Hierarchical Alignment: Block-Level LoRA-DPO for Preference Tuning in Large Language Models

Anonymous ACL submission

Abstract

Preference tuning is usually applied uniformly across all transformer layers, even though interpretability studies suggest that different depths play distinct functional roles. We ask a simple question: *where should preference updates be injected?* We propose *Hierarchical Alignment*, a layer-aware LoRA-DPO framework that partitions a model into shallow (*Local*), middle (*Intermediate*), and deep (*Global*) blocks and trains adapters only within a selected block. Across four models from two pretraining families on HH-RLHF, the middle region emerges as the most reliable fixed-budget target for preference adaptation: on Qwen2.5-7B, Mid-Align matches Full-DPO—LoRA-DPO applied to all transformer layers under the same recipe—within paired-bootstrap confidence intervals while training roughly one third of the trainable LoRA parameters; on Llama-3.1-8B, a residual Full-vs-Mid gap remains but is largely recovered by a middle-adjacent sliding window. Fine-grained depth scans show that low-loss windows concentrate around the middle of the network and consistently outperform random layer subsets at the same budget. Cross-judge evaluation, programmatic benchmarks, a HelpSteer2-Preference transfer check, and layer-wise gradient, update-norm, and representation-drift analyses together support a qualified conclusion: middle and middle-adjacent layers are the primary high-value region for fixed-budget preference adaptation, although generation-level rankings remain model-, dataset-, and rubric-dependent. These results make Mid-block LoRA-DPO a strong parameter-efficient default, with sliding-window refinement as a lightweight per-model adjustment.

1 Introduction

The alignment of Large Language Models (LLMs) with human preferences has transitioned from complex reinforcement learning pipelines to more stable, direct optimization objectives, most notably Direct Preference Optimization (DPO) (Rafailov et al., 2024). Despite this algorithmic progress, a "black-box" assumption persists: current alignment paradigms treat LLMs as monolithic entities and apply uniform parameter updates across the entire depth. This one-size-fits-all view sits at odds with a decade of interpretability evidence showing that Transformer layers are functionally specialized (van Aken et al., 2019; Nadipalli, 2025): lower layers govern local syntax and lexical tokens, middle layers integrate context and discourse, and upper layers carry out task-specific reasoning. Aligning a model by brute-force updates across all layers may therefore be inefficient in two ways at once—it spreads the preference signal thinly across blocks that play very different roles, and it risks unnecessary perturbation of pre-trained knowledge.

In this paper we ask a deliberately concrete question: *at what depth is preference learning most efficiently injected, and is the answer a single layer, a contiguous region, or arbitrary 1/3 of the network?* We propose *Hierarchical Alignment*, an architecture-aware DPO framework that partitions the model into three functionally motivated blocks—*Local* (shallow), *Intermediate* (middle), and *Global* (deep)—and applies targeted updates via Low-Rank Adaptation (LoRA, Hu et al., 2022) within a single block. We treat the depth-thirds partition as a structured starting hypothesis and validate it with fine-grained sliding-window scans, random-subset controls, multi-judge and benchmark triangulation, multi-seed and fair-hyperparameter analyses, a cross-dataset run on HelpSteer2-Preference, and layer-wise mechanistic evidence.

Findings. (i) On strong foundation models such as Qwen2.5-7B, Mid-Align matches Full-DPO—defined throughout as LoRA-DPO applied to all transformer layers under the same recipe—within paired-bootstrap 95% CIs that include zero in two

independent re-trainings, while training roughly one third of Full-DPO's trainable LoRA parameters. On Llama-3.1-8B a residual Full > Mid gap remains, indicating that the closeness between Mid and Full is *model-dependent* rather than universal. (ii) A fine-grained sliding-window sweep across the depth shows a clear valley-shaped training-loss profile whose minimum lies inside or immediately adjacent to the Mid tertile in both model families; this is reproduced for two pre-training lineages and is not recovered by random discrete-layer subsets of the same parameter budget. (iii) On our HH-RLHF reference comparisons (Qwen2.5-7B and Llama-3.1-8B), Global-only updates are the weakest *trained* strategy; the legacy Qwen3-8B row and the HelpSteer2 transfer results show that this deficit is not a universal property of deep-layer tuning. We provide layer-wise gradient, LoRA update, and representation-drift evidence that preference activity is concentrated in middle and middle-adjacent layers, consistent with this rubric-specific Global-only deficit.

Contributions.

- We formalize *Hierarchical Alignment* as a coarse-to-fine, architecture-aware DPO framework, with a clean LoRA-as-subspace-selector implementation that targets only the self-attention modules of a chosen block (§3).
- We stress-test the coarse block hypothesis with sliding-window scans, random-subset controls, multi-judge and benchmark triangulation, multi-seed and fair-hyperparameter analyses, and a HelpSteer2 cross-dataset sanity check (§4).
- We provide layer-wise mechanistic evidence—per-layer gradient RMS, LoRA update norm, and representation drift—that preference signals concentrate in middle and middle-adjacent layers, consistent with a layer-wise account of why Global-only tuning underperforms on HH-RLHF (§5).

Taken together, our results identify middle and middle-adjacent layers as the reliable high-value region for preference adaptation, and establish Mid-block LoRA-DPO as a strong default that captures most of Full-DPO's behavioural gains at roughly one-third of Full-DPO's trainable LoRA parameter count. The precise best window and the magnitude of the residual Full-vs-Mid gap are model- and dataset-dependent. The fixed-budget *training-loss* pattern—middle-adjacent windows are structurally privileged over both shallow/deep blocks and random sparse subsets—transfers across the two reference architectures and the two preference datasets we test. The corresponding *behavioural* conclusions are separately triangulated with multiple judges (Qwen3-Max, DeepSeek-v4-pro, GPT-5.5) and programmatic benchmarks (IFEval, AlpacaEval 2 LC).

2 Related Work

Our work sits at the intersection of preference alignment for LLMs and layer-specific / parameter-efficient editing. We deliberately keep this section focused on these two lines and on the question of where our novelty lies, rather than on broader hierarchical-network evidence.

2.1 Preference Alignment for LLMs

Modern LLM alignment has moved from Supervised Fine-Tuning to RLHF (Ouyang et al., 2022) and to simplified offline alternatives such as DPO (Rafailov et al., 2024) and SimPO (Meng et al., 2024). Recent variants explore hybrid pipelines combining SFT and DPO (Pant, 2025), group-relative weighting (Wang et al., 2025a), and dataset-level robustness through preference filtering (Zhang et al., 2025). A common assumption underlies essentially all of these: the alignment loss is applied uniformly across the entire network, treating the policy model as a monolithic black box. Our work questions exactly this monolithic update assumption.

2.2 Layer-Specific and Parameter-Efficient Tuning

A separate line of work argues that the importance of individual layers or modules is highly non-uniform, and that selective updates can be both efficient and effective.

Coarse-to-fine module selection. CoFiTune (Zhang et al., 2024) first performs coarse-grained tree search over modules and then applies fine-grained soft-masking, demonstrating that "which modules to update" is a non-trivial design choice in its own right.

Sparse layer updates. LISA (Pan et al., 2024) randomly freezes most layers during training and shows that sparse layer updates can match or exceed full fine-tuning under matched memory budgets. IST (Yao et al., 2024) learns explicit layer-importance scores and reports that random sparse

updates already provide a strong baseline that informed selection only needs to beat by a margin.

Single-layer or single-head steering. CogSteer (Wang et al., 2025b) steers behaviour with interpretability-driven single-layer interventions and reports that middle layers are often more effective intervention points than the final layer. Skean et al. (2025) argue that for many downstream tasks, intermediate-layer representations are more informative than the final layer.

Modular adapters at the value/persona level. ALIGNER (Ji et al., 2024) and Modular Pluralism (Feng et al., 2024) compose small external experts for fine-grained behavioural control without modifying the base model. LoRA (Hu et al., 2022) provides the parameter-efficient mechanism we adopt, but is itself agnostic to which layers should receive updates.

Probing evidence about Transformer depth. van Aken et al. (2019) and Nadipalli (2025) document that early layers capture syntax while middle and late layers encode semantics and task-specific logic; analogous results hold for state-space models such as Mamba (Sharma et al., 2024). These probing results motivate our partitioning scheme but do not themselves test how preference learning interacts with depth.

2.3 Positioning

Layer-aware editing is not novel in isolation; what is missing, and what we contribute, is a systematic study of *preference alignment* under contiguous block-level updates with matched LoRA-DPO budgets, with sliding-window and random-subset controls that falsify the "any 1/3 of layers will do" alternative head-on, plus mechanistic evidence tying behaviour back to per-layer gradient flow, LoRA update mass, and representation drift. **For preference alignment specifically, we show that the middle region is a verifiable high-value search prior, characterise its empirical shape, and offer a layer-wise mechanistic account.**

3 Methodology

3.1 Theoretical Foundations

Our approach is built upon the principle that a Transformer's internal architecture is not monolithic but functionally stratified. We formalize this as follows.

Definition 1 (Functional Stratification)**.** *Let a Transformer model be a sequence of N layers, $\mathcal{T} = \{L_1, L_2, \ldots, L_N\}$. A **functional stratification** is a partition of these layers into K disjoint blocks, $\Pi = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K\}$, where each block $\mathcal{S}_k = \{L_i : i \in I_k\}$ is hypothesized to perform a distinct functional role. The model's global computation F can thus be viewed as a composition of block-specific functions: $F(\mathbf{x}; \Theta) = f_K \circ f_{K-1} \circ \cdots \circ f_1(\mathbf{x})$.*

This framework rests on two foundational hypotheses derived from extensive interpretability research.

