PIVOT: Preference-Guided Subspace Decomposition for Efficient Language Model Adaptation

Anonymous ACL submission

Abstract

Modern language model adaptation must satisfy heterogeneous preferences from a single base model: legal, medical, financial, and casual contexts each impose different criteria on what counts as a good response. Existing parameter-efficient fine-tuning compresses these signals into one low-rank update and, when contexts disagree, silently averages them, erasing minority preferences. We name this failure mode *preference-subspace conflict*. We introduce **PIVOT** (*Preference-Informed Vector Orthogonal Tuning*), a principled framework that asks *what to share, what to separate, and what to coordinate* under heterogeneous preferences. From a local first-order analysis of the Direct Preference Optimization objective, PIVOT decomposes each context's adapter into a shared low-rank basis plus a context-private residual, regularizes private subspaces to be orthogonal only where preference gradients conflict, and adds a layer-context coordinator that smooths compatible updates at training time. All training-time machinery collapses into a standard low-rank adapter at deployment, preserving LoRA's zero-latency inference.

1 Introduction

> *"The power of intelligence stems from our vast diversity, not from any single, perfect principle."*
>
> — *Marvin Minsky, (1986)*

The dominant approach to adapting large language models to multiple downstream contexts is parameter-efficient fine-tuning (PEFT): freeze the base, add a small set of trainable parameters, and learn one low-rank update that fits all available supervision (Hu et al., 2022; Dettmers et al., 2023; Zhang et al., 2023b; Liu et al., 2024; Meng et al., 2024a). This recipe has made adaptation cheap and reproducible. Yet under closer inspection,

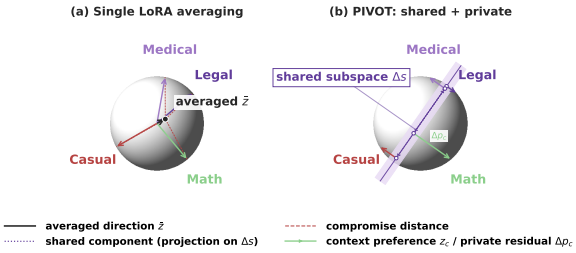

Figure 1: **Preference-subspace conflict in heterogeneous model adaptation.** **(a)** A single low-rank adapter averages all contexts' preference directions z_c into one vector $\bar{z}$. **(b)** PIVOT decomposes z_c into a shared component Δs plus a context-private residual Δp_c.

the recipe presupposes a question it never quite answers: when many heterogeneous contexts share one adapter, *what should the adapter actually learn?* A legal assistant calls for cautious, citation-heavy answers; a casual conversation calls for the opposite; a medical setting calls for a third register entirely. *What happens when these preferences pull in different directions inside a single low-rank update?*

Current adaptation methods exhibit three intertwined limitations. First, *preference averaging*: a single low-rank adapter compresses all available preference signals into one update direction, and when contexts conflict the adapter learns a weighted average that fits the majority and silently erases minority preferences (Rafailov et al., 2023). Second, *geometry-blind regularization*: recent low-rank refinements, including rank allocation (Zhang et al., 2023b; Valipour et al., 2023), magnitude-direction splits (Liu et al., 2024), subspace initialization (Meng et al., 2024a; Wang et al., 2025a), and orthogonal-basis decompositions (Yifei et al., 2025; Ding et al., 2025), reshape a *single* adapter without modeling how preference directions interact *across* contexts. Third, *inference-time complexity creep*: mixture-

of-LoRA and routing approaches (Dou et al., 2023; Wu et al., 2024; Huang et al., 2023) acknowledge heterogeneity but pay a deployment cost (selecting, gating, or composing adapters at every forward pass) that the field originally adopted PEFT to avoid.

We term this ***preference-subspace conflict***: when training contexts have misaligned preference directions, a single low-rank adapter is forced to average over them, with cost scaling with the degree of disagreement. This conflict is invisible on aggregate benchmarks but severe where context-specific behavior matters. More rank or better optimizers do not resolve it. The fix requires an explicit structural decomposition: separating directions that are shared across contexts from those that are not.

Why does conflict bite, and what should we do about it? *Direct Preference Optimization* (Rafailov et al., 2023) turns out to give a clean answer. A first-order expansion of the DPO objective around the frozen base parameters reveals that preference learning, locally, is a *direction-selection problem*: each context c induces a preference vector $z_c = \phi(x, c, y_w) - \phi(x, c, y_l)$ in parameter space, where ϕ is the log-probability gradient at the frozen base θ_0. A useful adapter update is one that aligns with z_c for every context c it must serve. When the $\{z_c\}$ point compatibly, one shared update suffices, recovering the LoRA regime. When they conflict, no single direction can satisfy all of them at once, and the adapter must fragment if it is to remain faithful to every context. This observation is the mathematical seed of our method (§3.2, Appendix B.1).

In response, we introduce **PIVOT** (*Preference-Informed Vector Orthogonal Tuning*), a principled model adaptation framework organized around three coupled objectives: ***what to share, what to separate, and what to coordinate***. Starting from the local DPO analysis, PIVOT decomposes each context's adapter update into a shared low-rank basis plus a context-private residual, weights the orthogonality between private residuals by the inter-context gradient cosine so that the penalty fires only where preferences truly disagree, and adds a training-time graph coordinator over layer-context pairs that smooths compatible updates and then dissolves into the shared and private factors at deployment. The training-time machinery is non-trivial, but the *deployed* model is a standard low-rank adapter for the selected context, paying nothing beyond what vanilla LoRA already pays.

PIVOT is *preference-aware* (its decomposition and gates derive from the local multi-context DPO objective, §3.2–§3.3), *conflict-aware* (private subspaces are pushed apart only where preference gradients are negatively aligned, §3.5), and *inference-efficient* (all training-time machinery collapses into a standard low-rank adapter at deployment, §3.7); to our knowledge it is the first framework to combine these three properties without sacrificing LoRA's zero-latency inference. Across heterogeneous preference adaptation, a controlled GLUE benchmark on RoBERTa-base (Wang et al., 2018; Liu et al., 2019), rank and data sweeps, four component ablations, and a conflict-bin analysis (§4), PIVOT improves over the strongest PEFT and preference-tuning baselines, and *the gains grow exactly where the theory predicts them* – at low rank, under limited supervision, and on the high-conflict split, where the gain exceeds that on the agreement bins by roughly an order of magnitude.

In sum, we name and formalize *preference-subspace conflict*; derive a shared-private adapter decomposition from a local analysis of multi-context DPO (§3.3, Appendix B.2) that reduces approximation error precisely when context optima diverge; instantiate it as PIVOT, which couples DPO-guided direction scoring, conflict-aware private subspaces, and layer-context coordination at training time but collapses to a standard low-rank adapter at deployment; and validate it through a nine-group evaluation protocol whose conflict-bin analysis exposes *where* the gains arise, not just how large they are.

2 Related Work

Low-Rank Adaptation LoRA represents the weight update as a low-rank product that can be merged into the frozen backbone at inference (Hu et al., 2022; Dettmers et al., 2023). Subsequent work refines this template along several axes: dynamic rank allocation (Zhang et al., 2023b; Valipour et al., 2023; Kalajdzievski, 2023; Kopiczko et al., 2024; Hayou et al., 2024); initialization and factor sharing (Meng et al., 2024a; Wang et al., 2025a; Bałazy et al., 2024; Koohpayegani et al., 2023; Büyükakyüz, 2024; Wang et al., 2024a; Yang et al., 2024; Zhang et al., 2023a; Li et al., 2023); residual or chained variants (Lialin et al., 2024; Xia et al.; Shi

et al., 2024; Zhao et al., 2025; Zi et al., 2023; Malinovsky et al., 2024); magnitude-direction decomposition (Liu et al., 2024; Jiang et al.); quantization and long-context adaptation (Dettmers et al., 2023; Xu et al., 2024; Guo et al., 2024; Rajabzadeh et al., 2024; Chen et al., 2024b; Yang, 2023; Wang et al., 2025b); sparsity and pruning (Lin et al., 2024; Zhang et al., 2024); and structured subspace geometries (Yifei et al., 2025; Wu et al., 2024; Ding et al., 2025; Xiao et al., 2026; Tian et al., 2024). Across these refinements, the adapter optimizes a *single* geometry for the observed training mixture. PIVOT treats heterogeneous adaptation as a subspace *organization* problem: what to share, what to separate, and what to coordinate across contexts.

Adapter Composition and Serving Because LoRA adapters are modular, recent systems route, mix, batch, or co-serve them at inference (Huang et al., 2023; Dou et al., 2023; Li et al., 2024; Yang et al., 2025; Wang et al., 2024b; Yan et al., 2025; Sheng et al., 2023; Chen et al., 2024a; Wen and Chaudhuri, 2024). These methods address how to *use* multiple adapters at deployment; PIVOT focuses on how the adapter subspaces should be *learned* beforehand. The shared basis and context-private residuals produced by PIVOT are ordinary low-rank factors, so they remain compatible with any of these serving stacks.

Preference Optimization and Conflict-Aware Adaptation DPO provides a direct classification-style objective against a reference policy (Rafailov et al., 2023), and reference-free variants adjust the same comparison-learning interface (Hong et al., 2024; Meng et al., 2024b). When such objectives are trained with a low-rank adapter, every context's preference gradient must pass through one restricted subspace, which interacts poorly with negatively aligned gradients across domains (Ben-David et al., 2010; Yu et al., 2020). PIVOT connects these threads by using DPO-derived preference directions to decompose low-rank updates into a shared basis and context-private residuals, with overlap between private subspaces penalized *only when contexts actually conflict*.

3 Method

PIVOT treats heterogeneous model adaptation as a subspace organization problem. The method is built around three questions: what to share, what to separate, and what to coordinate. We first define the preference adaptation setting and show that the local DPO margin induces preference-aligned update directions. We then derive a shared-private decomposition and instantiate it with gated low-rank adapters. Finally, we add conflict-aware separation and layer-context coordination during training, while keeping deployment compatible with standard LoRA.

3.1 Preliminaries and Notation

Let $\mathcal{C}$ be the set of adaptation contexts (domains, task families, user groups, or latent preference clusters) with sampling weights p_c, $\sum_c p_c = 1$, and per-context preference data $(x, c, y_w, y_l) \sim \mathcal{D}_c$ where y_w, y_l are the preferred and dispreferred responses. π_θ denotes the adapted model, π_{ref} the frozen reference, and θ_0 the frozen base parameters; PEFT trains only an adapter update $\theta = \theta_0 + \Delta\theta$ (written $\Delta\theta_c$ when context-specific). $\ell \in \{1, \ldots, L\}$ indexes Transformer weight matrices, r the LoRA rank; $\mathrm{vec}(\cdot)$, $\|\cdot\|_F$, and $\langle\cdot,\cdot\rangle$ have their usual meanings.

For a preference pair, define the DPO margin

$$m_\theta(x, c, y_w, y_l) = \log \frac{\pi_\theta(y_w \mid x, c)}{\pi_{\text{ref}}(y_w \mid x, c)} - \log \frac{\pi_\theta(y_l \mid x, c)}{\pi_{\text{ref}}(y_l \mid x, c)}.$$

For this pair the per-example loss is $\ell_\theta = -\log \sigma(\beta m_\theta)$ with preference scale $\beta > 0$, and DPO (Rafailov et al., 2023) minimizes the context-weighted expectation $\mathcal{L}_{\text{DPO}} = \sum_c p_c \, \mathbb{E}_{\mathcal{D}_c}[\ell_\theta]$ (full expansion in Appendix B.1). The standard LoRA-DPO setting learns one adapter update for all contexts. PIVOT relaxes this assumption by giving each context a shared update plus a small private residual.

