

AnchorLens: A Shared Coordinate System for Interpreting Intermediate Representations in Large Language Models

Giuseppe Trimigno and Gianfranco Lombardo and Stefano Cagnoni

University of Parma, Italy

{giuseppe.trimigno, gianfranco.lombardo, stefano.cagnoni}@unipr.it

Abstract

Lens methods aim to decode intermediate representations of large language models into vocabulary predictions by training independent translators for each layer. While this approach provides insights into how the final response emerges across layers, it overlooks the dominant structure shared among them. We identify that per-layer alignment problems share a dominant common structure: the first singular component of per-layer translators explains 31–64% of total spectral variance, 10–20 $\times$ above the unstructured baseline. Motivated by this evidence, we propose **AnchorLens**, which exploits a global-local decomposition: a shared full-rank matrix G maps layers into a common coordinate system, while per-layer low-rank adapters correct layer-specific residuals, trained in two stages. AnchorLens achieves over 90% parameter reduction relative to full-rank methods while matching TunedLens and outperforming LoRALens on next-token prediction and linear shortcut evaluations across eight benchmark tasks. These results hold across 16 models up to 120B parameters from five families: GPT-OSS, Qwen3, Qwen2.5, OLMo2, and Gemma2. Furthermore, the shared coordinate system induced by G enables geometric analyses inaccessible to independent per-layer methods, including quantification of inter-layer drift reduction and empirical validation of non-degeneracy: post-training adapter orthogonality confirms G absorbs the shared structure. Our code is publicly available on [GitHub](#).

1 Introduction

Large Language Models (LLMs) refine their internal representations across L layers before producing a final prediction. A central question in interpretability is what these intermediate states encode, and at which depth a model effectively commits to its output (Elhage et al., 2021; Jastrzebski et al., 2018). Answering this question requires decoding

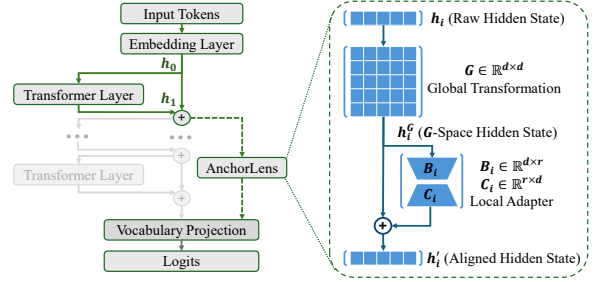


Figure 1: **AnchorLens** architecture. **Left:** AnchorLens is applied at each transformer layer, intercepting the hidden state h_i before the vocabulary projection to produce an intermediate prediction. **Right:** the two-stage transformation at layer i . A single shared, full-rank global matrix $G \in \mathbb{R}^{d \times d}$ maps h_i into the common G -space ($h_i^G = Gh_i + b_G$), placing all layers in the same coordinate system. A per-layer low-rank adapter $L_i = B_i C_i$ then applies a residual correction, yielding the aligned hidden state $h_i' = (I + L_i)h_i^G + b_i$.

intermediate representations into human-readable predictions, a task addressed by the growing family of *lens* methods. The LogitLens (Nostalgebraist, 2020) applies the model’s unembedding matrix directly to each intermediate hidden state, but implicitly assumes that the residual stream at layer i is already aligned with the final-layer representation, which does not generally hold across architectures. TunedLens (Belrose et al., 2025) addresses this by training a per-layer affine translator $A_i \in \mathbb{R}^{d \times d}$ for each layer via distillation from the model’s own final-layer output, substantially outperforming the LogitLens. However, TunedLens trains L independent full-rank matrices, at a cost of $L(d^2 + d)$ parameters that scales poorly with model depth L and hidden dimension d . LoRALens (Trimigno et al., 2026) reduces this by replacing each translator with a low-rank factorisation $A_i = B_i C_i$, achieving over 93% parameter reduction at modest performance cost, but it still treats each layer independently, leaving a source of statistical structure unexploited. **Motivation.** We observe that the per-layer align-

ment problems are not independent. Stacking the vectorised TunedLens translators as rows of $M \in \mathbb{R}^{L \times d^2}$, the first singular component of M accounts for 31–64% of the total spectral variance across all models (mean 42%), compared to the unstructured baseline of $\sim 1/L \approx 2\text{--}4\%$: the translators share a dominant common direction in the space of $d \times d$ matrices. Around 15% of available directions explain 80–87% of each individual translator’s energy throughout most of the network. This shared structure, consistent across 16 models from five families (1B–120B parameters), is left entirely unexploited by both TunedLens and LoRALens.

AnchorLens. We propose **AnchorLens**, a lens architecture that exploits this structure via a global-local decomposition (Figure 1). A single shared affine map G is applied as a direct transformation to the hidden state at every layer, mapping all representations into a common coordinate system: $h_i^G = Gh_i + b_G$. Per-layer, low-rank adapters then correct the remaining layer-specific residual: $h'_i = (I + L_i)h_i^G + b_i$, where I is the identity matrix, and $L_i = B_i C_i$ with rank $r = \lfloor d/32 \rfloor$. AnchorLens achieves over 90% parameter reduction relative to TunedLens and JumpToConclusion (Yom Din et al., 2024), at a parameter count nearly identical to LoRALens, while empirically matching TunedLens performance and surpassing LoRALens across all evaluated models and metrics. While LoRALens achieves comparable parameter efficiency, it treats each layer independently and cannot induce a shared coordinate system; the geometric analyses of §6.4, including cross-layer drift quantification and adapter orthogonality, are structurally unavailable to it, regardless of its predictive performance.

In fact, beyond efficiency, the direct application of G to every layer induces a shared coordinate system unavailable to per-layer methods, enabling three analyses that TunedLens cannot support: we show that G reduces inter-layer representational drift (mean covariance similarity $0.53 \rightarrow 0.76$ on average across 16 models); that local adapters become nearly orthogonal across layers after training (mean off-diagonal similarity reduced by 63% on average); and that per-layer adapter contributions cross-validate the spectral motivation of §3. Together, these results provide empirical support for the AnchorLens design: G captures the shared geometric structure, each L_i encodes layer-specific corrections, explaining why the global-local factorisation matches full-rank alignment at a fraction of the parameter cost.

Contributions. Our contributions are fourfold.

(i) We identify, through joint spectral analysis of TunedLens translator matrices, that the per-layer alignment problem has dominant shared structure. (ii) We propose AnchorLens, a global-local lens architecture exploiting this structure via a shared full-rank global transformation and per-layer low-rank adapters, trained via a two-stage procedure. (iii) AnchorLens achieves $>90\%$ parameter reduction over full-rank lens methods while matching TunedLens performance across 16 models from five families (GPT-OSS, Qwen3, Qwen2.5, OLMo2, and Gemma2) on next-token prediction and a linear shortcut evaluation on 8 benchmark tasks. (iv) The shared coordinate system induced by G enables unique cross-layer geometric analyses, including drift reduction quantification and exact validation of the factorisation’s non-degeneracy.

Paper organisation. §2 reviews background and related work. §3 presents the spectral motivation. §4 describes the AnchorLens architecture and training procedure. §5 details the experimental setup. §6 presents the results and post-training analyses.

2 Background and Related Work

An LLM processes an input sequence through L residual blocks, accumulating updates in a hidden state $h_{i+1} = h_i + F_i(h_i)$, where F_i captures the combined attention and feed-forward computation at block i (Elhage et al., 2021). The final prediction is $\text{softmax}(\text{LN}(h_L) W_U)$, where $W_U \in \mathbb{R}^{d \times |V|}$ is the unembedding matrix and LN is LayerNorm. We denote the model’s final-layer distribution as $f_{>i}(h_i)$, obtained by running the remaining $L-i$ blocks from h_i and applying the unembedding.

A *lens* at layer i is a parametric map $\phi_i : \mathbb{R}^d \rightarrow \Delta^{|V|}$ trained to approximate $f_{>i}(h_i)$ from h_i alone, without running the remaining blocks. TunedLens, LoRALens, and AnchorLens use the forward Kullback-Leibler distillation objective $\mathcal{L}_i = \mathbb{E}_x[D_{\text{KL}}(f_{>i}(h_i) \parallel \phi_i(h_i))]$, which ensures the lens decodes only information the model has encoded (Hewitt and Liang, 2019); JumpToConclusion (Yom Din et al., 2024) instead minimises the mean squared error between the aligned hidden state and the final-layer hidden state, operating in representation space rather than vocabulary space.

2.1 Related Work

Lens methods. The LogitLens (Nostalgebraist, 2020) sets $\phi_i(h_i) = \text{softmax}(\text{LN}(h_i) W_U)$, requir-

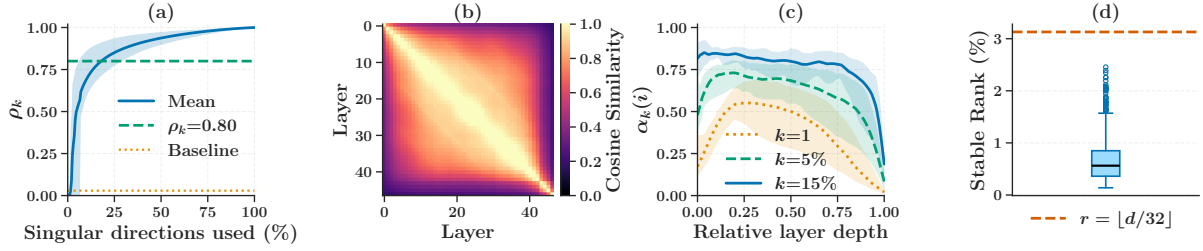


Figure 2: Spectral analysis of TunedLens translator matrices across 16 models. **(a)** Cumulative spectral energy ρ_k : the first direction explains 31–64% of variance (mean 42%); dotted = unstructured baseline; shaded = min–max. **(b)** Pairwise translator similarity (Qwen3-Next-80B-A3B-Instruct): banded structure confirms a shared global direction across all layer pairs. **(c)** Per-layer explained variance $\alpha_k(i)$ for the top 1%, 5%, and 15% of directions (shaded = min–max): the non-uniform depth profile motivates per-layer local adapters. **(d)** Stable rank of residuals $\Delta_i = A_i - \bar{A}$: all values fall strictly below $r = \lfloor d/32 \rfloor$ (dashed), justifying low-rank $\{L_i\}$.

ing no training but implicitly assuming the residual stream at layer i already approximates the final-layer representation, an assumption empirically false for most architectures. TunedLens (Belrose et al., 2025) addresses this with per-layer affine translators $\{(I + A_i), b_i\}$ trained via KL distillation over final-layer logits, establishing the design principles AnchorLens inherits: self-distillation and logit-lens warm-start. LoRALens (Trimigno et al., 2026) reduces the $L(d^2 + d)$ parameter cost of TunedLens by replacing each full-rank A_i with a low-rank factorisation, motivated by the spectral concentration of individual translator matrices. AnchorLens advances LoRALens by additionally identifying and factoring out the dominant structure *shared across all layers* into a jointly trained global matrix G , matching the performance of TunedLens while retaining the parameter efficiency of LoRALens. However, in LoRALens, cross-layer comparisons are not gauge-invariant because each layer-specific translator defines its own latent basis up to arbitrary factorisation symmetries; AnchorLens removes this ambiguity by introducing a single shared transformation G . JumpToConclusion (Yom Din et al., 2024) trains per-layer full-rank maps to minimise the mean squared error between intermediate hidden and final-layer hidden states, thus operating in the representation space rather than logit space.

Mechanistic interpretability. The residual stream formalism of Elhage et al. (2021) motivates the lens paradigm. If each layer writes incrementally to a shared information bus, a linear map from h_i to the vocabulary should recover the information accumulated up to layer i . The iterative inference view of Jastrzebski et al. (2018) and Lombardo et al. (2026), i.e., that residual blocks perform approxi-

mate gradient steps on the loss, further implies that intermediate representations encode task-relevant information at every depth, since each layer incrementally refines the prediction. AnchorLens operates at layer granularity, complementary to circuit-level interpretability (Wang et al., 2022; Conmy et al., 2023), which identifies model components at finer resolution.

Low-rank adaptation. LoRA (Hu et al., 2022) decomposes weight updates as low-rank products $\Delta W = BA$, motivated by the low intrinsic dimensionality of fine-tuning (Aghajanyan et al., 2021). AnchorLens’s local adapters $L_i = B_i C_i$ adopt the same factorisation structure and rank convention, applying it to representation alignment rather than weight updates. Adaptive rank allocation (Zhang et al., 2023) is a natural direction for future work, as the per-layer ΔKL analysis (§6.4) identifies the first layers as most underserved by the shared G .

Knowledge distillation and representational drift. Self-distillation from the model’s own output, as used by all trained lens methods, is related to the broader family of knowledge distillation methods (Hinton et al., 2015; Sanh et al., 2019). The inter-layer representational drift we quantify in §6.4 was first characterised by Belrose et al. (2025) and linked to rogue dimensions Timkey and van Schijndel (2021); AnchorLens’s global transformation G reduces this drift empirically. The sufficiency of affine alignment found in model stitching (Bansal et al., 2021; Csizsárik et al., 2021) further support the affine lens parameterisation.

3 Motivating AnchorLens

We ground the AnchorLens design in a spectral analysis of TunedLens translator matrices $\{A_i\}_{i=1}^L$ across 16 models, establishing two properties:

(i) the translators share dominant common structure, motivating a global component G ; and (ii) the layer-specific residuals are spectrally concentrated, justifying a low-rank parameterisation for the local adapters $\{L_i\}$.

Shared structure across translators. Stacking the vectorised translators as rows of $M \in \mathbb{R}^{L \times d^2}$ and computing the thin SVD $M = U\Sigma V^\top$, the cumulative spectral energy at rank k is:

$$\rho_k = \frac{\sum_{j=1}^k \sigma_j^2}{\sum_{j=1}^L \sigma_j^2} \quad (1)$$

Across all 16 models, the first singular component accounts for 31–64% of the total spectral variance (mean 42%), while the mean fraction of directions needed to reach 80% of variance is 16.3% of L (Figure 2a). By comparison, if the translators were unstructured, the baseline would give $\rho_1 \approx 1/L \approx 2\text{--}4\%$: the observed concentration is 10–20 $\times$ above. The stable rank $\|M\|_F^2 / \sigma_1^2$ ranges from 1.55% to 3.25% of L , confirming that the shared structure spans multiple components rather than a single dominant direction, motivating G being *full-rank* rather than a low-rank approximation.

This structure is not an artefact of TunedLens’s specific parameterisation: repeating the same analysis on LoRALens translators yields consistent results across all 16 models (first component: 29–56% of variance, mean 39%; stable rank: 2.15–4.27% of L). The shared structure therefore reflects a property of the underlying representations rather than of any single lens’s parameterisation.

