

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

Yishen Chen*

The Chinese University of
Hong Kong, Shenzhen
yishenchen@link.cuhk.edu.cn

Qi Dong*

Fudan University
19210980065@fudan.edu.cn

Yukun Zhang*

The Chinese University of
Hong Kong
215010026@link.cuhk.edu.cn

Abstract

Preference tuning typically distributes adapter updates across every Transformer layer. We study how the placement of a fixed LoRA budget affects Direct Preference Optimization (DPO). *Hierarchical Alignment* partitions decoder depth into *Local*, *Intermediate*, and *Global* regions and restricts LoRA updates to a chosen contiguous block. Across Qwen and Llama models, three-seed sliding-window sweeps reveal broad low-loss basins around intermediate depth. Under exact trainable-parameter matching, Middle achieves lower held-out DPO loss than an all-depth allocation on Qwen2.5-7B and Llama-3.1-8B. Generation-level evidence is more variable. LLM judges and a blinded human study leave Selected versus All unresolved on Qwen; on Llama, two pairwise judges favor All while the human interval includes zero. Trained-adapter interventions show that learned depth components affect the optimization objective; removing the Llama Selected window also lowers generation quality. Across Qwen2.5 at 3B, 7B, and 14B, Middle versus All remains unresolved under Qwen3-Max; at 3B and 14B, both improve over Base. In our implementation, Middle also uses less peak memory and takes shorter steps than a parameter-matched all-depth allocation. Together, the results support intermediate depth as a practical starting point for fixed-budget preference tuning, though the best window and its behavioral value still need per-setting validation.

1 Introduction

Direct Preference Optimization (DPO) provides a stable objective for aligning Large Language Models (LLMs) with preference data (Rafailov et al., 2023). Parameter-efficient implementations commonly place LoRA adapters at every Transformer layer. This convention leaves the depth allocation

of the trainable budget largely unexamined. Probing studies find substantial variation in representations across depth (van Aken et al., 2019; Nadipalli, 2025), motivating a controlled test of whether preference optimization is also sensitive to adapter placement.

We ask: *where should a fixed LoRA budget be placed for preference tuning?* *Hierarchical Alignment* divides decoder depth into three positional regions—*Local* (shallow), *Intermediate* (middle), and *Global* (deep)—and applies LoRA-DPO (Hu et al., 2022) to a selected block. These labels specify depth ranges and remain agnostic about exclusive linguistic or reasoning functions. We evaluate the depth-thirds partition with sliding windows (three training seeds each), exact-capacity all-depth controls, blinded human evaluation, trained-adapter interventions, a 3B/7B/14B scale study, external LLM judges, and efficiency profiling. This design separates *where* capacity is placed from *how much* is available.

Findings. Three-seed window sweeps produce broad low-loss basins around intermediate depth for Qwen2.5-7B and Llama-3.1-8B, with model-specific minima and frozen representative windows. Exact-capacity all-depth adapters have higher held-out DPO loss than the Intermediate block on both models. The objective-level difference between the selected window and the larger All configuration is unresolved. Behavioral results depend on the model. On Qwen2.5-7B, all four evaluators—three LLM judges and a blinded human panel—leave Selected versus All unresolved. On Llama, the two pairwise judges favor All; the human point estimate agrees but its interval includes zero. At 3B and 14B, both Middle and All improve over Base under Qwen3-Max and remain unresolved against each other. Adapter removals affect the objective; removing Llama’s Selected window also lowers generation quality.

*Equal contribution.

Contributions.

- We formulate preference tuning as a fixed-budget layer-allocation problem and implement contiguous-block LoRA-DPO with a coarse-to-fine depth search (§3).
- We separate objective-level and behavioral evidence using multi-seed windows, an exact-capacity all-depth baseline, external and human judges, adapter interventions, and a 3B/7B/14B scale study (§4).
- We report measured parameter, adapter, optimizer-state, peak-memory, and step-time costs, together with layer-wise gradients, updates, and representation drift (§5).

The evidence supports intermediate depth as a practical candidate region for fixed-budget LoRA-DPO, most clearly in the optimization objective and the matched-capacity comparison. Whether this objective advantage translates to better generations must be evaluated separately for each model scale, evaluator, and baseline allocation.

Three comparisons organize the study. Middle versus Matched isolates *placement* while holding the trainable parameter count exactly fixed. Selected versus All compares the window returned by the frozen search rule with the conventional, higher-capacity all-layer recipe. Middle versus Base at 3B, 7B, and 14B measures whether the restricted allocation produces a useful preference-tuning effect across scale. Separating these comparisons keeps placement at fixed capacity, the search outcome, and behavioral utility across scale from being treated as one claim.

The experimental workflow follows the same separation. Teacher-forced DPO loss on a frozen development split selects a window and quantifies objective effects across three training seeds. Generation from the seed-42 checkpoints is then evaluated on disjoint prompts by multidimensional, pairwise, programmatic, and human evaluators. Trained-adapter interventions test whether the learned depth allocation contributes to the observed effects. The resulting evidence supports a practical search procedure and shows where objective and behavioral rankings diverge.

2 Related Work

Our work sits at the intersection of preference alignment for LLMs and layer-specific, parameter-efficient tuning.

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., 2023) and SimPO (Meng et al., 2024). Recent work also studies how preference-data consistency affects multilingual DPO (Zhang et al., 2025). These methods typically apply the alignment loss to all trainable layers without examining how the update budget is distributed across depth. We treat this distribution as an explicit design variable.

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 identifies random sparse updates as a strong baseline for informed selection.

Single-layer or single-head steering. CogSteer (Wang et al., 2025) steers behavior 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 behavioral control without modifying the base model. LoRA (Hu et al., 2022) provides the parameter-efficient mechanism we adopt; its formulation is agnostic to which layers 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. Related depth-localized effects have also been observed for factual recall in Mamba (Sharma et al., 2024). These results motivate our depth partition. Unlike probing studies, our experiments directly test whether preference learning is sensitive to where updates are placed across depth.

2.3 Positioning

Our contribution is a controlled study of *preference alignment* under contiguous block-level updates and matched LoRA budgets. Through sliding windows, exact-capacity controls, adapter interventions, and a scale study, we characterize intermediate depth as a useful search prior and test how well its objective advantage carries over to behavioral outcomes.

Our setup also differs from sparse-update methods that choose a new random or learned subset during optimization. Each allocation in our study is fixed before training, uses the same DPO data and optimizer recipe, and produces an independently evaluable adapter. Contiguous windows preserve the network’s depth order and make location interpretable through a single coordinate. The all-depth Matched control then separates the value of concentration in one window from the value of merely reducing trainable capacity. This design brings together parameter-efficient tuning, layer-wise analysis, and preference evaluation in a single controlled setting.

3 Methodology

3.1 Layer-Allocation Hypotheses

We use decoder depth as the coordinate for allocating a fixed trainable budget.

Definition 1 (Depth Partition). *Let a Transformer model contain N ordered layers, $\mathcal{T} = \{L_1, L_2, \dots, L_N\}$. A **depth partition** divides these layers into K disjoint contiguous blocks, $\Pi = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K\}$, with $\mathcal{S}_k = \{L_i : i \in I_k\}$. The blocks preserve the model’s computational order, $F(\mathbf{x}; \Theta) = f_K \circ f_{K-1} \circ \dots \circ f_1(\mathbf{x})$.*

We test two empirical hypotheses.

Hypothesis 1 (Depth-Sensitive Allocation). *For a fixed LoRA parameter budget and training recipe, the DPO objective varies with the depth of a contiguous target window. A coarse depth partition can therefore narrow the search space for finer window selection. We test this in Sections 4.2–4.3 with three training seeds and exact-capacity controls.*

Hypothesis 2 (Objective–Behavior Relationship).

For a preference loss ℓ_m , optimization activity may concentrate in particular depth ranges. We quantify this activity as

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

and ask whether objective differences across depth carry over to generation preferences. Because objective and behavioral rankings can diverge, we evaluate them separately. Section 5 reports trained-adapter interventions alongside gradient RMS, LoRA update norms, and representation drift.

3.2 Implementation

3.2.1 Positional Block Partitioning

Following Hypothesis 1, we define Π using three positional blocks:

- **Local Block** ($\mathcal{S}_{\text{local}}$): the initial one-third of layers.
- **Intermediate Block** ($\mathcal{S}_{\text{mid}}$): the middle one-third of layers.
- **Global Block** ($\mathcal{S}_{\text{global}}$): the final one-third of layers.

These names are positional shorthand; they do not imply that each block serves an exclusive functional role. A deterministic rule distributes N layers as evenly as possible across the three blocks (exact indices in Appendix C.3). Section 4.2 refines the coarse partition with overlapping windows.