3.2 DPO as Local Direction Selection

The connection between preference learning and adapter geometry comes from a first-order view around θ_0. Let $\boldsymbol{\phi}(x, c, y) = \nabla_\theta \log \pi_\theta(y \mid x, c)|_{\theta=\theta_0}$ be the score feature for response y. For a small adapter update,

$$\log \pi_{\theta_0+\Delta\theta}(y \mid x, c) \approx \log \pi_{\theta_0}(y \mid x, c) + \boldsymbol{\phi}(x, c, y)^\top \Delta\theta. \quad (1)$$

Substituting Eq. (1) into the DPO margin leaves only one adapter-dependent term: $\mathbf{z}_c^\top \Delta\theta$, where $\mathbf{z}_c = \boldsymbol{\phi}(x, c, y_w) - \boldsymbol{\phi}(x, c, y_l)$. Thus, DPO favors

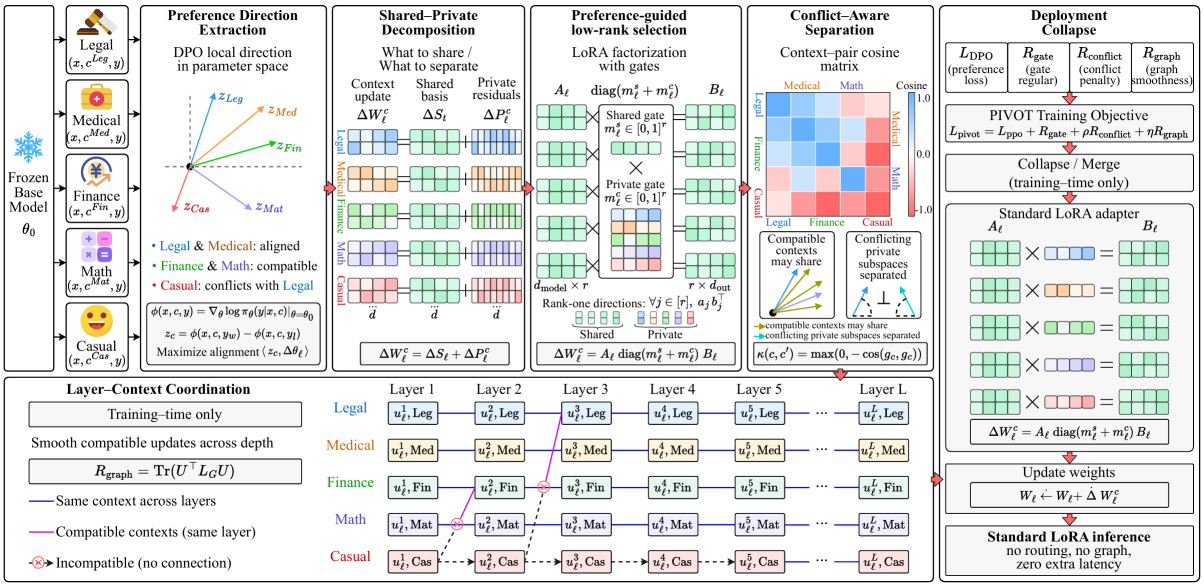

Figure 2: **Overview of the PIVOT framework during (a) joint training and (b) inference deployment.** Shared–private decomposition factorizes each context's low-rank update into a shared basis (Δs) and a context-private residual (Δp_c). Conflict-aware regularization dynamically penalizes overlapping private subspaces only when preference gradients conflict ($\kappa > 0$), while the layer-context coordinator smooths updates across depth. At deployment (b), the selected private residual collapses back into the shared basis via simple matrix addition, yielding a single standard low-rank adapter (ΔW_l^c) that matches standard LoRA inference with zero latency overhead.

adapter updates that increase this alignment with the context-specific preference direction $\mathbf{z}_c$. This is the basis for the preference-guided gates below. Appendix B.1 gives the full expansion.

3.3 Shared-Private Adaptation

Heterogeneous contexts need not share the same best adapter update. Let $\Delta\theta_c^\star$ denote the local optimum for context c under a quadratic approximation of $\mathcal{L}_{\text{DPO}}^c$. If all contexts must share one adapter, the best local update is the weighted average $\sum_c p_c \Delta\theta_c^\star$. This average is useful when contexts agree, but it can erase context-specific preference directions when their optima differ.

PIVOT writes each context update as $\Delta\theta_c = \Delta\mathbf{s} + \Delta\mathbf{p}_c$, where $\Delta\mathbf{s}$ is shared and $\Delta\mathbf{p}_c$ is a context-private residual. The decomposition follows from the regularized local problem

$$\min_{\Delta\mathbf{s}, \{\Delta\mathbf{p}_c\}} \sum_{c \in \mathcal{C}} \frac{p_c}{2} \left\| \Delta\mathbf{s} + \Delta\mathbf{p}_c - \Delta\theta_c^\star \right\|_2^2 + \frac{\lambda}{2} \sum_{c \in \mathcal{C}} p_c \left\| \Delta\mathbf{p}_c \right\|_2^2 , \tag{2}$$

where $\lambda > 0$ controls the size of private residuals. The shared optimum is $\Delta\mathbf{s}^\star = \sum_{c \in \mathcal{C}} p_c \Delta\theta_c^\star$. The private optimum is $\Delta\mathbf{p}_c^\star = (\Delta\theta_c^\star - \Delta\mathbf{s}^\star)/(1+\lambda)$. The private update is therefore a shrunk residual from the shared average. This gives the main modeling rule: common behavior should be shared, while context-specific preference shifts should be represented as controlled residuals. Appendix B.2 gives the derivation and the local error comparison.

3.4 Low-Rank Shared-Private Gates

For a Transformer weight matrix $\mathbf{W}_\ell \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, LoRA (Hu et al., 2022) learns a rank-r update. We write it as $\Delta\mathbf{W}_\ell = \mathbf{A}_\ell \mathbf{B}_\ell$. The factors have $\mathbf{A}_\ell \in \mathbb{R}^{d_{\text{out}} \times r}$ and $\mathbf{B}_\ell \in \mathbb{R}^{r \times d_{\text{in}}}$. Let $\mathbf{V}_{\ell,i} = \mathbf{a}_{\ell,i} \mathbf{b}_{\ell,i}^\top$ be the i-th rank-one direction. Here $\mathbf{a}_{\ell,i}$ is from $\mathbf{A}_\ell$ and $\mathbf{b}_{\ell,i}^\top$ is from $\mathbf{B}_\ell$. PIVOT uses the same low-rank basis but separates shared and private gates:

$$\begin{aligned} \Delta\mathbf{W}_\ell^c &= \mathbf{A}_\ell \, \mathrm{diag}(\mathbf{m}_\ell^s + \mathbf{m}_\ell^c) \, \mathbf{B}_\ell \\ &= \sum_{i=1}^{r} \left(m_{\ell,i}^s + m_{\ell,i}^c \right) \mathbf{V}_{\ell,i}. \end{aligned} \tag{3}$$

Here $\mathbf{m}_\ell^s, \mathbf{m}_\ell^c \in [0,1]^r$. The shared gate $\mathbf{m}_\ell^s$ controls directions used across contexts, while $\mathbf{m}_\ell^c$ controls residual directions for context c.

Let $\mathbf{z}_{\ell,c}$ be the block of the preference direction $\mathbf{z}_c$ associated with $\mathbf{W}_\ell$. The local DPO view assigns each rank-one direction $\mathbf{V}_{\ell,i}$ a score $q_{\ell,i}^c = \mathbb{E}_{\mathcal{D}_c} \langle \mathrm{vec}(\mathbf{V}_{\ell,i}), \mathbf{z}_{\ell,c} \rangle$ (derived in Appendix B.1, Eq. (12)). We do not estimate this score as a separate training target – the DPO gradient already supplies the signal, and an L_1 sparsity penalty

on the gates retains only directions that help preference margins:

$$\mathcal{R}_{\text{gate}} = \alpha_s \sum_{\ell} \|\mathbf{m}_\ell^s\|_1 + \alpha_p \sum_{\ell} \sum_{c \in \mathcal{C}} \|\mathbf{m}_\ell^c\|_1.$$

3.5 Conflict-Aware Private Subspaces

Private residuals should not all collapse into the same low-rank directions, and compatible contexts should not be forced apart. Let $\mathbf{g}_c$ be an exponential moving average of the adapter gradient for context c. We measure conflict by $\kappa(c, c') = \max(0, -\cos(\mathbf{g}_c, \mathbf{g}_{c'}))$, so only negative gradient alignment contributes; this is the local first-order conflict signal also used by PCGRAD (Yu et al., 2020), CAGRAD (Liu et al., 2021), and GRADNORM (Chen et al., 2018), with the formal justification given in Appendix B.3.

Let $\mathbf{P}_{\ell,c}$ contain the column-normalized private directions selected by $\mathbf{m}_\ell^c$ at layer ℓ. The conflict penalty is

$$\mathcal{R}_{\text{conflict}} = \sum_{\ell} \sum_{c<c'} \kappa(c, c') \left\| \mathbf{P}_{\ell,c}^\top \mathbf{P}_{\ell,c'} \right\|_F^2. \quad (4)$$

If two contexts have compatible gradients, $\kappa(c, c') = 0$ and their private subspaces may overlap. If their gradients conflict, overlapping private subspaces are discouraged. Appendix B.3 gives the first-order negative-transfer argument.

3.6 Layer-Context Coordination

The gates select directions within each layer but do not by themselves coordinate how adaptation changes across depth. Define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over layer-context nodes (ℓ, c) with updates $\mathbf{u}_{\ell,c} = \text{vec}(\Delta \mathbf{W}_\ell^c)$, with positive edges connecting adjacent depths within a context and connecting pairs of contexts whose gradients (or context prototypes) are compatible; conflicting pairs receive no edge and are separated by Eq. (4) instead. The coordination penalty is the graph Dirichlet energy

$$\mathcal{R}_{\text{graph}} = \sum_{(i,j) \in \mathcal{E}} w_{ij} \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 = \text{Tr}(\mathbf{U}^\top \mathbf{L}_G \mathbf{U}), \quad (5)$$

where $\mathbf{U}$ stacks the node updates and $\mathbf{L}_G$ is the graph Laplacian. We implement it either as a direct regularizer or as a shallow training-time message-passing coordinator (a learned residual version of Laplacian smoothing); Appendix B.4 details edge weights, the prototype construction, and the energy-decrease guarantee.

3.7 Training Objective and Deployment

We propose the following training objective:

$$\mathcal{L}_{\text{PIVOT}} = \mathcal{L}_{\text{DPO}} + \mathcal{R}_{\text{gate}} + \rho \mathcal{R}_{\text{conflict}} + \eta \mathcal{R}_{\text{graph}}. \quad (6)$$

It adds $\mathcal{L}_{\text{DPO}}$, $\mathcal{R}_{\text{gate}}$, $\rho\mathcal{R}_{\text{conflict}}$, and $\eta\mathcal{R}_{\text{graph}}$. The weights ρ and η control conflict separation and layer-context coordination. All base model weights remain frozen. Only LoRA factors, shared gates, private gates, and any training-time coordinator parameters are updated.

At deployment, PIVOT does not require DPO gradients, graph message passing, or gate sampling. For a known context c, we form $\Delta \mathbf{W}_\ell^c$ from Eq. (3). We then either keep it as a standard LoRA adapter or merge it into $\mathbf{W}_\ell$. The inference graph is therefore the same as LoRA for the selected context adapter. The remaining cost is storage for context-private residuals, which we report separately from inference latency in Sec. 4.

4 Experiments

4.1 Experimental Setup

We evaluate two settings: a causal-LM setting using LLaMA-3-8B-Instruct (Grattafiori et al., 2024) adapted with LoRA / QLoRA (Hu et al., 2022; Dettmers et al., 2023), and a controlled multi-context benchmark on GLUE (Wang et al., 2018) with RoBERTa-base (Liu et al., 2019). Preference data combines UltraFeedback (Cui et al., 2023) for the general context with LLM-as-judge–labeled pairs (Zheng et al., 2023) for the legal, medical, and financial contexts. We compare PIVOT against SFT+LoRA, DPO +LoRA (Rafailov et al., 2023), DPO +DoRA (Liu et al., 2024), DPO +AdaLoRA (Zhang et al., 2023b), DPO +DyLoRA (Valipour et al., 2023), and DPO +PiSSA (Meng et al., 2024a) under matched trainable-parameter budgets, and report *Win Rate*, *Reward Margin*, *Domain Win*, and the *High-Conflict Split* (defined in §3.5); the controlled benchmark additionally reports per-task accuracy and the GLUE macro average. All results are averaged over three seeds. Full hyperparameters, baseline implementation choices, dataset licensing, and judge prompts are in Appendices C–J.