The pairwise Frobenius cosine similarity of $\{A_i\}$ exhibits a banded, smoothly decaying structure with strictly positive off-diagonal entries (Qwen3-Next-80B-A3B-Instruct in Figure 2b), confirming that the shared structure is a smooth global property and not a localised cluster effect.

Depth-dependent structure. To resolve how the shared component distributes across individual layers, we compute the per-layer explained variance:

$$\alpha_k(i) = \frac{\sum_{j=1}^k (U_{ij} \sigma_j)^2}{\|A_i\|_F^2} \in [0, 1] \quad (2)$$

the fraction of layer i ’s translator energy captured by the top- k shared directions. As shown in Figure 2c, the single dominant shared direction ($k=1$) explains on average 45–55% of individual translator energy across middle layers ($i/L \in [0.15, 0.65]$), rising from $\sim 20\%$ at the first layer ($i/L=0$) to a broad plateau before declining toward the end. Including 15% of available

directions captures 80–87% of translator energy throughout most of the network. The last layer is a confirmed outlier across all models: all three curves collapse near 10%, consistent with its anomalous role at the boundary with the unembedding. The non-uniform depth profile confirms that G alone cannot uniformly align all layers and directly motivates per-layer local adapters.

Residual structure. Subtracting the mean translator $\bar{A} = \frac{1}{L} \sum_i A_i$ as a proxy for the shared component, we compute the stable rank of each residual $\Delta_i = A_i - \bar{A}$. The result is universal: $\text{srank}(\Delta_i) \leq 2.46\%$ of d across *every layer of every evaluated model* (Figure 2d). By the Eckart–Young–Mirsky theorem (Eckart and Young, 1936), a low-rank approximation therefore captures the dominant spectral energy of every residual, justifying the low-rank parameterisation of the local adapters introduced in §4.

Per-model analyses are given in Appendix A.

4 AnchorLens

The motivation analysis of §3 establishes that the per-layer alignment problems share dominant common structure. This motivates decomposing the alignment task into a layer-invariant global component and a low-rank layer-specific residual. Figure 1 illustrates the resulting architecture.

4.1 Architecture

Global transformation. AnchorLens applies a single shared affine map to the hidden state $h_i \in \mathbb{R}^d$ at every layer i :

$$h_i^G = Gh_i + b_G, \quad (3)$$

where $G \in \mathbb{R}^{d \times d}$ is a full-rank matrix and $b_G \in \mathbb{R}^d$ a shared bias, both trained in Stage 1 and then frozen. G must be full-rank: §3 shows that 7–13% of singular directions account for 80% of inter-layer spectral variance across evaluated models, so a low-rank G would systematically miss this energy. Unlike the per-layer residual form $(I + A_i)$ of TunedLens, G is a direct transformation: initialising $G=I_d$, $b_G=\mathbf{0}$ recovers the LogitLens as a principled warm-start, and applying the same full-rank bijection to every h_i maps all layers into the coordinate system defined by G^{-1} , enabling the cross-layer analyses of §6.4.

Local adapters. A single G cannot resolve layer-specific residuals (detailed in §6.4). The second

Data Source	Train Tokens	Eval Tokens
CommonCrawl	130.33M	10.83M
C4	65.72M	5.20M
ArXiv	11.52M	0.99M
GitHub	9.66M	0.92M
StackExchange	7.31M	0.54M
Wikipedia	1.95M	0.18M
Total	226.49M	18.66M

Table 1: Structure of the lens methods train and validation subsets, extracted from the SlimPajama corpus.

stage applies a per-layer, low-rank residual correction to the globally aligned state:

$$h'_i = (I + L_i) h_i^G + b_i, \quad L_i = B_i C_i \quad (4)$$

where $B_i \in \mathbb{R}^{d \times r}$, $C_i \in \mathbb{R}^{r \times d}$, $b_i \in \mathbb{R}^d$ absorbs layer-specific mean shifts, and $r = \lfloor d/32 \rfloor = 3.13\%$ of d , following (Trimigno et al., 2026). The rank constraint is spectrally justified: §3 establishes that the stable rank of $\Delta_i = A_i - \bar{A}$ is at most 2.46% of d across every layer of every evaluated model, strictly below 3.13%, confirming that a rank- r approximation captures the dominant spectral energy.

Complete transformation. Substituting Eq. 3 into Eq. 4 yields the full AnchorLens affine map:

$$h'_i = (G + L_i G) h_i + (b_G + L_i b_G + b_i) \quad (5)$$

with prediction $p_i(h_i) = \text{softmax}(\text{LN}(h'_i) W_U)$, where W_U and LN are frozen from the base LLM.

4.2 Two-Stage Training

Both stages minimise the forward KL distillation objective:

$$\mathcal{L}_i = \mathbb{E}_x [D_{\text{KL}}(f_{>i}(h_i) \parallel p_i(h_i))] \quad (6)$$

where $f_{>i}(h_i)$ is the LLM’s final-layer distribution. Distilling from the model’s own output ensures the lens decodes only information the model has encoded, avoiding the confound of external-label probing (Hewitt and Liang, 2019).

Stage 1: Global alignment. With $L_i = \mathbf{0}$ for all i , we jointly optimise $\{G, b_G\}$ over $\frac{1}{L} \sum_i \mathcal{L}_i$. Training across all layers simultaneously gives G access to L -fold aggregated gradient signal, making it a statistically more efficient estimator of the shared structure than any per-layer method (Baxter, 2000).

Stage 2: Local refinement. G and b_G are frozen; $\{B_i, C_i, b_i\}$ are initialised to zero via $B_i = \mathbf{0}$ and

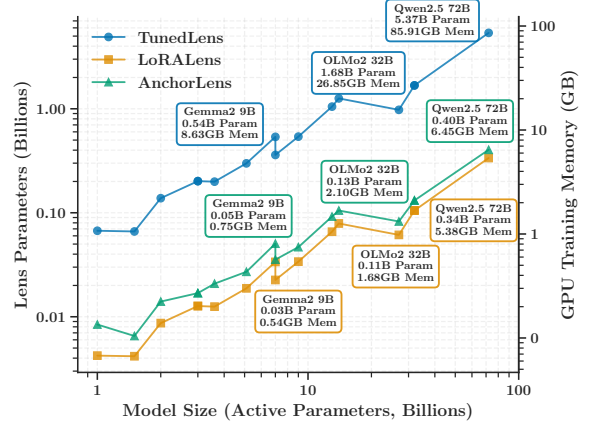


Figure 3: Lens parameter count (**left axis**, billions) and corresponding GPU training memory overhead (**right axis**, GB) as a function of LLM active parameter count (log-log scale). Active parameters are reported to account for MoE architectures. Annotations show exact values for three representative models. AnchorLens achieves >90% reduction relative to TunedLens across all scales (e.g., 5.4B $\rightarrow$ 0.40B parameters and 85.9 $\rightarrow$ 6.4 GB at 72B parameters); its overhead over LoRALens (one shared $d \times d$ matrix) diminishes with model scale.

optimised independently per layer. Freezing G prevents per-layer gradients from corrupting the global alignment and decouples Stage 2 into L independent sub-problems, a greedy coordinate descent over $(G, L_1, \dots, L_L)$. Full hyperparameters and training details are given in Appendix B. Sensitivity to these design choices is reported in Appendix F.

4.3 Parameter Efficiency

AnchorLens requires $(d^2 + d) + L(2dr + d)$ parameters. For large L , the per-layer count approaches $2dr$, yielding a reduction of $1 - 2r/d \approx 93.75\%$ relative to TunedLens and JTC (Belrose et al., 2025; Yom Din et al., 2024), which both require $L(d^2 + d)$. The overhead over LoRALens (Trimigno et al., 2026) is one shared $d \times d$ matrix, amortised over L layers and diminishing with model scale. Figure 3 shows the scaling across evaluated models: at 72B parameters, AnchorLens requires 0.40B parameters and 6.45 GB of GPU training memory compared to 5.37B and 85.91 GB for TunedLens: a 92.5% reduction in both quantities. Under fixed training settings, training time is nearly identical across methods; the GPU memory reduction can alternatively be allocated to larger batch sizes, reducing wall-clock training time proportionally. Per-model counts are in Appendix B.

Inference latency. At inference, AnchorLens applies two sequential affine maps per layer

Method	GPT-OSS 20B		GPT-OSS 120B	
	Train	Inference	Train	Inference
LogitLens	N/A	0.30	N/A	0.48
JTC	0.42	0.33	0.65	0.46
TunedLens	0.43	0.31	0.64	0.45
LoRALens	0.45	0.32	0.63	0.45
AnchorLens	0.42	0.31	0.64	0.46

Table 2: Training and inference latency (s/it) for all lens methods on the GPT-OSS family. Lens computation is negligible relative to the underlying LLM forward pass: AnchorLens matches other methods at both training and inference time despite its two-stage parameterisation.

($h_i \mapsto Gh_i + b_G \mapsto (I + B_i C_i)(\cdot) + b_i$) rather than TunedLens’s single map. We measure this overhead directly: across the GPT-OSS family, end-to-end inference latency for all methods is essentially the same, as lens computation is negligible relative to the cost of the underlying LLM forward pass (Table 2). More fundamentally, the two maps can be collapsed algebraically into one: recalling the combined transformation of Eq. 5, and rewriting it in TunedLens’s residual form $h'_i = (I + A_i)h_i + b_i$ gives

$$A_i^{\text{eq}} = (I + L_i)G - I, \quad \tilde{b}_i = (I + L_i)b_G + b_i, \quad (7)$$

so that $h'_i = (I + A_i^{\text{eq}})h_i + \tilde{b}_i$ exactly. Once trained, AnchorLens can therefore be deployed as a single per-layer affine map of exactly the same form as TunedLens, making its inference cost provably identical regardless of implementation.

5 Experimental Setup

Models. We evaluate all methods on 16 models from five families: Qwen2.5 Dense (1.5B, 7B, 14B, 32B, 72B), Qwen3 MoE (30B-A3B, 80B-A3B), GPT-OSS MoE (20B, 120B), OLMo2 Dense (1B, 7B, 13B, 32B), and Gemma2 Dense (2B, 9B, 27B) (Qwen, 2025a,b; OpenAI, 2025; Gemma, 2024). The set spans 1B–120B parameters, covering both dense and Mixture-of-Experts architectures. Full identifiers are in Appendix B.1.

Data. Training uses a 226.49M-token subset of SlimPajama (Soboleva et al., 2023); evaluation uses a held-out 18.66M-token subset from the same distribution. Table 1 reports the per-source breakdown. The same data are used for all methods.

Baselines. We compare against LogitLens (Nostalgebraist, 2020) (zero-parameter), TunedLens (Belrose et al., 2025), LoRALens (Trimigno et al.,

2026), and JumpToConclusion (Yom Din et al., 2024), which we will refer to as *JTC*. TunedLens, LoRALens, and AnchorLens share the same optimisation configuration: AdamW (Loshchilov and Hutter, 2019) with peak learning rate 10^{-3} , 25-step linear warmup, linear decay, and 2^{18} tokens per batch over 500 steps total. AnchorLens allocates these as 300 steps for Stage 1 and 200 for Stage 2 (§4.2). JTC follows the configuration of Yom Din et al. (2024) applied to our data. Full details and hyperparameters are in Appendix B.

Experiments and metrics. We conduct three experiments. *Next-token prediction*: KL divergence and cross-entropy between the lens prediction and the LLM’s final-layer distribution, evaluated layer-wise on the SlimPajama validation set. *Linear Shortcut Ability* (LSA): evaluated on eight benchmarks (MMLU-Pro, GPQA-Diamond, MATH-500, AIME-2024, AIME-2025, HumanEval, MBPP, and IFEval), covering reasoning, mathematics, coding, and instruction following (Wang et al., 2024; Rein et al., 2023; Hendrycks et al., 2021; Chen et al., 2021; Austin et al., 2021; Zhou et al., 2023), using three metrics. *Surprisal* measures $-\log f_{>i}(h_i)t^*$ where $t^* = \arg \max_t p_i(h_i)(t)$ is the lens’s top-predicted token at layer i and $f_{>i}$ is the final-layer distribution: lower surprisal indicates better agreement. *Precision@k* measures the fraction of examples for which t^* appears among the top- k tokens of the final-layer distribution ($k \in \{1, 5\}$): higher is better. Precision@1 reduces to exact agreement between the lens’s top token and the final layer’s top token. *Ablation*: four configurations as described in §6.3. Benchmark statistics are in Appendix B.

6 Results and Analysis

6.1 Next-Token Prediction

Figure 4 shows layer-wise KL divergence and cross-entropy on the SlimPajama validation set for Qwen3-Next-80B-A3B-Instruct. The method ranking is identical across all 16 evaluated models; per-model curves are in Appendix D. AnchorLens and TunedLens are nearly indistinguishable throughout on both metrics, confirming that the global-local decomposition preserves alignment quality at over 90% parameter reduction. LoRALens follows the same qualitative trend but sits above both across middle layers: these are precisely the layers where the shared structure identified in §3 is most dominant, and where absorbing it into G provides the largest gain over low-rank adapters alone. Log-

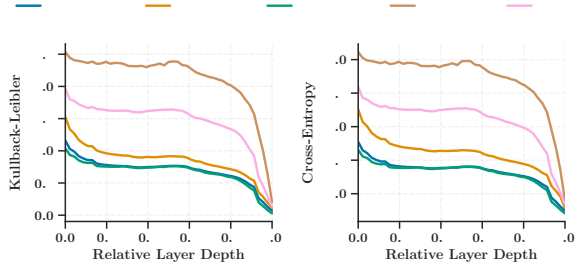


Figure 4: Layer-wise KL divergence (left) and cross-entropy (right) on the SlimPajama validation set for Qwen3-Next-80B-A3B-Instruct. AnchorLens and TunedLens are nearly indistinguishable throughout; LoRALens trails consistently; LogitLens and JumpToConclusion are substantially worse. Results for all 16 evaluated models follow the same ranking (Appendix D).

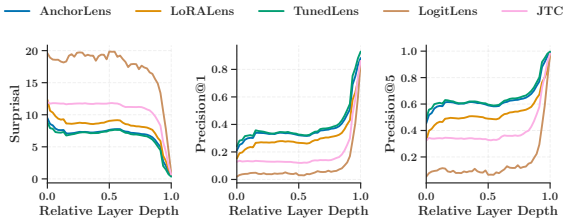


Figure 5: Linear Shortcut Ability on Qwen3-Next-80B-A3B-Instruct: Surprisal (left), Precision@1 (centre), and Precision@5 (right), averaged across 8 benchmarks. Per-benchmark curves are reported in Appendix E and are nearly identical across tasks. AnchorLens and TunedLens are indistinguishable on all three metrics; LoRALens trails; LogitLens and JTC are at or near chance for most of the network.

itLens remains essentially flat across most of the network, reflecting its inability to correct representational drift. JTC is high and erratic at early layers, converging only near the final layers; this instability is more pronounced on certain families.