Hypothesis 1 (Functional Specialization)**.** *In a sufficiently pre-trained LLM, a natural functional stratification exists where blocks process information hierarchically. This hierarchy manifests as a progression from lower-level linguistic features (e.g., syntax) in initial blocks to higher-level semantic and reasoning capabilities (e.g., factuality, intent) in final blocks. We test the depth-thirds instantiation of this hypothesis empirically via a fine-grained sliding-window scan with random-subset controls in §4.3.*

Hypothesis 2 (Objective-Function Correspondence)**.** *For a given alignment objective a_m with loss ℓ_m, preference gradients are expected to be* non-uniform *across depth and enriched in the parameter subspace Θ_k of the block(s) whose representations mediate the targeted behavioural features. Concretely,*

$$g_k(\ell_m) := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \left\| \frac{\partial \ell_m(\mathbf{x})}{\partial \Theta_k} \right\|$$

is hypothesised to vary substantially across blocks k, with mass biased towards a small set of mediating blocks rather than uniformly distributed; we do not claim a strict $\gg$ relation against every *other block. This motivates targeted intervention: restricting updates to the high-mass block(s) should recover most of the optimisation effect at a fraction of the parameter budget. We test this prediction in §5 using per-layer gradient RMS, LoRA update norm, and representation drift, and report both the dominant mid-region concentration and a model-specific late-layer cluster on Llama-3.1-8B as part of the evidence.*

3.2 Implementation

3.2.1 Functional Block Partitioning

Guided by Hypothesis 1, we instantiate the stratification Π with three functionally motivated blocks:

- **Local Block ($\mathcal{S}_{\text{local}}$):** The initial one-third of layers, responsible for syntax, grammar, and fluency.
- **Intermediate Block ($\mathcal{S}_{\text{mid}}$):** The middle one-third of layers, governing discourse coherence and local semantic consistency.
- **Global Block ($\mathcal{S}_{\text{global}}$):** The final one-third of layers, handling thematic relevance, instruction adherence, and high-level reasoning.

To keep the method model-agnostic and reproducible, we adopt a simple deterministic partitioning heuristic that distributes N layers as evenly as possible across the three blocks (exact per-model indices are listed in Appendix A). We treat this depth-thirds partition as a *coarse* hypothesis rather than the final answer: in §4.3 we validate it against a fine-grained sliding-window sweep and against random discrete-layer subsets of the same parameter budget.

3.2.2 Alignment Objective: Direct Preference Optimization (DPO)

We adopt the DPO loss as our alignment objective. For a preference tuple (x, y_w, y_l) where response y_w is preferred over y_l for prompt x, the loss is defined as:

$$\mathcal{L}_{\text{DPO}} = -\,\mathbb{E}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right] \tag{1}$$

where π_θ is the policy model being optimized, π_{ref} is a frozen reference model, and β is a temperature parameter.

3.2.3 Targeted Updates via Low-Rank Adaptation (LoRA)

To enforce the principle of targeted intervention from Hypothesis 2, we require a mechanism that confines parameter updates to a specific block $\mathcal{S}_k$. We employ LoRA for this purpose, treating it as a **subspace selector**.

We freeze the base model and inject trainable low-rank matrices *exclusively* into the self-attention modules of layers within target block $\mathcal{S}_k$, leaving all FFN weights frozen. Optimization operates on the LoRA parameters $\Theta_{k,\text{LoRA}}$:

$$\Theta_{k,\text{LoRA}}^{(t+1)} \leftarrow \Theta_{k,\text{LoRA}}^{(t)} - \eta\nabla_{\Theta_{k,\text{LoRA}}}\mathcal{L}_{\text{DPO}}. \tag{2}$$

4 Experiments

4.1 Experimental Setup and Evaluation Metrics

Models and Training. We evaluate four open-source foundation models: Llama-2-7B (Touvron et al., 2023), Llama-3.1-8B, Qwen2.5-7B, and Qwen3-8B, covering two pre-training families with varying depths (28–36 layers) and reasoning strength. All training uses the Anthropic HH-RLHF preference corpus (Bai et al., 2022) with 10k training pairs. Unless otherwise stated, DPO is trained for one epoch with learning rate 1×10^{-5}, $\beta = 0.1$, effective batch size 16, LoRA rank 16 and $\alpha = 32$ on $\{q, k, v, o\}_{\text{proj}}$, bfloat16, and random seed 42. **Throughout the paper, "Full-DPO" refers to LoRA-DPO applied to all transformer layers under this recipe** (not full-weight fine-tuning); "Mid-Align" refers to the same LoRA-DPO restricted to the middle block, which therefore trains ∼1/3 of Full-DPO's trainable LoRA parameters at matched hyperparameters. Inference for all new experiments uses greedy decoding to remove sampling noise from comparisons. Full hyperparameters are listed in Appendix B.

Data Splits. To prevent "selecting on the test set", we partition the HH-RLHF test split into three disjoint subsets via a fixed seed: `hh-test-main` ($n = 1000$, used only for final reporting), `hh-dev-select` ($n = 300$, reserved for intermediate generation sanity checks and hyperparameter-search diagnostics; the sliding-window ranking itself uses final DPO training loss on the 10k train subset, see §4.3), and `hh-human-pool` ($n = 180$, reserved for human evaluation). `hh-test-main` is index-compatible with the original ACL submission's evaluation set, so the legacy results in §4.2 can be reused.

Multidimensional Judge Protocol. Our *primary* evaluator is Qwen3-Max under a 16-dimensional LLM-as-Judge protocol grouped into six categories: *Language Quality* (grammar, fluency, token robustness, conciseness), *Logic & Coherence*, *Content Quality* (factuality, depth, cross-domain, creativity), *Safety & Ethics*, *Task Execution* (instruction following, tone), and *Meta-Level* (reflection, utility). Scoring uses a 5-point Likert scale anchored to a fixed mediocre baseline response (rated 3/5). The full prompt, dimension definitions, and output template are provided in Appendix D.

Table 1: Legacy block-level results: per-model Qwen-judge Overall mean across the five original strategies on `hh-test-main` ($n \approx 1000$). **Bold** indicates the best strategy within each row. Standard errors and full distributional statistics are reported in Appendix E.

Model	Base	Local	Mid	Global	Full
Llama-2-7B	1.84	2.43	2.36	1.92	**2.45**
Llama-3.1-8B	2.50	2.92	2.84	2.57	**3.04**
Qwen3-8B	3.31	3.24	3.27	**3.33**	3.26
Qwen2.5-7B	3.41	3.46	**3.53**	3.41	3.51
Avg.	2.77	3.01	3.00	2.81	**3.07**

Triangulation. Recognising the family-bias risk of a single judge, all headline claims in §4.4 are independently checked against (i) two non-Qwen judges (DeepSeek-v4-pro, GPT-5.5) using order-swapped pairwise A/B evaluation, (ii) the programmatic benchmarks IFEval (Zhou et al., 2023) and AlpacaEval 2 Length-Controlled (Dubois et al., 2025), and (iii) paired-bootstrap 95% CIs (10,000 resamples, prompt-level pairing). TruthfulQA (Lin et al., 2021) is used as an auxiliary factuality check in Appendix K.

4.2 Legacy Block Results: Initial Evidence

We first reproduce the four-model, five-strategy comparison from the original study as the starting evidence for the block-level effect. Table 1 reports the per-model Overall score for each strategy on `hh-test-main` under the Qwen3-Max rubric.

Three observations are worth stating honestly. (1) When averaged across the four models, Full-DPO has the highest mean (3.07), with Local-Align (3.01) and Mid-Align (3.00) close behind; Mid does not universally beat Full. (2) On the strongest base model in our pool—Qwen2.5-7B—Mid-Align (3.53) exceeds Full-DPO (3.51) and is the per-row winner, while Global-Align is the only trained strategy that fails to improve over Base on Llama-2 and Llama-3.1. (3) The Qwen3-8B row is largely flat: all strategies, including Base, are within ± 0.05 of one another, indicating that DPO on this base provides little headroom. The legacy 16-dimension heatmap (Appendix E) confirms that, behaviourally, Local sharpens lexical/fluency dimensions while Mid moves logical and coherence dimensions the most.

These initial signals are suggestive but not conclusive: the depth-thirds partition is heuristic, and aggregate scores have no error bars. §4.3–§4.5 address both concerns.

4.3 Fine-Grained Block Sweep

To test whether the apparent middle-block advantage is an artefact of the coarse depth-thirds partition or of selecting any 1/3 of layers, we run a fixed-parameter-budget sliding-window scan on the two reference models, Qwen2.5-7B (window size $w = 9$, $N = 28$) and Llama-3.1-8B ($w = 11$, $N = 32$). For each model we train 8 contiguous windows whose centres span shallow $\rightarrow$ deep, plus 5 random discrete-layer subsets of the same size (random seeds 11, 22, 33, 44, 55). Each window/subset is trained on the same 10k HH-RLHF train subset, and *window selection uses the final DPO training loss as a fixed-budget search prior*; `hh-test-main` contributes no signal to selection, and `hh-dev-select` is reserved for inference sanity-check generations and the lr$\times\beta$ grid (§4.5).

Valley-shaped loss profile. Figure 1 shows that the final DPO training loss is a clear U-shape over window-centre depth, with the minimum at the original Mid tertile for Qwen2.5-7B (window `10-18`, loss 0.6844) and at a middle-adjacent window for Llama-3.1-8B (window `6-16`, loss 0.6685 vs. Mid tertile `11-21`'s 0.6728). The depth-thirds partition is therefore close to optimal but not pixel-perfect: it is a good prior, not the final answer.

Contiguous middle beats random discrete layers. On both models, the random-subset band sits strictly above the middle windows: Qwen Mid (0.6844) vs. random mean 0.6883 ± 0.0015 (gap 0.0039); Llama training-loss minimum (`win06-16`, 0.6685) vs. random mean 0.6774 ± 0.0019 (gap 0.0089). The "any 1/3 of layers will do" alternative explanation is thus inconsistent with the data; the middle region is a structurally privileged target.