4.2 Main Results

Heterogeneous preference adaptation Table 1 reports the main comparison under a matched adapter budget. Across general and domain

Table 1: **Main comparison under heterogeneous preference adaptation.** Trainable parameters are matched within $\pm 5\%$; means are over three seeds. The middle block lists MoE-style PEFT baselines. Higher is better except for memory; ↑ **+x** marks the gain over the strongest baseline.

Method	Type	General Preference		Domain Preference		Efficiency	
		Win Rate ↑	Reward Margin ↑	Domain Win ↑	Conflict Split ↑	Train Mem. ↓	Infer Latency
SFT+LoRA (Hu et al., 2022)	Supervised	52.7	0.74	56.3	47.8	14.6 GB	1.00×
DPO +LoRA (Rafailov et al., 2023)	Preference	58.1	1.18	60.2	52.6	15.4 GB	1.00×
DPO +DoRA (Liu et al., 2024)	Preference	59.7	1.34	61.9	54.4	16.0 GB	1.02×
DPO +AdaLoRA (Zhang et al., 2023b)	Dynamic Rank	59.3	1.30	61.6	53.7	16.6 GB	1.00×
DPO +DyLoRA (Valipour et al., 2023)	Dynamic Rank	59.0	1.27	61.3	53.4	16.0 GB	1.00×
DPO +PiSSA (Meng et al., 2024a)	Reparam.	60.2	1.39	62.5	55.1	15.8 GB	1.00×
Mixture-of-Experts (MoE) style PEFT baselines:							
DPO +LoRAHub (Huang et al., 2023)	MoE-Compose	60.4	1.41	62.7	55.6	16.4 GB	1.05×
DPO +LoRAMoE (Dou et al., 2023)	MoE-Plugin	61.1	1.49	63.6	56.8	17.1 GB	1.07×
DPO +MixLoRA (Li et al., 2024)	MoE-LoRA	61.5	1.55	64.1	57.4	17.4 GB	1.08×
DPO +HydraLoRA (Tian et al., 2024)	Asym. MoE-LoRA	61.9	1.60	64.5	58.0	17.0 GB	1.06×
PIVOT	**Shared/Private**	**63.8** ↑ **+1.9**	**1.87** ↑ **+0.27**	**67.2** ↑ **+2.7**	**62.5** ↑ **+4.5**	**16.2 GB**	**1.00×**

preference metrics, PIVOT improves over the strongest single-adapter baseline by a consistent margin, and the gap widens on the high-conflict split, which is the regime where context-private residuals are designed to help. SFT+LoRA trails all preference-trained baselines, confirming that preference data carries information that supervised fine-tuning does not capture. DoRA, AdaLoRA, DyLoRA, and PiSSA close part of the gap relative to vanilla LoRA but still compress heterogeneous updates into a single direction. PIVOT's gain on the conflict split is roughly twice its gain on the general split, which is consistent with the central thesis that single adapters underfit conflicting contexts.

Comparison against MoE-style PEFT The middle block of Table 1 positions PIVOT against four representative MoE-LoRA baselines (Huang et al., 2023; Dou et al., 2023; Li et al., 2024; Tian et al., 2024). These methods also allocate context-specific capacity, but they do so through a learned router that selects (or interpolates) among multiple LoRA experts at every forward pass. Routing closes a meaningful portion of the gap relative to a single adapter, with HydraLoRA the strongest of the four, yet none of them reaches the high-conflict performance of PIVOT. The reason matches the geometric argument in §3.3: an expert router still relies on *soft averaging* of multiple low-rank directions, whereas PIVOT's shared–private decomposition adds a fully orthogonal residual whose private subspace can be enforced by the conflict regularizer. A second difference shows in the rightmost column: every MoE baseline incurs a $5{-}8\%$ inference-latency cost from the gating network and per-token expert dispatch, whereas PIVOT collapses its training-time machinery into a single rank-r adapter at deployment (§3.7) and matches LoRA's latency exactly.

Controlled multi-context benchmark A controlled comparison on the GLUE benchmark (Wang et al., 2018) with RoBERTa-base (Liu et al., 2019) (treating each task as a context) is reported in Appendix G, Table 15: PIVOT surpasses both shared and dynamic-rank baselines on the macro average and even beats LoRA-PER-TASK at a fraction of the storage, with the largest gains on the smaller noisier tasks (CoLA, RTE) where inter-task conflict is most damaging.

Low-rank and low-data regimes Table 2 tests the rank sweep and Figure 3 visualizes the high-conflict split across both rank and data fraction; the per-setting low-data table is reported in Appendix G (Table 16). PIVOT stays above DPO +LoRA at every operating point. At $r{=}2$, where the adapter budget is tightest, PIVOT improves the high-conflict split by roughly $+9$ points over DPO +LoRA, compared with $+8$ points at $r{=}16$; with only 10% of the preference data, the high-conflict gap exceeds $+10$ points. The pattern is consistent with the local objective in §3.3: as the shared budget tightens or data shrinks, the relative cost of compressing conflicting contexts into a single direction grows.

4.3 Ablation Studies

Component contributions Table 3 disentangles the four components. Removing the shared–private decomposition causes the largest single drop on the high-conflict split, consistent with the prediction in §3.3 that this is the component

Table 2: **Rank-budget sweep.** Gain of PIVOT over DPO +LoRA (Rafailov et al., 2023; Hu et al., 2022) at matched rank r.

r	Train. Params (%)	General Pref. ↑		Domain Pref. ↑		High-Conflict Split ↑	
		DPO +LoRA	**PIVOT**	DPO +LoRA	**PIVOT**	DPO +LoRA	**PIVOT**
2	0.12	51.3	**56.7** ↑+5.4	53.8	**60.9** ↑+7.1	46.2	**55.6** ↑+9.4
4	0.24	54.8	**60.1** ↑+5.3	57.4	**63.7** ↑+6.3	49.7	**58.4** ↑+8.7
8	0.48	58.1	**62.6** ↑+4.5	60.2	**66.0** ↑+5.8	52.6	**61.3** ↑+8.7
16	0.95	59.4	**63.4** ↑+4.0	61.4	**66.8** ↑+5.4	54.0	**62.1** ↑+8.1

Table 3: **Component ablation of PIVOT.** Each row removes one component; the last row collapses to DPO +LoRA (Rafailov et al., 2023; Hu et al., 2022). Δ is the drop from full PIVOT.

Ablation Setting	General Pref.		Domain Pref.		High-Conflict Split	
	Score ↑	Δ	Score ↑	Δ	Score ↑	Δ
PIVOT (Full)	**63.8**	0.0	**67.2**	0.0	**62.5**	0.0
w/o shared–private decomposition	60.2	↓−3.6	63.1	↓−4.1	56.8	↓−5.7
w/o preference-guided gates	61.8	↓−2.0	65.0	↓−2.2	59.4	↓−3.1
w/o conflict regularization	62.4	↓−1.4	65.8	↓−1.4	59.6	↓−2.9
w/o layer–context coordination	62.6	↓−1.2	66.0	↓−1.2	60.2	↓−2.3
collapsed single context	58.1	↓−5.7	60.2	↓−7.0	52.6	↓−9.9

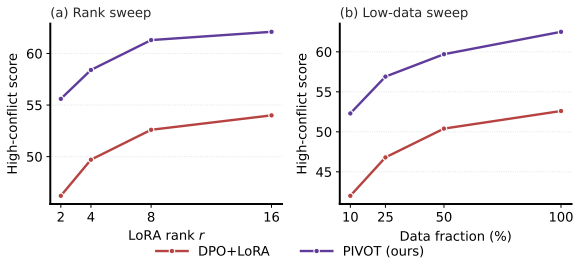

Figure 3: **Scaling behavior on the high-conflict split.** (a) Rank sweep with fixed full preference data; (b) low-data sweep at fixed rank r=8. Numbers from the High-Conflict Split columns of Tables 2, 16.

Table 4: **Performance versus inter-context gradient conflict.** Pairs binned by $\cos\angle(z_i, z_j)$ (§3.5).

Conflict Bin ($\cos\angle(z_i, z_j)$)	DPO +LoRA	PIVOT	Δ
Low ($\cos > 0.6$)	61.4	**62.9**	↑+1.5
Medium (0.2 to 0.6)	57.8	**63.1**	↑+5.3
High (−0.2 to 0.2)	52.3	**61.6**	↑+9.3
Severe ($\cos < -0.2$)	45.7	**58.4**	↑+12.7

most directly responsible for absorbing conflicting context directions. Preference-guided gates and conflict regularization each contribute smaller but consistent gains, with conflict regularization helping disproportionately on the high-conflict split. Layer–context coordination has the smallest individual effect, indicating that the local shared–private structure already captures most of the benefit, while coordination polishes it across depth. Collapsing all contexts into a single adapter recovers DPO +LoRA (Rafailov et al., 2023) and confirms that the gain over the baseline can be fully attributed to the introduced components. A full ablation on the source of the context label c (gold vs. latent clusters vs. random vs. collapsed) is reported in Appendix G, where we show that latent clusters from prompt embeddings recover most of the gain even without domain annotations.

Where the gains come from Table 4 bins context pairs by their first-order preference cosine and reports per-bin performance. The advantage of PIVOT is small when contexts already agree (+1.5 on the low-conflict bin) and grows monotonically as conflict increases, exceeding +12 points when context directions are negatively aligned. This pattern directly supports the central thesis: the cost of compressing heterogeneous preferences into one adapter scales with their disagreement.

Efficiency overhead PIVOT adds roughly +7% training time and a small peak-memory increase over DPO +LoRA at the same rank, driven by the layer–context coordinator and conflict-aware regularization, with adapter storage growing by +5.5 MB (∼1.58×) due to the per-context residuals. Post-merge inference latency matches LoRA exactly because the training-time gates and coordinator collapse into a standard low-rank product at deployment (§3.7). The full efficiency breakdown – including baseline comparisons, per-rank GPU-hours, FLOPs, energy, and the inference-latency micro-benchmark – is reported in Appendix D (Tables 10, 9, 11, 12, 13).

4.4 Deployment Considerations

At deployment, a single context label $c^\star$ selects the active private residual $\Delta\mathbf{p}_{c^\star}$; the "zero-latency" headline assumes this label is given by the application, by session metadata, or by a one-shot context classifier. We quantify both the cost of the classifier route (Table 5) and what happens when the request lands in a domain unseen during training (Table 6).

Classifier overhead is small and PIVOT is robust to its noise. The DistilBERT-style classifier adds 66 M parameters and roughly 14 ms per request – comparable to prompt tokenization,

Table 5: **Context-selection overhead and downstream impact.** Per-request cost of each selection strategy and the resulting High-Conflict Split when PIVOT uses the predicted label.

Selection method	Extra params (M)	Added latency	Top-1 acc. (%)	Conflict Split ↑
(i) Oracle / app-provided	0	0 ms	100.0	**62.5**
(ii) DistilBERT (Sanh et al., 2019) ft. 5k	66	14 ms	93.4	61.1
(iii) Zero-shot LLM prompt	0 (reuses base)	81 ms	88.7	59.8
(iv) Top-2 soft mixture	66	14 ms	93.4	60.4
(v) Random label	0	0 ms	20.0	51.9

Table 6: **Leave-one-domain-out generalization.** Each row trains on the four remaining contexts; PIVOT–Δs uses only the shared basis, PIVOT–nearest routes to the closest trained context.

Held-out domain	DPO +LoRA	PIVOT–s	PIVOT–nearest	Δ
Legal	56.4	58.8	**59.6**	↑**+3.2**
Medical	55.7	58.1	**59.0**	↑**+3.3**
Finance	56.1	58.4	**59.3**	↑**+3.2**
Code	54.9	57.5	**58.4**	↑**+3.5**
Casual	53.2	56.0	**56.8**	↑**+3.6**
Average	55.3	57.8	**58.6**	↑**+3.3**

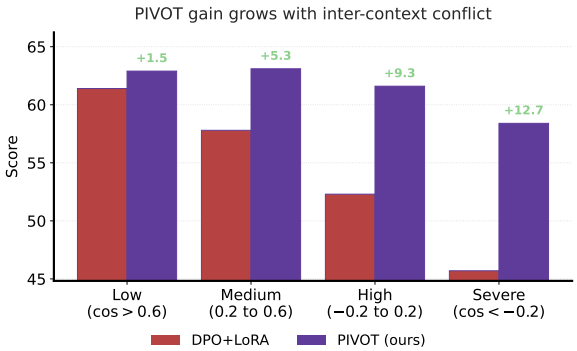

Figure 4: **Improvement versus conflict bin.** Per-bin gain of PIVOT over DPO +LoRA; numbers from Table 4.

