specific LoRA adapters, and its benefit may be limited when no useful neighboring adapters are available. The test-time search introduces additional inference cost compared with directly using a single adapter, although the search is low-dimensional and all model parameters remain frozen. Finally, the prompt-level energy is a surrogate objective and may be less aligned with task metrics for highly open-ended generation tasks.

## Ethical Considerations

LiST operates entirely on publicly available pretrained models and benchmarks, and introduces no new data collection. The method reduces per-task compute by reusing frozen adapters rather than retraining, lowering the carbon footprint of multi-task deployment. We foresee no direct ethical risks specific to LoRA composition; standard concerns around model misuse and bias inherited from base models apply equally here.

## Acknowledgement

This work is supported by Hong Kong RGC General Research Fund (Grant Nos. 15221123, 15216424, and 15211525), and the Hong Kong PolyU Internal Research Fund (Grant Nos. P0058468 and P0062168).

## References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Kecheng Chen, Pingping Zhang, Tiexin Qin, Shiqi Wang, Hong Yan, and Haoliang Li. 2025. Test-time adaptation for image compression with distribution regularization. In *International Conference on Learning Representations*, volume 2025, pages 95215–95232.

Yinghao Chen, Yeying Jin, Xiang Chen, Yanyan Wei, Ziyang Yan, and Yaowen Fu. 2026. Unpaired image deraining using reward-guided self-reinforcement strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1342–1354.

Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim K Goh, Ho-Kin Tang, Daojing He, and 1 others. 2024. Parameter competition balancing for model merging. *Advances in Neural Information Processing Systems*, 37:84746–84776.

Haiyang Guo, Fanhu Zeng, Ziwei Xiang, Fei Zhu, Da-Han Wang, Xu-Yao Zhang, and Cheng-Lin Liu. 2025. Hide-llava: Hierarchical decoupling for continual instruction tuning of multimodal large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13572–13586.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Chongcong Jiang, Tianxingjian Ding, Chuhan Song, Jiachen Tu, Ziyang Yan, Yihua Shao, Zhenyi Wang, Yuzhang Shang, Tianyu Han, and Yu Tian. 2026. Medical sam3: A foundation model for universal prompt-driven medical image segmentation. *arXiv preprint arXiv:2601.10880*.

Ansh Khurana, Sujoy Paul, Piyush Rai, Soma Biswas, and Gaurav Aggarwal. 2021. Sita: Single image test-time adaptation. *arXiv preprint arXiv:2112.02355*.

Hong Li, Chongjie Ye, Houyuan Chen, Weiqing Xiao, Ziyang Yan, Lixing Xiao, Zhaoxi Chen, Jianfeng Xiang, Shaocong Xu, Xuhui Liu, and 1 others. 2026. Near: Coupled neural asset-renderer stack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 29834–29844.

Minwen Liao, Haobo Dong, Xinyi Wang, Kurban Ubul, Yihua Shao, and Ziyang Yan. 2025. Gm-moe: Low-light enhancement with gated-mechanism mixture-of-experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8766–8776.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

Xinyu Luo, Jie Liu, Kecheng Chen, Junyi Yang, Bo Ding, Arindam Basu, and Haoliang Li. 2025. Efficient test-time adaptation through latent subspace coefficients search. *arXiv preprint arXiv:2510.11068*.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. 2023. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*.

Nazanin Padkan, Ziyan Yan, and Fabio Remondino. 2025. Evaluating monocular depth estimation methods on industrial objects. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48:175–181.

Reza Qorbani, Gianluca Villani, Theodoros Panagiotakopoulos, Marc Botet Colomer, Linus Härenstam-Nielsen, Mattia Segu, Pier Luigi Dovesi, Jussi Karlgren, Daniel Cremers, Federico Tombari, and 1 others. 2025. Semantic library adaptation: Lora retrieval and fusion for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9804–9815.

Yihua Shao, Yan Gu, Minxi Yan, Siyu Chen, Haiyang Liu, Ziyang Yan, Yongjia Li, Yan Wang, Qun SONG, Hao Tang, and 1 others. 2026a. Gradient enhancement task aware post-training quantization. In *35th International Joint Conference on Artificial Intelligence (IJCAI 2026)*.

Yihua Shao, Haojin He, Sijie Li, Siyu Chen, Xinwei Long, Fanhu Zeng, Yuxuan Fan, Muyang Zhang, Ziyang Yan, Ao Ma, and 1 others. 2025a. Eventvad: Training-free event-aware video anomaly detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 2586–2595.

Yihua Shao, Deyang Lin, Minxi Yan, Siyu Chen, Fanhu Zeng, Minwen Liao, Ao Ma, Ziyang Yan, Haozhe Wang, Yan Wang, and 1 others. 2026b. Tr-dq: Time-rotation diffusion quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 8869–8877.

Yihua Shao, Xiaofeng Lin, Xinwei Long, Siyu Chen, Minxi Yan, Yang Liu, Ziyang Yan, Ao Ma, Hao Tang, and Jingcai Guo. 2026c. Icm-fusion: In-context meta-optimized lora fusion for multi-task adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 8860–8868.