Held-out evaluation. We transfer each model's Stage-1 training-loss-minimum window to `hh-test-main` and evaluate it alongside Full, original Mid, Base, and a random representative (Global is included for Llama in Stage 2 and for Qwen in the independent Round-2 headline comparison; see Appendix G). On **Qwen2.5-7B**, the training-loss-minimum window *coincides with the original Mid block* (`win10-18`), which reaches a held-out Overall of 3.037, matching Full (3.030) within the paired-bootstrap 95% CI (CI includes zero; §4.4). On **Llama-3.1-8B**, the training-loss minimum is `win06-16` (0.6685, shifted left of the original three-equal-partition Mid `win11-21`); it

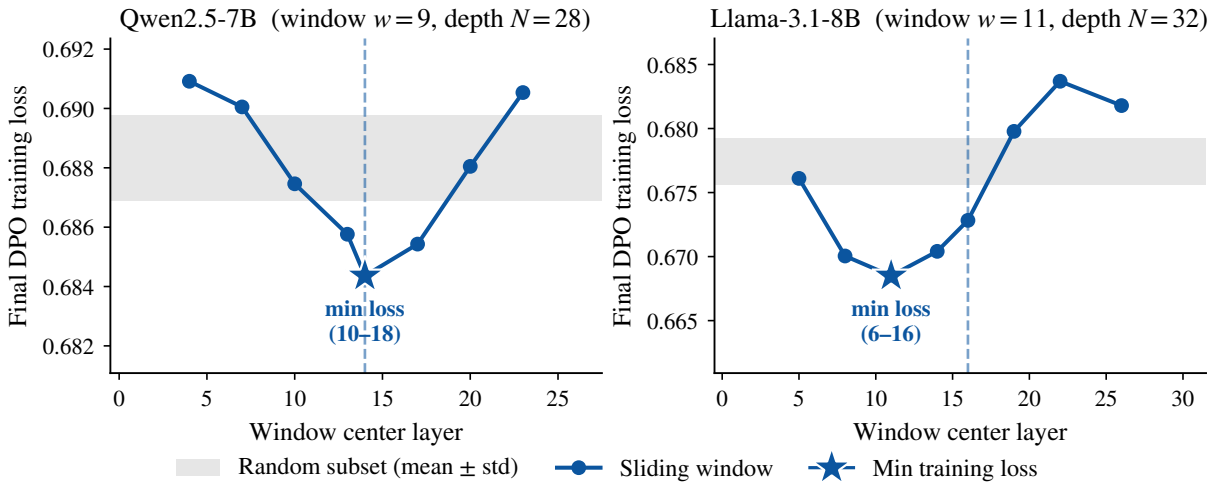

Figure 1: Final DPO training loss, used as a matched-budget Stage-1 optimization diagnostic, vs. sliding-window centre (Qwen2.5-7B: $w=9$, $N=28$; Llama-3.1-8B: $w=11$, $N=32$). Window size matches the original Mid tertile, holding parameter budget constant. Shaded band: mean±std of 5 random discrete-layer subsets at the same budget. Dashed line: centre of Mid tertile. Blue star: minimum-loss window. Both models show a valley-shaped profile whose minimum lies inside or immediately adjacent to the Mid tertile, and contiguous middle windows consistently beat random subsets at the same budget.

reaches 2.709 on the held-out set, comparable to Full (2.694) and clearly above the original Mid (2.403). The selection rule uses Stage-1 training loss only; `hh-test-main` contributes no signal to window selection. Full numbers (all 8 windows, 5 random subsets, additional middle-neighbourhood probes) are in Appendix F.

Why a coarse-to-fine search matters. The fixed-budget loss scan shows that contiguous middle-adjacent windows reliably beat random discrete-layer subsets at the same parameter budget. The original depth-tertile Mid block, however, is not always the best *generation-level* middle allocation: on Llama-3.1-8B, original Mid reaches only 2.403 Overall on `hh-test-main`, lower than the random representative (2.593; non-Qwen pairwise also gives Mid<Random by ~ 8 pp, Appendix G). This is precisely why a coarse depth-thirds prior is followed by a lightweight sliding-window refinement: the training-loss-selected window `win06-16` closes this gap and matches Full within the noise.

4.4 Evaluation Validation: Headline Result

We now check whether the relative ordering between Mid, Full, and Global is robust to the judge family, to standard benchmarks, and to statistical noise. Figure 2 shows the headline Base/Full/Mid/Global comparison on `hh-test-main` for Llama-3.1-8B (stage-2 evaluation) and Qwen2.5-7B (an independent re-trained, re-inferred, re-scored Round-2 replication).

Cross-judge triangulation. Pairwise order-swapped A/B comparisons (Mid vs. Full, Mid vs. Global, Mid vs. Random) with DeepSeek-v4-pro and GPT-5.5 agree in *direction* with the Qwen rubric on all 6 model-comparison cells. On Qwen2.5-7B, Mid is mildly preferred over Full (net +2.5/+2.1 pp) and clearly preferred over Global (+10.3/+11.1 pp) and Random (+7.7/+6.3 pp). On Llama-3.1-8B, Full is clearly preferred over Mid (−18.0/−21.2 pp net to Mid) while Mid is still preferred over Global (+31.1/+50.9 pp). The "Qwen-judge family bias" alternative is therefore not the dominant explanation. Full per-cell numbers and tie rates are in Appendix G.

Programmatic benchmarks. On IFEval (541 prompts, strict and loose program judgement), no trained strategy strays more than 1–2 pp from Full; Mid is within 1.1 pp of Full on every metric variant for both models (the worst case is Qwen Loose-P, where Full 36.6 vs. Mid 35.5), indicating that restricting LoRA-DPO to the middle block does not harm instruction-following. On AlpacaEval 2 Length-Controlled (805 prompts, `gpt-4o-mini` annotator vs. `text-davinci-003`), Llama Full and the Llama training-loss-selected window `win06-16` are within 0.2 LC pp; on Qwen the LC win rates saturate at the AlpacaEval regulariser ceiling, so we report only the raw Mid ≈ Full pattern. Detailed tables (and TruthfulQA) are in Appendix G.

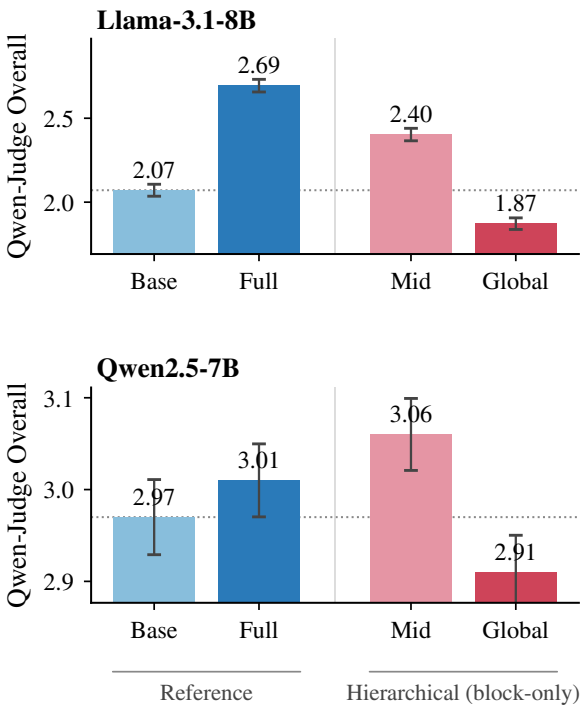

Figure 2: Headline 16-dim Qwen-judge Overall on `hh-test-main` ($n \approx 998$); Qwen2.5-7B uses an *independent Round-2 replication*. Error bars: ± 1 SE; dashed = Base. Llama: Full > Mid (paired-bootstrap $\Delta = -0.292$, 95% CI $[-0.365, -0.222]$). Qwen2.5-7B: Mid $\approx$ Full (R1 CI $[-0.045, +0.058]$; R2 CI $[-0.004, +0.104]$). For the two headline models shown here, Global-only is the weakest trained strategy under the HH-RLHF rubric; this ordering can shift on other rubrics (§4.5). Local-only and sliding-window variants are in the appendix.

Paired-bootstrap CIs. On `hh-test-main`, the 95% paired-bootstrap CIs of Mid$-$Full are $[-0.045, +0.058]$ in Round 1 and $[-0.004, +0.104]$ in the independent Round-2 replication for Qwen2.5-7B, and $[-0.365, -0.222]$ for Llama-3.1-8B; Mid$-$Global is positive on both models (Qwen $+0.148^\star$ in R2, Llama $+0.532^\star$; $\star$ = excludes 0). The Mid$\approx$Full closeness on Qwen2.5-7B is thus robust under multiple judges and across independent replications; the Mid<Full gap on Llama-3.1-8B is significant.

4.5 Robustness and Cross-Dataset Transfer

Seed and hyperparameter robustness. On Qwen2.5-7B, Full and Mid are stable across three independent training seeds (13, 42, 77), reaching final DPO training losses of 0.6821 ± 0.0006 and 0.6844 ± 0.0004, respectively. Random is evaluated across three layer-selection seeds (11, 22, 33) under the same training recipe and reaches 0.6876 ± 0.0012; this variance reflects subset-selection rather than training stochasticity, but we report it for comparison. The Full < Mid < Random (mean) ordering does not flip on any seed, and Mid's per-seed standard deviation is one-third of Random's. A 2×2 learning-rate $\times \beta$ grid (scored by final DPO training loss on the same 10k train subset, with `hh-dev-select` reserved for inference sanity check) selects *the same* optimum (lr $= 1 \times 10^{-5}$, $\beta = 0.1$) for both Full and Mid, ruling out the "default hyperparameters happen to favour Mid" alternative explanation. Full numbers are in Appendix H.

Cross-dataset sanity check on HelpSteer2-Preference. To check whether the layer-ordering pattern is HH-RLHF-specific, we re-train Qwen2.5-7B and Llama-3.1-8B on 6,766 non-tie HelpSteer2-Preference pairs (Wang et al., 2025c) and evaluate on 300 held-out HelpSteer2 validation prompts (same Qwen3-Max rubric, paired bootstrap CIs). The training-loss ordering largely transfers: averaged across random seeds, Full < Mid < Random < Global holds on both models. Mid and Full are again not significantly different in held-out generation (paired-bootstrap 95% CIs cover 0 for Mid$-$Full on both models; Qwen $[-0.110, +0.063]$, Llama $[-0.107, +0.040]$). We also report two phenomena *honestly*: (i) on HelpSteer2 the Mid vs. Global direction reverses under Qwen3-Max's general-helpfulness rubric, and (ii) DPO on HelpSteer2 does not raise absolute Qwen3-Max Overall scores above Base on either model. We therefore treat HelpSteer2 as evidence for the *relative* layer-allocation pattern, not for an absolute generation-quality gain; §7 discusses this caveat. Full 8-variant tables (including sliding-window probes) are in Appendix I.