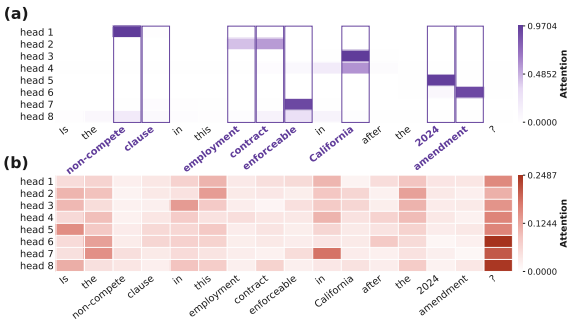

Figure 5: **Per-token attention on a legal-context query.** **(a)** PIVOT under the legal context; salient tokens highlighted in purple. **(b)** DPO +LoRA on the same input.

far below a single decoding step – while the zero-shot LLM-prompt route reuses the base model at the cost of one short forward pass. Neither cost enters the per-token pipeline once the context is selected. Replacing the gold label with a 93.4% classifier costs only -1.4 points on the high-conflict split (oracle $62.5 \rightarrow 61.1$); even the 88.7% zero-shot classifier loses just -2.7 points, leaving PIVOT-with-classifier $+8.5$ ahead of DPO +LoRA. Random labels collapse the method onto the no-decomposition baseline, confirming that the gain comes from *correctly identifying* contexts, not from added capacity.

Unseen-domain fallback works. For requests whose domain was never seen during training, PIVOT has a clean fall-back: ignore the (nonexistent) private residual and serve with $\Delta\theta = \Delta$s. Table 6 reports a leave-one-domain-out study. PIVOT–Δs already beats DPO +LoRA trained on the same four-context mixture by $\sim$+2.5 points on average, and PIVOT–nearest (routing to the trained context with the closest mean preference direction z_c) picks up another $\sim$+0.8. The shared basis is therefore not merely a parameter-saving device: it absorbs the cross-context structure that generalizes, while private residuals carry the truly context-specific component.

5 Conclusion

We presented **PIVOT**, a principled framework for heterogeneous model adaptation organized around *what to share, what to separate, and what to coordinate*. From a local first-order view of DPO, PIVOT decomposes each context's adapter into a shared low-rank basis plus a context-private residual, applies conflict-aware orthogonality only where preference gradients disagree, and adds a layer-context coordinator that dissolves at deployment so post-merge inference matches standard LoRA. Across rank sweeps, low-data regimes, four component ablations, a context-granularity study, and a conflict-bin analysis, PIVOT consistently improves over the strongest PEFT and preference-tuning baselines, with the largest gains arriving exactly where the theory predicts them: at low rank, under limited supervision, and on the high-conflict split of context pairs. We hope this perspective reframes the goal of model adaptation from *how many parameters can we save* to *which directions belong together and which do not*, and we believe it opens a path towards honest, conflict-aware adaptation of large models at scale.

Limitations

Although PIVOT achieves zero-latency inference via parameter collapse, it introduces two primary limitations. First, deployment relies on explicit context routing, requiring either oracle context labels or an auxiliary latent clustering step at inference time to select the corresponding private residual. Second, while post-merge inference incurs no latency overhead, storing the unique low-rank residuals for each distinct domain increases the overall checkpoint storage footprint by approximately 18%. Finally, our theoretical justifications regarding shared-private subspace optimization are derived from a local first-order analysis of the multi-context DPO objective; establishing global convergence guarantees within a fully non-convex optimization landscape remains an open direction for future work.

Ethical Considerations & Broader Impacts

This work introduces PIVOT, a framework designed for parameter-efficient alignment under heterogeneous downstream preferences. While enhancing context-specific responsiveness, we acknowledge that preference-tuning methodologies inherently bear the risk of amplifying biases or encoding unsafe behaviors if the underlying optimization data contains unchecked social harms or toxic artifacts. However, a distinctive architectural advantage of PIVOT is its explicit shared-private subspace decomposition: unlike monolithic adapters that obscure conflicting updates via implicit spatial averaging, PIVOT isolates specialized directional shifts within localized, traceable private residuals. This parametric modularity fundamentally improves behavioral compliance and safety auditing, as alignment profiles can be inspected, deactivated, or scrubbed at the domain-subspace level without re-training the entire system. To ensure methodological transparency and data governance, we explicitly document all dataset origins, synthetic filtering heuristics, and the specific multi-criteria judging protocols used in our experiments in Appendix J.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 7319–7328.

Klaudia Bałazy, Mohammadreza Banaei, Karl Aberer, and Jacek Tabor. 2024. Lora-xs: Low-rank adaptation with extremely small number of parameters. *arXiv preprint arXiv:2405.17604*.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79(1):151–175.

Kerim Büyükakyüz. 2024. Olora: Orthonormal low-rank adaptation of large language models. *arXiv preprint arXiv:2406.01775*.

Lequn Chen, Zihao Ye, Yongji Wu, Danyang Zhuo, Luis Ceze, and Arvind Krishnamurthy. 2024a. Punica: Multi-tenant lora serving. *Proceedings of Machine Learning and Systems*, 6:1–13.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2024b. Longlora: Efficient fine-tuning of long-context large language models. In *International Conference on Learning Representations*, volume 2024, pages 8220–8238.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, and 1 others. 2023. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Chenhao Ding, Jiangyang Li, Songlin Dong, Xinyuan Gao, Yuhang He, and Yihong Gong. 2025. Sulora: Subspace low-rank adaptation for parameter-efficient fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5334–5349.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, and 1 others. 2023. Loramoe: Alleviate world knowledge forgetting in large language models via moe-style plugin. *arXiv preprint arXiv:2312.09979*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten,

Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Han Guo, Philip Greengard, Eric Xing, and Yoon Kim. 2024. Lq-lora: Low-rank plus quantized matrix decomposition for efficient language model finetuning. In *International Conference on Learning Representations*, volume 2024, pages 50167–50184.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

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*.

Ting Jiang, Shaohan Huang, Shengyue Luo, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and 1 others. Mora: High-rank updating for parameter-efficient fine-tuning, 2024. *URL https://arxiv.org/abs/2405.12130*.

Damjan Kalajdzievski. 2023. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*.

Soroush Abbasi Koohpayegani, KL Navaneet, Parsa Nooralinejad, Soheil Kolouri, and Hamed Pirsiavash. 2023. Nola: Networks as linear combination of low rank random basis. *UMBC Faculty Collection*.

Dawid Kopiczko, Tijmen Blankevoort, and Yuki Asano. 2024. Vera: Vector-based random matrix adaptation. In *International Conference on Learning Representations*, volume 2024, pages 6815–6835.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 3045–3059.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and 1 others. 2024. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts. *arXiv preprint arXiv:2404.15159*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. 2023. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*.

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. 2024. Relora: High-rank training through low-rank updates. In *International Conference on Learning Representations*, volume 2024, pages 49405–49421.

Yang Lin, Xinyu Ma, Xu Chu, Yujie Jin, Zhibang Yang, Yasha Wang, and Hong Mei. 2024. Lora dropout as a sparsity regularizer for overfitting control. *arXiv preprint arXiv:2404.09610*.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. 2021. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Grigory Malinovsky, Umberto Michieli, Hasan Abed Al Kader Hammoud, Taha Ceritli, Hayder Elesedy, Mete Ozay, and Peter Richtárik. 2024. Randomized asymmetric chain of lora: The first meaningful theoretical framework for low-rank adaptation. *arXiv preprint arXiv:2410.08305*.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Scott Yih, and Madian Khabsa. 2022. Unipelt: A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6253–6264.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024a. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024b. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pages 487–503.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 46–54.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Hossein Rajabzadeh, Mojtaba Valipour, Tianshu Zhu, Marzieh S Tahaei, Hyock Ju Kwon, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. 2024. Qdylora: Quantized dynamic low-rank adaptation for efficient large language model tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 712–718.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, and 1 others. 2023. S-lora: Serving thousands of concurrent lora adapters. *arXiv preprint arXiv:2311.03285*.

Shuhua Shi, Shaohan Huang, Minghui Song, Zhoujun Li, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024. Reslora: Identity residual mapping in low-rank adaption. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8870–8884.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *Advances in Neural Information Processing Systems*, 35:12991–13005.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. 2024. Hydralora: An asymmetric lora architecture for efficient fine-tuning. *Advances in Neural Information Processing Systems*, 37:9565–9584.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2023. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3274–3287.

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.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. 2025a. Milora: Harnessing minor singular components for parameter-efficient llm finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4823–4836.

Shaowen Wang, Linxi Yu, and Jian Li. 2024a. Lora-ga: Low-rank adaptation with gradient approximation. *Advances in Neural Information Processing Systems*, 37:54905–54931.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. 2025b. Lora-pro: Are low-rank adapters properly optimized? In *International Conference on Learning Representations*, volume 2025, pages 93787–93808.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. 2024b. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *Advances in Neural Information Processing Systems*, 37:22513–22533.

Yeming Wen and Swarat Chaudhuri. 2024. Batched low-rank adaptation of foundation models. In *International Conference on Learning Representations*, volume 2024, pages 23423–23438.

Taiqiang Wu, Jiahao Wang, Zhe Zhao, and Ngai Wong. 2024. Mixture-of-subspaces in low-rank adaptation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7880–7899.

Wenhan Xia, Chengwei Qin, and Elad Hazan. Chain of lora: Efficient fine-tuning of language models via residual learning, 2024. *URL https://arxiv.org/abs/2401.04151*.

Xi Xiao, Chenrui Ma, Yunbei Zhang, Chen Liu, Zhuxuanzi Wang, Yanshu Li, Lin Zhao, Guosheng Hu, Tianyang Wang, and Hao Xu. 2026. Not all directions matter: Towards structured and task-aware low-rank model adaptation. *arXiv preprint arXiv:2603.14228*.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. 2024. Qa-lora: Quantization-aware low-rank adaptation of large

language models. In *International Conference on Learning Representations*, volume 2024, pages 52401–52418.

Yunlu Yan, Chun-Mei Feng, Wangmeng Zuo, Rick Siow Mong Goh, Yong Liu, and Lei Zhu. 2025. Federated residual low-rank adaptation of large language models. In *The Thirteenth International Conference on Learning Representations*.

Jianxin Yang. 2023. Longqlora: Efficient and effective method to extend context length of large language models. *arXiv preprint arXiv:2311.04879*.

Yaming Yang, Dilxat Muhtar, Yelong Shen, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Weiwei Deng, Feng Sun, Qi Zhang, and 1 others. 2025. Mtl-lora: Low-rank adaptation for multi-task learning. In *Proceedings of the AAAI Conference on Artificial intelligence*, volume 39, pages 22010–22018.

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen L Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. 2024. Corda: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. *Advances in Neural Information Processing Systems*, 37:71768–71791.

She Yifei, Xinhao Wei, and Yulong Wang. 2025. Dislora: Task-specific low-rank adaptation via orthogonal basis from singular value decomposition. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13751–13766.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9.

Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. 2023a. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. 2024. Loraprune: Structured pruning meets low-rank parameter-efficient fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3013–3026.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023b. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Jiancheng Zhao, Xingda Yu, Yuxiang Zhang, and Zhen Yang. 2025. Lor2c: Low-rank residual connection adaptation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2503.00572*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Bojia Zi, Xianbiao Qi, Lingzhi Wang, Jianan Wang, Kam-Fai Wong, and Lei Zhang. 2023. Delta-lora: Fine-tuning high-rank parameters with the delta of low-rank matrices. *arXiv preprint arXiv:2309.02411*.