6.2 Linear Shortcut Ability

Figure 5 reports Surprisal, Precision@1, and Precision@5 as a function of layer depth, averaged across all eight LSA benchmarks for Qwen3-Next-80B-A3B-Instruct. Per-benchmark curves are in Appendix E: they are nearly identical across tasks, confirming that the method ranking is task-agnostic and that averaging is a faithful summary. AnchorLens and TunedLens are again indistinguishable on all three metrics. LoRALens is notably weaker, with the gap most visible in Precision@1 and Precision@5 at early and middle layers, where the absence of a shared component leaves layer-specific alignment underspecified. LogitLens and JTC re-

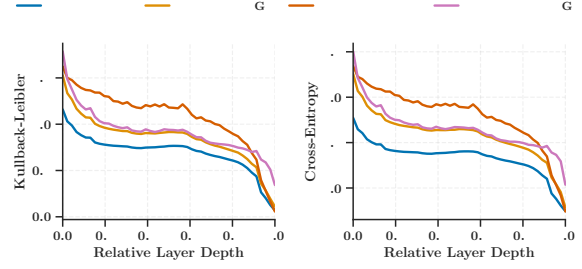


Figure 6: Ablation study: layer-wise KL (left) and CE (right) for Qwen3-Next-80B-A3B-Instruct. AnchorLens-G uses the global transformation only; TunedLens-AvgG replaces all translators with the mean $\bar{A}$; TunedLens-LR replaces each translator with its rank- r truncated-SVD approximation ($r = \lfloor d/32 \rfloor$). Both TunedLens variants show instability at early layers that is more severe on other model families (Appendix C).

main at or near chance throughout most of the network. A consistent pattern emerges across all methods: LSA metrics improve slowly through the middle of the network and rise sharply only in the final 15% of layers, confirming that task-specific predictions become linearly accessible from intermediate representations predominantly at late depth. AnchorLens tracks this transition as faithfully as TunedLens, demonstrating that the shared coordinate system of §4 does not compromise the layer-wise progression of task-relevant information.

6.3 Ablation Study

We compare four configurations to isolate the contribution of each AnchorLens component: the full model, **AnchorLens-G** (global transformation G only, no local adapters), **TunedLens-AvgG** (TunedLens using the mean translator $\bar{A}$ as a shared global, without adapters), and **TunedLens-LR** (TunedLens with each translator replaced by its best rank- r truncated-SVD approximation). Figure 6 shows the KL/CE curves for Qwen3-Next-80B-A3B-Instruct; all models are given in Appendix C. Three findings emerge. *First*, local adapters are individually necessary: AnchorLens-G underperforms the full model throughout middle layers, confirming that the layer-specific residual cannot be absorbed by G alone, consistent with the non-uniform depth profile of §3. *Second*, the jointly trained G outperforms the post-hoc mean $\bar{A}$: TunedLens-AvgG is worse than AnchorLens-G despite both being single-matrix translators applied uniformly to all layers. *Third*, joint training outperforms post-hoc compression: TunedLens-LR is not consistently better than TunedLens-AvgG

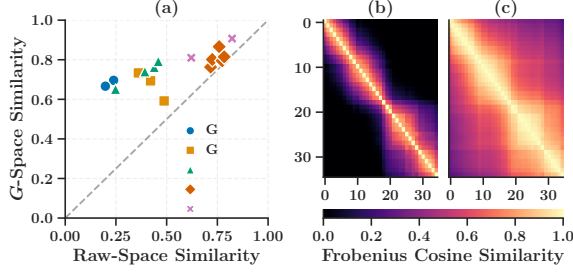


Figure 7: **(a)** Pairwise covariance similarity in raw-space (x -axis) vs G -space (y -axis) for all models. Points above the diagonal indicate drift reduction; average shift $0.53 \rightarrow 0.76$. Models with low raw similarity (GPT-OSS, OLMo) gain most; models already highly aligned (Qwen2.5–3) show limited change. **(b,c)** Covariance similarity in raw-space (b) vs G -space (c) for GPT-OSS-120B. The gain (0.47) suggests strong drift reduction.

and exhibits erratic behaviour at early layers on several model families (Appendix C), whereas AnchorLens is stable across all evaluated models.

6.4 Analysis of the Shared Coordinate System

Because G is a fixed, full-rank bijection applied identically to every layer, all G -transformed representations $\{Gh_i\}_{i=1}^L$ share a common coordinate system, a structural property not naturally induced by per-layer methods. We exploit this to examine what G achieves geometrically and how the local adapters specialise within the shared space.

G Reduces Representational Drift. Following the drift characterisation of Belrose et al. (2025), we compute the mean pairwise Frobenius cosine similarity between hidden-state covariance matrices at each layer, in both raw and G -transformed space. Across all 16 evaluated models, the mean off-diagonal similarity increases from 0.53 (raw) to 0.76 (G -space) on average. Figure 7(a) shows the per-model scatter, while Figures 7(b,c) show the similarity heatmaps for GPT-OSS-120B. The improvement is concentrated in models with high raw drift (< 0.6): GPT-OSS and OLMo families gain 0.32–0.47. For models whose translators are already highly aligned (Qwen2.5–3 families; > 0.7), G has a limited effect (gain 0.02–0.19), which is expected. Per-model results are in Appendix D.

Adapter Orthogonality. Figures 8(a–b) contrast the pairwise similarity of $\{A_i\}$ and $\{L_i\}$ on GPT-OSS-120B. The TunedLens translators form the banded, slowly-decaying structure identified in §3; had G failed to absorb the shared structure, $\{L_i\}$ would be forced to recapture it, reproducing the same pattern. Instead, after AnchorLens train-

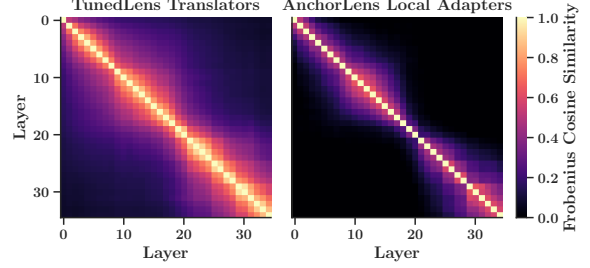


Figure 8: Pairwise Frobenius cosine similarity of TunedLens translators $\{A_i\}$ (banded, slowly decaying) and AnchorLens local adapters $\{L_i\}$ (near-diagonal) for GPT-OSS-120B. After training, G has absorbed the shared banded structure, leaving $\{L_i\}$ orthogonal across layers (75% reduction in mean off-diagonal similarity). Per-model heatmaps in Appendix D.

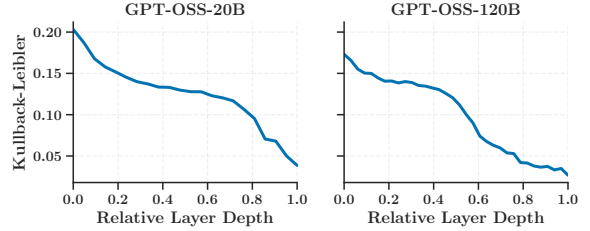


Figure 9: Per-layer KL improvement ΔKL_i over G -only for GPT-OSS 20B and 120B. First layer benefits most, consistent with its lowest per-layer explained variance in §3. The qualitative trend is consistent across all 16 models (Appendix D).

ing, the mean off-diagonal similarity of the local adapters falls by 75% relative to $\{A_i\}$ (range 41–83% across 16 models; Appendix D), leaving each L_i to encode strictly layer-specific information. This near-orthogonality is only meaningful under a shared coordinate system: without a common reference space, as in TunedLens or LoRALens, cross-layer adapter similarity is confounded by the dominant shared direction. G has therefore absorbed the shared alignment structure, and the factorisation is non-degenerate across every model.

Per-layer adapter contribution. Let denote $\Delta\text{KL}_i = \text{KL}_i(G \text{ only}) - \text{KL}_i(G + L_i)$ the improvement provided by each local adapter over G alone. Figure 9 shows the depth profile for GPT-OSS 20B and 120B; the qualitative trend is consistent across all 16 models (Appendix D). Layer 0 consistently provides the largest benefit ($\Delta\text{KL}_0 \approx 0.17\text{--}0.20$), consistent with its lowest per-layer explained variance ($\alpha_1(0) \approx 0.18$ §3): it is the layer most underserved by the shared component G . The benefit decreases with depth, reflecting that G provides progressively better coverage as representations ap-

Layer	LogitLens	TunedLens	AnchorLens-G	AnchorLens
Prompt 1: “The capital of France is” (model output: Paris)				
5 (11%)	Capital / capital / Capital_	the / a / its	the / capital / located	the / capital / a
10 (21%)	capital / Capital / Le	the / a / _	the / Paris / a	Paris / the / a
23 (49%)	吗 / ____ / ____	the / a / not	the / not / a	the / a / not
37 (79%)	这座城市 / Paris / ____	Paris / the / _	the / Paris / a	Paris / the / located
44 (94%)	the / Paris / 巴黎	Paris / the / a	Paris / the / a	Paris / Paris_ / _
Prompt 2: “She opened the door and walked” (model output: out)				
0 (0%)	walk / walk_ / walks	into / to / away	into / away / in	away / into / in
6 (13%)	walk / 走了 / walks	up / over / through	up / through / over	up / through / over
14 (30%)	进去 / 在里面 / into	out / into / down	out / into / down	out / into / down
23 (49%)	进去 / 几步 / into	down / out / into	down / into / out	down / out / into
44 (94%)	进去 / 进来 / 出去	into / out / in	into / out / in	out / into / in

Table 3: Top-3 predictions at five relative depths for two prompts (“The capital of France is” and “She opened the door and walked”) on Qwen3-Next-80B-A3B-Instruct. “_” denotes a space token. Correct predictions are bolded.

proach the final-layer distribution. The agreement between the spectral prediction of §3 and the empirical per-layer contribution confirms the internal consistency of the AnchorLens design.

6.5 Qualitative Decoding Examples

To complement the quantitative results above, we show layer-wise top-3 predictions for two prompts on Qwen3-Next-80B-A3B-Instruct, comparing LogitLens, TunedLens, AnchorLens-G (G -only, no local adapters), and AnchorLens (Table 3).

For “The capital of France is” (model output: *Paris*), LogitLens fails systematically from layers 13–36, producing non-English tokens rather than coherent predictions: a direct consequence of the representational drift characterised in §6.4. At layers 10 and 37, AnchorLens predicts *Paris* as its top-1 token whereas AnchorLens-G predicts it only as top-2: the local adapter provides a concrete, token-level advantage at exactly the layers where G alone is insufficient, consistent with the per-layer Δ KL analysis of §6.4.

For “She opened the door and walked” (model output: *out*), LogitLens copies the last input token (*walk*) at layer 0, then collapses to non-English tokens from layer 6 onward, never recovering until layer 45. AnchorLens and TunedLens instead produce semantically valid English continuations (*into*, *away*, *up*, *through*, *out*, *down*) from the very first layer and are qualitatively indistinguishable throughout: a token-level confirmation of the layer-wise KL/CE parity reported in §6.1.

7 Conclusion

We introduced AnchorLens, a parameter-efficient lens architecture grounded in a spectral observa-

tion: the per-layer alignment problems solved by TunedLens share a dominant common structure, with the first singular component of the stacked translator matrix explaining 31–64% of the total spectral variance (mean 42%) across 16 diverse models with up to 120B parameters from five families (GPT-OSS, Qwen3, Qwen2.5, OLMo2, and Gemma2), 10–20 $\times$ above the unstructured baseline. AnchorLens exploits this by decomposing the alignment into a shared, full-rank global transformation G and per-layer low-rank adapters $\{L_i\}$, trained via a principled two-stage procedure. The result is over 90% parameter reduction relative to full-rank methods, while matching TunedLens performance and surpassing LoRALens across all evaluated models and metrics. Beyond efficiency, the shared coordinate system induced by G enables geometric analyses unavailable to per-layer methods, including a quantified reduction in inter-layer representational drift and empirical validation of the non-degeneracy of the global-local factorisation.

Several directions remain open. Adaptive rank allocation, i.e., assigning higher rank to layers where G provides least coverage, as revealed by the per-layer Δ KL profile, could potentially surpass TunedLens at minimal parameter cost. Extending the spectral motivation and AnchorLens architecture to multimodal language models, where the residual stream geometry may differ substantially due to the interplay of visual and textual modalities, is a natural direction: the per-layer explained variance $\alpha_k(i)$ and local adapter analyses of §6.4 could identify layers that specialise in specific modalities, with the shared G potentially capturing cross-modal alignment structure.

Limitations

Model-dependent drift reduction. The drift reduction result in §6.4 is heterogeneous across model families. The drift reduction is concentrated in models with high raw drift (GPT-OSS, OLMo2 families; raw covariance similarity < 0.6), achieving gain in 0.32–0.47. For models with high raw covariance similarity (Qwen2.5–3 families; raw covariance similarity > 0.7), G provides a limited drift reduction (gain 0.02–0.19), as it is optimised for KL alignment rather than covariance similarity. Therefore, it should not be interpreted as a universal property.

Uniform rank allocation. The rank $r = \lfloor d/32 \rfloor$ is fixed uniformly across all layers. The per-layer ΔKL analysis (§6.4) shows that initial layers benefit most from their local adapters, suggesting that non-uniform rank allocation, i.e., assigning higher rank to underserved layers, could improve performance at equivalent or lower total parameter count. This is not explored in the current work.

Evaluation scope. All evaluated models are publicly available, pre-trained, instruct, or reasoning language models. Multimodal models are not evaluated; their residual stream geometry may differ in ways that affect the spectral motivation and the performance of the global-local decomposition. This will be explored in future work.