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:

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

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)

We use LoRA to confine parameter updates to a chosen block S_k , treating it as a **subspace selector** for the depth-sensitive allocation in Hypothesis 1.

We freeze the base model and inject trainable low-rank matrices into the self-attention projections of layers in S_k only, leaving all FFN weights frozen. Optimization updates the LoRA parameters $\Theta_{k,\text{LoRA}}$:

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

3.2.4 Fixed-Budget Allocations

An allocation specifies a set of adapted layers I and a rank r_i for each $i \in I$. All experiments target the q , k , v , and o attention projections and keep $\alpha_i/r_i = 2$. **Middle** denotes the allocation that adapts the Intermediate block at rank 16. **Selected** assigns rank 16 to the representative window returned by the procedure below. **All** assigns rank 16 to every layer and serves as the conventional higher-capacity reference.

Matched spreads Middle’s aggregate rank across every layer. For Qwen2.5-7B, Middle adapts nine layers and has rank-sum $9 \times 16 = 144$; Matched uses rank 5 in 24 layers and rank 6 in layers 3, 10, 17, and 24. For Llama-3.1-8B, Middle adapts eleven layers and has rank-sum $11 \times 16 = 176$; Matched alternates ranks 5 and 6 across the 32 layers. These constructions yield exactly 3.244M and 4.686M trainable parameters for both members of each Middle–Matched pair. Automated checks verify the targeted projections, layer indices, and parameter counts before each training run.

3.2.5 Coarse-to-Fine Window Selection

The depth thirds provide a model-independent coarse partition. We refine the middle candidate with overlapping windows whose width equals the model’s middle-third width. Every window is trained with seeds 13, 42, and 77 and ranked by mean teacher-forced dev DPO loss. The window with the lowest mean loss serves as the basin reference. Window i belongs to the low-loss basin if its 95% seed- t interval overlaps the reference interval or if $|\bar{\ell}_i - \bar{\ell}_{\min}| < s_{\text{pool}}$, where $s_{\text{pool}} = \sqrt{(s_i^2 + s_{\min}^2)/2}$. Among basin members, the frozen representative is the one whose center is closest to the fixed middle-third center, with lower

mean loss as a tiebreaker. Preference accuracy, implicit margin, and chosen-token KL provide complementary diagnostics but do not override the loss criterion. Generation results play no role in window selection and are computed only after the representative has been frozen.

This rule identifies a stable region despite seed variability and gives the fixed depth-thirds prior a concrete operational role: Middle is the default candidate, and the window scan provides a lightweight adjustment when neighboring windows fall in the same loss basin. Section 4.2 reports the full three-seed results, and Appendix D lists every window.

4 Experiments

4.1 Experimental Setup and Evaluation Metrics

Models and training. The controlled study uses Qwen2.5-7B (Qwen et al., 2025) and Llama-3.1-8B (Grattafiori et al., 2024) as reference models and adds Qwen2.5-3B/14B for scale. Models train for one epoch on 10,000 Anthropic HH-RLHF pairs (Bai et al., 2022) with learning rate 10^{-5} , $\beta = 0.1$, effective batch size 16, and bfloat16 precision. LoRA adapts the $\{q, k, v, o\}$ attention projections ($r = 16$, $\alpha = 32$, dropout 0.05). Maximum prompt and total sequence lengths are 1,024 and 2,048 tokens; training seeds are 13, 42, and 77. Generation uses greedy decoding from the seed-42 checkpoint with a 512-token cap. Appendix C.1 records the complete software and training configuration.

Controlled-pipeline integrity. All headline results use a corrected data pipeline that preserves exact prompt–completion boundaries under the corresponding tokenizer (Appendix B). Earlier single-seed artifacts from a prior pipeline remain in the exploratory appendix for provenance but do not enter the controlled comparisons.

Splits and uncertainty. Frozen, disjoint subsets contain 10,000 training pairs, 300 dev prompts for teacher-forced DPO loss and window selection, 1,000 main-test prompts for generation, and 180 human-pool prompts. Row indices are unique within each split, and prompt text is verified before paired evaluation. Objective effects use 95% seed- t intervals over the three training seeds. Generation effects use paired prompt bootstrap intervals; human intervals resample prompts so that the three votes for one prompt remain clustered. An interval

containing zero is reported as unresolved. Split-integrity checks and valid-count accounting appear in Appendix A.

Teacher-forced evaluation. On the frozen 300-prompt dev set, we compute DPO loss, preference accuracy, implicit margin, and chosen-token $\text{KL}(\pi_\theta \parallel \pi_{\text{ref}})$ from each loaded checkpoint. DPO loss is the selection criterion because it directly matches the training objective; the remaining metrics diagnose whether a lower loss coincides with consistent preference movement. For each checkpoint, the policy pass enables the adapter while the reference pass disables it, keeping tokenization and base weights fixed.

Behavioral evaluation. Qwen3-Max scores 16 dimensions on a 1–5 scale, grouped into language quality, logic and coherence, content quality, safety, task execution, and meta-level utility. DeepSeek-v4-pro and GPT-5.5 provide pairwise judgments with mandatory A/B order swapping. IFEval (Zhou et al., 2023), AlpacaEval 2 Length-Controlled (Dubois et al., 2024), and TruthfulQA (Lin et al., 2022) add programmatic and auxiliary checks. Empty, truncated, and failed generations remain in the denominators of all reported rates. Full endpoint and prompt details appear in Appendix B.

4.2 Multi-Seed Window Search

We train eight overlapping windows per reference model at a fixed parameter budget: nine layers for Qwen2.5-7B and eleven for Llama-3.1-8B. Qwen starts are $\{0, 3, 6, 9, 10, 13, 16, 19\}$ and Llama starts are $\{0, 3, 6, 9, 11, 14, 17, 21\}$. Every window uses the same data order, optimizer, rank, projections, and three training seeds. Applying the selection rule from §3, the procedure selects Qwen layers 10–18 and Llama layers 9–19 as frozen representatives; the minimum-mean windows are 13–21 and 6–16, respectively.

Both models exhibit a broad low-loss basin around intermediate depth, though the exact minimum varies. Qwen window means drop from 0.6919 (layers 0–8) to a minimum of 0.6753 (layers 13–21) before rising again; Llama’s minimum is 0.6559 (layers 6–16), with the two deepest windows rising above 0.681. Qwen’s basin contains windows 9–17, 10–18, and 13–21; Llama’s contains 3–13, 6–16, and 9–19. The center-nearest rule therefore returns the fixed Middle for Qwen and a middle-adjacent window for Llama. Across

seeds, the result supports a region-level prior with model-specific refinement.

Preference accuracy, implicit margin, and chosen-token KL provide complementary consistency checks but do not change the selected representative. Crucially, the selector was frozen before any generation took place: the behavioral test set supplies no search signal, and each judge evaluates a single predeclared checkpoint per model.

4.3 Matched-Capacity Objective Control

Matched allocates an aggregate rank of 144 across all 28 Qwen layers and 176 across all 32 Llama layers, exactly matching Middle’s 3.244M and 4.686M trainable parameters. Rank and alpha vary together so that $\alpha/r = 2$ in every adapted layer. The Qwen pattern uses rank 6 at four evenly spaced layers and rank 5 elsewhere; the Llama pattern alternates ranks 5 and 6. The result is an all-depth adapter with exactly the same trainable budget as Middle.

Middle has lower dev loss than Matched on Qwen ($\Delta = -0.0086$, 95% CI $[-0.0095, -0.0078]$) and Llama (-0.0057 , $[-0.0109, -0.0006]$). The interval excludes zero for both models, establishing a placement effect at fixed capacity. Selected versus the larger All configuration is unresolved on Qwen ($+0.0028$, $[-0.0010, +0.0066]$) and Llama ($+0.0036$, $[-0.0024, +0.0096]$). Concentrating capacity in the middle region thus improves the objective over an equal-capacity all-depth spread. The Selected–All gap remains unresolved with three seeds, even though All uses roughly three times as many trainable parameters.

4.4 Behavioral Validation

Table 1 reports behavioral effects alongside the objective results. On Qwen2.5-7B, Selected improves over Base by $+0.152$ on the Qwen3-Max Overall scale ($[+0.093, +0.212]$). Against All, Selected is unresolved across all four evaluators: Qwen3-Max (-0.020), DeepSeek-v4-pro (-0.043), GPT-5.5 (-0.029), and the human panel. In this setting, the behavioral gap between Selected and All is small relative to prompt-level uncertainty.