5 Mechanistic Evidence

We now report layer-wise measurements consistent with the prediction of Hypothesis 2 that the middle region is mechanistically distinguished during preference learning. We collect three signals on Qwen2.5-7B and Llama-3.1-8B for both Full-DPO and Mid-only training: (a) per-layer gradient RMS over the first 100 training steps (averaged over q, k, v, o projections), (b) the layer-summed Frobenius norm of the LoRA $\Delta W = (\alpha/r)BA$ at the end of training, and (c) per-layer cosine drift $1 - \cos(h_l^{\text{base}}, h_l^{\text{aligned}})$ of last-token hidden states on 256 held-out prompts.

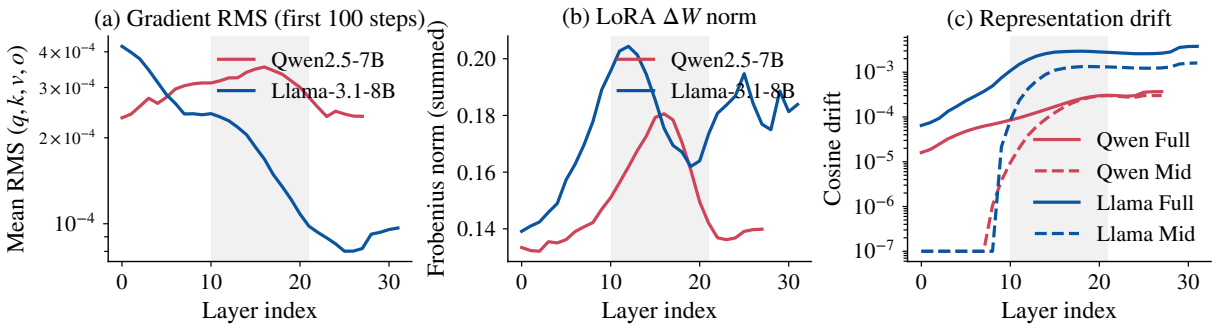

Figure 3: Three views of preference-tuning localisation along depth. **(a)** Per-layer gradient RMS during Full-DPO (first 100 steps). **(b)** Layer-summed Frobenius norm of LoRA ΔW at end of Full-DPO. **(c)** Per-layer cosine drift between base and aligned hidden states on 256 held-out prompts; solid = Full-DPO, dashed = Mid-only. Shaded: union of both models' Mid tertile (layers 10–21). Middle and middle-adjacent layers repeatedly emerge as high-activity regions; Llama-3.1-8B has an additional late-layer LoRA-norm cluster (layers 28–31) mirroring its Full-vs-Mid gap in Figure 2. Mid-only drift starts at the LoRA injection range and propagates downstream.

Gradient flow is concentrated in middle and middle-adjacent layers. In Full-DPO training, Qwen2.5-7B's per-layer gradient RMS forms a clear peak in the middle band (layers 14–17); Llama-3.1-8B maintains a high plateau from early through middle layers and drops sharply after layer 21. In both cases late layers receive substantially less gradient mass. This pattern is consistent with the HH-RLHF Global-only deficit on our reference models (§4.2, §4.4); the HelpSteer2 reversal (§4.5) indicates that Global-only behaviour remains rubric- and dataset-dependent.

LoRA update mass peaks in the middle, with a model-specific late-layer cluster on Llama. The final LoRA ΔW Frobenius norm peaks at the Mid tertile on both models. Llama-3.1-8B uniquely shows a secondary spike at layers 28–31; we report this honestly rather than hide it, because it correlates with the surviving Full $>$ Mid gap on Llama (Figure 2) and suggests Llama distributes preference learning across more layers than Qwen.

Mid-only updates propagate downstream, but not equally across models. Under Mid-only training, per-layer cosine drift starts at the LoRA injection range ($\sim$layer 10) and propagates into later layers. The extent of this propagation is model-dependent: in the late region, Mid-only reaches roughly $0.91\times$ of Full-DPO drift on Qwen2.5-7B (0.00030 vs. 0.00033), but only about $0.45\times$ on Llama-3.1-8B (0.00138 vs. 0.00304). This asymmetry mirrors the behavioural results: Qwen admits a near Mid$\approx$Full solution, whereas Llama retains a larger all-layer advantage and also exhibits a late-layer LoRA-norm cluster.

Together, these signals show that preference activity concentrates in middle and middle-adjacent layers, propagates downstream, and varies by model in ways that track the residual Full-vs-Mid gap. Per-layer values are in Appendix J.

6 Discussion and Conclusion

The middle region as a robust default. Across block-level results, sliding-window scans, cross-judge checks, programmatic benchmarks, cross-dataset transfer, and layer-wise mechanistic analyses, our evidence points to a practical lesson: middle and middle-adjacent layers are the most reliable fixed-budget region for LoRA-DPO preference adaptation. On Qwen2.5-7B, Mid-block LoRA-DPO matches all-layer LoRA-DPO within paired-bootstrap CIs while training roughly one third of the trainable LoRA parameters. On Llama-3.1-8B, fixed Mid leaves a significant residual gap, but a middle-adjacent sliding window recovers much of it. Thus, Mid-block LoRA-DPO is a strong default, and sliding-window refinement is a model-specific adjustment when Full-vs-Mid gaps matter.

Conclusion. Hierarchical Alignment reframes preference tuning as a layer-allocation problem rather than an all-layer default. The depth-thirds Mid block is a strong prior, sliding-window scans refine it when needed, and gradient, update-norm, and representation-drift evidence explain why the middle is privileged. The exact best window remains model- and rubric-dependent, but the broader lesson is stable: exploiting depth-wise structure yields a more parameter-efficient default for preference alignment.

7 Limitations

Our findings are based on four LLMs with broadly similar Transformer decoder architectures; whether the same middle-region pattern holds for state-space, retentive, or mixture-of-experts models remains open. The layer partitioning we study is depth-based and static; although sliding-window refinement (§4.3) softens the heuristic concern, fully data-driven block assignment via probing or gradient analysis is left to future work. Our primary automated evaluation uses Qwen3-Max as a judge; we validate all headline claims with two non-Qwen judges (DeepSeek-v4-pro, GPT-5.5) and programmatic benchmarks (IFEval, AlpacaEval 2 LC), and a small human blind-eval pack has been prepared but not annotated. The cross-dataset run on HelpSteer2-Preference reveals two limits to a strong universal claim: (i) under Qwen3-Max's general-helpfulness rubric, DPO on HelpSteer2 does not raise scores above the base model, and (ii) the Mid-vs-Global direction reverses on Qwen, indicating the Global-only deficit is partly rubric- and dataset-specific. Finally, our efficiency claim is restricted to *parameter* efficiency and to the smaller layer-search space we suggest; we make no wall-clock-speedup claim because our current implementation runs a full forward pass even when LoRA is restricted to a single block (a partial-backprop implementation would be required to convert this into real training speed-up).

8 Ethical Considerations

Our work raises several ethical considerations. First, the Anthropic/hh-rlhf dataset may contain social biases related to gender, race, or ideology, which could be amplified during alignment—even if not introduced by our method. Second, while our approach improves parameter efficiency, training and evaluation still demand significant computational resources; we encourage future work on lightweight replications to improve accessibility. Third, layer-specific tuning enables fine-grained behavioral control, posing dual-use risks: such techniques could be misused to induce harmful behaviors while maintaining fluent output. We advocate for transparency in model editing and research into detection mechanisms for stealthy manipulations. Finally, our LLM-as-Judge evaluation uses Qwen3-Max, whose training data and biases are not fully transparent. We caution against overreliance on automated metrics and recommend human validation for high-stakes applications.

Use of AI Assistants

We used large language models (e.g., Qwen3-Max) only to assist with language polishing of the manuscript. All core ideas, methodology design, experiment execution, statistical analysis, and writing were performed by the authors. Every model-suggested edit was reviewed, fact-checked, and revised by the authors before inclusion. No part of the technical content—including the layer-block design, the evaluation protocol, the mechanistic analyses, and the reported numerical results—was produced autonomously by an AI system. (Acknowledgements of funding, infrastructure, and individuals are omitted in the anonymous submission and will be added to the camera-ready version.)

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2025. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. *Preprint*, arXiv:2404.04475.

Shangbin Feng, Taylor Sorensen, Yuhan Liu, Jillian Fisher, Chan Young Park, Yejin Choi, and Yulia Tsvetkov. 2024. Modular pluralism: Pluralistic alignment via multi-llm collaboration. *Preprint*, arXiv:2406.15951.

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

Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Juntao Dai, Tianyi Qiu, and Yaodong Yang. 2024. Aligner: Efficient alignment by learning to correct. *Preprint*, arXiv:2402.02416.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods, 2022. *URL https://arxiv.org/abs/2109.07958*, 1.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple preference optimization with a

reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

Suneel Nadipalli. 2025. Layer-wise evolution of representations in fine-tuned transformers: Insights from sparse autoencoders. *Preprint*, arXiv:2502.16722.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. 2024. Lisa: Layer-wise importance sampling for memory-efficient large language model fine-tuning. *Advances in Neural Information Processing Systems*, 37:57018–57049.

Piyush Pant. 2025. Improving LLM safety and helpfulness using SFT and DPO: A study on OPT-350m. *Preprint*, arXiv:2509.09055.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Arnab Sen Sharma, David Atkinson, and David Bau. 2024. Locating and editing factual associations in mamba. *Preprint*, arXiv:2404.03646.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 55854–55875. PMLR.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. 2019. How does BERT answer questions?: A layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pages 1823–1832. ACM.