References

- Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. [Intrinsic dimensionality explains the effectiveness of language model fine-tuning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328. Online. Association for Computational Linguistics.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. [Program synthesis with large language models](#). *Preprint*, arXiv:2108.07732.
- Yamini Bansal, Preetum Nakkiran, and Boaz Barak. 2021. [Revisiting model stitching to compare neural representations](#). *CoRR*, abs/2106.07682.
- Jonathan Baxter. 2000. A model of inductive bias learning. *J. Artif. Int. Res.*, 12(1):149–198.
- Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. 2025. [Eliciting latent predictions from transformers with the tuned lens](#). *Preprint*, arXiv:2303.08112.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. [Evaluating large language models trained on code](#).
- Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. [Towards automated circuit discovery for mechanistic interpretability](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 16318–16352. Curran Associates, Inc.
- Adrián Csiszárík, Péter Kőrösi-Szabó, Ákos K. Matszangosz, Gergely Papp, and Dániel Varga. 2021. Similarity and matching of neural network representations. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS ’21, Red Hook, NY, USA. Curran Associates Inc.
- C. Eckart and G. Young. 1936. [The approximation of one matrix by another of lower rank](#). *Psychometrika*, 1(3):211–218.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. [A mathematical framework for transformer circuits](#). *Transformer Circuits Thread*.
- Gemma. 2024. [Gemma 2: Improving open language models at a practical size](#). *Preprint*, arXiv:2408.00118.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). *CoRR*, abs/2103.03874.
- John Hewitt and Percy Liang. 2019. [Designing and interpreting probes with control tasks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. [Distilling the knowledge in a neural network](#). *Preprint*, arXiv:1503.02531.
- 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*.
- Stanisław Jastrzębski, Devansh Arpit, Nicolas Ballas, Vikas Verma, Tong Che, and Yoshua Bengio. 2018. [Residual connections encourage iterative inference](#). *Preprint*, arXiv:1710.04773.
- Gianfranco Lombardo, Giuseppe Trimigno, and Stefano Cagnoni. 2026. [A geometric perspective on next-token prediction in large language models: Three emerging phases](#). *Preprint*, arXiv:2605.09011.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Nostalgebraist. 2020. Interpreting gpt: The logitlens. <https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens>.
- OpenAI. 2025. [gpt-oss-120b & gpt-oss-20b model card](#). *Preprint*, arXiv:2508.10925.
- Qwen. 2025a. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Qwen. 2025b. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. [Gpqa: A graduate-level google-proof q&a benchmark](#). *Preprint*, arXiv:2311.12022.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter](#). *CoRR*, abs/1910.01108.
- Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. [SlimPajama: A 627B token cleaned and deduplicated version of RedPajama](#).
- William Timkey and Marten van Schijndel. 2021. [All bark and no bite: Rogue dimensions in transformer language models obscure representational quality](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Giuseppe Trimigno, Gianfranco Lombardo, and Stefano Cagnoni. 2026. [Low-rank lens for scalable llms interpretability](#). pages 35–40.
- Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. [Interpretability in the wild: a circuit for indirect object identification in gpt-2 small](#). *Preprint*, arXiv:2211.00593.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024. [Mmlu-pro: A more robust and challenging multi-task language understanding benchmark](#). *Preprint*, arXiv:2406.01574.
- Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2024. [Jump to conclusions: Short-cutting transformers with linear transformations](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9615–9625, Torino, Italia. ELRA and ICCL.
- Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. [Adaptive budget allocation for parameter-efficient fine-tuning](#). In *The Eleventh International Conference on Learning Representations*.
- 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 Supplementary Motivation Analyses

Table 4 and Figure 10 report per-model spectral statistics and pairwise translator similarity heatmaps complementing the cross-model analysis of §3.

Per-model spectral statistics. Table 4 reports four quantities for each evaluated model: the first singular component’s share of total spectral energy (ρ_1), the fraction of directions needed to explain 80% of energy ($\rho_{80\%}$, as % of L), the stable rank of the stacked translator matrix M ($\text{srank}(M)$, as % of L), and the stable rank of the per-layer residuals $\Delta_i = A_i - \bar{A}$ ($\text{srank}(\Delta)$, as % of d). Spectral concentration is consistent across all model families and scales: ρ_1 is elevated across all models, and $\text{srank}(\Delta)$ is strictly below the LoRA rank $r = \lfloor d/32 \rfloor = 3.13\%$ of d for every layer of every evaluated model, confirming the universality of the low-rank motivation for $\{L_i\}$.

Pairwise translator similarity. Figure 10 shows the pairwise Frobenius cosine similarity $S_{ij} = \langle A_i, A_j \rangle_F / (\|A_i\|_F \|A_j\|_F)$ for all 16 evaluated models. The banded, slowly-decaying structure is consistent across families: nearby layers are highly similar and similarity decreases smoothly with $|i - j|$, confirming that the shared structure identified in §3 is a global property of the translator set rather than a localised cluster effect. The last

layer is a mild outlier in several models (darker final row and column), consistent with its anomalous role at the boundary with the unembedding.

B Implementation Details

B.1 HuggingFace Model Identifiers

Table 5 reports the HuggingFace identifier for each of the 16 evaluated models.

B.2 Training Hyperparameters

All trained lens methods (TunedLens, LoRALens, and AnchorLens) are optimised with AdamW (Loshchilov and Hutter, 2019) with peak learning rate 10^{-3} , weight decay 0.01, gradient clipping to norm 1, 25-step linear warmup, linear decay to zero, and 2^{18} tokens per batch over 500 steps total. AnchorLens allocates these as 300 steps for Stage 1 (optimising G and b_G jointly across all layers) and 200 steps for Stage 2 (optimising $\{B_i, C_i, b_i\}$ independently per layer with G frozen).

Initialisation. G is initialised to I_d and b_G to $\mathbf{0}$, recovering the logit lens as a warm-start. In Stage 2, B_i is initialised to $\mathbf{0}$ and C_i with the standard LoRA initialisation (Hu et al., 2022) (random Gaussian scaled by $1/\sqrt{r}$), so that $L_i = B_i C_i = \mathbf{0}$ at the start of Stage 2. Per-layer biases b_i are initialised to $\mathbf{0}$.

Hardware. All experiments are conducted on a single NVIDIA H200 NVL 141 GB GPU. JumpToConclusion is trained following the original procedure of Yom Din et al. (2024) applied to our data.

Software. All experiments are implemented using Pytorch and transformers libraries.

B.3 Per-Model Lens Parameter Counts

Table 6 reports exact lens parameter counts and GPU training memory overhead (parameters, optimizer state, and gradients under AdamW) for TunedLens, LoRALens, and AnchorLens across all 16 evaluated models, complementing Figure 3 of the main paper.

B.4 LSA Benchmark Statistics

Table 7 reports statistics for the eight benchmarks used in the Linear Shortcut Ability evaluation of §6.

C Full Ablation Results

Figures 13–12 report layer-wise KL divergence and cross-entropy for all four ablation conditions (AnchorLens, AnchorLens-G, TunedLens-LR, TunedLens-AvgG) across all 16 evaluated models, grouped by model family. Results are consistent with the findings reported in §6: AnchorLens outperforms all ablations throughout the network depth on every model, confirming that both the global transformation G and the local adapters $\{L_i\}$ contribute individually to performance. AnchorLens-G (global-only) is consistently second, confirming that G alone provides a strong baseline from which the local adapters correct the remaining layer-specific residuals.

The TunedLens-based ablations reveal qualitative instabilities that vary in severity across model families. TunedLens-AvgG exhibits large errors at early layers in most models, reflecting that the mean translator $\bar{A}$ is a poor approximation to the KL-optimal shared component for the first transformer layers. TunedLens-LR is the most erratic: while it performs comparably to TunedLens-AvgG on some models (GPT-OSS, Qwen2.5-72B), it shows sharp, localised spikes in KL and cross-entropy on Gemma2-27B, OLMo2-7B, and OLMo2-1B, reaching values several times higher than AnchorLens at the same depth. These instabilities confirm that post-hoc rank- r SVD truncation of per-layer translators is not a reliable substitute for jointly training a low-rank alignment from scratch.

D Per-Model Empirical Results

This appendix provides complete per-model results for the three empirical analyses reported in §6 and §6.4: layer-wise next-token prediction (§D.1), pairwise translator and adapter similarity (§D.2), and per-layer adapter contribution (§D.3).

D.1 Layer-Wise Next-Token Prediction

Figures 16–20 show layer-wise KL divergence and cross-entropy for all five methods across all 16 models, grouped by family. The method ranking is consistent throughout: AnchorLens and TunedLens are nearly indistinguishable; LoRALens follows closely; LogitLens and JumpToConclusion are substantially worse and exhibit erratic behaviour, most pronounced at early layers. LogitLens reaches particularly high values on the Gemma2 family (up to ~ 12 nats KL on Gemma2-27B), reflecting that this

Model	ρ_1 (%)	$\rho_{80\%}$ (%)	$\text{srnk}(M)$ (%)	$\text{srnk}(\Delta)$ (%)
Gemma2 2B	37.91	24.00	2.64	0.87
Gemma2 9B	31.36	26.83	3.19	0.76
Gemma2 27B	32.46	20.00	3.08	0.80
GPT-OSS 20B	34.39	17.39	2.91	0.38
GPT-OSS 120B	30.74	22.86	3.25	0.22
OLMo2 1B	52.02	20.00	1.92	1.84
OLMo2 7B	46.88	12.90	2.13	1.15
OLMo2 13B	42.05	12.82	2.38	0.52
OLMo2 32B	42.40	9.52	2.36	0.42
Qwen2.5 1.5B	40.00	25.93	2.50	0.86
Qwen2.5 7B	43.32	18.52	2.31	0.84
Qwen2.5 14B	37.63	14.89	2.66	0.37
Qwen2.5 32B	43.95	11.11	2.28	0.60
Qwen2.5 72B	39.94	8.86	2.50	0.23
Qwen3 30B-A3B	53.47	10.64	1.87	0.74
Qwen3-Next 80B-A3B	64.37	4.26	1.55	1.76

Table 4: Per-model spectral statistics of TunedLens translator matrices. ρ_1 : share of total spectral energy in the first singular component (% of $\|M\|_F^2$). $\rho_{80\%}$: fraction of singular directions required to explain 80% of spectral energy (% of L). $\text{srnk}(M)$: stable rank of the stacked translator matrix M (% of L). $\text{srnk}(\Delta)$: stable rank of residuals $\Delta_i = A_i - \bar{A}$ (% of d); all values are strictly below the LoRA rank $r = \lfloor d/32 \rfloor = 3.13\%$ of d , confirming the universality of the low-rank motivation for $\{L_i\}$.

Model	HuggingFace Identifier
Gemma2 2B	google/gemma-2-2b-it
Gemma2 9B	google/gemma-2-9b-it
Gemma2 27B	google/gemma-2-27b-it
GPT-OSS 20B	openai/gpt-oss-20b
GPT-OSS 120B	openai/gpt-oss-120b
OLMo2 1B	allenai/OLMo-2-0425-1B-Instruct
OLMo2 7B	allenai/OLMo-2-1124-7B-Instruct
OLMo2 13B	allenai/OLMo-2-1124-13B-Instruct
OLMo2 32B	allenai/OLMo-2-0325-32B-Instruct
Qwen2.5 1.5B	Qwen/Qwen2.5-1.5B-Instruct
Qwen2.5 7B	Qwen/Qwen2.5-7B-Instruct
Qwen2.5 14B	Qwen/Qwen2.5-14B-Instruct
Qwen2.5 32B	Qwen/Qwen2.5-32B-Instruct
Qwen2.5 72B	Qwen/Qwen2.5-72B-Instruct
Qwen3 30B-A3B	Qwen/Qwen3-30B-A3B-Instruct-2507
Qwen3-Next 80B-A3B	Qwen/Qwen3-Next-80B-A3B-Instruct

Table 5: HuggingFace model identifiers for all 16 evaluated models.

family’s residual stream geometry at early layers departs most strongly from the final-layer distribution. JumpToConclusion is consistently erratic across families, confirming its poor empirical performance relative to trained lens methods.

D.2 Pairwise Similarity: Full-Rank Translators versus Low-Rank Adapters

Figure 21 shows pairwise Frobenius cosine similarity heatmaps for both TunedLens translators $\{A_i\}$ and AnchorLens local adapters $\{L_i\}$ for all 16 models. In every case, $\{A_i\}$ exhibits the banded, slowly-decaying structure identified in §3, while $\{L_i\}$ is near-diagonal, confirming that G has absorbed the shared structure and that the factorisation is non-degenerate across all evaluated models and architectures. Two models display unusual $\{A_i\}$ patterns: Qwen2.5-72B shows a cross-shaped similarity structure, and OLMo2-1B shows a distinct block in the first few layers; in both cases the $\{L_i\}$ heatmap remains near-diagonal, demonstrating that AnchorLens adapts correctly regardless of the global similarity structure of the translators.

D.3 Per-Layer Adapter KL Contribution

Figure 22 shows the per-layer KL improvement $\Delta\text{KL}_i = \text{KL}_i(G \text{ only}) - \text{KL}_i(G + L_i)$ for all 16 models. The finding reported in §6.4 generalises universally: layer 0 provides the largest benefit in every single model, and the contribution decreases with depth throughout the network. The magnitude

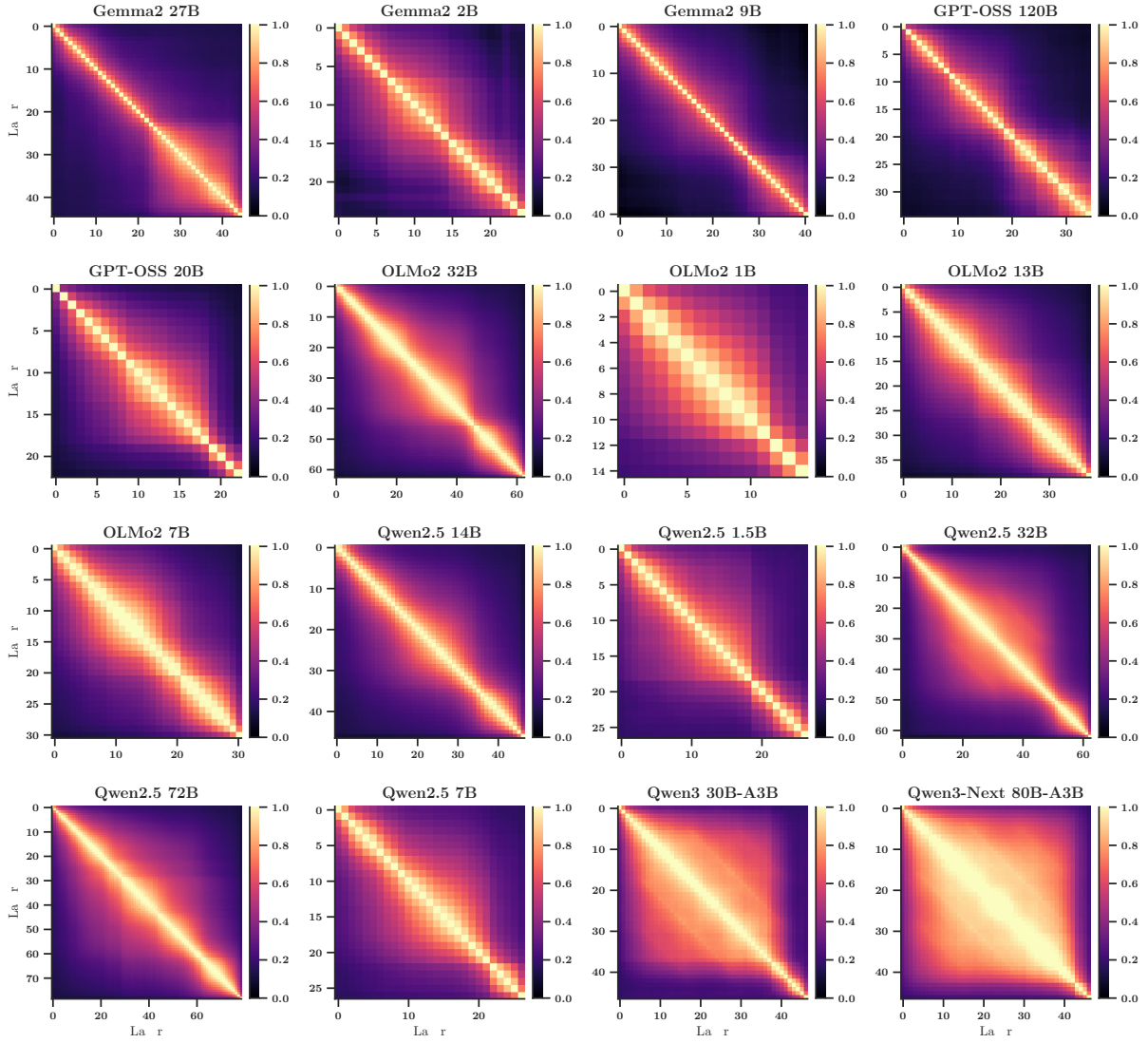


Figure 10: Pairwise Frobenius cosine similarity of TunedLens translator matrices $\{A_i\}$ for all 16 evaluated models, grouped by family. The banded, slowly-decaying structure is consistent across all architectures and scales: nearby layers share high similarity, which decreases smoothly with $|i - j|$, confirming that cross-layer shared structure is a universal property of trained per-layer translators. The last layer is a mild outlier in several models (darker final row and column), consistent with its anomalous role identified in §3.