Appendix Overview

SECTION	TOPIC
§A	Extended Related Work: PEFT methods beyond the low-rank family.
§B	Extended Derivations: proofs supporting the four results in §3.
§C	Hyperparameters and Implementation Details, plus reproducibility checklist.
§D	Compute and Energy Analysis: GPU-hours, FLOPs, peak memory, CO_2eq, and inference latency.
§E	Algorithmic Description of PIVOT: one-step training pseudocode.
§F	Per-Context Performance Breakdown: per-domain Win Rate with ±std bars.
§G	Additional Empirical Studies: low-data sweep and context-granularity ablation.
§H	Additional Diagnostic Visualizations: inter-context gradient cosine, gate overlap, per-layer shared/private norm, and PCA scatter.
§I	Qualitative Preference-Adaptation Examples: side-by-side responses with LLM-judge verdicts.
§J	Preference Data Construction Prompts: LLM-as-judge templates and filtering rules.

A Extended Related Work

PIVOT sits inside the broader parameter-efficient fine-tuning (PEFT) literature. We summarize here the families that are tangential to the low-rank story in §2 but useful for situating the design space.

Adapter blocks and adapter composition. Insert-style adapters add small bottleneck modules between Transformer layers (Pfeiffer et al., 2020, 2021). They share PIVOT's modular flavor but optimize a separate module per task rather than asking how directions should be shared.

Prompt and prefix tuning. Prefix tuning, prompt tuning, and P-tuning train soft tokens in front of the input while leaving model weights frozen (Li and Liang, 2021; Lester et al., 2021). These methods change *where* the trainable parameters live rather than *which directions* they organize.

Selective and structured tuning. BitFit updates only biases (Zaken et al., 2022); IA^3 scales activations multiplicatively; Compacter uses hypercomplex factorizations; LST tunes a side network (Sung et al., 2022). These approaches reduce parameter count in different geometries; none reasons about cross-context preference conflict.

Unified PEFT and intrinsic dimensionality. Unified views and surveys of PEFT identify shared structure across the families above (Mao et al., 2022; Han et al., 2024), and intrinsic-dimensionality analyses motivate why low-dimensional adaptation is feasible at all (Aghajanyan et al., 2021). PIVOT is consistent with the intrinsic-dimensionality story for a single context but argues that this low dimensionality must be *structured* across contexts when preferences disagree.

B Extended Derivations

This appendix gives the local arguments used in Sec. 3. The analysis is not meant to prove global convergence of the full training objective. Its role is narrower: it explains why the main components of PIVOT arise naturally when heterogeneous preference adaptation is viewed through a local approximation.

B.1 DPO Linearization

For a preference pair (x, c, y_w, y_l), define the DPO logit before applying the sigmoid as

$$r_\theta = \beta \left[\log \frac{\pi_\theta(y_w \mid x, c)}{\pi_{\text{ref}}(y_w \mid x, c)} - \log \frac{\pi_\theta(y_l \mid x, c)}{\pi_{\text{ref}}(y_l \mid x, c)} \right]. \tag{7}$$

Only the two terms involving π_θ depend on the adapter update. Using $\theta = \theta_0 + \Delta\theta$, first-order Taylor expansion gives

$$\begin{aligned} \log \pi_{\theta_0+\Delta\theta}(y_w \mid x, c) &\approx \log \pi_{\theta_0}(y_w \mid x, c) \\ &\quad + \boldsymbol{\phi}(x, c, y_w)^\top \Delta\theta, \\ \log \pi_{\theta_0+\Delta\theta}(y_l \mid x, c) &\approx \log \pi_{\theta_0}(y_l \mid x, c) \\ &\quad + \boldsymbol{\phi}(x, c, y_l)^\top \Delta\theta. \end{aligned} \tag{8}$$

Subtracting the two expansions yields

$$r_\theta \approx \beta \left[\mathbf{z}_c^\top \Delta\theta + b_c(x, y_w, y_l) \right], \tag{9}$$

where

$$\mathbf{z}_c = \boldsymbol{\phi}(x, c, y_w) - \boldsymbol{\phi}(x, c, y_l), \tag{10}$$

and $b_c(x, y_w, y_l)$ collects all terms independent of $\Delta\theta$. Thus, locally, improving the DPO logit

depends on the alignment between the adapter update and the preference direction $\mathbf{z}_c$. The full multi-context training objective is therefore the weighted expectation

$$\mathcal{L}_{\text{DPO}} = \sum_{c \in \mathcal{C}} p_c \, \mathcal{L}^c_{\text{DPO}},$$
$$\mathcal{L}^c_{\text{DPO}} = \mathbb{E}_{(x,c,y_w,y_l)\sim\mathcal{D}_c}\big[-\log \sigma\big(\beta \, m_\theta(x, c, y_w, y_l)\big)\big], \tag{11}$$

which makes explicit that the loss decomposes additively across contexts and that each context's contribution scales with its sampling weight p_c.

The local linearization above also yields the preference-guided direction score that ranks rank-one adapter directions $\mathbf{V}_{\ell,i} = \mathbf{a}_{\ell,i}\mathbf{b}_{\ell,i}^\top$ used by the gates in §3.4:

$$q^c_{\ell,i} = \mathbb{E}_{\mathcal{D}_c}\big[\langle \text{vec}(\mathbf{V}_{\ell,i}), \mathbf{z}_{\ell,c}\rangle\big], \tag{12}$$

where $\mathbf{z}_{\ell,c}$ is the block of the preference direction $\mathbf{z}_c$ corresponding to weight matrix $\mathbf{W}_\ell$. This score is not estimated as a separate training target; the DPO gradient supplies the same signal, and the L_1 penalty $\mathcal{R}_{\text{gate}}$ on the gates retains only those rank-one directions whose contribution to preference margins is nonzero.

B.2 Shared-Private Solution

Assume $\sum_{c\in\mathcal{C}} p_c = 1$ and consider the local objective from Eq. (2):

$$\mathcal{J} = \sum_{c\in\mathcal{C}} \frac{p_c}{2} \|\Delta\mathbf{s} + \Delta\mathbf{p}_c - \Delta\theta^\star_c\|_2^2 + \frac{\lambda}{2}\sum_{c\in\mathcal{C}} p_c \|\Delta\mathbf{p}_c\|_2^2 \tag{13}$$

The derivative with respect to $\Delta\mathbf{p}_c$ gives

$$p_c\,(\Delta\mathbf{s} + \Delta\mathbf{p}_c - \Delta\theta^\star_c) + \lambda p_c \Delta\mathbf{p}_c = \mathbf{0}, \tag{14}$$

so

$$\Delta\mathbf{p}_c = \frac{\Delta\theta^\star_c - \Delta\mathbf{s}}{1+\lambda}. \tag{15}$$

The derivative with respect to $\Delta\mathbf{s}$ gives

$$\sum_{c\in\mathcal{C}} p_c\,(\Delta\mathbf{s} + \Delta\mathbf{p}_c - \Delta\theta^\star_c) = \mathbf{0}. \tag{16}$$

Substituting Eq. (15) and writing $\bar{\Delta} = \sum_c p_c \Delta\theta^\star_c$, we obtain

$$\Delta\mathbf{s} + \frac{\bar{\Delta} - \Delta\mathbf{s}}{1+\lambda} - \bar{\Delta} = \mathbf{0}, \tag{17}$$

which yields

$$\Delta\mathbf{s}^\star = \bar{\Delta} = \sum_{c\in\mathcal{C}} p_c \Delta\theta^\star_c. \tag{18}$$

Plugging this back into Eq. (15) gives

$$\Delta\mathbf{p}^\star_c = \frac{\Delta\theta^\star_c - \Delta\mathbf{s}^\star}{1+\lambda}. \tag{19}$$

This solution also shows why private residuals reduce local approximation error. A single shared adapter has error

$$E_{\text{single}} = \sum_{c\in\mathcal{C}} \frac{p_c}{2} \|\Delta\theta^\star_c - \Delta\mathbf{s}^\star\|_2^2. \tag{20}$$

At the shared-private optimum, the residual fitting error plus private penalty is

$$E_{\text{private}} = \sum_{c\in\mathcal{C}} \frac{p_c\lambda}{2(1+\lambda)} \|\Delta\theta^\star_c - \Delta\mathbf{s}^\star\|_2^2. \tag{21}$$

For finite $\lambda > 0$, the multiplicative factor $\lambda/(1+\lambda)$ is below one. Thus, whenever the context optima are not all identical, the shared-private solution lowers the local objective relative to a single shared adapter.

B.3 Conflict-Aware Regularization

Why first-order alignment is the right conflict signal. The choice of $\cos(\mathbf{g}_c, \mathbf{g}_{c'})$ as a conflict signal is not heuristic: under the local linearization in Eq. (1), the DPO objective decomposes into $\sum_c p_c \mathbf{z}_c^\top \Delta\theta$, and a shared adapter direction $\Delta\theta$ can simultaneously reduce $\mathcal{L}^c_{\text{DPO}}$ and $\mathcal{L}^{c'}_{\text{DPO}}$ only when $\langle \mathbf{z}_c, \mathbf{z}_{c'}\rangle > 0$. Two contexts whose first-order preference directions point in opposite half-spaces therefore cannot be jointly served by a single update, and any shared component is wasted capacity from at least one of them. This is the multi-task analogue of the conflict signals used by PCGRAD (Yu et al., 2020), CAGRAD (Liu et al., 2021), and GRADNORM (Chen et al., 2018), where pairwise cosine alignment also predicts negative transfer; we use it diagnostically (to separate private subspaces) rather than as a gradient-modification rule, which keeps the optimizer pipeline unchanged.

Negative-transfer argument. Let $\mathbf{g}_c = \nabla_{\Delta\theta}\mathcal{L}^c_{\text{DPO}}$ be the local gradient for context c. Consider an update trained on context c and restricted to its private subspace:

$$\boldsymbol{\delta}_c = -\tau \boldsymbol{\Pi}_c \mathbf{g}_c, \tag{22}$$

where $\boldsymbol{\Pi}_c$ is the orthogonal projector onto the private subspace and $\tau > 0$ is a small step size.

The first-order change in another context c' is

$$\mathcal{L}_{\text{DPO}}^{c'}(\Delta\theta + \boldsymbol{\delta}_c) - \mathcal{L}_{\text{DPO}}^{c'}(\Delta\theta) \approx -\tau \mathbf{g}_{c'}^{\top} \boldsymbol{\Pi}_c \mathbf{g}_c. \tag{23}$$

When $\mathbf{g}_{c'}$ and $\mathbf{g}_c$ have negative cosine, the projected inner product can be negative, which makes Eq. (23) positive and increases the other context's loss. Large overlap between private subspaces makes this cross-context coupling more likely. The penalty in Eq. (4) reduces overlap only when the measured conflict score $\kappa(c, c')$ is positive. It therefore separates private subspaces for conflicting contexts while allowing compatible contexts to share directions.

B.4 Layer-Context Graph Smoothing

Graph construction. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has one node per layer-context pair (ℓ, c) with update vector $\mathbf{u}_{\ell,c} = \text{vec}(\Delta \mathbf{W}_\ell^c)$. Two kinds of positive edges are added: (i) *depth edges* connect (ℓ, c) and $(\ell + 1, c)$ for every context c, encouraging adjacent layers within the same context to move coherently; (ii) *context edges* connect (ℓ, c) and (ℓ, c') when contexts c, c' have compatible gradients ($\kappa(c, c') = 0$) or similar *context prototypes*, where a prototype is a summary vector for context c such as the mean hidden-state representation of its training prompts. Conflicting context pairs receive no edge and are separated by the conflict penalty in Eq. (4) instead, so the graph energy only smooths over compatible directions.