Haowen Wang, Yun Yue, Zhiling Ye, Shuowen Zhang, Lei Fan, Jiaxin Liang, Jiadi Jiang, Cheng Wei, Jingyuan Deng, Xudong Han, Ji Li, Chunxiao Guo, Peng Wei, Jian Wang, and Jinjie Gu. 2025a. Learning to align, aligning to learn: A unified approach for self-optimized alignment. *Preprint*, arXiv:2508.07750.

Xintong Wang, Jingheng Pan, Liang Ding, Longyue Wang, Longqin Jiang, Xingshan Li, and Chris Biemann. 2025b. Cogsteer: Cognition-inspired selective layer intervention for efficiently steering large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25507–25522.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2025c. Helpsteer2-preference: Complementing ratings with preferences. In *The Thirteenth International Conference on Learning Representations*.

Kai Yao, Penglei Gao, Lichun Li, Yuan Zhao, Xiaofeng Wang, Wei Wang, and Jianke Zhu. 2024. Layer-wise importance matters: Less memory for better performance in parameter-efficient fine-tuning of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1977–1992.

Hengyuan Zhang, Yanru Wu, Dawei Li, Sak Yang, Rui Zhao, Yong Jiang, and Fei Tan. 2024. Balancing speciality and versatility: a coarse to fine framework for supervised fine-tuning large language model. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7467–7509.

Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025. Cm-align: Consistency-based multilingual alignment for large language models. *Preprint*, arXiv:2509.08541.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

A Model Layer Partition Details

To ensure reproducibility and cross-architecture generalizability, we implement a *dynamic layer splitting* strategy that partitions each Transformer model into three functionally distinct blocks: Local, Intermediate, and Global. Given a total depth N, we define the base layers per block as $n = N \mathbin{//} 3$ and the remainder as $r = N \bmod 3$. The blocks are then sequentially allocated such that the Local Block comprises the first $n + [r > 0]$ layers, the Intermediate Block covers the subsequent $n + [r > 1]$ layers, and the Global Block contains the remaining layers, where $[\cdot]$ denotes the Iverson bracket. The specific layer indices for each evaluated model are detailed in Table 2. All LoRA adapters are applied exclusively within these predefined ranges to ensure a rigorous and isolated comparison of hierarchical alignment effects.

Table 2: Layer partitioning for each model under the dynamic splitting strategy.

Model	Total Layers	Local Block	Intermediate Block	Global Block
Llama-2-7b	32	0–10	11–21	22–31
Llama-3.1-8B	32	0–10	11–21	22–31
Qwen3-8B	36	0–11	12–23	24–35
Qwen2.5-7B	28	0–9	10–18	19–27

Note that layer indexing starts at 0. For example, in Qwen2.5-7B ($N = 28$), $n = 9$, $r = 1$, so only the Local Block receives an extra layer.

This systematic approach ensures consistent methodology across models of varying depths, enabling fair comparison of hierarchical alignment strategies.

B Complete Training Hyperparameters

We provide the full list of hyperparameters used in our Direct Preference Optimization (DPO) training pipeline to guarantee reproducibility. Unless otherwise specified, all models and conditions share the same training configuration.

All experiments use the Hugging Face `Transformers` and `TRL` libraries with fixed random seeds for data loading, shuffling, and parameter initialization. The above configuration applies to the HH-RLHF DPO experiments reported in §4.3–§5; the HelpSteer2-Preference run (§4.5) keeps the same effective batch size and optimisation hyperparameters but uses a smaller per-device batch with larger gradient accumulation and gradient checkpointing enabled because its prompts are longer (Appendix I). Legacy results in §4.2 were trained under an earlier sampling-decode pipeline whose outputs we reuse without modification but do *not* mix with the new outputs in any held-out comparison.

C Training Convergence (Reference Only)

For completeness we report DPO training-loss convergence across the four legacy models and five strategies in Figure 4. All configurations converge within a single epoch, indicating stable optimization under the hyperparameters in Appendix B.

We *do not* report wall-clock training time as evidence of efficiency. Our current LoRA-on-a-single-block implementation still runs a full forward pass through every layer, so observed wall-clock differences are dominated by hardware noise rather than by structural sparsity; a real wall-clock speedup would require partial-backprop or structured-sparsity engineering that is outside the scope of this paper. Accordingly, all efficiency claims in the main text are framed in terms of *parameter* efficiency and *search-space* efficiency only.

D Detailed Evaluation Protocol

To ensure transparency, consistency, and full reproducibility of our automated evaluation process, we provide the complete structured prompt used to elicit judgments from the *Qwen3-Max* model. This protocol enforces a blind evaluation, anchors scoring to a fixed reference point, and mandates a strict machine-readable output format.

Evaluation Dimensions Table 4 details the 16 specialized dimensions used for scoring. Each dimension is designed to capture specific linguistic, logical, or ethical attributes of the generated responses.

LLM-as-Judge Prompt Template The evaluation prompt provided to *Qwen3-Max* is structured as follows. We use a Likert-scale approach (1–5) where a score of 3 is explicitly anchored to a mediocre baseline case to minimize model-specific bias.

> ***[System Role]***
> *You are a rigorous and objective text quality analyst. Your task is to evaluate a single AI-generated response based on predefined dimensions and scoring standards, anchored to a baseline example.*
>
> ***[Baseline Case (Anchor for Score = 3)]***
> *- Prompt: What are some good types of hats to shade myself from the sun?*
> *- Response: There are many kinds of hats, like baseball caps, sun hats, bucket hats and so on. You can choose one you like.*
>
> ***[Dimension Definitions]***
> *(The dimensions defined in Table 4 are passed here in JSON format...)*
>
> ***[Scoring Criteria (Relative to Baseline)]***
> *- 1: Significantly worse than baseline (severe factual errors or topic drift).*

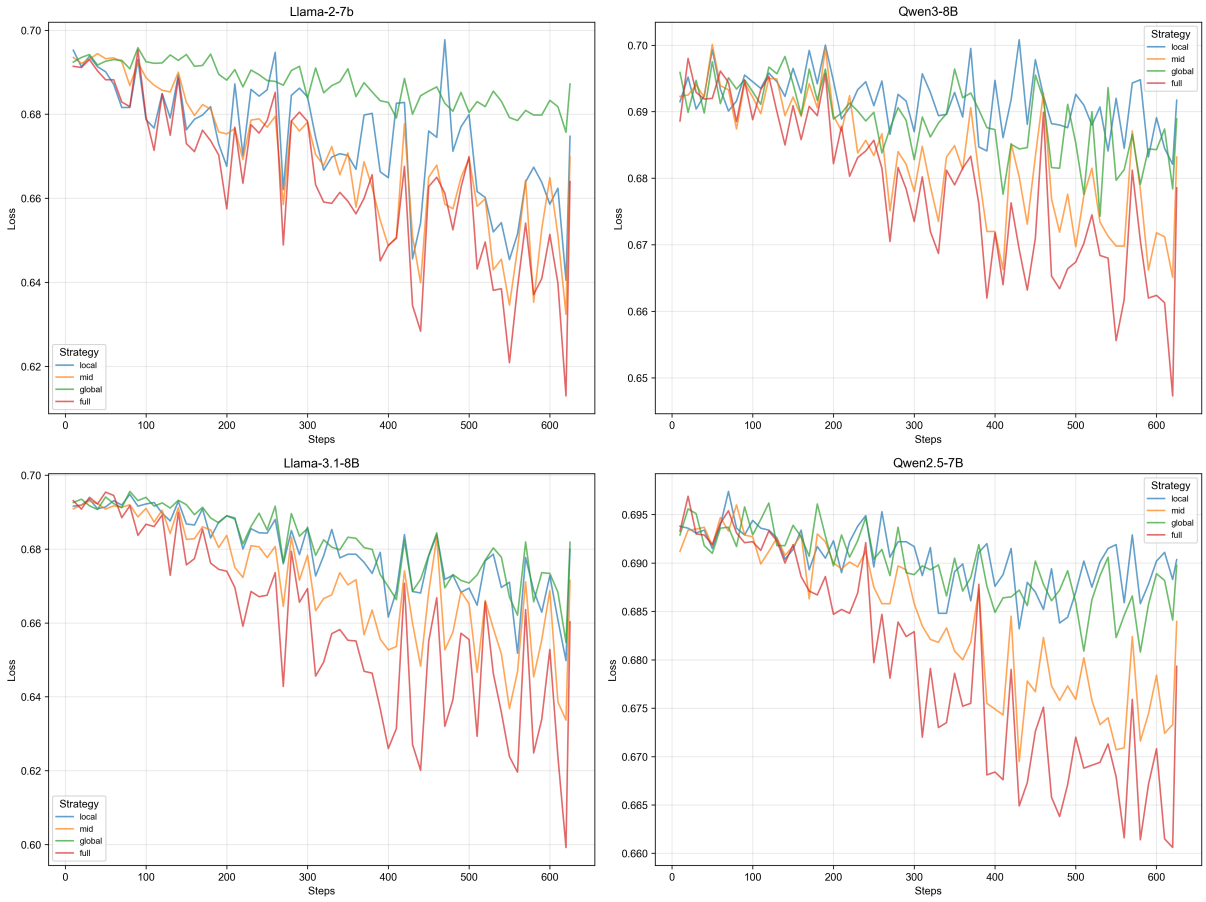

Figure 4: Training loss curves across all four legacy models and five strategies. Each subplot corresponds to one base model. All trained variants (*local*, *mid*, *global*, *full*) exhibit monotonic loss reduction; base models (no training) maintain flat trajectories.

- 2: Slightly worse than baseline (minor shortcomings in logic or content).
- 3: On par with baseline (meets minimum requirements, lacks refinement).
- 4: Slightly better than baseline (improved detail or structure).
- 5: Clearly superior to baseline (substantial enhancement in multiple dimensions).