of ΔKL_0 varies across models and families (from ~ 0.15 on GPT-OSS-120B to ~ 0.60 on Gemma2-9B), reflecting differences in how much the first transformer layer’s representation departs from the shared global alignment. Qwen2.5-72B exhibits a noisier profile, consistent with the cross-shaped translator similarity structure noted in §D.2. In all cases the final layer shows the smallest ΔKL , confirming that G provides its best coverage at the network boundary closest to the vocabulary projection.

E Linear Shortcut Ability: Full Results

This appendix provides per-model LSA results aggregated across benchmarks (§E.1) and per-benchmark LSA results for the representative model (§E.2), completing the analysis of §6.

E.1 Per-Model Results (Benchmark-Averaged)

Figures 23–27 report layer-wise Surprisal, Precision@1, and Precision@5, averaged across all eight benchmarks, for all 16 evaluated models grouped by family. The method ranking is consistent with the main text: AnchorLens and TunedLens are near-indistinguishable on all three

Model	TunedLens		LoRALens		AnchorLens	
	Params (B)	Mem (GB)	Params (B)	Mem (GB)	Params (B)	Mem (GB)
Gemma2 2B	0.14	2.21	0.01	0.14	0.01	0.22
Gemma2 9B	0.54	8.63	0.03	0.54	0.05	0.75
Gemma2 27B	0.98	15.63	0.06	0.98	0.08	1.32
GPT-OSS 20B	0.20	3.19	0.01	0.20	0.02	0.33
GPT-OSS 120B	0.30	4.78	0.02	0.30	0.03	0.43
OLMo2 1B	0.07	1.07	0.00	0.07	0.01	0.13
OLMo2 7B	0.54	8.59	0.03	0.54	0.05	0.81
OLMo2 13B	1.05	16.78	0.07	1.05	0.09	1.47
OLMo2 32B	1.68	26.85	0.11	1.68	0.13	2.10
Qwen2.5 1.5B	0.07	1.06	0.00	0.07	0.01	0.10
Qwen2.5 7B	0.36	5.76	0.02	0.36	0.04	0.57
Qwen2.5 14B	1.26	20.14	0.08	1.26	0.11	1.68
Qwen2.5 32B	1.68	26.85	0.11	1.68	0.13	2.10
Qwen2.5 72B	5.37	85.91	0.34	5.38	0.40	6.45
Qwen3 30B-A3B	0.20	3.22	0.01	0.20	0.02	0.27
Qwen3 80B-A3B	0.20	3.22	0.01	0.20	0.02	0.27

Table 6: Lens parameter counts (billions) and GPU training memory overhead (GB; parameters + optimizer state + gradients under AdamW) for TunedLens, LoRALens, and AnchorLens across all 16 evaluated models. Memory is computed in bfloat16.

Benchmark	Num. Samples	Task Type	Answer Type
MMLU-Pro	12,032	Multi-domain reasoning / QA	10-choice multiple choice
GPQA-Diamond	198	Graduate-level scientific reasoning	4-choice multiple choice
MATH-500	500	Mathematical reasoning	Open-ended numerical / symbolic
AIME 2024	30	Competition mathematics	Integer answer (0–999)
AIME 2025	30	Competition mathematics	Integer answer (0–999)
HumanEval	164	Code generation	Python function completion
MBPP	974	Code generation	Short Python program
IFEval	541	Instruction following	Constrained free-form text

Table 7: Statistics for the eight LSA benchmarks. Task types cover reasoning, mathematics, coding, and instruction following.

metrics across every model; LoRALens trails both but remains well above LogitLens and JumpToConclusion.

The universal pattern of a sharp late-layer rise in Precision@1 and Precision@5 is present in every model: task-relevant information becomes linearly accessible predominantly in the final 15–25% of layers, regardless of model family, size, or architecture. LogitLens and JumpToConclusion fail to decode this information effectively throughout most of the network, with Precision@1 near zero at early and middle depths. Two models show a more extreme version of this pattern: OLMo2-1B (Figure 25) and Gemma2-2B (Figure 24), where all methods produce near-zero Precision@1 until $i/L \approx 0.80$, after which performance rises steeply. This suggests that these smaller models concentrate task-relevant information more sharply in the final layers than larger models in the same families.

E.2 Per-Benchmark Results

Figure 28 shows Surprisal, Precision@1, and Precision@5 for each of the eight LSA benchmarks individually on the representative model (Qwen3-80B-A3B). The curves are nearly identical across all benchmarks for every method, confirming that averaging across benchmarks in §6 is a faithful summary and that the method ranking is not task-specific. The only modest variation across benchmarks is in the absolute level of Surprisal at early layers: AIME-2024 and AIME-2025 exhibit higher Surprisal throughout, reflecting the greater difficulty of competition mathematics relative to multiple-choice tasks; the Precision@1 and Precision@5 rankings are nevertheless preserved across all benchmarks and depth levels.

F Design Choice Sensitivity

This appendix reports sensitivity analyses for four design choices fixed in the main text: the two-stage training procedure (§F.1), the Stage 1/Stage 2 step

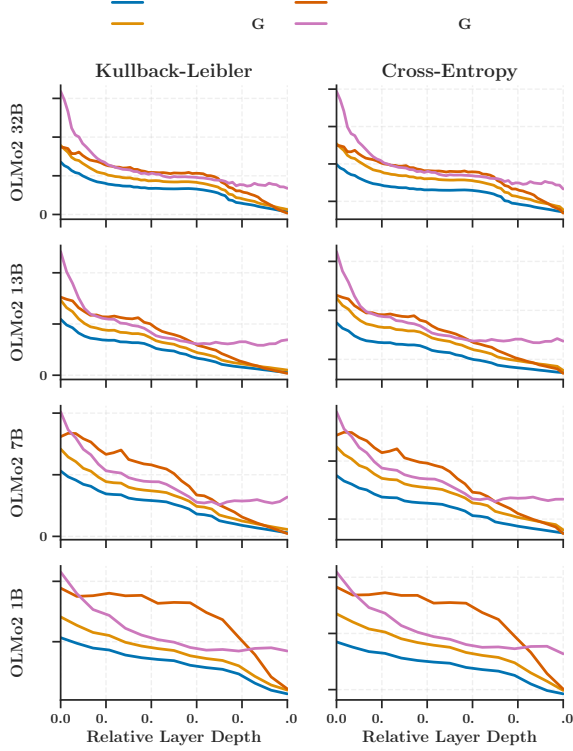


Figure 11: Ablation: KL divergence (left) and cross-entropy (right) for the OLMo2 family. Instability of TunedLens-LR and TunedLens-AvgG increases with decreasing model size: both ablations are relatively smooth on OLMo2-32B and increasingly erratic on OLMo2-7B and OLMo2-1B. AnchorLens remains stable and consistently lowest across all four models.

allocation (§F.2), the local adapter rank (§F.3), and the rank of the global transformation G (§F.4). All experiments are conducted on GPT-OSS 20B and GPT-OSS 120B, using the same training configuration as the main text (§4.2) except for the varied factor under study. In addition to KL and CE on the SlimPajama validation set, we report two auxiliary metrics introduced in §6.4: $G\text{-Sim}$, the mean off-diagonal pairwise covariance similarity in G -space (higher indicates a stronger shared coordinate system), and $L\text{-Spec}$, the mean off-diagonal pairwise Frobenius cosine similarity of the local adapters $\{L_i\}$ (lower indicates more specialised, less redundant adapters).

The overall conclusion across all four sweeps is that the design choices adopted in the main text are near-optimal: results are robust to moderate perturbation, and degrade only under substantial deviations from the defaults, each in an interpretable way that reinforces the design rationale of §4.

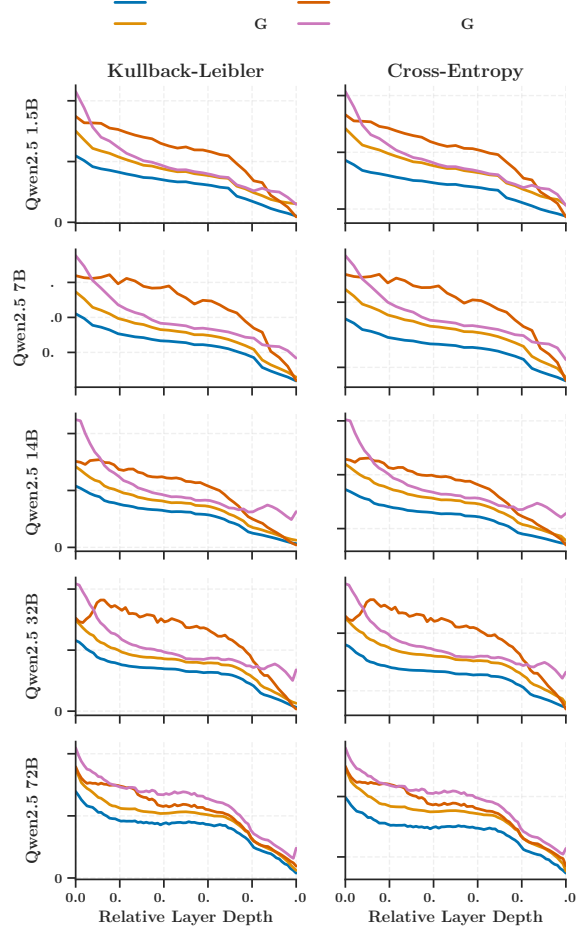


Figure 12: Ablation: KL divergence (left) and cross-entropy (right) for the Qwen2.5 family across five model scales. TunedLens-LR shows erratic behaviour on smaller models (1.5B, 7B, 14B) and becomes smoother at larger scales (32B, 72B), while remaining above AnchorLens at every scale. The gap between AnchorLens and AnchorLens-G is consistent across scales.

F.1 Is the Two-Stage Procedure Necessary?

We compare the original two-stage procedure (Stage 1: G only, 300 steps; Stage 2: $\{L_i\}$ only, G frozen, 200 steps) against two alternatives: **JointStage2**, in which G remains trainable throughout Stage 2 alongside $\{L_i\}$; and **JointStages12**, in which G and $\{L_i\}$ are optimised jointly from step 1, with no separate global-alignment phase.

Fully joint training (JointStages12) is worse on every metric on both models: KL and CE both degrade, G -space alignment weakens substantially ($G\text{-Sim}$ falls from 0.743 to 0.618 on 20B; 0.606 to 0.540 on 120B), and adapters are less specialised (higher $L\text{-Spec}$). This is consistent with the coordinate-descent argument of §4.2: allowing per-layer gradients to influence G throughout training

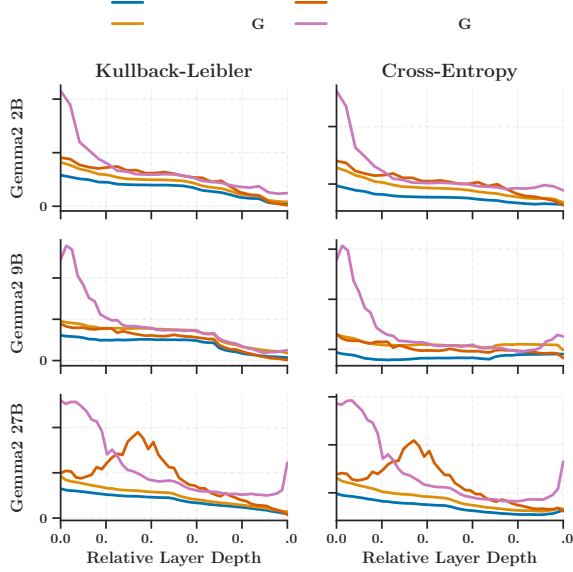


Figure 13: Ablation: KL divergence (left) and cross-entropy (right) for the Gemma2 family. TunedLens-AvgG exhibits large early-layer errors on all three models; TunedLens-LR shows pronounced instability on Gemma2-27B, with sharp spikes throughout the middle of the network. AnchorLens is stable and lowest across all depths.

prevents it from converging to a stable shared alignment before the adapters begin fitting layer-specific residuals.

Keeping G trainable in Stage 2 (JointStage2) is a more subtle case. It achieves a small KL/CE improvement over the original procedure on both models (e.g., KL 0.601 vs. 0.604 on 20B; 0.438 vs. 0.444 on 120B, a difference of under 1% in both cases). However, its adapters are consistently less specialised than the original procedure’s (L-Spec 0.178 vs. 0.164 on 20B; 0.095 vs. 0.086 on 120B). We interpret this as a genuine, small trade-off: allowing G to continue adapting during Stage 2 lets it absorb a small amount of layer-specific signal that would otherwise be captured by $\{L_i\}$, marginally improving raw predictive performance at the cost of the clean separation between global and local structure that the two-stage procedure is designed to produce. Since a central contribution of AnchorLens is the interpretability enabled by this separation (§6.4), we adopt the original two-stage procedure, freezing G before training the local adapters.