Llama tells a different story. DeepSeek-v4-pro and GPT-5.5 both favor All over Selected by 6.1 and 7.7 net-preference points, with intervals excluding zero; the human point estimate agrees, though its wider interval includes zero. Against the equal-capacity Matched control, Qwen Selected is fa-

All-window three-seed results and frozen held-out selection

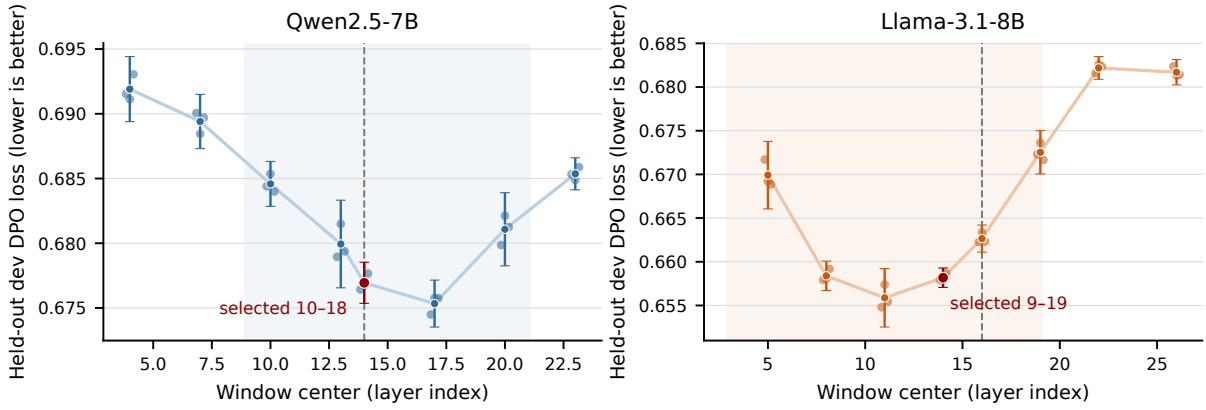


Figure 1: Held-out DPO loss for every contiguous window across seeds 13, 42, and 77. Small points are seed values; large points and bars show means and 95% seed- t intervals. Gray dashed lines mark the middle-third center; red points mark the frozen representative windows. Qwen’s low-loss basin contains 9–17, 10–18, and 13–21; Llama’s contains 3–13, 6–16, and 9–19.

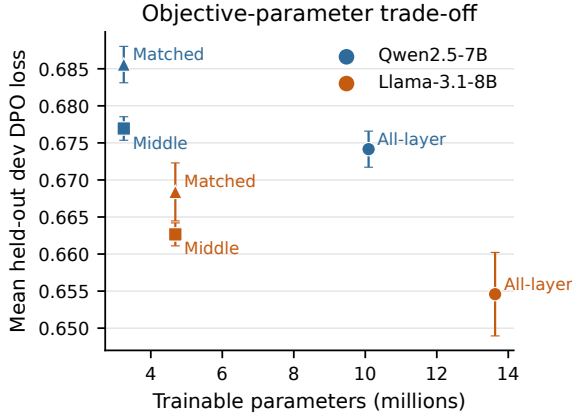


Figure 2: Mean held-out DPO loss versus trainable parameters; bars are 95% seed- t intervals. Middle improves on the exact-capacity all-depth Matched control for both models; All is the higher-capacity all-depth reference.

vored by DeepSeek (+0.081) but unresolved under GPT-5.5 and Qwen3-Max; Llama Middle is disfavored against Matched by both judges. IFEval differences are mostly within two percentage points; 27 of 28 intervals include zero, with one resolved Qwen2.5-14B Middle–Matched result on loose instruction accuracy (Appendix B.3). AlpacaEval 2 assigns All a descriptively higher length-controlled win rate on both models (Qwen 53.98 vs. 50.00; Llama 4.99 vs. 4.28), but the Qwen estimator places several conditions at its 50.00 regularization floor, so we draw no inferential conclusion.

Blind human evaluation. Three independent annotators compare 120 prompts per model, with ran-

domized A/B order, ties allowed, and six repeated quality-control prompts per annotator. Model family, allocation, and response orientation are hidden. Annotators read temperature-zero Chinese translations of each prompt and output; identical source strings always receive the same canonical translation. The comparison therefore measures preference over translated Chinese renderings. Formal estimates exclude the repeated prompts and retain one Llama item for which both outputs are empty as a tie.

Qwen yields 114 Selected wins, 100 ties, and 146 All wins; Llama yields 133 Selected wins, 69 ties, and 158 All wins. The tie-inclusive net preferences are -0.089 ($[-0.203, +0.028]$) and -0.069 ($[-0.197, +0.061]$), respectively. Fleiss’ κ is 0.401/0.514 and repeated-item consistency is 0.833/0.889 for Qwen/Llama. Three-way prompt-level agreement—exact agreement between the human-majority A/B/tie label and the LLM judge’s aggregated label—is 0.400/0.492 for Qwen and 0.583/0.658 for Llama under DeepSeek/GPT-5.5. The higher Llama agreement is consistent with the shared direction toward All across evaluators; on Qwen, evaluator judgments are more dispersed.

4.5 Scale and Efficiency

Qwen scale study. The same deterministic thirds give Middle layers 12–23 at 3B, 10–18 at 7B, and 16–31 at 14B. At 3B, dev loss favors All over Middle ($\Delta_{\text{Mid-All}} = +0.0072$, 95% CI $[+0.0035, +0.0109]$); at 14B the difference is unresolved ($+0.0014$, $[-0.0058, +0.0085]$). Qwen3-

Table 1: Representative controlled behavioral effects. Effects are net preference for the first allocation, except Qwen3-Max (difference on its 1–5 Overall score). * marks a 95% interval excluding zero. Complete judge and benchmark tables are in Appendix B.3.

Model	Comparison	Evaluator	Effect [95% CI]	<i>n</i>
Qwen2.5-7B	Selected – All	Qwen3-Max	−0.020 [−0.073, +0.031]	1000
Qwen2.5-7B	Selected – All	DeepSeek-v4-pro	−0.043 [−0.087, +0.002]	1000
Qwen2.5-7B	Selected – All	Human	−0.089 [−0.203, +0.028]	120
Llama-3.1-8B	Selected – All	DeepSeek-v4-pro	−0.061 [−0.110, −0.013]*	1000
Llama-3.1-8B	Selected – All	Human	−0.069 [−0.197, +0.061]	120
Qwen2.5-7B	Selected – Matched	DeepSeek-v4-pro	+0.081 [+0.035, +0.126]*	1000
Llama-3.1-8B	Middle – Matched	DeepSeek-v4-pro	−0.076 [−0.124, −0.030]*	1000

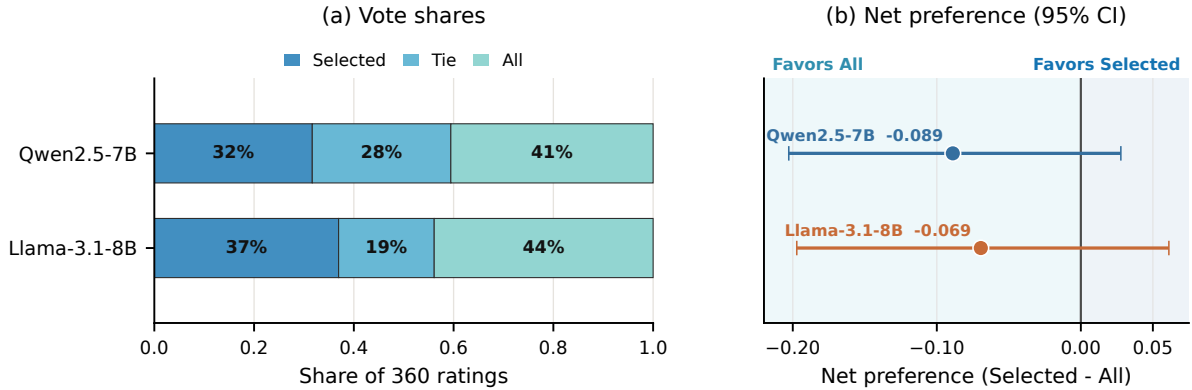


Figure 3: Blinded human comparison of the frozen Selected window against All. Left: fractions of 360 formal votes per model. Right: tie-inclusive net preference with 95% prompt-clustered bootstrap intervals. Both intervals include zero.

Max behavior is also unresolved between Middle and All at both scales: -0.005 $[-0.063, +0.053]$ at 3B and $+0.001$ $[-0.049, +0.051]$ at 14B. Middle and All each improve over Base at 3B ($+0.201/ +0.206$) and 14B ($+0.429/ +0.428$).