[Response to Evaluate]
- User Prompt: [INSERT PROMPT]
- Model Response: [INSERT RESPONSE]

[Output Requirements]
*Return a JSON array where each element contains "dimension", "score" (integer 1-5), and "reason". Output **only** the raw JSON array.*

E Legacy Heatmap and Full Distributional Statistics

This appendix gathers the detailed legacy results that motivate §4.2. Figure 5 shows the 15-dimension Qwen-judge heatmap originally reported in the ACL submission; Table 5 reports the strategy-level aggregate statistics ($N = 4{,}000$, four models pooled) and Table 6 reports the top-10 model-strategy combinations on `hh-test-main`.

F Block Sweep: Full Results

This appendix reports the full sliding-window scan numbers behind Figure 1 and the held-out reporting in §4.3. All windows are trained with the same parameter budget as the original Mid tertile (Qwen2.5-7B: $w = 9$ layers; Llama-3.1-8B: $w = 11$ layers) under the standard DPO recipe (lr 1×10^{-5}, $\beta = 0.1$, LoRA $r = 16/\alpha = 32$ on $\{q, k, v, o\}_{\text{proj}}$, one epoch).

Stage-1 sliding-window training loss (Qwen2.5-7B, $N = 28$).

Window	Layer range	Final train loss	Δ vs Mid
`win00-08`	0–8	0.6909	+0.0065
`win03-11`	3–11	0.6901	+0.0057
`win06-14`	6–14	0.6875	+0.0031
`win09-17`	9–17	0.6858	+0.0014
`win10-18`*	10–18 (=Mid)	**0.6844**	baseline
`win13-21`	13–21	0.6854	+0.0010
`win16-24`	16–24	0.6880	+0.0036
`win19-27`	19–27	0.6905	+0.0061

* Original Mid tertile; for Qwen2.5-7B this is also the Stage-1 training-loss minimum, so the training-

Table 3: Complete set of training hyperparameters used in the HH-RLHF DPO experiments (§4.3–§5). Values match the actual training code (corrected relative to the ACL submission for four entries: weight decay, batch size, gradient checkpointing, and precision). HelpSteer2 deviations are listed in Appendix I; SimPO uses its recommended optimisation settings in Appendix L.

Hyperparameter	**Value**
Learning Rate	1.0×10^{-5}
Optimizer	AdamW
Weight Decay	0.0
β_1, β_2	0.9, 0.999
Per-device Batch Size	4
Gradient Accumulation Steps	4
Effective Batch Size	16
Number of Epochs	1
DPO Beta (β)	0.1
LoRA Rank (`r`)	16
LoRA Alpha (α)	32
LoRA Dropout	0.05
Target Modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`
Precision	bfloat16
Gradient Checkpointing	Disabled
Max Prompt Length	1024
Max Sequence Length	2048
Inference Decoding (new exp.)	greedy
Training Seed	42

loss-selected sliding-window transfer for Qwen coincides with the original Mid block. Other windows in the scan (`win06-14`, `win09-17`, `win13-21`, etc.) are reported as middle-neighbourhood probes for robustness, not as training-loss-selected candidates. Full-DPO (all 28 layers) reaches 0.6825 on the same recipe.

Random discrete-layer baselines (Qwen2.5-7B, 9 layers each).

Seed	**Final train loss**
11	0.6875
22	0.6889
33	0.6867
44	0.6876
55	0.6908
Mean±std	0.6883 ± 0.0015

Stage-1 sliding-window training loss (Llama-3.1-8B, $N = 32$).

Table 4: Definitions of the 16 evaluation dimensions grouped by functional categories.

Dimension	**Definition**
GrammarFluency	Grammatical accuracy, sentence completeness, and general fluency.
TokenRobustness	Consistency in semantic integrity across input variations.
Conciseness	Balance between brevity and information completeness.
SyntacticComplexity	Diversity of syntactic structures and linguistic richness.
CoherenceLogic	Logical flow and causal relationships between sentences.
TopicCoherence	Relevance and focus on the central query themes.
Tone	Contextual appropriateness of the response style.
Factuality	Empirical accuracy of claims and information.
KnowledgeDepth	Detail level, including material properties or design features.
CrossDomain	Generalizability across diverse scenarios (e.g., sports, urban).
Creativity	Novelty and originality of the proposed suggestions.
Safety	Absence of hazardous advice or high-risk content.
EthicsSensitivity	Adherence to ethical norms and exclusionary-free language.
MetaReflection	Self-awareness and explicit statement of guidance limitations.
InstructionFollowing	Degree to which the user's specific constraints are addressed.
Overall	Holistic quality assessment synthesized across all dimensions.

Window	**Layer range**	**Final train loss**	**Δ vs Mid**
`win00-10`	0–10	0.6761	+0.0033
`win03-13`	3–13	0.6700	−0.0028
`win06-16`†	6–16	**0.6685**	−0.0043
`win09-19`	9–19	0.6704	−0.0024
`win11-21`⋆	11–21 (=Mid)	0.6728	baseline
`win14-24`	14–24	0.6798	+0.0070
`win17-27`	17–27	0.6837	+0.0109
`win21-31`	21–31	0.6818	+0.0090

⋆ Original three-equal-partition Mid tertile. † Stage-1 training-loss minimum window for Llama-3.1-8B; used as the training-loss-selected sliding-window transfer variant. Other windows (e.g., `win03-13`) are reported as middle-neighbourhood probes for robustness, not as training-loss-selected candidates. Full-DPO (all 32 layers) reaches 0.6607.

Random discrete-layer baselines (Llama-3.1-8B, 11 layers each).

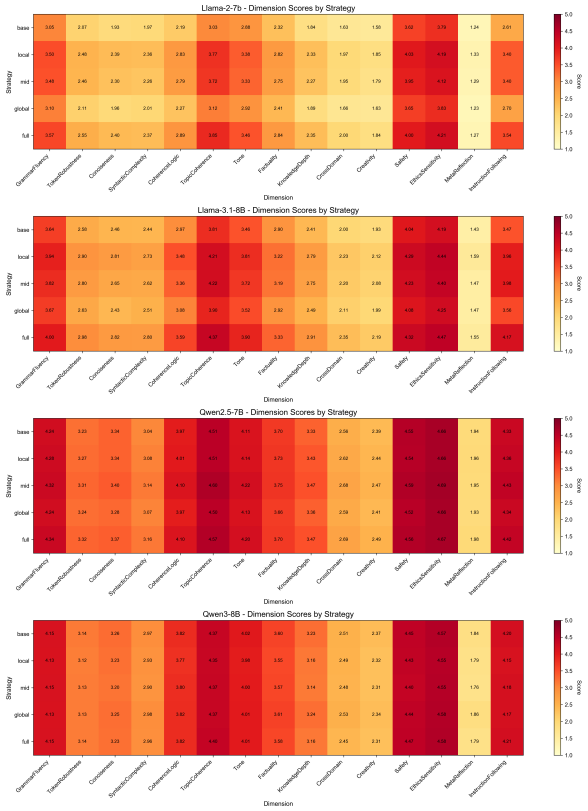

Figure 5: Heatmap of average Qwen-judge scores across 15 evaluation dimensions, grouped by model and strategy. Color intensity reflects score magnitude (light yellow = low, dark red = high). Local sharpens lexical/fluency dimensions; Mid moves logical and coherence dimensions the most.

Seed	Final train loss
11	0.6800
22	0.6750
33	0.6774
44	0.6760
55	0.6787
Mean±std	0.6774 ± 0.0019

Stage-2 held-out main-test results (Qwen3-Max 16-dim Overall, `hh-test-main`). For held-out reporting, we transfer each model's Stage-1 training-loss-minimum window (Qwen2.5-7B: `win10-18`, identical to the original Mid block; Llama-3.1-8B: `win06-16`) and evaluate it alongside the relevant baselines. We additionally report pre-specified middle-neighbourhood probes to show local stability around the middle region; these probe windows are not training-loss-selected candidates and are not used as model-selection signals.

Table 5: Pooled-across-models aggregate Overall by strategy ($N = 4{,}000$).

Strategy	*Mean*	*Std*	*Median*	*Count*
Base	2.765	1.340	3.000	4,000
Local	3.013	1.269	3.000	4,000
Mid	3.002	1.273	3.000	4,000
Global	2.808	1.338	3.000	4,000
Full	3.067	1.253	3.000	4,000

Table 6: Top 10 model-strategy combinations by mean Overall ($N = 1{,}000$).

Rank	*Model*	*Strategy*	*Mean*
1	Qwen2.5-7B	Mid	3.534
2	Qwen2.5-7B	Full	3.514
3	Qwen2.5-7B	Local	3.462
4	Qwen2.5-7B	Base	3.414
5	Qwen2.5-7B	Global	3.413
6	Qwen3-8B	Global	3.329
7	Qwen3-8B	Base	3.306
8	Qwen3-8B	Mid	3.270
9	Qwen3-8B	Full	3.264
10	Qwen3-8B	Local	3.240

Strategy	n	**Overall**	**SE**
Llama-3.1-8B			
`win03-13`†	998	**2.766**	0.038
`win06-16`‡	998	2.709	0.039
Full	997	2.694	0.038
Random ($s = 33$)	999	2.593	0.039
Mid (original)	998	2.403	0.037
Base	999	2.071	0.036
Global	999	1.872	0.034
Qwen2.5-7B			
`win09-17`†	998	**3.054**	0.039
Mid (original)‡	999	3.037	0.039
`win06-14`	999	3.035	0.040
Full	999	3.030	0.039
Random ($s = 33$)	999	3.023	0.040
Base	999	2.973	0.040

‡ Stage-1 training-loss minimum (Llama: `win06-16`; Qwen: original Mid block = `win10-18`). † Middle-neighbourhood probe: a window near (but not equal to) the training-loss minimum, included for robustness; *not* a training-loss-selected candidate and *not* used as a model-selection signal. Bold marks the highest held-out Overall per model for descriptive reporting only; selection follows the Stage-1 training-loss criterion (‡ rows), not the held-out maximum.