F.2 Training Step Allocation

We vary the allocation of the fixed 500-step training budget between Stage 1 and Stage 2, sweeping $\{25/475, 100/400, 200/300, 300/200,$

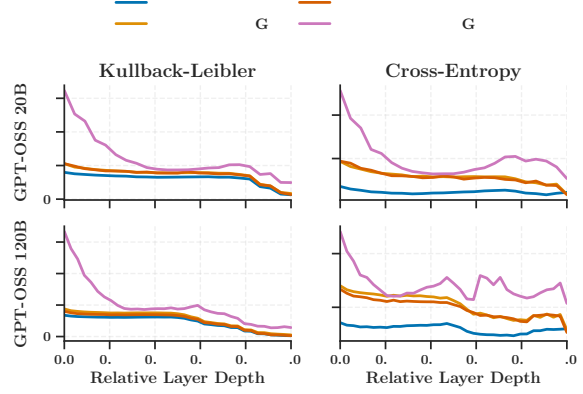


Figure 14: Ablation: KL divergence (left) and cross-entropy (right) for the GPT-OSS family. Both TunedLens ablations are substantially above AnchorLens; TunedLens-AvgG is particularly high at early layers. AnchorLens-G is close to full AnchorLens, indicating that G absorbs most of the alignment benefit for this family.

Condition	KL	CE	G-Sim	L-Spec
GPT-OSS 20B				
Original (2-stage)	0.604	2.230	0.743	0.164
JointStage2	0.601	2.206	0.715	0.178
JointStages12	0.628	2.284	0.618	0.184
GPT-OSS 120B				
Original (2-stage)	0.444	2.603	0.606	0.086
JointStage2	0.438	2.580	0.602	0.095
JointStages12	0.463	2.625	0.540	0.127

Table 8: Two-stage training vs. joint alternatives. G-Sim: mean off-diagonal covariance similarity in G -space (higher is better). L-Spec: mean off-diagonal adapter similarity (lower is better).

400/100, 475/25}; the main text uses 300/200.

Results are robust across a wide central range: allocations from 200/300 to 400/100 all achieve KL within 0.008 of the 300/200 default on both models, with G-Sim consistently above 0.72 (20B) and 0.59 (120B). Only the two most extreme allocations degrade performance substantially, and in each case the degradation is interpretable. At 25/475, Stage 1 is too short for G to converge: G-Sim collapses to 0.510 (20B) and 0.522 (120B), the lowest values in the sweep, and the resulting poorly-aligned G forces the local adapters to compensate, yielding the highest L-Spec values observed (0.308 and 0.100). At 475/25, the opposite failure occurs: with only 25 steps for Stage 2, the local adapters barely move from their zero initialisation. This is visible directly in L-Spec, which falls to near zero (0.033 on 20B; 0.002 on 120B), not because the adapters

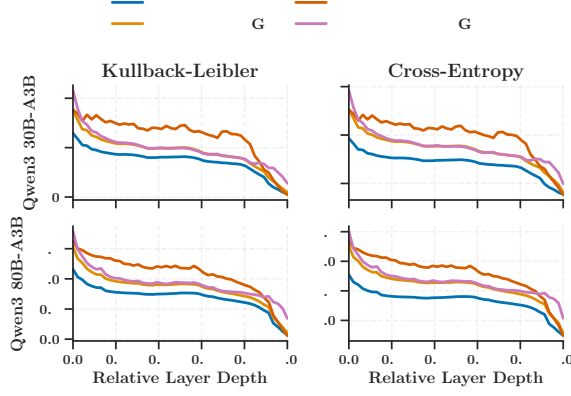


Figure 15: Ablation: KL divergence (left) and cross-entropy (right) for the Qwen3 MoE family. TunedLens-LR exhibits persistent bumpiness throughout the network on both models; TunedLens-AvgG is smoother but remains well above AnchorLens. The gap between AnchorLens and AnchorLens-G is small, consistent with the Qwen3 family having high raw translator similarity (§6.4).

have achieved ideal specialisation, but because they remain close to $B_i = 0$ and are therefore trivially dissimilar to one another. This undertraining is confirmed by the substantially degraded KL at 475/25 (0.697 on 20B, 0.503 on 120B, both far above the 300/200 default). The 300/200 allocation used in the main text is near-optimal by KL, CE, and G-Sim on both models, and is not an arbitrary choice.

F.3 Local Adapter Rank

We vary the local adapter rank $r = \lfloor d/k \rfloor$ for $k \in \{8, 16, 32, 64\}$; the main text uses $k=32$, motivated by the universal stable-rank bound of §3 ($\text{srnk}(\Delta_i) \leq 2.46\%$ of d , strictly below $r=3.13\%$ of d at $k=32$).

On GPT-OSS 120B, KL is essentially flat across the full range ($\Delta \leq 0.004$ nats from $k=8$ to $k=64$), and CE shows no consistent trend with rank. On GPT-OSS 20B, there is a small but consistent improvement with higher rank: moving from $k=32$ to $k=8$ (four times the rank) improves KL by 0.014 nats, at $2.78\times$ the parameter cost (56.1M vs. 20.2M). This is a favourable trade-off in neither direction: the marginal gain from additional rank is small relative to its parameter cost, and the default $k=32$ already captures the great majority of achievable performance on both models. Higher-rank adapters are also consistently less specialised (higher L-Spec) at both scales, indicating that as rank increases, adapters increasingly absorb structure that could otherwise be captured by G

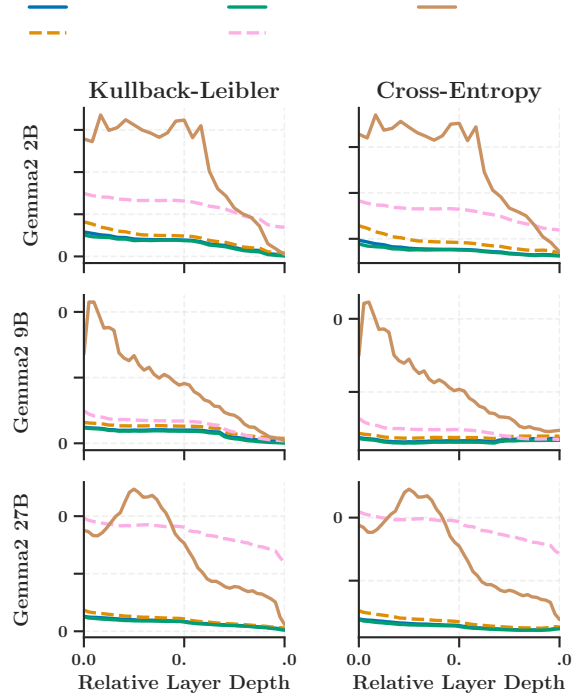


Figure 16: Layer-wise KL (left) and CE (right) for the Gemma2 family. LogitLens reaches up to ~ 12 nats on Gemma2-27B; JumpToConclusion is high and erratic throughout. AnchorLens and TunedLens are near-identical across all three models.

alone. This is consistent with the interpretation that a larger local adapter partially duplicates, rather than complements, the shared coordinate system. These results confirm empirically that the stable-rank bound motivating $r = \lfloor d/32 \rfloor$ in §3 is not merely a conservative theoretical bound but closely tracks the point of diminishing empirical returns.

F.4 Global Transformation Rank

We vary the rank of G itself, $r_G = \lfloor d/k \rfloor$ for $k \in \{1, 8, 64\}$, where $k=1$ (full rank) is used in the main text. For $k > 1$, G is parameterised as a low-rank factorisation with the same total parameter budget structure as the local adapters.

Low-rank G causes severe performance degradation: at $k=8$, KL increases by 42% on 20B and 66% on 120B relative to full-rank G , and CE increases even more sharply on 120B ($2.603 \rightarrow 3.510$). This is by far the largest effect observed across all four sensitivity sweeps in this appendix, and directly supports the spectral argument of §3 that the shared structure across translators spans multiple singular directions and cannot be captured by a low-rank G .

The G-Sim column requires care in interpretation. Low-rank G achieves *higher* G-Sim than

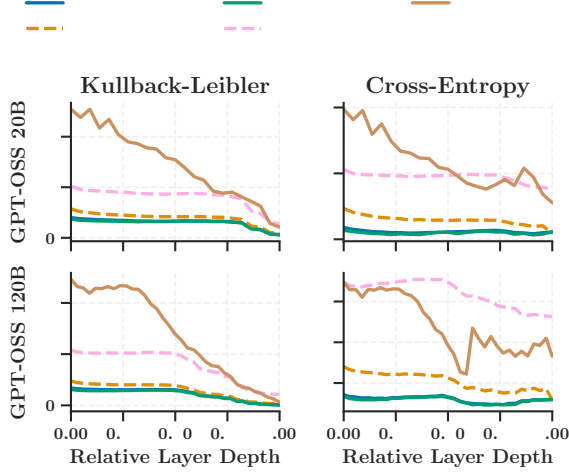


Figure 17: Layer-wise KL (left) and CE (right) for the GPT-OSS family. JumpToConclusion shows pronounced instability in the final layers on GPT-OSS-120B (cross-entropy plot). AnchorLens and TunedLens track closely throughout both models.

full-rank G (0.947–0.970 vs. 0.743 on 20B), which might appear to indicate a stronger shared coordinate system. We do not believe this reflects genuine improvement: a low-rank G necessarily projects all hidden states into a subspace of dimension $r_G \ll d$, which mechanically inflates covariance similarity between layers regardless of whether meaningful alignment has occurred, since all representations are confined to the same low-dimensional subspace. This is corroborated by the L-Spec column, which is substantially *worse* (higher) for low-rank G (0.400–0.427 vs. 0.164 on 20B): the local adapters, denied a well-aligned global component to build on, must instead compensate for G ’s inadequacy, and in doing so lose their layer-specific specialisation. Taken together, the KL/CE degradation and the L-Spec increase indicate that the high G-Sim under low-rank G is an artefact of representational collapse rather than of improved alignment, and that full-rank G is a necessary component of the design rather than a conservative default.

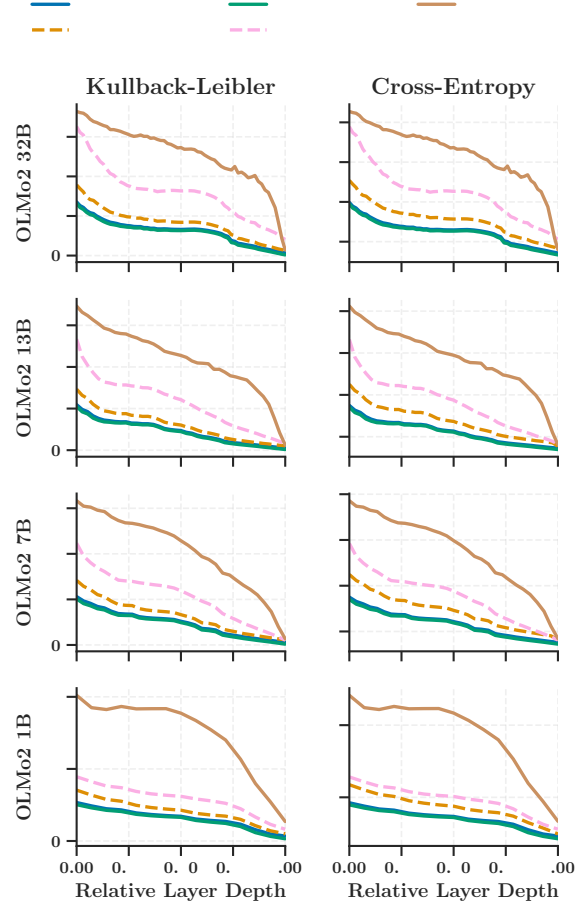


Figure 18: Layer-wise KL (left) and CE (right) for the OLMo2 family across four scales. JumpToConclusion is erratic on all four models; the instability is most pronounced on OLMo2-1B. AnchorLens and TunedLens are near-identical throughout.

Stage1/Stage2	KL	CE	G-Sim	L-Spec
GPT-OSS 20B				
25/475	0.812	2.652	0.510	0.308
100/400	0.631	2.264	0.667	0.201
200/300	0.608	2.239	0.728	0.159
300/200	0.604	2.230	0.743	0.164
400/100	0.612	2.234	0.743	0.159
475/25	0.697	2.325	0.677	0.033
GPT-OSS 120B				
25/475	0.598	3.032	0.522	0.100
100/400	0.461	2.637	0.592	0.115
200/300	0.442	2.603	0.599	0.090
300/200	0.444	2.603	0.606	0.086
400/100	0.456	2.638	0.604	0.065
475/25	0.503	2.790	0.592	0.002

Table 9: Sensitivity to the Stage 1/Stage 2 step allocation (budget fixed at 500 steps). G-Sim and L-Spec as in Table 8.

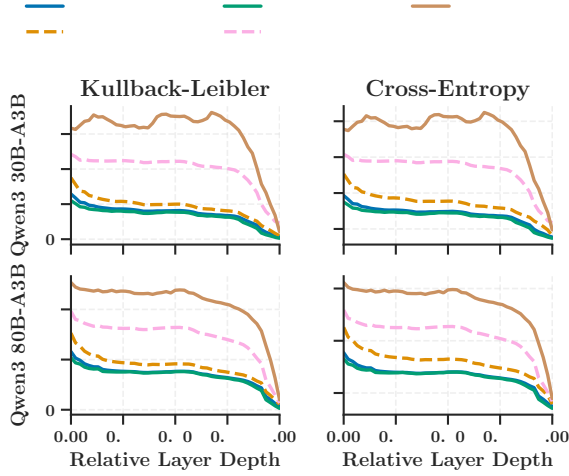


Figure 19: Layer-wise KL (left) and CE (right) for the Qwen3 MoE family. LogitLens is flat and high throughout, consistent with the Qwen3 residual stream requiring substantial alignment. JumpToConclusion shows moderate but persistent elevation above AnchorLens. AnchorLens and TunedLens are essentially on top of one another.

k	KL	CE	Params	L-Spec
GPT-OSS 20B				
8	0.590	2.219	56.1M	0.198
16	0.599	2.225	32.2M	0.188
32	0.604	2.230	20.2M	0.164
64	0.610	2.241	14.3M	0.153
GPT-OSS 120B				
8	0.444	2.603	81.0M	0.097
16	0.443	2.606	44.7M	0.091
32	0.444	2.603	26.5M	0.086
64	0.447	2.611	17.5M	0.080

Table 10: Sensitivity to the local adapter rank $r = \lfloor d/k \rfloor$. Params reports total local adapter parameters (all L layers).

k	KL	CE	Params	G-Sim	L-Spec
GPT-OSS 20B					
1 (full-rank)	0.604	2.230	20.2M	0.743	0.164
8	0.855	2.733	14.1M	0.947	0.400
64	0.897	2.965	12.3M	0.970	0.427
GPT-OSS 120B					
1 (full-rank)	0.444	2.603	26.5M	0.606	0.086
8	0.738	3.510	20.3M	0.978	0.291
64	0.784	3.742	18.5M	0.973	0.324

Table 11: Sensitivity to the rank of the global transformation G . Params reports the total parameter count of G plus all local adapters combined.