The 14B exact-capacity comparison adds a behavioral check on placement. Middle exceeds Matched by $+0.094$ ($[+0.031, +0.156]$), and All exceeds Matched by $+0.093$ ($[+0.033, +0.154]$), on 999 valid prompts. These point estimates are nearly identical, while the direct Middle–All difference remains unresolved. The equal-capacity all-depth distribution falls behind both. Across the three scales, the depth prior remains useful, though the precise objective gap relative to All varies by scale.

Measured efficiency. Appendix Table 15 reports the seed-42 NVIDIA H800 implementation profile over steps 50–600. Because Middle and Matched have the same trainable parameter count and nearly the same serialized adapter size, the runtime gap between them isolates the effect of placing adapters in a contiguous block versus spreading them across

every layer. Middle records 17.6–18.1 GiB lower peak memory, 18.6–19.2% shorter steps, and 22.8–23.7% higher throughput. All uses roughly three times the trainable parameters and optimizer-state storage of Middle. Every allocation still executes a full forward pass through the frozen base; the measured savings are specific to this implementation, hardware, and optimizer-state layout. Supplementary HelpSteer2, TruthfulQA, SimPO, and layer-wise diagnostics serve as exploratory evidence outside the headline comparisons.

5 Trained-Adapter Interventions

We intervene on the trained seed-42 All checkpoint by masking adapter contributions in selected depth blocks while keeping all other weights fixed. A *removal* masks one block’s adapters; a *retention* keeps only one block and masks the other two. Both operations are post-training, so they measure how much the trained model depends on adapter contributions from each depth region.

Removing Middle increases dev DPO loss on Qwen by $+0.0124$ ($[+0.0050, +0.0194]$) and on

Llama by $+0.0157$ ($[+0.0032, +0.0278]$); removing Llama’s selected 9–19 window increases it by $+0.0207$ ($[+0.0044, +0.0364]$). Among the three depth thirds, retaining only Middle preserves the largest fraction of All’s objective gain over Base: 0.765 for Qwen and 0.645 for Llama (Shallow: 0.157/0.497; Deep: 0.291/0.127). Retaining Llama’s frozen Selected window (layers 9–19) preserves an even larger fraction, 0.801.

Generation-level effects are less consistent. For middle-block removal, the Qwen3-Max Overall change is unresolved on both models. On Llama, however, removing the Selected 9–19 adapters reduces Overall by -0.258 ($[-0.395, -0.120]$), and removing the shallow block reduces it by -0.455 ($[-0.605, -0.311]$). Teacher-forced effects thus peak at middle or middle-adjacent depth, while removing Llama’s shallow block also lowers generation quality. Gradient RMS, update norms, and representation drift provide complementary correlational depth profiles (Appendix H.1).

6 Discussion and Conclusion

What the objective results establish. The multi-seed windows show that held-out DPO loss changes systematically with allocation depth. Their minima differ across Qwen and Llama, yet both low-loss basins cover middle or middle-adjacent windows. The matched-capacity controls strengthen this result: concentrating the same trainable parameter count in Middle improves the objective over distributing it across every layer. The Selected–All loss differences remain unresolved with three seeds, which puts allocation and capacity on the same practical scale but does not imply that the two strategies are equivalent.

What behavioral evaluation adds. A lower teacher-forced loss is a useful training signal and selection criterion, but generated responses introduce additional sources of variation: decoding strategy, prompt content, evaluator choice, and rubric design. Qwen Selected versus All remains unresolved across four evaluator views. Llama’s external judges favor All, with human votes pointing in the same direction at lower precision. At 3B and 14B, Middle and All each improve over Base and remain unresolved against each other under Qwen3-Max. These outcomes make behavioral validation an explicit second stage of the search procedure.

Interpreting depth. The positional labels provide a stable coordinate shared across architectures. Adapter removal and retention establish that the location of trained components affects the DPO objective, and selected Llama removals also affect generation. Gradient RMS, update norms, and representation drift show related depth structure, with a secondary late-layer update cluster in Llama. Together, these measurements show where preference-learning signals concentrate in the studied checkpoints and help explain why the same fixed window need not work across the models studied.

Practical workflow. A resource-constrained practitioner should first check whether all-layer DPO improves over Base on the target generation rubric. If it does, rank-16 Middle LoRA-DPO is a natural starting point. A small set of overlapping neighboring windows can then be trained on the same budget and screened for a low-loss basin using held-out preference data across seeds. The selected adapter should finally be compared with Base and the all-layer reference on the target generation rubric. An exact-capacity all-depth control is especially informative when the goal is to attribute gains to placement. This sequence keeps the layer search small while preserving a direct behavioral decision for deployment.

Conclusion. Hierarchical Alignment frames preference tuning as a layer-allocation problem. Multi-seed windows, exact-capacity controls, interventions, and scale results show that allocation changes the objective and can alter generated behavior. Intermediate depth provides a useful starting point for the search; the final window should be confirmed by behavioral evaluation on the target task.

7 Limitations

Our controlled experiments cover decoder-only Qwen models at 3B, 7B, and 14B and Llama at 8B. The main training corpus is HH-RLHF; HelpSteer2-Preference (Wang et al., 2024) serves as a smaller transfer check. A Qwen3-8B case from an earlier pipeline shows no detectable All-versus-Base behavioral headroom and is retained only as a boundary diagnosis, not as evidence about layer ordering (Appendix H.2). Broader architectures, model families, preference domains, and languages may yield different depth profiles.

The objective analysis uses three training seeds, whereas generation and most judge evaluations use

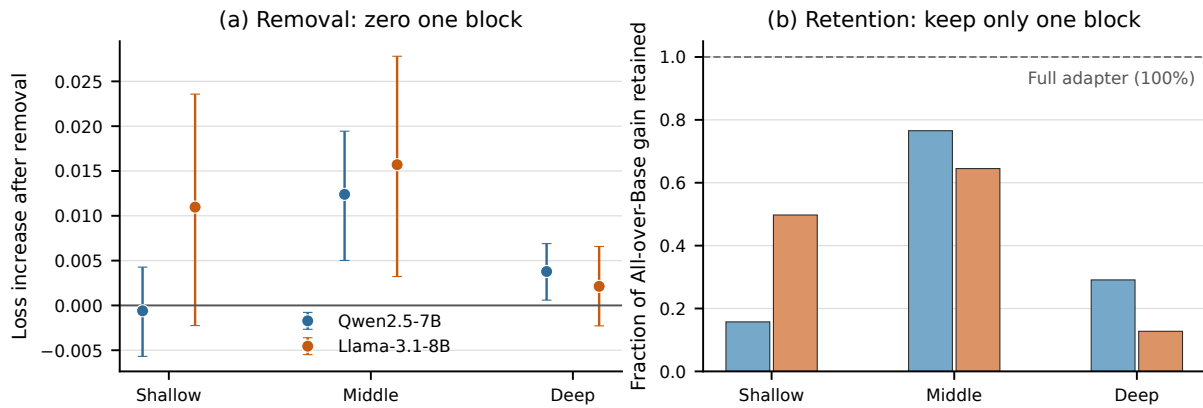


Figure 4: Trained-adapter interventions on the seed-42 All checkpoint. Within each panel, higher values indicate larger measured intervention effects. Left: dev-loss increase after removing one fixed depth-third block, with paired-prompt 95% intervals. Right: fraction of the All-over-Base dev-loss gain retained when only one fixed depth-third block remains. The Llama Selected 9–19 intervention is reported in the text and Appendix F.

the seed-42 checkpoints; prompt-level bootstrap intervals cover prompt uncertainty while leaving training-seed uncertainty unmeasured. Our frozen window selector uses held-out DPO loss, and objective and behavioral rankings sometimes diverge.

Qwen3-Max supplies the multidimensional score, while DeepSeek-v4-pro and GPT-5.5 provide pairwise checks. These proprietary judges limit reproducibility and may share unknown biases. The blinded human study covers 120 prompts per model comparison with three annotators; its intervals include zero and agreement is moderate. Because annotators judged canonical Chinese translations, these preferences cannot be equated directly with the English automated evaluations, and translation may alter nuance.

We do not compare against learned masks, gradient-selected subsets, or random layer subsets under the same LoRA-DPO budget. The Matched control isolates the effect of concentrating capacity versus spreading it across all layers, but does not test learned layer selection. The adapter-removal experiments establish dependence on trained components, while nonlinear interactions between adapter blocks limit functional localization claims. Efficiency measurements come from a single implementation and hardware environment. Middle uses fewer LoRA parameters than All and shows lower peak memory and shorter steps than the exact-capacity Matched control, but every model still runs a full forward pass through the frozen base, so a general systems-level speedup remains unestablished.