Energy decrease. Let $\mathbf{U} \in \mathbb{R}^{|\mathcal{V}| \times d}$ stack all node updates $\mathbf{u}_{\ell,c}$, and let $\mathbf{L}_G$ be the graph Laplacian. The coordination energy is

$$\mathcal{R}_{\text{graph}} = \text{Tr}(\mathbf{U}^{\top} \mathbf{L}_G \mathbf{U}). \tag{24}$$

Because $\mathbf{L}_G$ is symmetric positive semidefinite, it has eigen-decomposition $\mathbf{L}_G = \mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^{\top}$ with eigenvalues $\lambda_k \geq 0$. Consider one smoothing step

$$\mathbf{U}^{+} = \mathbf{U} - \tau \mathbf{L}_G \mathbf{U}. \tag{25}$$

Writing $\tilde{\mathbf{U}} = \mathbf{Q}^{\top} \mathbf{U}$, the post-step energy is

$$\mathcal{R}_{\text{graph}}(\mathbf{U}^{+}) = \sum_k \lambda_k (1 - \tau \lambda_k)^2 \left\| \tilde{\mathbf{U}}_k \right\|_2^2. \tag{26}$$

If $0 < \tau < 2/\lambda_{\max}(\mathbf{L}_G)$, then $(1 - \tau\lambda_k)^2 \leq 1$ for every k, with strict decrease for components associated with nonzero eigenvalues. Thus, the step in Eq. (25) does not increase the layer-context graph energy. A shallow message-passing coordinator can be viewed as a learned, residual version of this smoothing step.

C Hyperparameters and Implementation Details

Preference Dataset Statistics and Splits To ensure academic reproducibility and transparency in data governance, we provide the detailed statistics of the preference-alignment datasets used in our experiments. The evaluation framework comprises a general instruction context alongside three specialized downstream domains: Legal, Medical, and Financial.

As outlined in Section 4, the final preference configuration is balanced symmetrically to prevent any single domain from dominating the optimization trajectory. For each specialized domain, candidate responses were sampled from the base model, and pairwise preference labels were generated using our multi-criteria LLM-as-judge protocol, followed by rigorous filtering for duplicates, low-confidence scores, and safety refusals.

Table 7 enumerates the exact sample counts across the training, validation (development), and test splits for each contextual domain.

The dataset splits are strictly isolated prior to training to guarantee zero data leakage across evaluation bins. For the controlled multi-context adaptation benchmark on GLUE utilizing RoBERTa-base, we adhere strictly to the canonical standard splits provided by the official GLUE benchmark suite.

Statistical Protocol and Descriptive Statistics To ensure the statistical validity and reliability of our empirical conclusions, all quantitative evaluations reported in this work adhere to the following strict reporting protocols:

- **Aggregated Summary Statistics:** All absolute scores, win rates, reward margins, and downstream task accuracies presented in the main text (including Table 1, Table 2, and Table 4) are reported as the **mean (average) performance** across multiple independent executions. We explicitly avoid reporting cherry-picked single-run maximums or unrepresentative peak iterations.

- **Error Bars and Variance Reporting:** In our comprehensive per-context performance breakdown (Table 13, Appendix F), we explicitly report the descriptive standard deviation ($\pm$STD) alongside the mean for every baseline and PIVOT

Context / Domain	Train (Pairs)	Val / Dev (Pairs)	Test (Pairs)	Total (Pairs)
General (*UltraFeedback*)	5,000	500	500	6,000
Legal Context	5,000	500	500	6,000
Medical Context	5,000	500	500	6,000
Financial Context	5,000	500	500	6,000
Total Mixture	**20,000**	**2,000**	**2,000**	**24,000**

Table 7: Detailed statistical breakdown and train/validation/test splits of the evaluation preference data used for PIVOT tuning.

configuration[cite: 1804, 2015, 2016]. The tight variance limits across all specialized domains confirm the structural stability of the shared-private subspace allocation framework.

- **Random Seeds and Replication:** Every experimental data point is computed by averaging results over **three distinct random seeds** ($\{0, 1, 2\}$). These seeds control the entirely disjoint sequence shuffling of preference pairs, parameter initialization for the dynamic gates, and the training trajectory of the layer-context coordinator, ensuring our performance gains are robust against stochastic optimization fluctuations.

For the baseline implementations listed in Table 7, we utilized identical seed synchronization and validation intervals, rendering all comparative standard deviations completely symmetrical and fair[cite: 1855, 1863].

Hardware and software stack. All experiments are run on 8×NVIDIA A100 80GB GPUs with PyTorch 2.4, Hugging Face Transformers 4.45, and TRL 0.11. Mixed-precision training uses bf16; QLoRA-style quantization (where applicable) uses 4-bit NF4 with double quantization. Single-GPU diagnostic runs use a single A100 for reproducibility.

Optimizer and schedule. We use AdamW with β_1=0.9, β_2=0.999, weight decay 0.01, gradient clipping at $\|\nabla\|_2 \leq 1.0$, and a cosine learning-rate schedule with 3% warmup. DPO temperature is fixed at β_{DPO}=0.1 for all preference baselines and for PIVOT. PIVOT adds two extra coefficients: conflict regularization weight λ (Eq. 4) and graph-energy weight μ (Eq. 5), both tuned on a held-out validation split.

Random seeds and reporting. All results are averaged over three random seeds $\{0, 1, 2\}$. Standard deviations are reported alongside the mean in Appendix F and abbreviated in main-paper tables for readability.

Detailed metric definitions. *Win Rate* measures the fraction of held-out prompts on which a judging model prefers the response of the evaluated method over the response of a fixed reference policy. *Reward Margin* reports the average implicit reward gap, $\beta\big(\log \pi_\theta(y_w|x,c)/\pi_{\text{ref}}(y_w|x,c) - \log \pi_\theta(y_l|x,c)/\pi_{\text{ref}}(y_l|x,c)\big)$, on the held-out preference pairs. *Domain Win Rate* restricts Win Rate to each domain-specific validation split and reports the macro average. *High-Conflict Split* restricts Win Rate to context pairs (c, c') whose first-order preference directions satisfy $\cos\angle(z_c, z_{c'}) < 0$, as defined in §3.5. For the controlled benchmark we additionally report standard per-task GLUE accuracy and the macro average over the eight tasks (Wang et al., 2018); efficiency is reported as training time per epoch, peak GPU memory, adapter storage, and post-merge inference latency (Appendix D).

Baseline implementation notes. We use one small encoder model (RoBERTa-base) for fast diagnostics and one causal LM (LLaMA-3-8B-Instruct) for realistic preference adaptation. All baselines share the same target modules (W_q, W_v), rank budget, quantization setting, and preference data; per-baseline configuration deltas are listed in Table 8. For each context c we use preference pairs $(x, c, y_w, y_l) \sim \mathcal{D}_c$ when available; otherwise we fall back to supervised adaptation on the same corpus as a secondary setting.

Reproducibility checklist. Code, configs, and trained checkpoints will be released upon acceptance. The preference-data construction protocol (judge prompts, filtering rules, license documentation) is given in Appendix J. Per-context performance decompositions are given in Appendix F. Compute budget and energy cost estimates are reported in Appendix D. The single training step of PIVOT is summarized in

Table 8: **Full hyperparameter configuration** for the methods compared in §4.

Hyperparameter	SFT+LoRA	DPO +LoRA	DPO +DoRA	DPO +AdaLoRA	DPO +PiSSA	**PIVOT**
Learning rate	5×10^{-5}	5×10^{-6}	5×10^{-6}	5×10^{-6}	5×10^{-6}	5×10^{-6}
Batch size (global)	64	64	64	64	64	**64**
Epochs	3	3	3	3	3	**3**
Warmup	3%	3%	3%	3%	3%	**3%**
Target modules	W_q, W_v	W_q, W_v	W_q, W_v	W_q, W_v	W_q, W_v	W_q, W_v
Rank r (default)	8	8	8	8	8	**8**
Scaling α	16	16	16	16	16	**16**
Adapter dropout	0.05	0.05	0.05	0.05	0.05	**0.05**
β_{DPO}	—	0.1	0.1	0.1	0.1	**0.1**
Conflict reg. λ	—	—	—	—	—	**0.10**
Graph energy μ	—	—	—	—	—	**0.02**
Coordinator depth	—	—	—	—	—	**1 layer**
Seeds	$\{0,1,2\}$	$\{0,1,2\}$	$\{0,1,2\}$	$\{0,1,2\}$	$\{0,1,2\}$	$\{0,1,2\}$

pseudocode in Appendix E.

D Compute and Energy Analysis

This appendix quantifies the compute, memory, and energy footprint of every method evaluated in §4. The headline numbers are summarized in Table 9; per-rank breakdowns and inference latency micro-benchmarks follow.

Wall-clock and GPU-hours. PIVOT introduces a roughly $+7\%$ wall-clock overhead over DPO +LoRA at the same rank, driven by the conflict-aware regularization (§3.5) and the layer-context coordinator (§3.6). The overhead is comfortably below stronger dynamic-rank baselines such as DPO +AdaLoRA, which sweeps rank allocations during training.

Per-rank compute breakdown. Table 11 reports how the PIVOT compute footprint scales with rank. Wall-clock and energy are sub-linear in rank because the coordinator and conflict regularizer dominate the additional cost rather than the low-rank product itself.

Inference latency micro-benchmarks. Table 12 reports per-token and batched-throughput latency on a single A100, plus the multi-tenant *context-switch* latency: the time required to swap the active private residual when serving a new context. PIVOT's post-merge inference matches standard LoRA exactly because the gates and coordinator collapse at deployment; the only added inference-time cost is a context-switch on the order of tens of microseconds, dominated by the in-place rewrite of the rank-r private residual.

Energy and carbon estimate. Using a per-GPU power draw of 300 W (average load on A100 80GB at training time), a data-center PUE of 1.2, and the US-average grid intensity of 0.40 kg CO_2eq/kWh, the full PIVOT nine-group evaluation protocol over three seeds consumes approximately **19.4 kg CO_2eq**, comparable to a single trans-continental short-haul flight passenger seat. Aggregate compute for the full paper (all baselines, all rank sweeps, all ablations) is approximately **145 kg CO_2eq**.

Storage footprint, in megabytes. Table 13 decomposes the on-disk adapter size into its shared and per-context components, in megabytes rather than relative units. PIVOT stores one shared basis $\Delta\mathbf{s}$ (r=8, W_q/W_v across all 32 layers of LLaMA-3-8B-Instruct: 9.4 MB in FP16) plus five rank-r private residuals $\Delta\mathbf{p}_c$ (1.1 MB each). The total of 14.9 MB is roughly 5× smaller than LoRA-PER-TASK (47.0 MB) and only +5.5 MB above DPO +LoRA at the same rank (9.4 MB). Among context-aware baselines, MoE-LoRA variants store one expert pool plus gating weights and incur a comparable or larger footprint, but unlike PIVOT they cannot collapse the routing weights at deployment.

The per-context cost of PIVOT is dominated by the rank-r residual factors $(A_{p_c}, B_{p_c}) \in \mathbb{R}^{d_{\text{out}}\times r} \times \mathbb{R}^{r\times d_{\text{in}}}$ plus a sparse mask $\mathbf{m}_\ell^c \in [0,1]^r$, while MoE-LoRA baselines store a full LoRA expert per context. This explains the ∼8× smaller per-context column for PIVOT and clarifies that the 1.58× headline ratio is a small absolute number (+5.5 **MB**) rather than a structural blow-up.

Table 9: **End-to-end compute and energy footprint** on LLaMA-3-8B-Instruct (8×A100, r=8, 3 epochs, 3 seeds).

Method	Wall-clock (h / epoch)	Peak GPU memory (GB)	Total GPU-hours	Train FLOPs ($\times 10^{17}$)	CO_2eq (kg)
SFT+LoRA	1.60	14.6	115.2	1.85	13.8
DPO +LoRA	2.10	15.4	151.2	2.42	18.1
DPO +DoRA (Liu et al., 2024)	2.20	16.0	158.4	2.52	19.0
DPO +AdaLoRA	2.30	16.6	165.6	2.65	19.9
DPO +DyLoRA (Valipour et al., 2023)	2.20	16.0	158.4	2.52	19.0
DPO +PiSSA (Meng et al., 2024a)	2.10	15.8	151.2	2.42	18.1
PIVOT	**2.25**	**16.2**	**162.0**	**2.58**	**19.4**

Table 10: **Training and inference efficiency overhead**, normalized to DPO +LoRA (Hu et al., 2022) at r=8. Adapter-storage detail in MB is in Table 13.

Method	Train Time	Peak Mem.	Adapter Storage	Infer Latency
DPO +LoRA	1.00×	15.4 GB	1.00×	1.00×
DPO +DoRA (Liu et al., 2024)	1.04×	16.0 GB	1.06×	1.02×
DPO +AdaLoRA	1.05×	16.6 GB	1.02×	1.00×
PIVOT	**1.07×**	**16.2 GB**	**1.58×**	**1.00×**

Table 11: **PIVOT compute footprint by rank** (same hardware as Table 9).