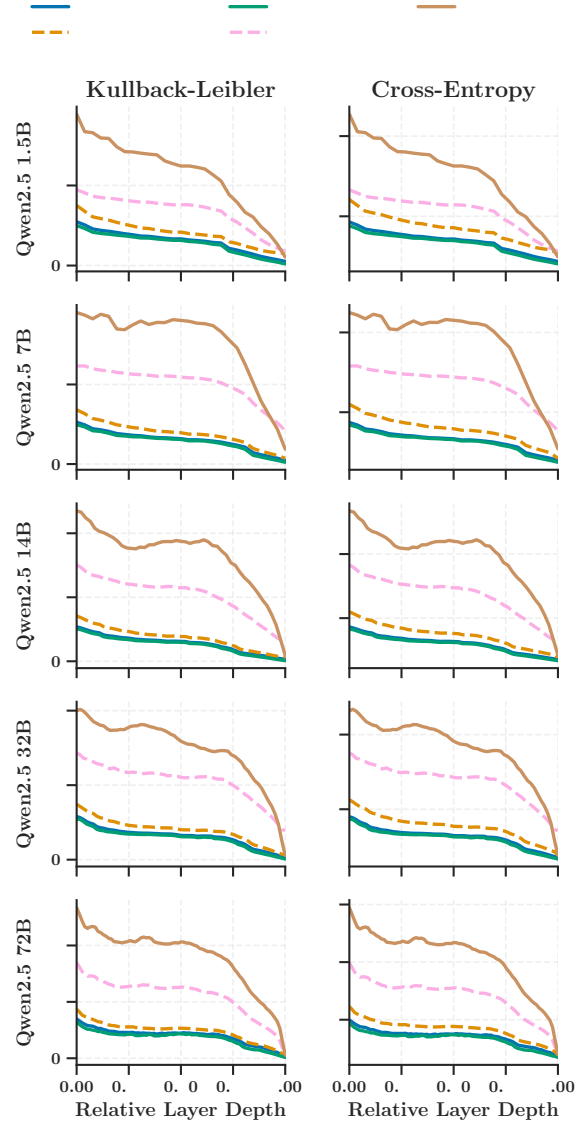


Figure 20: Layer-wise KL (left) and CE (right) for the Qwen2.5 family across five scales. JumpToConclusion is particularly erratic on Qwen2.5-7B, -14B, and -32B. LogitLens reaches high cross-entropy values on Qwen2.5-72B at the final layers. AnchorLens and TunedLens are near-identical at every scale.

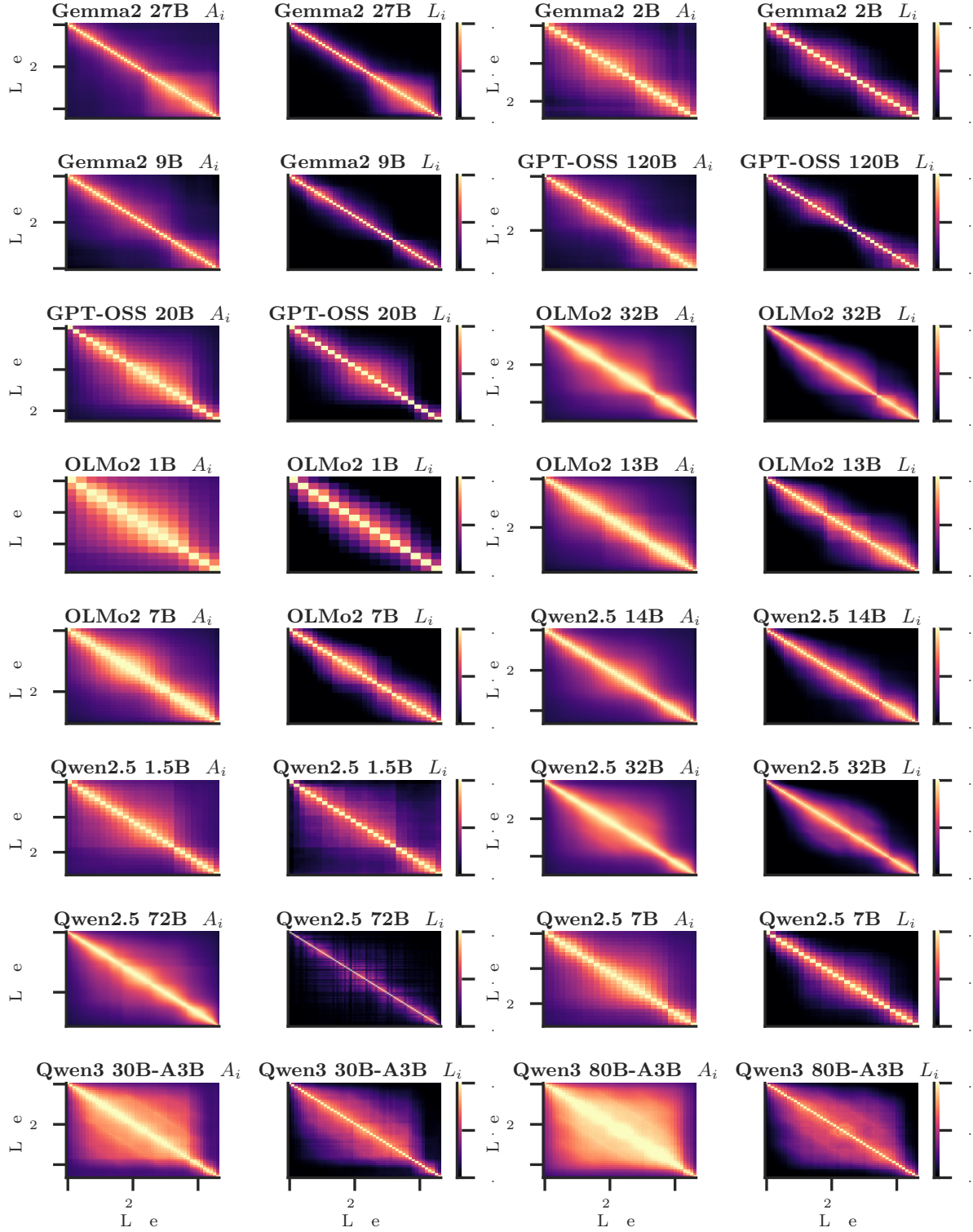


Figure 21: Pairwise Frobenius cosine similarity of TunedLens translators $\{A_i\}$ (odd columns) and AnchorLens local adapters $\{L_i\}$ (even columns) for all 16 evaluated models. $\{A_i\}$ consistently shows banded structure; $\{L_i\}$ is near-diagonal in every case, confirming that G absorbs the shared alignment structure universally.

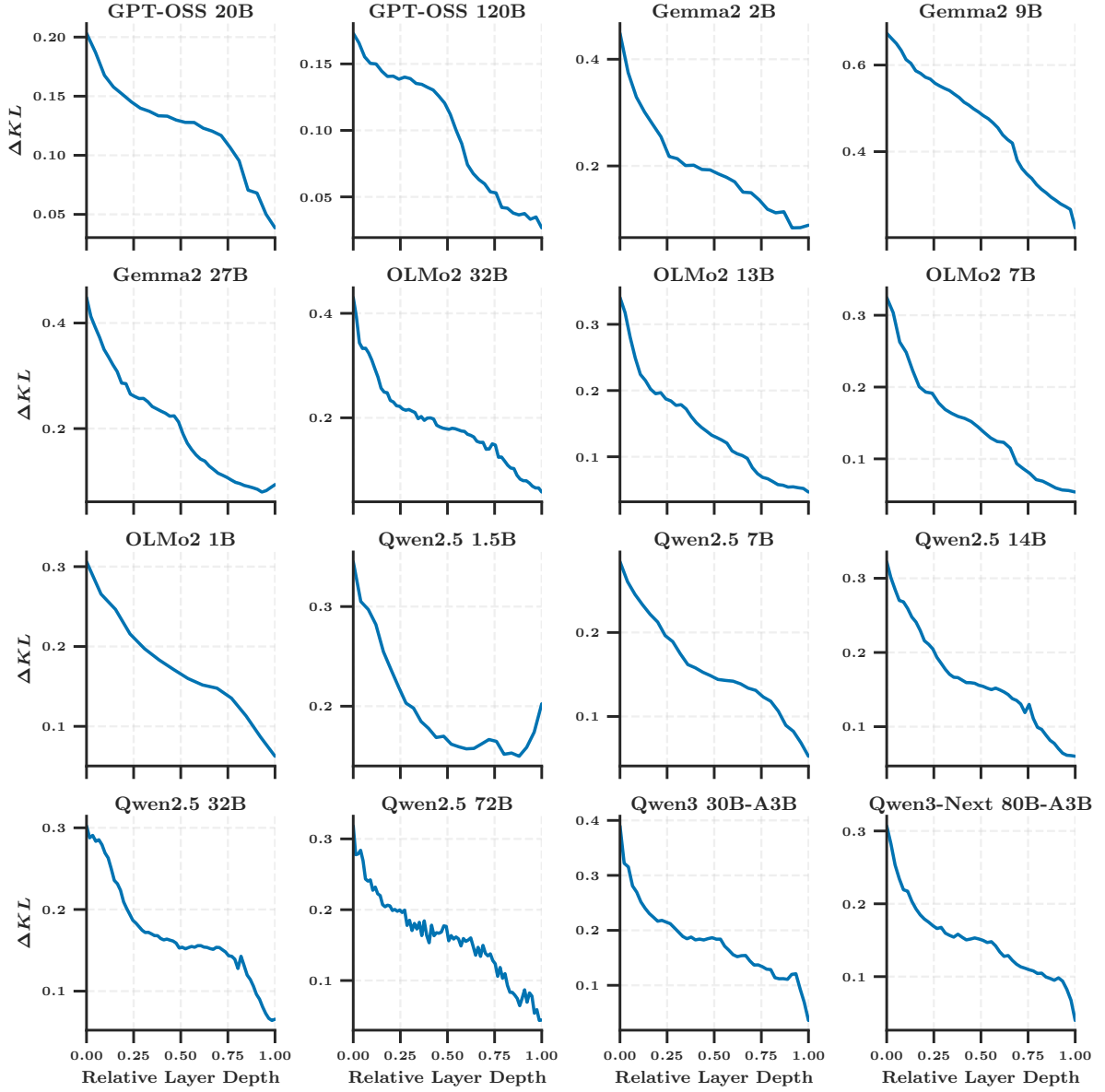


Figure 22: Per-layer KL improvement ΔKL_i of local adapters over G -only for all 16 evaluated models. Layer 0 consistently provides the largest benefit; the profile decreases with depth in every model, cross-validating the spectral prediction of §3.

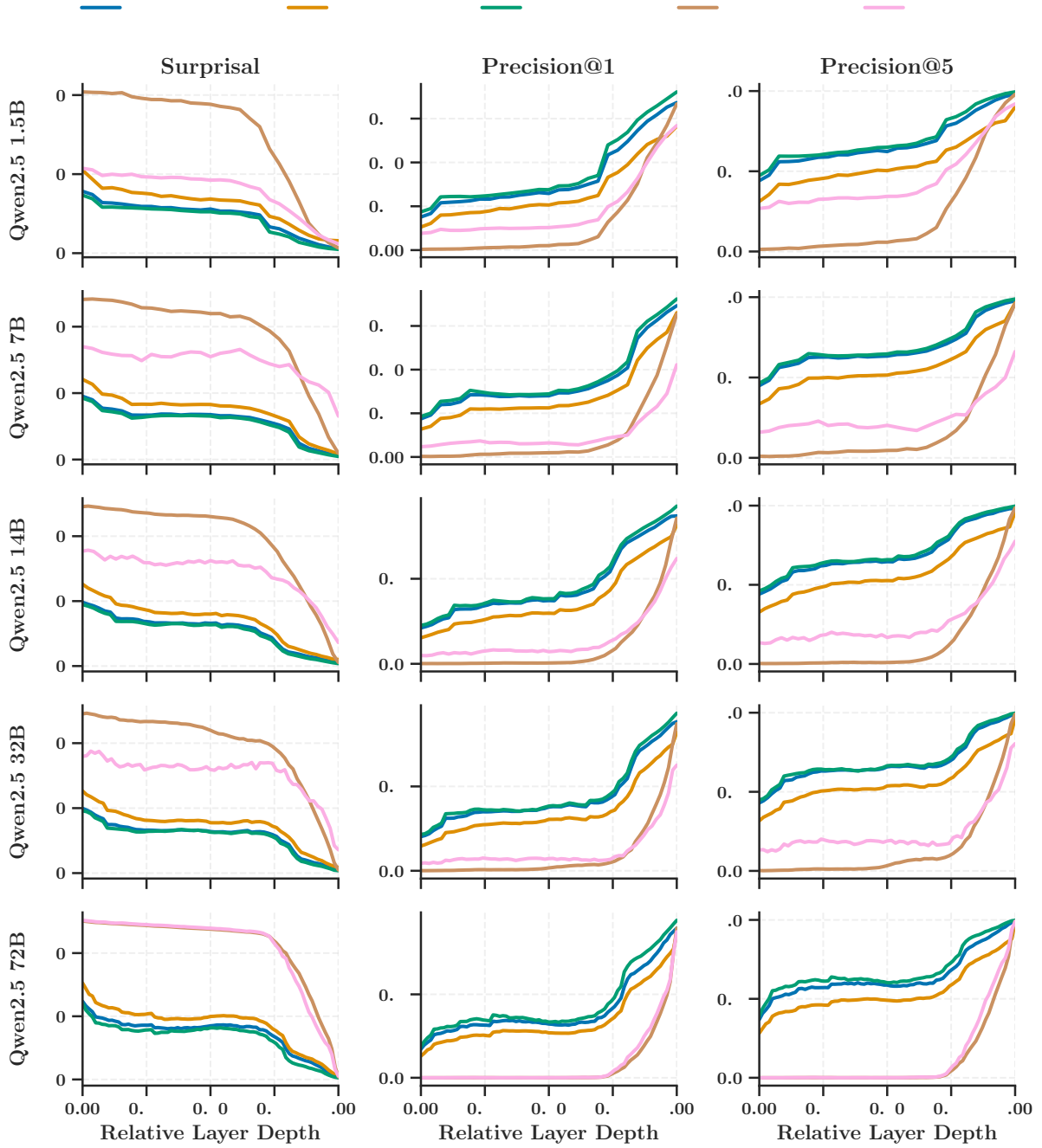


Figure 23: LSA metrics (Surprisal, Precision@1, Precision@5) averaged across 8 benchmarks for the Qwen2.5 family. AnchorLens and TunedLens are near-indistinguishable across all five scales; the performance gap over LoRALens is consistent throughout the network.