8 Ethical Considerations

HH-RLHF and HelpSteer2 may contain biases that preference tuning can amplify, and layer-selective tuning carries dual-use risks. We therefore report uncertainty and evaluator disagreement throughout, and pair proprietary judges with programmatic checks and blinded human evaluation.

All three authors, who are adults and native speakers of Chinese, served as annotators. No external participants were recruited or paid. Model identity, allocation, and response orientation were hidden during annotation, and results are reported without attributing labels to individual authors. Because the study involved no external human participants, no separate participant-consent procedure or institutional ethics review was sought.

Use of AI Assistants

AI assistants supported editing and consistency checks. Temperature-zero deepseek-chat produced the Chinese translations used in the human study but did not generate any labels (Appendix E). The authors retain full responsibility for the research.

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](#). *arXiv preprint arXiv:2204.05862*.

- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. [Length-controlled AlpacaEval: A simple way to debias automatic evaluators](#). *arXiv preprint arXiv:2404.04475*.
- Shangbin Feng, Taylor Sorensen, Yuhao Liu, Jillian Fisher, Chan Young Park, Yejin Choi, and Yulia Tsvetkov. 2024. [Modular pluralism: Pluralistic alignment via multi-LLM collaboration](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4151–4171.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The Llama 3 herd of models](#). *arXiv preprint arXiv:2407.21783*.
- 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](#). In *Advances in Neural Information Processing Systems*, volume 37.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [TruthfulQA: Measuring how models mimic human falsehoods](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. [SimPO: Simple preference optimization with a reference-free reward](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 124198–124235.
- Suneel Nadipalli. 2025. [Layer-wise evolution of representations in fine-tuned transformers: Insights from sparse autoencoders](#). *arXiv 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](#). In *Advances in Neural Information Processing Systems*, volume 35.
- 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](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 57018–57049.
- Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. [Qwen2.5 technical report](#). *arXiv preprint arXiv:2412.15115*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems*, volume 36.
- Arnab Sen Sharma, David Atkinson, and David Bau. 2024. [Locating and editing factual associations in Mamba](#). In *First Conference on Language Modeling*.
- Oscar Srean, 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.
- 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*, pages 1823–1832. ACM.
- Xintong Wang, Jingheng Pan, Liang Ding, Longyue Wang, Longqin Jiang, Xingshan Li, and Chris Bie-mann. 2025. [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. 2024. [HelpSteer2-Preference: Complementing ratings with preferences](#). *arXiv preprint arXiv:2410.01257*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.
- 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](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25689–25702.

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](#). *arXiv preprint arXiv:2311.07911*.

A Data Integrity and Split Roles

A.1 Data Roles and Integrity

The controlled study follows a protocol frozen before the formal run matrix. The training set contains 10,000 HH-RLHF preference pairs. Three mutually disjoint subsets drawn from the HH-RLHF test pool have fixed roles: 300 prompts for teacher-forced development metrics and window selection, 1,000 prompts for generation-based evaluation, and 180 prompts reserved for human evaluation. The stored row order is fixed. Every original row index is unique within its split, and disjointness is verified using both source row IDs and prompt text before paired evaluation.

B Generation and Judge Protocols

B.1 Judges, Endpoints, and Rubrics

All API evaluations were completed and audited on July 14–15, 2026 UTC. The absolute rubric uses the model identifier qwen3-max through Aliyun’s OpenAI-compatible endpoint. Pairwise evaluation uses deepseek-v4-pro through the official DeepSeek endpoint and gpt-5.5 through the ZYAI OpenAI-compatible relay. Each pairwise comparison uses one provider throughout, verifies prompt identity, presents both A/B orders, and retains ties. The final pairwise matrix has 1,000 valid prompts per row. Incomplete files and failed API rows are excluded; rows from different providers are never merged. No credential is stored with the results.

The 16 Qwen3-Max dimensions are Grammar/Fluency, Token Robustness, Conciseness, Syntactic Complexity, Coherence/Logic, Topic Coherence, Tone, Factuality, Knowledge Depth, Cross-Domain Generalization, Creativity, Safety, Ethics Sensitivity, Meta-Reflection, Instruction Following,

and Overall. Each dimension is scored from 1 to 5. The system message defines the model as a professional text-quality evaluator. The generated user template inserts the definitions of all 16 dimensions, five anchored score levels, a fixed score-3 example, the evaluated prompt and response, and an output instruction requiring a JSON array of dimension, integer score, and short reason objects. Validity checks require all 16 dimensions and an integer score in range.

For pairwise evaluation, the system message is: “You are a fair and expert evaluator comparing two assistant responses. You pay careful attention to helpfulness, correctness, instruction following, and safety. You never favor longer or more verbose answers if they are not clearly more useful.” The user template inserts the user prompt and responses A/B, asks which response better answers the request without factual or logical errors, is more useful, safe, and appropriate, and requires only `{"winner": "A" | "B" | "tie", "reason": "<1 short sentence>"}`. Every prompt is judged again with the response order swapped; a pair is valid only when both calls parse. The reverse-order label is mapped back to the original A/B frame. If the two calls agree, their shared label is the prompt label; if they disagree, the prompt is recorded as a tie. Thus each of the n prompts contributes one aggregated A/B/tie label, net preference is $(A \text{ wins} - B \text{ wins})/n$, and bootstrap resampling operates at the prompt level.

B.2 Complete Qwen3-Max and Pairwise Effects

B.3 Programmatic Benchmarks

IFEval uses 541 prompts containing 834 separately scored instructions. Table 4 reports all controlled allocation comparisons. Effects are percentage-point differences for A minus B. Twenty-seven of 28 intervals contain zero; the one resolved result is Qwen2.5-14B Middle over Matched on loose instruction accuracy.

AlpacaEval 2 LC uses the `weighted_alpaca_eval_gpt-4o-mini-2024-07-18` configuration, with 805 prompts per condition. The LC estimator can regularize near-reference raw win rates to exactly 50.00; this value is not an equivalence test.

For plain-text records, the controlled pipeline treats the final space at the prompt-completion boundary as the first character of each completion. This preserves the exact concatenated dia-

Table 2: Complete controlled Qwen3-Max Overall effects. Effects are A minus B on the 1–5 scale.

Comparison (A–B)	Effect [95% CI]	<i>n</i>	Interpretation
Qwen2.5-7B Selected–Base	+0.152 [+0.093, +0.212]	1000	A favored
Qwen2.5-7B Selected–All	−0.020 [−0.073, +0.031]	1000	unresolved
Qwen2.5-7B Selected–Matched	+0.036 [−0.018, +0.090]	1000	unresolved
Qwen2.5-7B All–Matched	+0.056 [−0.001, +0.113]	1000	unresolved
Qwen2.5-3B Middle–Base	+0.201 [+0.135, +0.266]	1000	A favored
Qwen2.5-3B All–Base	+0.206 [+0.137, +0.273]	1000	A favored
Qwen2.5-3B Middle–All	−0.005 [−0.063, +0.053]	1000	unresolved
Qwen2.5-14B Middle–Base	+0.429 [+0.355, +0.505]	1000	A favored
Qwen2.5-14B All–Base	+0.428 [+0.353, +0.503]	1000	A favored
Qwen2.5-14B Middle–All	+0.001 [−0.049, +0.051]	1000	unresolved
Qwen2.5-14B Matched–Base	+0.335 [+0.265, +0.404]	999	A favored
Qwen2.5-14B Middle–Matched	+0.094 [+0.031, +0.156]	999	A favored
Qwen2.5-14B All–Matched	+0.093 [+0.033, +0.154]	999	A favored

Table 3: Complete controlled order-swapped pairwise effects. Effects are tie-inclusive net preference for A over B. Each row has 1,000 valid prompt pairs.

Comparison (A–B)	Judge	Effect [95% CI]	<i>n</i>	Interpretation
Qwen2.5-7B Selected–All	DeepSeek-v4-pro	−0.043 [−0.087, +0.002]	1000	unresolved
	GPT-5.5	−0.029 [−0.079, +0.021]	1000	unresolved
Qwen2.5-7B Selected–Matched	DeepSeek-v4-pro	+0.081 [+0.035, +0.126]	1000	A favored
	GPT-5.5	+0.049 [−0.002, +0.100]	1000	unresolved
Llama-3.1-8B Selected–All	DeepSeek-v4-pro	−0.061 [−0.110, −0.013]	1000	B favored
	GPT-5.5	−0.077 [−0.130, −0.023]	1000	B favored
Llama-3.1-8B Middle–Matched	DeepSeek-v4-pro	−0.076 [−0.124, −0.030]	1000	B favored
	GPT-5.5	−0.083 [−0.136, −0.030]	1000	B favored
Qwen2.5-14B Middle–All	DeepSeek-v4-pro	+0.018 [−0.027, +0.062]	1000	unresolved
	GPT-5.5	+0.042 [−0.008, +0.091]	1000	unresolved

Table 4: Complete IFEval effects in percentage points. SP/SI are strict prompt/instruction accuracy; LP/LI are loose prompt/instruction accuracy.