Yihua Shao, Yeling Xu, Xinwei Long, Siyu Chen, Ziyang Yan, Haoting Liu, Yan Wang, Hao Tang, and Yang Yang. 2025b. Accidentblip: Agent of accident warning based on ma-former. In *2025 IEEE Intelligent Vehicles Symposium (IV)*, pages 2156–2161. IEEE.

Yihua Shao, Minxi Yan, Yang Liu, Siyu Chen, Wenjie Chen, Xinwei Long, Ziyang Yan, Lei Li, Chenyu Zhang, Nicu Sebe, Hao Tang, Yan Wang, Hao Zhao, Mengzhu Wang, and Jingcai Guo. 2025c. In-context meta lora generation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 6138–6146. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Nan Wang, Lixing Xiao, Yuantao Chen, Weiqing Xiao, Pierre Merriaux, Lei Lei, Ziyang Yan, Saining Zhang, Shaocong Xu, Bohan Li, and 1 others. 2026. Unifying appearance codes and bilateral grids for driving scene gaussian splatting. *Advances in Neural Information Processing Systems*, 38:29827–29858.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115.

Ziyang Yan, Wenzhen Dong, Yihua Shao, Yuhang Lu, Haiyang Liu, Jingwen Liu, Haozhe Wang, Zhe Wang, Yan Wang, Fabio Remondino, and 1 others. 2025. Renderworld: World model with self-supervised 3d label. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6063–6070. IEEE.

Ziyang Yan, Yihua Shao, Minwen Liao, Siyu Chen, Nan Wang, Muyuan Lin, Jenq-Neng Hwang, Hao Zhao, Fabio Remondino, and Lei Li. 2026. 3dsceneeditor: Controllable 3d scene editing with gaussian splatting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1852–1863.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, volume 2024, pages 22743–22763.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Longhui Yuan, Binhui Xie, and Shuang Li. 2023. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15922–15932.

Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. 2025. Robustmerge: Parameter-efficient model merging for mllms with direction robustness. *Advances in Neural Information Processing Systems*, 38:71071–71095.

Marvin Zhang, Sergey Levine, and Chelsea Finn. 2022. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642.

# A Appendix

## A.1 Benchmark Details

The GLUE benchmark includes SST-2, MRPC, RTE, QNLI, QQP, CoLA, MNLI-m, MNLI-mm, STS-B, and WNLI. MM-MergeBench includes SciQA, Image, VQA, REC, OCR, VizWiz, Flickr, IconQA, AVQA, Image-R, S2W, and TabMWP. We further divide MM-MergeBench into seen and unseen domains: the seen domains include SciQA, Image, VQA, REC, OCR, VizWiz, Flickr, and IconQA, while the unseen domains include AVQA, Image-R, S2W, and TabMWP. Unseen domains contain only test samples and provide no training samples for training LoRA adapters. Consequently, no target-domain LoRA adapter is trained or used for AVQA, Image-R, S2W, or TabMWP. During unseen-domain evaluation, LiST reuses only the LoRA adapters trained on seen domains and constructs a local simplex from retrieved seen-domain adapters. This setting evaluates cross-domain generalization of adapter composition rather than adaptation with an unseen-domain LoRA.

## A.2 Hyperparameter Sensitivity

**Sensitivity to Neighbor Count.** Table 7 analyzes the effect of the number of neighboring adapters $K$. When $K = 0$ , the model reduces to the target-task adapter and avoids negative transfer, but it cannot use complementary information from related tasks. A very small neighborhood, such as $K = 1$, is not sufficiently expressive and may still select an imperfect neighbor, leading to degraded performance. The best result is obtained at $K = 3$, while $K = 5$ gives nearly identical performance with slightly higher latency. However, further increasing $K$ hurts both accuracy and negative transfer, and using all adapters leads to the worst performance among fusion settings. This trend supports the local-simplex assumption: useful transfer comes from a small set of related adapters, whereas large or global neighborhoods introduce unrelated task directions and increase computational cost.

**Sensitivity to Monte Carlo Forward Count.** Table 8 studies the number of stochastic forward passes used for estimating the prompt-level energy and stochastic consistency. With only one forward pass, the energy estimate is noisy and the performance is poor. Increasing $M$, from 2 to 4 substantially improves performance and JS estimate stability, showing that multiple stochastic evaluations are important for reliable candidate scoring. The performance continues to improve at $M = 8$, where the JS estimate becomes highly stable. Increasing $M$ further to 16 brings only marginal improvement but introduces additional latency. We therefore use $M = 8$ as the default setting, which provides a good balance between stability, accuracy, and inference cost.

**Sensitivity to Acceptance Margin.** Table 9 evaluates the energy-improvement margin $m_{accept}$ in the safe acceptance rule. When the margin is zero, the model accepts many searched candidates, but this leads to low performance and high negative transfer, suggesting that small energy reductions are not always reliable. Increasing the margin to 1.0 improves both performance and robustness by filtering weak improvements. The best result is achieved at $m_{accept} = 3.0$ where the accept rate remains moderate and negative transfer is substantially reduced. A larger margin, $m_{accept} = 5.0$, becomes too conservative: it rejects many beneficial candidates and slightly lowers the final performance. These results show that the acceptance margin controls a trade-off between exploiting useful test-time search results and avoiding unreliable low-energy candidates.