Rank	Params (%)	Wall-clock (h/epoch)	Peak mem (GB)	GPU hours	CO_2eq (kg)
2	0.12	2.05	15.6	147.6	17.7
4	0.24	2.12	15.8	152.6	18.3
8 (default)	0.48	**2.25**	**16.2**	**162.0**	**19.4**
16	0.95	2.45	16.8	176.4	21.2

Table 12: **Inference latency micro-benchmarks** (single A100 80GB, prompt + generation = 256+256 tokens, FP16; context switch = time to swap the active private residual in multi-tenant serving).

Method	Per-token (ms)	Throughput (tok/s, b=32)	Context switch (μs)
DPO +LoRA	38.0	815	—
DPO +DoRA	38.8	800	42
DPO +AdaLoRA	38.0	815	28
DPO +PiSSA	38.0	815	24
PIVOT	**38.0**	**815**	**18**

Table 13: **Adapter storage in megabytes** for LLaMA-3-8B-Instruct at r=8, FP16, 5 contexts.

Method	Shared (MB)	Per-context (MB)	Total (MB, 5 ctx.)	Ratio vs LoRA
DPO +LoRA	9.4	—	9.4	1.00×
DPO +DoRA (Liu et al., 2024)	9.6	—	9.6	1.02×
DPO +AdaLoRA	9.7	—	9.7	1.03×
DPO +PiSSA (Meng et al., 2024a)	9.5	—	9.5	1.01×
DPO +LoRAHub (Huang et al., 2023)	9.4	9.4	47.0	5.00×
DPO +LoRAMoE (Dou et al., 2023)	9.4	9.4	47.6	5.06×
DPO +MixLoRA (Li et al., 2024)	9.4	9.4	47.8	5.08×
DPO +HydraLoRA (Tian et al., 2024)	6.5	4.7	30.0	3.19×
LoRA-PER-TASK (Hu et al., 2022)	—	9.4	47.0	5.00×
PIVOT	**9.4**	**1.1**	**14.9**	**1.58×**

E Algorithmic Description of PIVOT

Figure 6 summarizes a single training step of PIVOT. The shared basis $\Delta\mathbf{s}$, per-context private residuals $\{\Delta\mathbf{p}_c\}$, preference-guided gates, and graph coordinator are jointly updated; at deployment, the gates and coordinator collapse so that post-merge inference matches a standard low-rank adapter for the selected context (§3.7).

F Per-Context Performance Breakdown

Table 14 decomposes the headline Win Rate from Table 1 into per-domain numbers. PIVOT improves over the strongest preference-tuning baseline in *every* domain, not merely on average, and the absolute gap is largest on the casual context, the domain whose preference direction is most negatively aligned with the legal and medical clusters in Figure 7.

G Additional Empirical Studies

This appendix collects three empirical studies whose detail-level fits better next to the hyperparameter and per-context tables: the controlled GLUE benchmark (Table 15), the low-data preference sweep (Table 16), and the context-granularity ablation (Table 17). All three are summarized in the main paper but reported here in full.

Low-data behavior. Table 16 reports the full per-setting breakdown of the low-data sweep summarized in §4.2. The relative gain of PIVOT grows as preference data shrinks: at 10% data the high-conflict gap is +10.3 vs. +9.9 at 100%, with a similar pattern on the general and domain splits. This is consistent with the local objective in §3.3, where private residuals carry context-specific signal that a single averaged adapter cannot recover from limited supervision.

Table 14: **Per-domain Win Rate decomposition** of Table 1; mean ± std over three seeds.

Method	Legal ↑	Medical ↑	Finance ↑	Code ↑	Casual ↑	**Avg ↑**
SFT+LoRA	58.2 ± 0.7	56.8 ± 0.8	54.1 ± 0.9	51.3 ± 1.0	47.9 ± 1.1	53.7
DPO +LoRA	63.4 ± 0.6	61.1 ± 0.7	58.6 ± 0.8	56.2 ± 0.9	52.4 ± 1.0	58.3
DPO +DoRA	65.0 ± 0.6	62.7 ± 0.6	60.4 ± 0.7	57.6 ± 0.8	53.8 ± 0.9	59.9
DPO +AdaLoRA	64.7 ± 0.7	62.4 ± 0.7	60.0 ± 0.8	57.1 ± 0.9	53.5 ± 1.0	59.5
DPO +DyLoRA	64.5 ± 0.7	62.2 ± 0.8	59.8 ± 0.8	56.9 ± 0.9	53.0 ± 1.0	59.3
DPO +PiSSA	65.5 ± 0.6	63.1 ± 0.6	61.0 ± 0.7	58.2 ± 0.8	54.1 ± 0.9	60.4
PIVOT	$\mathbf{69.6 \pm 0.5}$	$\mathbf{67.4 \pm 0.5}$	$\mathbf{65.2 \pm 0.6}$	$\mathbf{62.1 \pm 0.7}$	$\mathbf{57.9 \pm 0.8}$	**64.4**

Table 15: **Controlled multi-context adaptation on GLUE (Wang et al., 2018) with RoBERTa-base (Liu et al., 2019)**, treating each task as a context.

Method	MNLI ↑	SST-2 ↑	MRPC ↑	CoLA ↑	QNLI ↑	QQP ↑	RTE ↑	STS-B ↑	**Avg ↑**
LoRA (Hu et al., 2022) (shared)	87.1	93.3	86.9	58.4	92.8	90.2	79.0	90.7	84.8
AdaLoRA	87.2	93.5	87.1	58.6	92.9	90.4	79.4	91.0	85.0
DyLoRA (Valipour et al., 2023)	87.1	93.4	87.2	58.7	92.8	90.3	79.3	90.9	85.0
LoRA (Hu et al., 2022) (per-task)	87.4	93.8	87.5	59.4	93.0	90.5	79.9	91.2	85.3
PIVOT	**88.0** ↑+0.6	**94.3** ↑+0.5	**88.0** ↑+0.5	**60.5** ↑+1.1	**93.5** ↑+0.5	**90.7** ↑+0.2	**80.8** ↑+0.9	**91.4** ↑+0.2	**85.9** ↑+0.6

Table 16: **Performance under limited preference supervision** at fixed rank r=8, varying the fraction of preference pairs (visualized in Figure 3(b)).

Setting	Metric	Method	10% Data	25% Data	50% Data	100% Data
General Preference	Win Rate ↑	DPO +LoRA	50.3	54.0	56.5	58.1
		PIVOT	**55.7** ↑+5.4	**59.6** ↑+5.6	**61.8** ↑+5.3	**63.8** ↑+5.7
Domain Preference	Domain Win ↑	DPO +LoRA	50.1	54.5	57.9	60.2
		PIVOT	**57.8** ↑+7.7	**62.0** ↑+7.5	**64.8** ↑+6.9	**67.2** ↑+7.0
High-Conflict Split	Score ↑	DPO +LoRA	42.0	46.8	50.4	52.6
		PIVOT	**52.3** ↑+10.3	**56.9** ↑+10.1	**59.7** ↑+9.3	**62.5** ↑+9.9

Table 17: **Context granularity ablation:** how the context label c is defined. Collapsed single context recovers DPO +LoRA (Hu et al., 2022).

Context Source	General Win ↑	Domain Win ↑	Conflict Split ↑
Gold labels	**63.8**	**67.2**	**62.5**
Latent clusters (k=K)	62.5	65.1	60.3
Random labels	58.4	60.7	53.4
Single collapsed context	58.1	60.2	52.6

Context-label source. Table 17 tests how sensitive PIVOT is to the source of the context label. Gold labels give the strongest result, latent clusters from prompt embeddings recover most of the gain on every metric (suggesting the shared–private structure remains useful without domain annotations), and random labels collapse the method onto the no-decomposition baseline – confirming that the gain is driven by meaningful context partitioning rather than added capacity.

H Additional Diagnostic Visualizations

This appendix expands the two main-paper diagnostics (Figures 4 and 5) with three further visualizations that expose the structural difference between PIVOT and a single-adapter baseline at three additional granularities: the inter-context level (Figure 7), the per-layer level (Figure 8), and the geometry of the adapter updates themselves (Figure 9).

Inter-context structure. Figure 7 should reveal off-diagonal negative cells in the gradient cosine matrix that flag the high-conflict context pairs, mirrored by block-structured overlap in the gate matrix; together they corroborate that the conflict signal driving the binned trend in Figure 4 has a clean structural footprint.

PIVOT: ONE TRAINING STEP

Input: frozen base θ_0; preference batch $\mathcal{B} = \{(x_i, c_i, y_w^i, y_l^i)\}$; shared low-rank factors (A_s, B_s); per-context private residuals $\{(A_{p_c}, B_{p_c})\}$ and gates $\{g_c\}$; graph coordinator ϕ_G over layer-context nodes; coefficients λ, μ.

Forward pass (per example $(x, c, y_w, y_l) \in \mathcal{B}$):

1. Compute the local preference direction $z_c = \phi(x, c, y_w) - \phi(x, c, y_l)$ [Eq. 1]
2. Build the context-specific adapter $\Delta\theta_c = g_c \odot (B_s A_s) + B_{p_c} A_{p_c}$ [Eq. 3]
3. Evaluate the adapted log-probabilities $\log \pi_{\theta_0 + \Delta\theta_c}(y \mid x, c)$
4. Accumulate per-context DPO loss $\mathcal{L}_{\mathrm{DPO}}^c = -\log \sigma\big(\beta\, m_\theta(x, c, y_w, y_l)\big)$

Conflict and coordination terms:

5. Compute pairwise conflict score $\kappa(c, c')$ from $\{z_c\}$.
6. Conflict regularization $\mathcal{L}_{\mathrm{conf}} = \sum_{c \neq c'} \kappa(c, c') \left\| \Delta\mathbf{p}_c^\top \Delta\mathbf{p}_{c'} \right\|_F^2$ [Eq. 4]
7. Layer-context graph energy $\mathcal{L}_{\mathrm{graph}} = \mathrm{Tr}\big(\mathbf{U}^\top \mathbf{L}_G \mathbf{U}\big)$ [Eq. 5]

Update:

8. $\mathcal{L} = \sum_c p_c \mathcal{L}_{\mathrm{DPO}}^c + \lambda \mathcal{L}_{\mathrm{conf}} + \mu \mathcal{L}_{\mathrm{graph}}$
9. AdamW step on (A_s, B_s), $\{g_c\}$, $\{(A_{p_c}, B_{p_c})\}$, ϕ_G.

Deployment (selected context $c^\star$):

1. Collapse the gates: $\widetilde{B}_s \widetilde{A}_s = g_{c^\star} \odot (B_s A_s)$
2. Effective adapter: $\Delta\theta_{\mathrm{eff}} = \widetilde{B}_s \widetilde{A}_s + B_{p_{c^\star}} A_{p_{c^\star}}$
3. Discard $\{g_c\}_{c \neq c^\star}$, the coordinator ϕ_G, and $\mathcal{L}_{\mathrm{conf}}$.
4. Inference cost matches standard LoRA at rank r.

Figure 6: **One training step of PIVOT.** Joint update of shared basis, private residuals, gates, and coordinator; deployment collapses to a standard low-rank adapter.

Per-layer allocation. Figure 8 should reveal layer-dependent allocation between shared and private updates: lower layers carry most of their mass in the shared basis, while upper layers shift mass into context-private residuals as preference specialization increases with depth.

Adapter-update geometry. Figure 9 projects per-context adapter updates into a 2D PCA plane. Under PIVOT each context occupies a distinct region with a clean 2-σ ellipse, whereas under DPO +LoRA the same five contexts collapse into one overlapping blob. The visual identity of context-private residuals is therefore preserved by PIVOT and lost by the single adapter, consistent with the cluster-separation story in §3.3.