Comparison (A–B)	SP [95% CI]	SI [95% CI]	LP [95% CI]	LI [95% CI]
Qwen2.5-7B Selected–All	−0.92 [−2.96, +1.11]	−1.80 [−3.89, +0.35]	−0.37 [−2.77, +2.03]	−1.80 [−4.19, +0.50]
Qwen2.5-7B Selected–Matched	−1.29 [−3.51, +0.92]	−0.72 [−2.81, +1.34]	−0.74 [−3.33, +1.85]	−0.72 [−3.06, +1.67]
Llama-3.1-8B Selected–All	−0.92 [−2.59, +0.74]	−0.72 [−2.00, +0.59]	−1.11 [−2.77, +0.55]	−0.84 [−2.14, +0.48]
Llama-3.1-8B Middle–Matched	−0.37 [−1.85, +1.11]	−0.24 [−1.53, +0.97]	−0.55 [−2.22, +0.92]	−0.24 [−1.59, +1.08]
Qwen2.5-3B Middle–All	+0.18 [−1.85, +2.22]	−0.72 [−2.44, +0.97]	+0.55 [−1.48, +2.77]	−0.48 [−2.32, +1.35]
Qwen2.5-14B Middle–All	−0.18 [−2.03, +1.66]	−0.36 [−2.06, +1.32]	+0.37 [−1.85, +2.59]	+0.36 [−1.58, +2.36]
Qwen2.5-14B Middle–Matched	+0.92 [−1.66, +3.51]	+1.68 [−0.60, +3.98]	+1.66 [−1.11, +4.44]	+2.76 [+0.37, +5.16]

Table 5: Controlled AlpacaEval 2 results. Empty counts include empty or immediate-EOS generations.

Model	Allocation	Empty	Raw WR ± SE	LC WR ± SE	Avg. chars
Qwen2.5-7B	Base	20	52.22 ± 1.72	50.00 ± 0.00	2893
	Matched	19	51.77 ± 1.72	50.00 ± 0.00	2943
	Selected	19	52.45 ± 1.72	50.00 ± 0.00	2720
	All	13	54.16 ± 1.72	53.98 ± 0.02	2801
Llama-3.1-8B	Base	67	5.81 ± 0.81	3.44 ± 0.07	6223
	Matched	58	6.03 ± 0.82	3.52 ± 0.07	6135
	Middle	56	7.59 ± 0.91	4.59 ± 0.08	5978
	Selected	60	7.19 ± 0.89	4.28 ± 0.07	5902
	All	53	7.47 ± 0.90	4.99 ± 0.10	5955

logue and ensures that the prompt token IDs are a prefix of both tokenized prompt–completion sequences. An audit found a tokenizer-boundary mismatch in legacy adapters; no legacy adapter or generation enters the controlled headline tables.

C Training and Allocation Protocol

C.1 Training and Decoding

Table 6 gives the complete shared recipe. Qwen2.5-14B uses micro-batch 2, accumulation 8, and gradient checkpointing for memory; its effective

Table 6: Training and generation settings for controlled HH-RLHF experiments.

Item	Value
Training pairs / epochs	10,000 / 1
Learning rate / weight decay	$1 \times 10^{-5} / 0$
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$)
DPO β	0.1
Precision	bfloat16
Micro-batch / accumulation	4 / 4
Effective batch	16
LoRA targets	q, k, v, o projections
Standard LoRA	$r = 16, \alpha = 32$, dropout 0.05
Prompt / total length	1,024 / 2,048
Training seeds	13, 42, 77
Generation	greedy, seed 42, max 512 new tokens

batch and all optimization settings remain unchanged. The recorded software environment is Python 3.11.15, PyTorch 2.10.0+cu128, Transformers 5.6.1, TRL 1.2.0, PEFT 0.19.1, and Datasets 4.8.4.

C.2 Compute and Artifact Terms

The recorded controlled run matrix contains 77 completed single-GPU training runs. Four NVIDIA H800 GPUs were used as independent single-GPU workers; summed recorded training time is approximately 22.1 GPU-hours. This training-only total excludes generation and API-based judging.

HH-RLHF is distributed under the MIT License; HelpSteer2-Preference under CC BY 4.0; Qwen2.5 and Qwen3 under Apache 2.0; and Llama 3.1 under the Llama 3.1 Community License. We used these artifacts for research preference tuning and evaluation.

Teacher-forced metrics compare the adapted policy with the unadapted reference on the same preferred and dispreferred completions. Objective intervals are two-sided 95% Student- t intervals across the three training seeds ($df = 2$). Generation uses the seed-42 checkpoint. Pairwise and rubric effects use paired-prompt bootstrap intervals; human intervals resample prompts while keeping each prompt’s three votes together.

C.3 Positional Partitions and Exact-Capacity Allocation

Layer indices begin at zero. Local, Intermediate, and Global denote contiguous depth ranges only.

Table 7 lists every controlled partition. The selected Llama window is middle-adjacent and differs from the deterministic middle third.

Matched preserves $\alpha/r = 2$ in every layer and matches the aggregate rank of Middle. For Qwen2.5-7B, Middle has rank-sum $9 \times 16 = 144$; Matched uses rank 5 in 24 layers and rank 6 at layers 3, 10, 17, and 24. For Llama-3.1-8B, Middle has rank-sum $11 \times 16 = 176$; Matched uses rank 5 in the 16 even layers and rank 6 in the 16 odd layers. Automated checks verified the targeted projections and the exact trainable-parameter count before each run.

D Complete Objective-Level Results

D.1 Multi-Seed Window Search

The frozen selector ranks mean dev DPO loss across seeds 13, 42, and 77. Let the minimum-mean reference have mean $\bar{\ell}_{\min}$ and sample standard deviation $s_{\min}$. Window i enters the low-loss basin if its 95% Student- t interval overlaps the reference interval or if $|\bar{\ell}_i - \bar{\ell}_{\min}| < \sqrt{(s_i^2 + s_{\min}^2)/2}$. Among basin members, the selected representative is the one whose center is closest to the fixed middle-third center, with lower mean loss as a tiebreaker. Preference accuracy, mean implicit margin, and chosen-token KL are consistency checks and do not replace the loss criterion.

D.2 Matched Capacity and Scale

Paired seed differences confirm that Middle improves on Matched for Qwen (-0.0086 , 95% CI $[-0.0095, -0.0078]$) and Llama (-0.0057 , $[-0.0109, -0.0006]$). Selected versus All is unresolved for Qwen ($+0.0028$, $[-0.0010, +0.0066]$) and Llama ($+0.0036$, $[-0.0024, +0.0096]$).

E Blind Human Evaluation

The human study compares the frozen Selected window with All for Qwen2.5-7B and Llama-3.1-8B. For each comparison independently, 120 source IDs are sampled without replacement from the aligned 180-prompt human pool using a deterministic pseudorandom seed constructed from global seed 4242, the comparison code, and formal-sample; sampled IDs are then sorted. The Qwen and Llama formal sets can therefore differ. Each comparison has three independent annotators and six repeated quality-control prompts per annotator. Model identity, allocation, and response orientation are hidden. The public sheets contain ran-

Table 7: Depth partitions in the controlled study.

Model	Layers	Local	Intermediate	Global	Selected
Qwen2.5-3B	36	0–11	12–23	24–35	12–23
Qwen2.5-7B	28	0–9	10–18	19–27	10–18
Llama-3.1-8B	32	0–10	11–21	22–31	9–19
Qwen2.5-14B	48	0–15	16–31	32–47	16–31

Table 8: Qwen2.5-7B held-out window results across three training seeds.