# B More Analyses

**Energy–performance alignment on feasible candidates.** In the main analysis, we show that energy reduction is generally aligned with downstream metric improvement, but this alignment can be weakened by candidates with large constraint violations. We further examine this relationship

| $K$ | Avg. | Neg. Transfer ↓ | Latency |
|---|---|---|---|
| 0 | 87.35 | **0.00** | **123.32** |
| 1 | 85.96 | 0.13 | 306.81 |
| 3 | **87.70** | 0.02 | 318.56 |
| 5 | 87.68 | 0.04 | 329.62 |
| 7 | 84.92 | 0.10 | 340.56 |

Table 7: **Sensitivity analysis on neighbor count $K$.**

| $M$ | Avg | JS Estimate Stability ↑ | Latency |
|---|---|---|---|
| 1 | 80.92 | – | **172.88** |
| 2 | 83.19 | 94.28 | 244.23 |
| 4 | 86.83 | 96.07 | 292.94 |
| 8 | 87.70 | 98.89 | 331.53 |
| 16 | **87.72** | **99.02** | 378.64 |

Table 8: **Sensitivity of MC forward count $M$.**

| $m_{\text{accept}}$ | Avg | Accept Rate | Neg. Transfer ↓ |
|---|---|---|---|
| 0.00 | 80.85 | **57.63** | 0.132 |
| 1.00 | 83.69 | 53.37 | 0.073 |
| 3.00 | **87.70** | 42.76 | **0.024** |
| 5.00 | 86.80 | 11.96 | 0.025 |

Table 9: **Sensitivity of acceptance margin $m_{\text{accept}}$.**

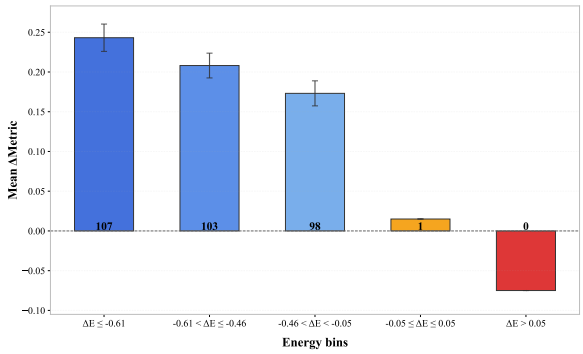



Figure 3: Energy–performance alignment on feasible candidates only. After filtering high-violation candidates, larger energy decreases correspond more clearly to downstream metric gains, supporting the feasibility-aware interpretation of the prompt-level energy.

after applying the feasibility filter used by the safe acceptance rule. Figure 3 groups feasible candidate perturbations by their relative energy change, $\Delta\mathcal{E} = \mathcal{E}(\delta) - \mathcal{E}(0)$. After high-violation candidates are removed, larger energy decreases show a cleaner positive association with metric gains. This supports the role of prior, geometric, and stochastic-consistency constraints in making the prompt-level energy a more reliable test-time selection signal.

**Effect of safe acceptance rules.** We also visualize how different deployment rules affect final performance and fallback behavior. Figure 4 compares prior-only inference, always accepting searched candidates, energy-only acceptance, constraint-only acceptance, and the full acceptance rule. Always accepting searched candidates improves over the prior-only baseline, confirming that test-time search can find useful sample-specific LoRA compositions. However, the full acceptance rule achieves the best average performance while triggering more fallback, indicating that fallback is not merely conservative. Instead, it rejects unreliable searched candidates whose energy improvement is not supported by feasibility constraints.

## C LLM Usage

We used ChatGPT and Claude to assist with drafting and refining text. All AI-assisted content was reviewed, revised, and verified by the authors. These tools contributed to wording and phrasing suggestions only, and played no independent role in research ideation, experimental design, or data analysis. The authors bear full responsibility for all content in this manuscript.

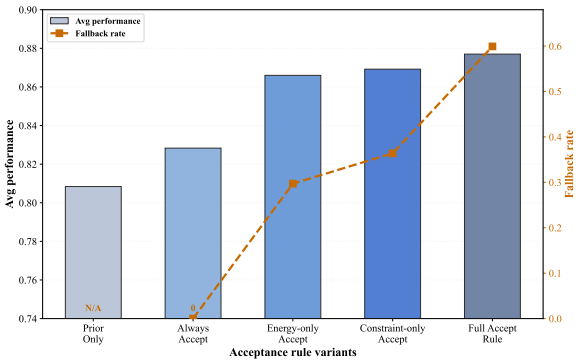



Figure 4: Effect of safe acceptance rules. The full rule achieves the best average performance while triggering more fallback, showing that feasibility-aware rejection of unreliable searched candidates improves robustness.