G Evaluation Validation: Full Tables

This appendix reports the cross-judge pairwise, IFEval, AlpacaEval 2 LC, and paired-bootstrap CI numbers behind Figure 2 and the triangulation paragraphs in §4.4.

Pairwise A/B-swap evaluation (non-Qwen judges). Order-swapped pairing; Δ = (Mid win) − (Other win) in percentage

points. Cross-judge directional agreement (DeepSeek vs. GPT-5.5) is $6/6 = 100\%$ across model$\times$comparison cells.

Paired-bootstrap 95% CIs (Qwen3-Max Overall, `hh-test-main`). $\Delta = \text{mean}(A) - \text{mean}(B)$; $\star$ = 95% CI excludes zero. R1 = Stage-2 main evaluation; R2 = independent re-train + re-infer + re-score on Qwen2.5-7B.

Round-2 independent replication (Qwen2.5-7B).

Strategy	n	Overall	SE
Mid	999	**3.060**	0.039
Full	999	3.010	0.040
Local	999	2.996	0.040
Base	999	2.970	0.041
Global	997	2.910	0.040

IFEval (541 prompts; values in %). No trained strategy strays more than 1–2 pp from Full; Mid is within 1.1 pp of Full on every metric variant for both models (worst case: Qwen Loose-P, Full 36.6 vs. Mid 35.5), and the Llama training-loss-selected window `win06-16` is within 0.5 pp of Full. The Qwen `win09-17` row is included as a middle-neighbourhood probe (not a training-loss-selected candidate) and shown for completeness.

AlpacaEval 2 Length-Controlled (805 prompts, `gpt-4o-mini` annotator vs. `text-davinci-003`). The Qwen LC column saturates at 50.00 because raw WR $\approx 50\%$ triggers AlpacaEval's L1-LR-CV maximum regulariser; this is a known conservative behaviour, not a bug.

H Robustness: Multi-Seed and Hyperparameter Grid

This appendix reports the multi-seed and 2×2 hyperparameter sweep behind the robustness paragraph in §4.5.

Training-seed robustness for Full and Mid (Qwen2.5-7B). Full and Mid are each re-trained with three independent training seeds; identical recipe otherwise. The Full $<$ Mid ordering does not flip on any seed; both strategies show per-seed standard deviation ≤ 0.0006.

Layer-subset variability for Random (Qwen2.5-7B). For the Random baseline, the three "seeds" below are *layer-selection* seeds (the training seed is fixed at 42). The resulting variance therefore reflects which 9 layers are sampled, not training stochasticity. Mid's per-training-seed std (0.0004) is one-third of Random's per-layer-subset std (0.0012), indicating that contiguous middle-block updates are more deterministic than random sparse layer choices, even after accounting for the different sources of variance.

2×2 hyperparameter grid by final training loss (Qwen2.5-7B). Both Full and Mid select the same optimum (lr$= 1 \times 10^{-5}$, $\beta = 0.1$), which coincides with our default recipe; the "default hyperparameters happen to favour Mid" alternative explanation is therefore inconsistent with the data. Full remains slightly better than Mid at every grid point (gap 0.002–0.003); the Mid$\approx$Full relation is therefore a stable property of the optimisation landscape, not a peculiarity of a single hyperparameter choice.

I HelpSteer2-Preference: Full Results

This appendix reports the full 8-variant HelpSteer2-Preference results behind the cross-dataset paragraph in §4.5. Training uses 6,766 non-tie preference pairs from `split=train`, `preference_strength`$\neq$`0`; evaluation uses 300 prompts from `split=val` (no overlap with training pairs) under the same Qwen3-Max 16-dimension rubric.

Training configuration deviations. HelpSteer2 prompts include multi-turn context and longer assistant placeholders than HH-RLHF, so per-device memory pressure is higher. We therefore keep the optimisation hyperparameters in Table 3 unchanged but adopt three reproducibility-relevant deviations, summarised in Table 14; the effective batch size remains 16, matching the HH-RLHF setup.

Final training loss (last-3-log average).

Variant	Qwen2.5-7B	Llama-3.1-8B
Full	**0.6621**	**0.6532**
Sliding-window	0.6671	0.6584
Mid	0.6689	0.6663
Random ($s = 33$)	0.6806	0.6721
Random ($s = 11$)	0.6827	0.6794
Random ($s = 22$)	0.6843	0.6661
Global	0.6929	0.6888

The Full $<$ {sliding-window, Mid} $<$ Random (mean) $<$ Global ordering observed on HH-RLHF largely transfers to HelpSteer2. Llama random seed 22 happens to land on a partially middle-overlapping subset (gap to Mid almost vanishes), consistent with the random-subset variability documented in Appendix F.

Table 7: Cross-judge pairwise A/B-swap evaluation. Bold Δ indicates a ≥ 5 pp net direction.

Model	Comparison	Judge	Mid win %	Other win %	Tie %	Δ (pp)
Qwen2.5-7B	Mid vs. Full	DeepSeek-v4-pro	29.1	26.6	44.3	$+2.5$
Qwen2.5-7B	Mid vs. Full	GPT-5.5	33.1	31.0	36.0	$+2.1$
Qwen2.5-7B	Mid vs. Global	DeepSeek-v4-pro	34.0	23.7	42.3	$\mathbf{+10.3}$
Qwen2.5-7B	Mid vs. Global	GPT-5.5	41.0	29.9	29.1	$\mathbf{+11.1}$
Qwen2.5-7B	Mid vs. Random	DeepSeek-v4-pro	30.3	22.6	47.1	$\mathbf{+7.7}$
Qwen2.5-7B	Mid vs. Random	GPT-5.5	34.5	28.2	37.3	$\mathbf{+6.3}$
Llama-3.1-8B	Mid vs. Full	DeepSeek-v4-pro	24.0	42.0	34.0	$\mathbf{-18.0}$
Llama-3.1-8B	Mid vs. Full	GPT-5.5	29.2	50.4	20.4	$\mathbf{-21.2}$
Llama-3.1-8B	Mid vs. Global	DeepSeek-v4-pro	49.0	17.9	33.1	$\mathbf{+31.1}$
Llama-3.1-8B	Mid vs. Global	GPT-5.5	68.8	17.9	13.3	$\mathbf{+50.9}$
Llama-3.1-8B	Mid vs. Random	DeepSeek-v4-pro	25.5	34.3	40.2	$\mathbf{-8.8}$
Llama-3.1-8B	Mid vs. Random	GPT-5.5	32.8	40.9	26.3	$\mathbf{-8.1}$

Table 8: Paired-bootstrap 95% CIs on `hh-test-main`.

Round	Model	Comparison	n	Δ	95% CI	Sig.
R1	Llama-3.1-8B	Mid vs. Full	996	-0.292	$[-0.365, -0.222]$	$\star$
R1	Qwen2.5-7B	Mid vs. Full	999	$+0.007$	$[-0.045, +0.058]$	—
R2	Qwen2.5-7B	Mid vs. Full	999	$+0.050$	$[-0.004, +0.104]$	—
R1	Llama-3.1-8B	Mid vs. Global	997	$+0.532$	$[+0.459, +0.606]$	$\star$
R2	Qwen2.5-7B	Mid vs. Global	997	$+0.148$	$[+0.087, +0.209]$	$\star$

Table 9: IFEval strict/loose prompt- and instruction-level accuracy on 541 prompts. Bold = best per column within each model.

Model + Strategy	Strict P	Strict I	Loose P	Loose I
Qwen2.5-7B Base	28.3	39.8	35.5	47.0
Qwen2.5-7B Full	28.1	39.3	36.6	47.6
Qwen2.5-7B Mid	28.5	39.6	35.5	46.6
Qwen2.5-7B `win09-17`	**29.6**	**41.0**	**37.5**	**48.4**
Llama-3.1-8B Base	23.1	32.5	26.1	35.4
Llama-3.1-8B Full	22.2	31.9	24.9	34.5
Llama-3.1-8B Mid	22.9	32.3	25.9	35.0
Llama-3.1-8B `win06-16`	22.7	31.7	25.9	34.5

Table 10: AlpacaEval 2 raw and length-controlled win rates.

Model + Strategy	Raw WR %	LC v1 %	Avg len
Llama-3.1-8B Base	5.79	3.50	6292
Llama-3.1-8B Full	7.69	**5.21**	5807
Llama-3.1-8B `win06-16`	**7.74**	5.03	5988
Llama-3.1-8B Mid	7.00	4.19	6097
Llama-3.1-8B Global	5.44	3.24	6417
Qwen2.5-7B Mid	**52.23**	50.00	2832
Qwen2.5-7B Full	50.97	50.00	2959
Qwen2.5-7B Global	50.68	50.00	3041

Table 11: Final DPO training loss on Qwen2.5-7B for Full and Mid across three training seeds.

Strategy	Seed 13	Seed 42	Seed 77	Mean	Std
Full	0.6819	0.6827	0.6816	**0.6821**	0.0006
Mid	0.6844	0.6847	0.6840	0.6844	0.0004

Table 12: Final DPO training loss on Qwen2.5-7B for Random across three layer-selection seeds (training seed fixed at 42).

Strategy	Layer-seed 11	Layer-seed 22	Layer-seed 33	Mean	Std
Random	0.6874	0.6889	0.6865	0.6876	0.0012

Table 13: 2×2 lr$\times\beta$ grid: final DPO training loss on the 10k HH-RLHF train subset for Qwen2.5-7B (values are the last-logging-step training loss). Bold = grid optimum for each row.

	$\beta=0.05$		$\beta=0.1$	
Strategy	**lr=5e-6**	**lr=1e-5**	**lr=5e-6**	**lr=1e-5**
Full	0.6901	0.6799	0.6875	**0.6726**
Mid	0.6908	0.6829	0.6881	**0.6754**

Table 14: HelpSteer2 training-configuration deviations from Table 3. All other hyperparameters (lr, β, LoRA r/α, target modules, precision, decoding) are unchanged.