Layers	Dev loss $\pm$ SD	95% CI	Pref. acc.	Margin	Chosen KL
0–8	0.691907 $\pm$ 0.001012	[0.689394, 0.694420]	0.5322 $\pm$ 0.0168	0.00441 $\pm$ 0.00177	0.00247 $\pm$ 0.00051
3–11	0.689406 $\pm$ 0.000844	[0.687311, 0.691502]	0.5278 $\pm$ 0.0096	0.01027 $\pm$ 0.00161	0.00350 $\pm$ 0.00094
6–14	0.684585 $\pm$ 0.000699	[0.682849, 0.686320]	0.5544 $\pm$ 0.0084	0.02136 $\pm$ 0.00200	0.00486 $\pm$ 0.00133
9–17	0.679940 $\pm$ 0.001362	[0.676556, 0.683324]	0.5656 $\pm$ 0.0139	0.03267 $\pm$ 0.00347	0.00620 $\pm$ 0.00120
10–18 [†]	0.676943 $\pm$ 0.000640	[0.675354, 0.678532]	0.5822 $\pm$ 0.0069	0.03910 $\pm$ 0.00057	0.00634 $\pm$ 0.00108
13–21 [*]	0.675337 $\pm$ 0.000730	[0.673524, 0.677150]	0.6156 $\pm$ 0.0150	0.04068 $\pm$ 0.00129	0.00406 $\pm$ 0.00029
16–24	0.681079 $\pm$ 0.001137	[0.678254, 0.683904]	0.6056 $\pm$ 0.0255	0.02662 $\pm$ 0.00249	0.00223 $\pm$ 0.00024
19–27	0.685363 $\pm$ 0.000498	[0.684127, 0.686599]	0.5900 $\pm$ 0.0033	0.01664 $\pm$ 0.00091	0.00111 $\pm$ 0.00006

* minimum mean; [†] frozen representative. Basin: 9–17, 10–18, 13–21.

Table 9: Llama-3.1-8B held-out window results across three training seeds.

Layers	Dev loss $\pm$ SD	95% CI	Pref. acc.	Margin	Chosen KL
0–10	0.669917 $\pm$ 0.001556	[0.666051, 0.673782]	0.5800 $\pm$ 0.0033	0.09603 $\pm$ 0.01146	0.05121 $\pm$ 0.01632
3–13	0.658393 $\pm$ 0.000675	[0.656716, 0.660070]	0.5922 $\pm$ 0.0038	0.12261 $\pm$ 0.00336	0.04683 $\pm$ 0.00266
6–16 [*]	0.655876 $\pm$ 0.001348	[0.652528, 0.659223]	0.5967 $\pm$ 0.0115	0.12351 $\pm$ 0.00579	0.03954 $\pm$ 0.00280
9–19 [†]	0.658179 $\pm$ 0.000448	[0.657067, 0.659291]	0.5889 $\pm$ 0.0019	0.11208 $\pm$ 0.00318	0.03009 $\pm$ 0.00272
11–21	0.662655 $\pm$ 0.000623	[0.661106, 0.664203]	0.5944 $\pm$ 0.0077	0.09177 $\pm$ 0.00340	0.02014 $\pm$ 0.00160
14–24	0.672529 $\pm$ 0.000999	[0.670049, 0.675010]	0.5878 $\pm$ 0.0241	0.05508 $\pm$ 0.00292	0.00756 $\pm$ 0.00102
17–27	0.682186 $\pm$ 0.000523	[0.680887, 0.683485]	0.5722 $\pm$ 0.0139	0.02993 $\pm$ 0.00135	0.00329 $\pm$ 0.00040
21–31	0.681695 $\pm$ 0.000585	[0.680240, 0.683149]	0.5522 $\pm$ 0.0107	0.03134 $\pm$ 0.00111	0.00550 $\pm$ 0.00040

* minimum mean; [†] frozen representative. Basin: 3–13, 6–16, 9–19.

Table 10: Controlled three-seed dev metrics for the reference models.

Model	Allocation	Trainable M	Dev loss	Pref. acc.	Margin	Chosen KL
Qwen2.5-7B	Middle/Selected	3.244	0.676943 $\pm$ 0.000640	0.5822	0.03911	0.00634
	Matched	3.244	0.685571 $\pm$ 0.000989	0.5611	0.01748	0.00253
	All	10.093	0.674154 $\pm$ 0.000980	0.5944	0.04830	0.00913
Llama-3.1-8B	Middle	4.686	0.662655 $\pm$ 0.000623	0.5944	0.09177	0.02014
	Matched	4.686	0.668384 $\pm$ 0.001575	0.5889	0.07464	0.02038
	All	13.631	0.654588 $\pm$ 0.002268	0.5956	0.15084	0.06002

Table 11: Qwen scale results. Objective effects are three-seed dev-loss differences; behavior effects are Qwen3-Max Overall differences from the seed-42 generations. Effects are Middle minus All.

Scale	Middle dev loss	All dev loss	Middle–All [95% CI]
3B	0.674580 $\pm$ 0.000772	0.667378 $\pm$ 0.000748	+0.0072 [+0.0035, +0.0109]
14B	0.663891 $\pm$ 0.001124	0.662537 $\pm$ 0.001771	+0.0014 [−0.0058, +0.0085]
<i>Qwen3-Max behavioral effect (Middle–All)</i>			
3B	−0.005		[−0.063, +0.053]
14B	+0.001		[−0.049, +0.051]

domized LEFT/RIGHT/TIE choices; private keys restore the allocation identity after annotation.

Annotators read Chinese translations of the English prompts and outputs. A temperature-zero deepseek-chat translation instruction requires faithful translation with no summarization, correction, comparison, or source guessing. Identical source strings always receive the same canonical translation. Across all six packs, 2,268 text cells map to 623 unique source strings with no conflict-

ing translations. This design evaluates preferences over translated renderings, so stylistic and semantic changes introduced by translation remain a limitation.

All 756 submitted choices are valid. The formal estimate excludes the 36 repeated QC rows and retains them only for self-consistency. Confidence intervals use 100,000 prompt-cluster bootstrap samples with seed 42; the three votes from each prompt are resampled together.

Table 12: Complete blind human-evaluation results. A is Selected and B is All. Net preference includes ties and is $(A - B)/360$.

Model	Raw A/Tie/B	Majority A/Tie/B	Net A - B [95% CI]	A win, no ties [CI]	Fleiss κ	Repeat consistency
Qwen2.5-7B	114/100/146	38/39/43	-0.089 [-0.203, +0.028]	0.438 [0.358, 0.519]	0.401	0.833
Llama-3.1-8B	133/69/158	41/29/50	-0.069 [-0.197, +0.061]	0.457 [0.378, 0.537]	0.514	0.889

Both point estimates favor All, and both intervals include zero. Leave-one-annotator-out estimates preserve the same direction. Three-way agreement is defined as an exact prompt-level match between the human-majority A/B/tie label and the order-swapped LLM judge’s aggregated A/B/tie label. It is higher for Llama; evaluator disagreement remains a limitation.

F Trained-Adapter Interventions

Interventions operate on the seed-42 All checkpoint after training. A removal intervention masks adapter contributions from one depth region while retaining all other trained adapters. A retention intervention keeps contributions from only one region. Teacher-forced effects compare the intervened model with intact All on the same 300 dev prompts; behavioral effects use Qwen3-Max on 299–300 paired generations.

Among the fixed depth thirds, Middle-only retention preserves 0.765 of the Qwen All-over-Base dev-loss gain and 0.645 of the Llama gain, the largest fraction for each model. Llama Selected-only retention preserves 0.801 for the frozen 9–19 window. These results establish dependence on trained adapter components. Simultaneously trained blocks can interact nonlinearly, so the intervention does not identify an exclusive function for any block.

G Implementation-Level Efficiency

The profiling run uses the seed-42 training configuration. Adapter size and peak allocated CUDA memory are measured. Adapter tensors are float32; one float32 gradient per trainable parameter and two float32 Adam moments make the reported gradient and optimizer-state quantities exact. Step time and tokens per second use timestamp and token deltas from steps 50 through 600, excluding setup and warm-up.

Middle and Matched have equal trainable counts and optimizer-state accounting. Their measured peak-memory and step-time differences arise in this implementation and hardware environment. The model still executes every frozen base layer, so

the table does not establish a general training-time speedup.

H Exploratory Analyses from an Earlier Pipeline

The analyses in this section use an earlier pipeline or single-seed diagnostics. They provide scope checks and mechanistic context, and none enters the controlled headline comparisons.

H.1 Layer-Wise Diagnostics

Gradient RMS is recorded every five steps from step 0 through step 95. For each layer, the plotted value averages the four q, k, v, o projections in the final record. Update norms use $\Delta W = (\alpha/r)BA$ and sum the four projections in each layer. Representation drift uses the last-token hidden state on 256 held-out dev prompts and reports $1 - \cos(h_l^{\text{base}}, h_l^{\text{aligned}})$.

These diagnostics are correlational. The trained-adapter interventions in §F provide the direct component-dependence tests.