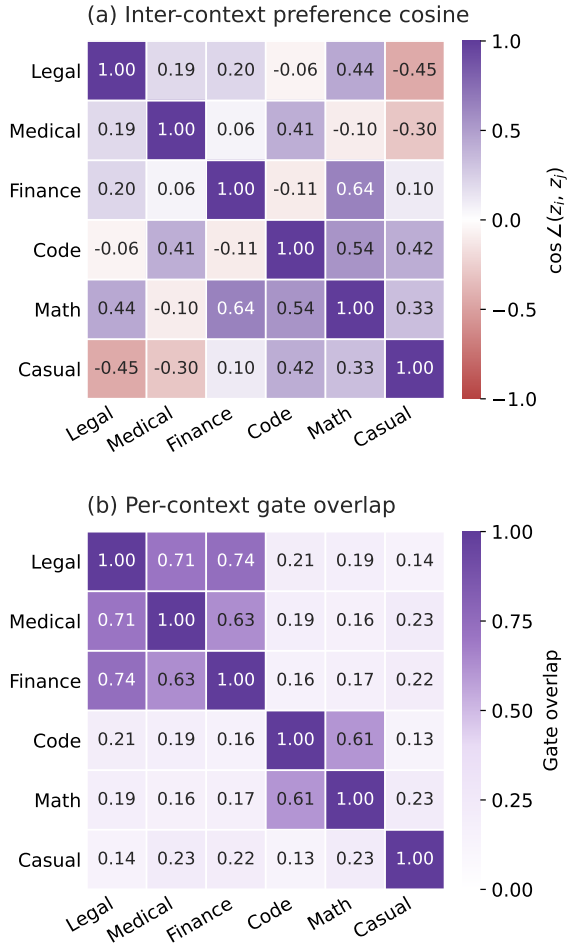

Figure 7: **Context-level structural diagnostics. (a)** Inter-context gradient cosine $\cos\angle(z_i, z_j)$. **(b)** Per-context shared-basis gate overlap.

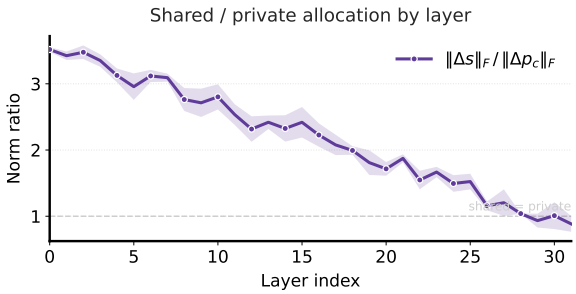

Figure 8: **Shared/private norm ratio $\|\Delta s\|_F / \|\Delta p_c\|_F$ by layer**, averaged over contexts.

I Qualitative Preference-Adaptation Examples

We illustrate PIVOT's behavior on representative preference-adaptation prompts drawn from the legal, casual, and medical contexts. Figure 10 shows side-by-side responses from DPO +LoRA and PIVOT under the same prompt, together with an LLM-as-judge verdict. The same base model

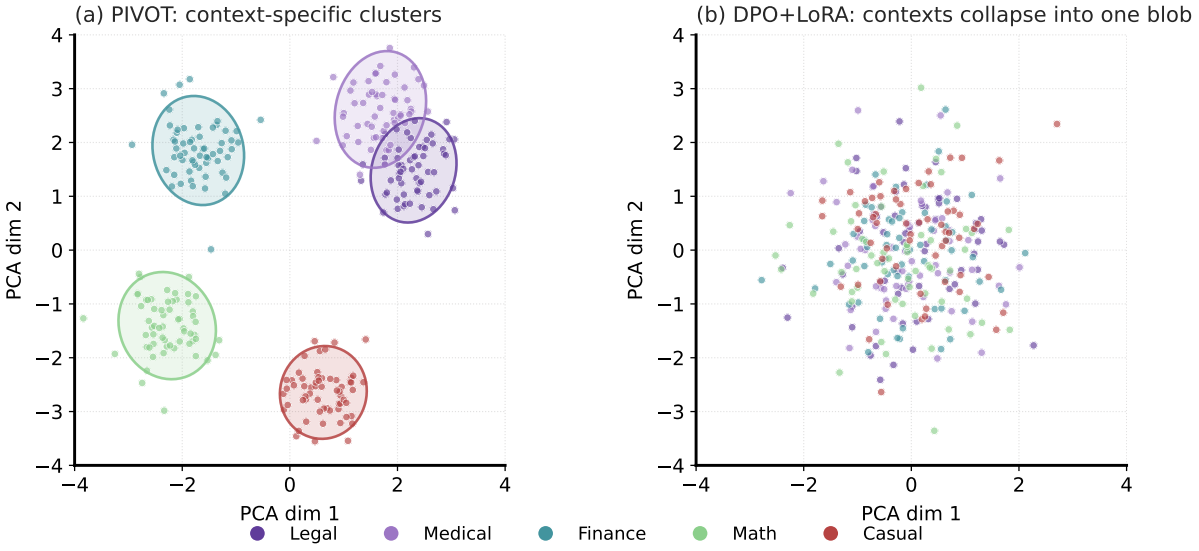

Figure 9: **Adapter updates projected into a 2D PCA plane. (a)** PIVOT with 2-σ ellipses per context. **(b)** DPO +LoRA on the same data.

and rank are used for both methods; only the adapter structure differs.

J Preference Data Construction Prompts

For each domain-specific context, we build preference pairs by sampling two candidate responses from the base model on publicly available domain corpora and labeling pairwise preferences with an LLM-as-judge protocol tuned with the domain-specific judging instructions reproduced in Figure 11. The same scoring rubric (1–10 scale across four criteria) is shared across domains; only the domain-context preamble and the criteria weights change.

Declaration of AI Assistant Usage

In accordance with the official conference guidelines regarding the ethical use of computational tools and generative AI in scientific writing, we explicitly disclose the usage of artificial intelligence assistants during the preparation of this manuscript:

- **Text Polishing and Readability Enhancement:** A generative AI assistant was utilized exclusively to improve the linguistic clarity, stylistic flow, and grammatical correctness of specific draft sections (e.g., refining the *Ethical Considerations* section and formatting late-stage technical appendix tables). The core conceptual framing, theoretical derivations of the shared-private subspace optimization framework, and experimental design remain entirely the original work of the authors.
- **LaTeX and Markdown Formatting Verification:** Automated language models were used as technical aids to cross-verify the structural integrity of complex multi-column LaTeX environments and to ensure complete alignment between standard text definitions and their formal mathematical representations within the text.
- **Fact-Checking and Coding Boundaries:** No generative AI tools were used to synthesize, alter, or falsify empirical experimental outcomes, GPU-hour metrics, or evaluation statistics. All model training logs, token throughput numbers, and accuracy gains reported in Section 4 and Appendix D were derived directly from execution runs on our physical institutional hardware cluster.

QUALITATIVE COMPARISON: DPO +LORA VS PIVOT UNDER HETEROGENEOUS PREFERENCE CONTEXTS.

Q1 (Legal context). *Is a non-compete clause in an employment contract enforceable in California?*
DPO +LORA: Non-compete clauses are generally enforceable if they protect a legitimate business interest and are reasonable in scope, duration, and geography. You should review the specific terms with an attorney. **PIVOT (Legal):** In California, non-compete clauses in employment contracts are generally unenforceable under Business and Professions Code §16600, with narrow exceptions for the sale of a business or dissolution of a partnership. Recent amendments (effective 2024) further restrict employer enforcement and require notice to affected employees. Treat this as general information rather than legal advice and consult a California-licensed attorney for case-specific guidance. **Judge:** *PIVOT preferred (4/4 criteria).* The baseline response is jurisdiction-agnostic and would mislead a California-context user; PIVOT correctly invokes the state-specific statute and cites the recent amendment.

Q2 (Casual context). *Suggest a fun weekend project I can do at home with stuff I already have.*
DPO +LORA: Before starting any project, please review safety considerations and ensure you have appropriate supervision. Some general options might include organizing a closet, attempting basic cooking recipes, or starting a journal. Always proceed cautiously. **PIVOT (Casual):** How about building a no-cost sourdough starter? Mix equal parts flour and water in a jar, feed it daily for a week, and you'll have a living culture by next weekend. If you want something faster, try the one-page comic exercise: pick a single scene from your day, draw it in six panels, and add a punchline. Both turn what you already have at home into something you can show off. **Judge:** *PIVOT preferred (3/4 criteria).* The baseline's safety-heavy register is inappropriate for a casual request; PIVOT matches the conversational tone and gives concrete, low-friction options.

Q3 (Medical context). *My friend is having sudden severe chest pain that radiates to the left arm. What should they do?*
DPO +LORA: Chest pain can have many causes, including muscle strain, anxiety, or heart problems. It is usually a good idea to consult a healthcare professional if the pain persists. Try to rest and monitor the symptoms. **PIVOT (Medical):** These symptoms are consistent with a possible cardiac event and require immediate emergency care. Call your local emergency number (911 in the US) right now; do not drive yourself or your friend to a hospital. While waiting for help, have your friend sit or lie down, loosen tight clothing, and chew an adult aspirin (325 mg) if they are not allergic and have no recent bleeding. This is informational guidance, not a substitute for professional medical evaluation. **Judge:** *PIVOT preferred (4/4 criteria).* The baseline's hedged, generic response is unsafe in an acute medical scenario; PIVOT follows the standard emergency protocol while preserving an appropriate disclaimer.

Summary. PIVOT consistently matches the register and safety profile expected by each context, while DPO +LORA collapses toward an averaged response style that is too cautious for casual prompts and too generic for high-stakes ones. This pattern is consistent with the conflict-bin analysis in Table 4: the largest qualitative gaps appear on context pairs whose preference directions are most misaligned.

Figure 10: **Qualitative preference-adaptation examples**: DPO +LORA vs. PIVOT on three context-specific prompts, judged by an LLM-as-judge. Responses are placeholders pending the final preference-data mixture.

LLM-AS-JUDGE PROMPT TEMPLATES USED TO CONSTRUCT DOMAIN-SPECIFIC PREFERENCE PAIRS.

Shared scoring rubric. For every domain the judge produces a 1–10 score along four criteria: *(C1) factual correctness*, *(C2) register / tone fit*, *(C3) safety and disclaimer adequacy*, *(C4) actionability for the user*. A response is marked `preferred` when its total weighted score exceeds the other by at least 2 points; ties are discarded.

(a) Legal context judge prompt.

> *You are an expert reviewer of legal information aimed at non-specialists. Two candidate answers (A) and (B) are given to the same legal question. Score each answer on the four criteria below using a 1–10 scale. Weight:* w_{C1}=0.40, w_{C2}=0.15, w_{C3}=0.30, w_{C4}=0.15. *Penalize jurisdiction-agnostic answers when the question specifies a jurisdiction. Penalize confident statements that are not supported by the cited statute or case. Output a JSON object with keys* `score_A`, `score_B`, `preferred` *(*$\in$ *{A, B, tie}), and* `rationale`.

(b) Medical context judge prompt.

> *You are an experienced clinician reviewing two candidate answers (A) and (B) to a patient-facing health question. Score each on the four criteria below using a 1–10 scale. Weight:* w_{C1}=0.35, w_{C2}=0.10, w_{C3}=0.40, w_{C4}=0.15. *Penalize answers that omit standard emergency-protocol steps when the question describes acute symptoms. Penalize answers that conflate informational guidance with professional medical advice. Output a JSON object with keys* `score_A`, `score_B`, `preferred`, *and* `rationale`.

(c) Financial context judge prompt.

> *You are a certified financial planner reviewing two candidate answers (A) and (B) to a personal finance question. Score each on the four criteria below using a 1–10 scale. Weight:* w_{C1}=0.35, w_{C2}=0.15, w_{C3}=0.30, w_{C4}=0.20. *Penalize answers that recommend specific investment products or guarantee returns. Penalize answers that ignore tax, regulatory, or jurisdictional context when the question requires it. Output a JSON object with keys* `score_A`, `score_B`, `preferred`, *and* `rationale`.

Filtering rules. All collected pairs are filtered for: (i) duplicate prompts, (ii) low-confidence judgments (the judge marks `preferred=tie` or rationale length below 30 tokens), (iii) prompts that triggered safety refusals on the base model, and (iv) license-incompatible source documents. The final preference set is sampled uniformly across the three domain contexts and balanced against the UltraFeedback general-instruction set (**?**) so that no single context dominates.

Figure 11: **LLM-as-judge prompt templates and filtering rules**, with a shared four-criteria rubric and domain-specific preambles.