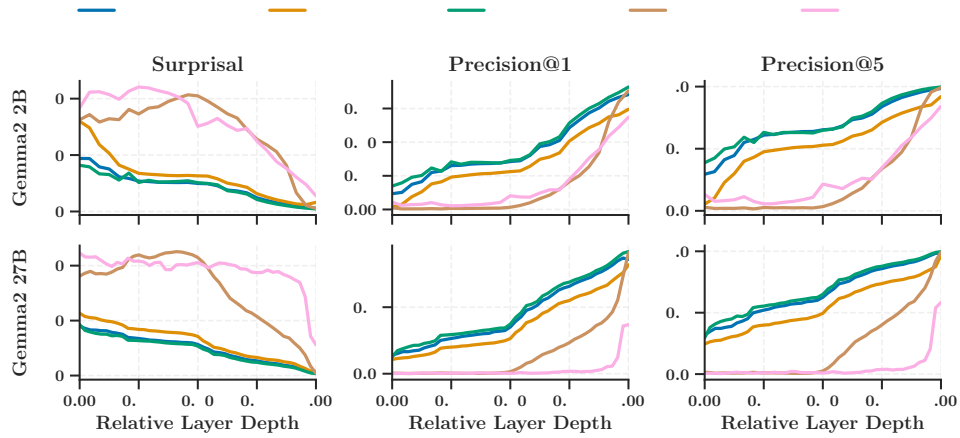


Figure 24: LSA metrics averaged across 8 benchmarks for Gemma2-2B and Gemma2-27B (Gemma2-9B follows the same trend and is omitted for space). Gemma2-2B shows near-zero Precision@1 until $i/L \approx 0.75$, with all methods rising sharply in the final layers. JumpToConclusion is particularly erratic on Gemma2-27B.

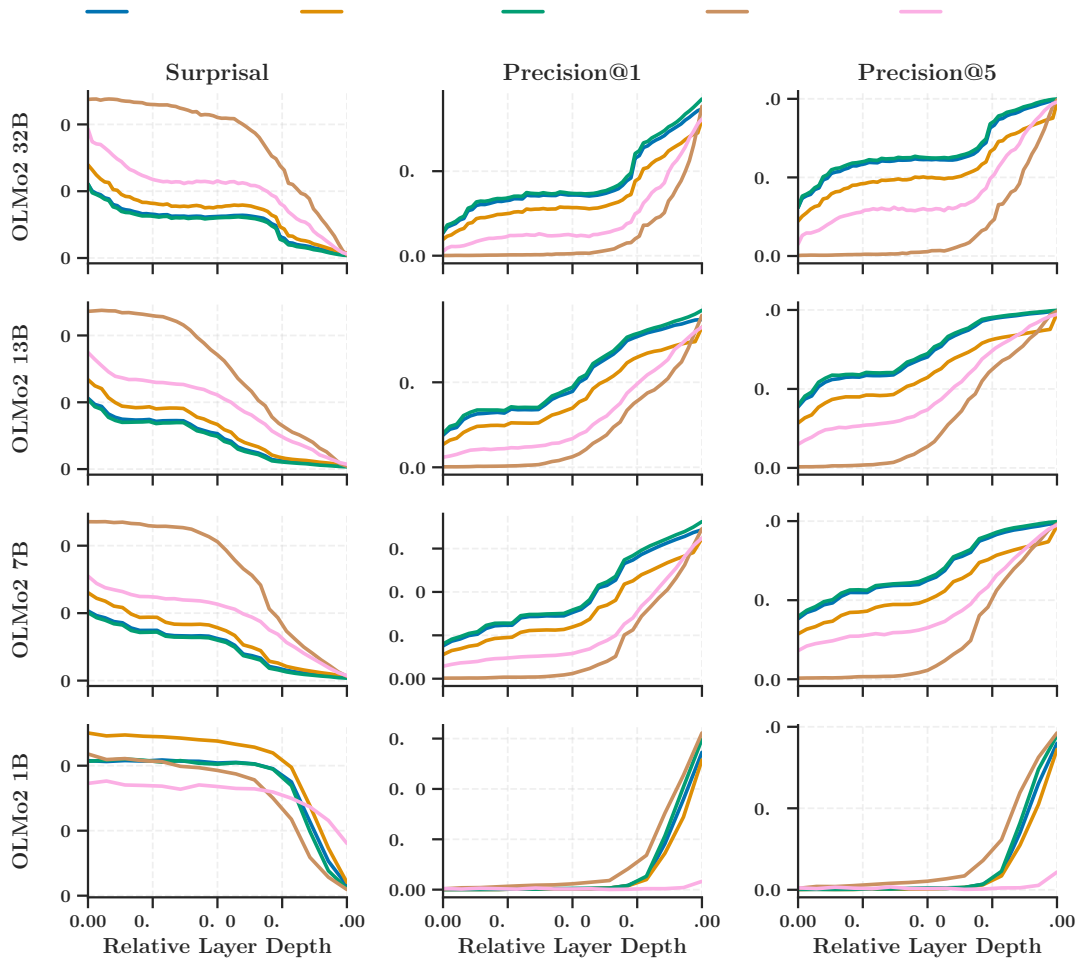


Figure 25: LSA metrics averaged across 8 benchmarks for the OLMo2 family. OLMo2-1B shows the most abrupt late-layer rise: Precision@1 remains near zero until $i/L \approx 0.85$, then rises steeply. The pattern is progressively earlier with increasing model size. AnchorLens and TabledLens are near-identical at every scale.

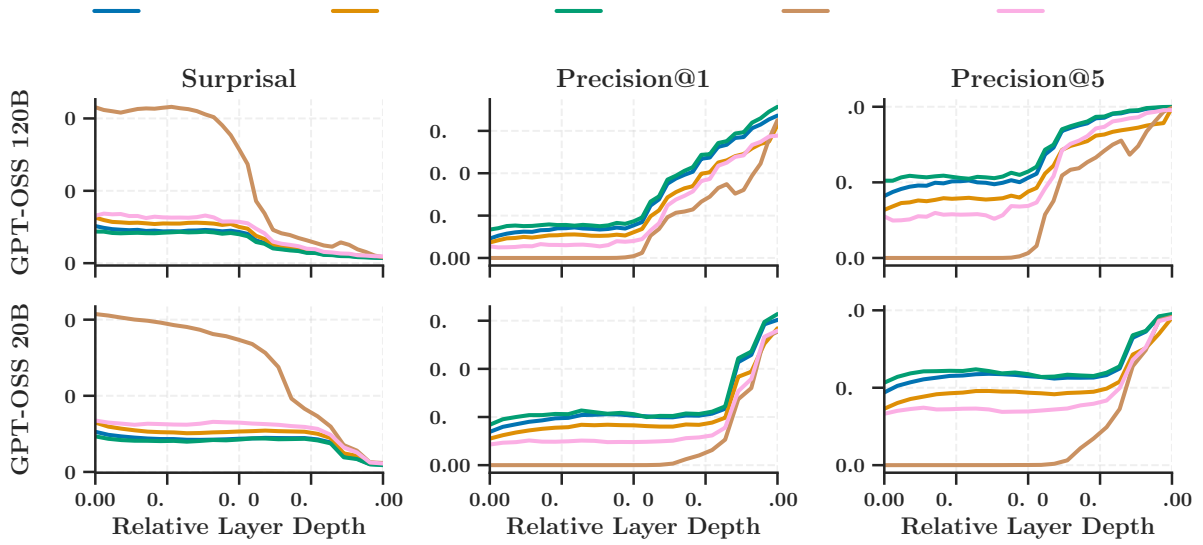


Figure 26: LSA metrics averaged across 8 benchmarks for the GPT-OSS family. AnchorLens and TunedLens are essentially indistinguishable on all three metrics for both models. LogitLens shows a pronounced Surprisal drop only in the final 25% of layers, consistent with its poor intermediate alignment.

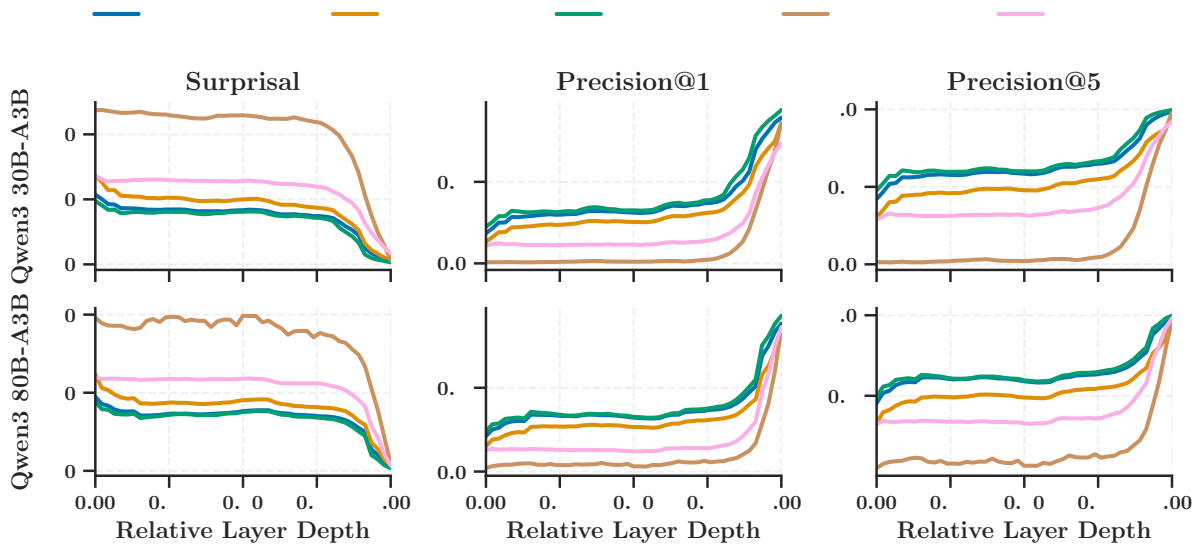


Figure 27: LSA metrics averaged across 8 benchmarks for the Qwen3 MoE family. The late-layer rise occurs at $i/L \approx 0.80$ for both models. AnchorLens and TunedLens are near-identical; LoRALens is visibly below on Precision@1 and Precision@5 throughout the network.

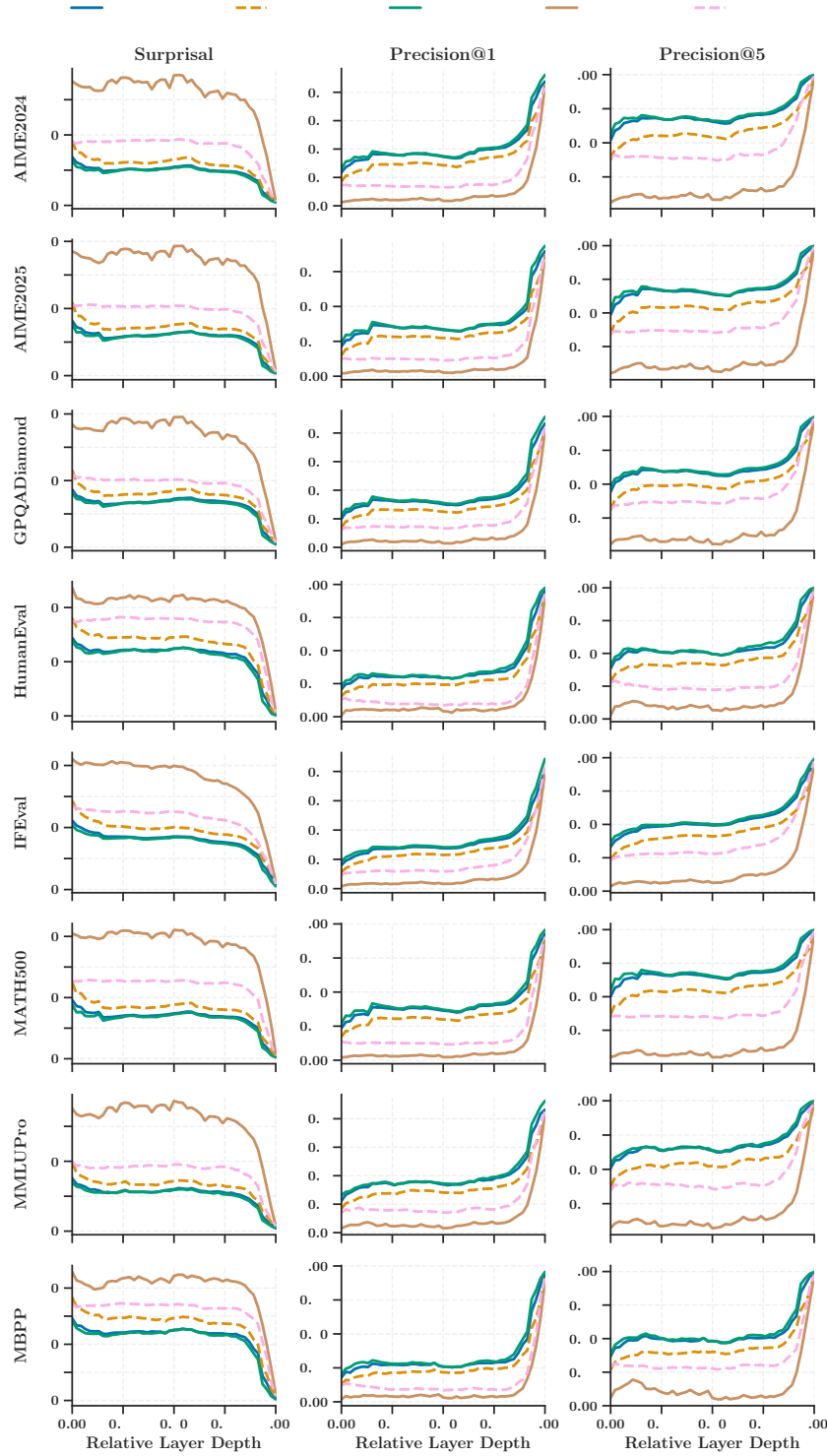


Figure 28: Per-benchmark LSA metrics (Surprisal, Precision@1, Precision@5) for Qwen3-80B-A3B across all evaluated benchmarks. Curves are near-identical across benchmarks for every method, confirming that the benchmark-averaged results in §6 are representative. AIME benchmarks show slightly higher Surprisal at early layers, consistent with their greater task difficulty; the method ranking is preserved throughout.