H.2 Qwen3-8B Headroom Boundary

Qwen3-8B (Yang et al., 2025) is retained only as a headroom boundary case from the earlier pipeline. All-layer LoRA-DPO lowers teacher-forced DPO loss from the unadapted reference value $\ln 2 = 0.693147$ to 0.678554, while Qwen3-Max Overall is 3.264 for All and 3.306 for Base. Order-swapped pairwise checks on the same 1,000 stored prompts find no reliable All-versus-Base preference: DeepSeek-v4-pro gives +0.009 $([-0.044, +0.061])$ and GPT-5.5 gives +0.022 $([-0.035, +0.078])$. Because all-layer tuning itself has no detectable behavioral gain in this case, Qwen3-8B does not support an inference about the ordering of layer allocations.

H.3 HelpSteer2 Transfer

The exploratory HelpSteer2 run uses 6,766 non-tie training pairs and 300 held-out validation prompts. Its effective batch is 16 with micro-batch 2, accumulation 8, and gradient checkpointing. Table 17 shows that the training-loss ordering transfers more clearly than the Qwen3-Max behavioral ordering.

Table 13: Complete intervention effects for removal experiments. Positive dev-loss effects indicate worse objective fit after removal; negative Qwen3-Max effects indicate lower Overall.

Model	Removed adapters	Metric	Effect [95% CI]	<i>n</i>
Qwen2.5-7B	Shallow	dev DPO loss	−0.0006 [−0.0057, +0.0043]	300
Qwen2.5-7B	Middle	dev DPO loss	+0.0124 [+0.0050, +0.0194]	300
Qwen2.5-7B	Deep	dev DPO loss	+0.0038 [+0.0006, +0.0069]	300
Llama-3.1-8B	Shallow	dev DPO loss	+0.0110 [−0.0022, +0.0236]	300
Llama-3.1-8B	Middle	dev DPO loss	+0.0157 [+0.0032, +0.0278]	300
Llama-3.1-8B	Deep	dev DPO loss	+0.0021 [−0.0023, +0.0066]	300
Llama-3.1-8B	Selected 9–19	dev DPO loss	+0.0207 [+0.0044, +0.0364]	300
Qwen2.5-7B	Middle	Qwen3-Max Overall	+0.0000 [−0.0967, +0.1000]	300
Qwen2.5-7B	Shallow	Qwen3-Max Overall	−0.1000 [−0.2067, +0.0067]	300
Qwen2.5-7B	Deep	Qwen3-Max Overall	+0.0700 [−0.0230, +0.1670]	300
Llama-3.1-8B	Middle	Qwen3-Max Overall	−0.0301 [−0.1538, +0.0903]	299
Llama-3.1-8B	Selected 9–19	Qwen3-Max Overall	−0.2575 [−0.3946, −0.1204]	299
Llama-3.1-8B	Shallow	Qwen3-Max Overall	−0.4548 [−0.6054, −0.3110]	299
Llama-3.1-8B	Deep	Qwen3-Max Overall	+0.0167 [−0.0736, +0.1070]	299

Table 14: Complete retention interventions. Fractions are descriptive ratios of the Base-to-All dev-loss improvement preserved when only the named trained adapters remain.

Model	Retained adapters	Dev DPO loss	Δ loss vs. All [95% CI]	Gain retained
Qwen2.5-7B	Shallow	0.690307	+0.015197 [+0.006988, +0.023018]	0.157
	Middle	0.679340	+0.004230 [−0.001447, +0.009677]	0.765
	Deep	0.687899	+0.012790 [+0.002875, +0.022162]	0.291
Llama-3.1-8B	Shallow	0.673010	+0.020347 [+0.005990, +0.034291]	0.497
	Middle	0.667030	+0.014368 [−0.000187, +0.028303]	0.645
	Deep	0.687986	+0.035323 [+0.012951, +0.057318]	0.127
	Selected 9–19	0.660704	+0.008041 [−0.003422, +0.018608]	0.801

Table 15: Complete NVIDIA H800 efficiency accounting. All runs execute a full forward pass through the base model.

Model	Allocation	Trainable M	Adapter MiB	Grad MiB	Adam MiB	Peak GiB	s/step	tok/s
Qwen2.5-7B	Middle	3.244	23.29	12.38	24.75	59.73	1.315	5422.4
	Matched	3.244	23.31	12.38	24.75	77.30	1.615	4415.8
	All	10.093	49.43	38.50	77.00	77.44	1.585	4498.3
Llama-3.1-8B	Middle	4.686	34.31	17.88	35.75	58.37	1.452	4904.2
	Matched	4.686	34.33	17.88	35.75	76.42	1.796	3965.5
	All	13.631	68.45	52.00	104.00	76.59	1.770	4023.2

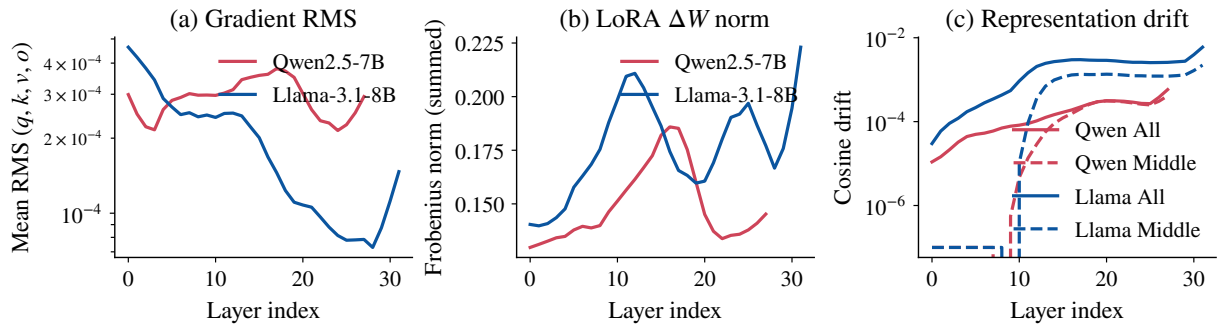


Figure 5: Correlational diagnostics along depth. (a) Per-layer gradient RMS at recorded step 95 of All training. (b) Layer-summed Frobenius norm of LoRA ΔW after All training. (c) Per-layer cosine drift between base and aligned hidden states on 256 held-out prompts; solid lines show All and dashed lines show Middle. Shading covers the union of the two models’ middle thirds.

The behavioral results do not show an absolute gain over Base for either model. On Qwen, Global (3.047) also exceeds Middle (2.880), reversing the corresponding HH-RLHF direction. We therefore

treat HelpSteer2 as a boundary case for dataset and rubric transfer, rather than evidence that a fixed behavioral ordering generalizes.

Table 16: Qualitative summary of the correlational depth diagnostics.

Signal	Qwen2.5-7B	Llama-3.1-8B
Gradient RMS	Peak at layers 14–17; lower values in late layers	Broad early-to-middle plateau; drop after layer 21
LoRA update norm	Middle peak	Middle peak with a secondary late cluster
Representation drift	Rises from middle depth and propagates downstream	Jumps near middle entry and propagates downstream

Table 17: Exploratory HelpSteer2 results. Window denotes the earlier sliding-window choice.

Allocation	Qwen2.5-7B		Llama-3.1-8B	
	Train loss	Overall	Train loss	Overall
Base	—	3.067	—	1.680
All	0.6621	2.903	0.6532	1.613
Middle	0.6689	2.880	0.6663	1.582
Window	0.6671	2.933	0.6584	1.577
Global	0.6929	3.047	0.6888	1.680

Table 18: Exploratory Qwen2.5-7B TruthfulQA results.

Allocation	MC1	MC2	Δ MC1 vs. Base [95% CI]
Base	0.3097	0.4207	—
Local	0.3146	0.4205	+0.0049 [−0.0024, +0.0122]
Middle	0.3158	0.4203	+0.0061 [−0.0012, +0.0135]
Global	0.3133	0.4203	+0.0037 [−0.0012, +0.0098]
All	0.3146	0.4198	+0.0049 [−0.0012, +0.0122]

H.4 TruthfulQA and SimPO Checks

TruthfulQA uses 817 multiple-choice questions in zero-generation log-likelihood mode. All 95% intervals for the MC1 change relative to Base include zero.

An exploratory SimPO check holds the Qwen2.5-7B data and LoRA configuration fixed ($\text{lr} = 5 \times 10^{-7}$, $\beta = 2.0$, $\gamma = 1.0$). Qwen3-Max gives Middle minus All as -0.020 with 95% CI $[-0.067, +0.029]$; DeepSeek yields 22.6%/25.2% Middle/All wins and 52.2% ties. Both comparisons are unresolved.