Item	HH-RLHF default	HelpSteer2
Per-device batch size	4	2
Gradient accumulation	4	8
Gradient checkpointing	Disabled	Enabled
Effective batch size	16	16

Validation Qwen3-Max 16-dim Overall (300 held-out prompts). DPO on HelpSteer2 does *not* raise absolute Qwen3-Max scores above the base model: Qwen `full−base` $\Delta = -0.163^{\star}$ ($[-0.263, -0.063]$); on Llama, Base and Global tie at the top.

Paired-bootstrap 95% CIs (HelpSteer2 val Overall). $\star$ = 95% CI excludes zero.

Discussion of the Mid-vs-Global reversal. On HelpSteer2, both models show Mid − Global < 0 with CI excluding zero, reversing the HH-RLHF direction. A plausible interpretation: HelpSteer2 prompts (more technical, writing-focused, longer context) differ stylistically from HH-RLHF; the middle-block adaptation learned on HelpSteer2 produces a style drift that is penalised under Qwen3-Max's general-helpfulness rubric, while Global-only (deep-layer-only updates) preserves more of the base model's general behaviour. We therefore treat HelpSteer2 as evidence for the *relative* layer-allocation pattern (training loss: Full $<$ window/Mid $<$ Random $<$ Global; generation: Mid≈Full) rather than for an absolute generation-quality gain.

J Mechanistic Analysis: Per-Layer Numerics

This appendix reports the per-layer numerical values aggregated into Figure 3, together with the instrumentation used to collect them.

Instrumentation. For each model in {Qwen2.5-7B, Llama-3.1-8B} and each strategy in {Full, Mid}, we re-train the standard recipe with the following options enabled:

- `log_grad_rms=True`: in `TrainerCallback.on_pre_optimizer_step`, the RMS of all LoRA gradients is recorded per layer and per projection in $\{q, k, v, o\}$. The first 100 steps are logged, sampled every 5 steps (20 records). Each layer's gradient RMS is then averaged over the 20 records and over the four projections.
- `export_lora_norm=True`: after training, $\|\Delta W_l\|_F$ with $\Delta W = (\alpha/r)BA$ is exported per layer, summing the q, k, v, o submodules.
- Representation drift: 256 prompts from `hh-dev-select` are run through the base and aligned models; the last-token hidden state is extracted at every decoder layer and the per-layer cosine drift $1\text{–}\cos(h_l^{\text{base}}, h_l^{\text{aligned}})$ is averaged over the 256 prompts.

Representation drift, averaged over depth thirds.

Region	Full drift avg	Mid-only drift avg
Qwen2.5-7B		
Early (L0–8)	0.00004	≈ 0
Mid (L9–17)	0.00013	0.00007
Late (L18–27)	0.00033	0.00030
Llama-3.1-8B		
Early (L0–9)	0.00023	≈ 0
Mid (L10–19)	0.00242	0.00086
Late (L20–31)	0.00304	0.00138

Under Mid-only training, the drift in the Late region propagates downstream despite no LoRA being injected past layer 17 (Qwen) or layer 19 (Llama). The extent of propagation is, however, model-dependent: Mid-only reaches roughly

Table 15: HelpSteer2-Preference validation Qwen3-Max 16-dim Overall (300 held-out prompts). Bold = best per column.

	Qwen2.5-7B		Llama-3.1-8B	
Variant	**Overall**	**SE**	**Overall**	**SE**
Base	**3.067**	0.075	**1.680**	0.058
Full	2.903	0.075	1.613	0.056
Mid	2.880	0.076	1.582	0.053
Global	3.047	0.075	1.680	0.060
Sliding-window	2.933	0.076	1.577	0.055
Random ($s=11$)	2.967	0.080	1.653	0.058
Random ($s=22$)	3.013	0.075	1.649	0.054
Random ($s=33$)	3.007	0.077	1.595	0.054

Table 16: HelpSteer2 paired-bootstrap 95% CIs on validation Qwen3-Max Overall. Sample size $n \approx 300$ for all rows (exact $n = 298$–300).

Comparison ($A-B$)	**Model**	Δ	**95% CI**	**Sig.**
Mid − Full	Qwen2.5-7B	-0.023	$[-0.110, +0.063]$	—
Mid − Global	Qwen2.5-7B	-0.171	$[-0.268, -0.074]$	$\star$
window − Full	Qwen2.5-7B	$+0.030$	$[-0.067, +0.127]$	—
window − Mid	Qwen2.5-7B	$+0.053$	$[-0.040, +0.147]$	—
Full − Base	Qwen2.5-7B	-0.163	$[-0.263, -0.063]$	$\star$
Mid − Random	Qwen2.5-7B	-0.127	$[-0.217, -0.033]$	$\star$
Mid − Full	Llama-3.1-8B	-0.033	$[-0.107, +0.040]$	—
Mid − Global	Llama-3.1-8B	-0.100	$[-0.191, -0.010]$	$\star$
window − Full	Llama-3.1-8B	-0.037	$[-0.120, +0.047]$	—
window − Mid	Llama-3.1-8B	-0.003	$[-0.077, +0.070]$	—
Full − Base	Llama-3.1-8B	-0.067	$[-0.150, +0.013]$	—
Mid − Random	Llama-3.1-8B	-0.023	$[-0.097, +0.047]$	—

$0.91\times$ of Full-DPO Late-region drift on Qwen2.5-7B but only about $0.45\times$ on Llama-3.1-8B. The middle's effect therefore propagates through the residual stream rather than requiring direct intervention, but the strength of this propagation tracks the behavioural Mid≈Full vs. Full>Mid difference between the two models.

Three-signal summary. The Llama late-layer LoRA-norm cluster is the only model-specific deviation; it correlates with the surviving Full > Mid gap in generation quality (Figure 2) and suggests Llama distributes preference learning across more layers than Qwen.

Released artefacts. The per-step per-layer time series of gradient RMS, the end-of-training LoRA ΔW Frobenius norms, and the per-layer cosine drift tensors are released under `results_v2/exp5_mechanism/` together with the figure-generation script for full reproducibility of Figure 3.

K TruthfulQA Auxiliary Results

We run TruthfulQA (Lin et al., 2021) as an auxiliary factuality probe on Qwen2.5-7B. 817 multiple-choice questions are scored in zero-generation log-likelihood mode (MC1, MC2); no LLM generation or LLM judge is involved.

Strategy	**MC1**	**MC2**	**Δ MC1 vs. Base [95% CI]**
Base	0.3097	0.4207	—
Local	0.3146	0.4205	$+0.0049$ $[-0.0024, +0.0122]$
Mid	**0.3158**	0.4203	$+0.0061$ $[-0.0012, +0.0135]$
Global	0.3133	0.4203	$+0.0037$ $[-0.0012, +0.0098]$
Full	0.3146	0.4198	$+0.0049$ $[-0.0012, +0.0122]$

All MC1 deltas vs. Base are below 0.007 and all 95% CIs contain zero. None of the trained strategies significantly improves or harms factual accuracy under TruthfulQA's zero-generation evaluation, consistent with the observation that HH-RLHF preference learning is primarily about helpfulness and style rather than factual knowledge.

L SimPO Robustness Check

To check whether the Mid ≈ Full closeness on Qwen2.5-7B is specific to the DPO objective, we retrain Qwen2.5-7B with SimPO (Meng et al., 2024) (reference-free, length-normalised log-likelihood with a margin γ). Recommended SimPO hyperparameters are used: lr$= 5\times10^{-7}$, $\beta = 2.0$, $\gamma = 1.0$;

Table 17: Three mechanistic signals aggregated by model. All three signals concentrate in middle/middle-adjacent layers on both models.

Signal	**Qwen2.5-7B**	**Llama-3.1-8B**
Per-layer gradient RMS (Full-DPO, first 100 steps)	Clear peak at layers 14–17; late layers substantially lower	High plateau from early through middle layers; sharp drop after layer 21
LoRA ΔW Frobenius norm (end of Full-DPO)	Clean middle peak	Middle peak plus secondary spike at layers 28–31
Per-layer cosine drift (Full vs. Mid)	Rises from middle and propagates monotonically downstream	Jumps at the mid-layer entry and propagates downstream

LoRA configuration is identical to the DPO experiments. Both Full and Mid are trained on the same HH-RLHF 10k pairs.

Qwen3-Max 16-dimension Overall.

Configuration	n	**Mean**	**Std**
DPO-Full	999	3.010	1.257
DPO-Mid	999	**3.060**	1.239
SimPO-Full	999	2.954	1.258
SimPO-Mid	997	2.934	1.290

Paired-bootstrap 95% CIs. $\star$ = 95% CI excludes zero.

Comparison ($A-B$)	Δ	**95% CI**	**Sig.**
DPO-Mid − DPO-Full	$+0.050$	$[-0.004, +0.104]$	—
SimPO-Mid − SimPO-Full	-0.020	$[-0.067, +0.029]$	—
SimPO-Mid − DPO-Mid	-0.127	$[-0.189, -0.067]$	$\star$
SimPO-Full − DPO-Full	-0.056	$[-0.120, +0.005]$	—

Sample size $n \approx 999$ for all rows.

DeepSeek pairwise A/B-swap (tie inclusive).

Comparison (A **vs.** B)	A **win %**	B **win %**	**Tie %**
SimPO-Mid vs. SimPO-Full	22.6	25.2	52.2
SimPO-Mid vs. DPO-Mid	25.0	34.4	40.6
SimPO-Full vs. DPO-Full	29.8	33.2	37.0

Sample sizes: $n = 712$–762 depending on row.

Takeaway. SimPO-Mid is not significantly different from SimPO-Full under paired-bootstrap 95% CIs (CI contains zero; DeepSeek tie rate 52.2%), reproducing the Mid $\approx$ Full closeness observed under DPO; we treat this as failure to reject the null rather than positive equivalence. SimPO is overall slightly weaker than DPO under our recipe (SimPO-Mid − DPO-Mid $\Delta = -0.127^{\star}$), but the *relative* Mid-vs-Full relation is preserved across both objectives, suggesting the closeness is an architectural property of preference adaptation rather than an artefact of the DPO loss.