Tracing Logit Trajectories Across Layer Depth: Dataset-Level Explainability for Language Models

Jeesu Jung
KAIST
jisu.jung5@gmail.com

Sangkeun Jung*
Chungnam National University
hugmanskj@gmail.com

Abstract

Sentence-level explanations can miss the bigger picture of how a black-box model behaves across data, which matters most for complex criteria like safety that cannot be defined by a single rule. We **trace Logit-Trajectory**, which tracks adjacent-layer logit updates as vectors and aggregates them into a reproducible dataset-level trajectory pattern, enabling *depth-wise explainability* through signals such as coherence and angular rotation. Across 6 languages and 5 NLP tasks, we show these trajectory summaries reveal consistent depth-wise patterns that divergence- and similarity-based baselines often wash out due to scalarization. As a case study where dataset-level intermediate decision structure matters, we evaluate safety classification, reporting both trajectory-level visual separability and classification performance.

1 Introduction

Recent work on black-box explainability for language models has largely focused on per-sentence or per-example analyses(Vaswani et al., 2017; Brown et al., 2020; Ouyang et al., 2022). However, explanations grounded in individual instances rarely yield a consistent account of model behavior at the dataset level(Ouyang et al., 2022; Bai et al., 2022; Wei et al., 2023; Gehman et al., 2020). This limitation becomes more acute as modern LLMs are deployed across diverse tasks and conditions within a single model. In particular, when the target criterion is inherently multi-faceted—such as safety, which cannot be captured by a single scalar objective—per-example explanations are insufficient. What is needed instead is a dataset-level analysis that traces the intermediate structure of decision formation as it unfolds across layer depth.

Lens-style methods(Figure 1(a)) such as the logit lens (nostalgebraist, 2020) and tuned lens (Belrose et al., 2023) decode intermediate predictions but

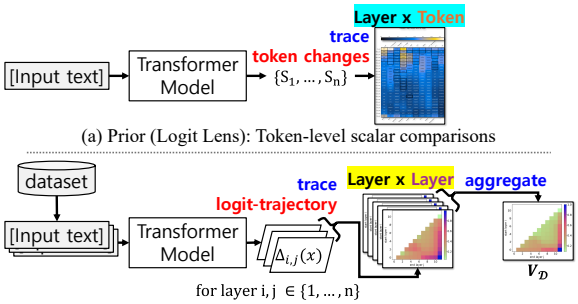


Figure 1: (a) Prior logit-lens readouts(nostalgebraist, 2020) trace a *layer*×*token* logit for a *single input*; such readouts wash out directional transitions, and aggregating them at the dataset level is difficult. (b) Our dataset-level trajectory: for each x, build a *layer*×*layer* Δ-map $\Delta_{i,j}(x)$ and aggregate over $\mathcal{D}$ to obtain a dataset-level pattern.

are essentially layerwise snapshots, typically analyzed per example. Layerwise metrics reduce each transition to a single number, e.g., KL divergence on decoded outputs (Kishino et al., 2025) or similarity measures in representation space (Kornblith et al., 2019; Morcos et al., 2018). At the dataset level, *depth-wise transition patterns can wash out under scalarization*, obscuring stable structure in how decisions evolve across depth.

A natural concern is that summing adjacent updates recovers an endpoint statistic $z^{(L)} - z^{(0)}$. Our focus is the *intermediate structure*—centroid coherence and angular rotation—which is not recoverable from the endpoint alone and becomes stable under dataset-level aggregation.

We introduce ***Logit-Trajectory***, a dataset-level framework that preserves depth-wise transitions by tracking adjacent logit updates. These patterns enable practitioners to (1) *localize* where target task decisions stabilize, (2) *highlight* high-impact transition points for debugging and intervention, and (3) *produces* task-discriminative trajectory for comparing models and detecting distribution shift—signals scalar summaries often miss. Instead of decod-

*Corresponding Author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17800–17823

Metric	What it captures (and misses)	Where it can fail (dataset-level)	What we observe
KL-divergence (nostalgebraist (2020))	Scalar *change distribution*; misses update *direction/turns*.	Averaging collapses mixed dynamics; emphasizes endpoint spikes over last-layer structure.	High `within` and high `between` (gap ≤ 0.3); spikes near final layers, weak mid-layer signal.
Cosine similarity (Jiang et al. (2025a))	Scalar *angular similarity*; misses *transition geometry* across depth.	Angular rotation average out; differences show mainly at early/late layers.	High `within` and high `between` (gap ≤ 0.05); mostly flat mid-layers.
Endpoint statistic Δ^{end}	Vector updates $\Delta^{(0\to L)}$; misses update *trajectories*.	No capturing internal-change→weaker dataset-level discrimination; variation by model is very large.	Smaller `within`–`between` gaps (up to 0.578); Failure of the p-value to converge for models with 70B+ parameters.
Logit-Trajectory (Ours)	Vector updates $\Delta^{(\ell\to\ell+1)}$; keeps directional structure (Centroid coherence/Angular rotation).	Retains turns/alignment under averaging; Provide learnable diagnostic signal	Larger `within`–`between` gaps (up to 0.636); learnable with logistic regression (pre-filter).

Table 1: Under dataset averaging, scalar metrics(KL-divergence, cosine similarity) can blur condition-specific structure; our Logit-Trajectory retains directional dynamics and yields larger within–between gaps. L is the model's total number of layers, and ℓ is an internal layer index with $1 \leq \ell < L$.

ing per-layer predictions, we model each layer-to-layer difference as a **vector update** in vocabulary space and aggregate updates across examples into a **dataset-averaged vector field** (centroid). We then measure **centroid-direction coherence** and **angular rotation across depth**, yielding reproducible signatures comparable across datasets, models, and conditions (Figure 1).

Finally, we use safety classification as a downstream case study in a diagnostic setting to demonstrate that trajectory features are *learnable*: safe and harmful subsets exhibit systematic differences in their trajectories, and a simple classifier can leverage early-layer transitions as a pre-filtering signal. We present this experiment as evidence that trajectory signals are learnable, not as a claim of state-of-the-art safety filtering.

Our main contributions are:

- **Logit-Trajectory:** A dataset-level trajectory pattern that aggregates adjacent logit updates into a centroid field and measures coherence and angular rotation across depth.
- **Trajectory pattern vs. scalar summaries:** Evidence that scalar transition metrics can wash out dataset-level structure through aggregation and mixing, whereas trajectory maps preserve separability.
- **Downstream case study:** A case study demonstrating that trajectory features are discriminative and support early-layer prediction in a safety classification setting.

2 Related Work

Lens-based decoding. Prior work decodes intermediate hidden states into vocabulary space (logit lens) and mitigates representation drift with learned per-layer decoders (tuned lens) (nostalgebraist, 2020; Belrose et al., 2023; Pal et al., 2023). These methods primarily provide per-layer snapshots; we instead model *adjacent-layer logit updates* and aggregate them to obtain stable dataset-level transitions.

Logit attribution and component decomposition. Logit-decomposition methods explain outputs by attributing contributions to tokens, layers, or internal components (Ferrando et al., 2023; Nguyen). In contrast, we focus on *layerwise logit update vectors* and their aggregation to capture dataset-level decision dynamics.

Representation similarity across layers. Layerwise similarity metrics (e.g., cosine, CKA) are widely used to characterize redundancy or stage boundaries (Kornblith et al., 2019; Jiang et al., 2025b). However, they capture geometry, not the *direction* of movement in vocabulary space; we instead track direction via aggregated adjacent-layer logit updates. Probing networks can also make such representations linearly or with a lightweight MLP separable (Agarwal et al., 2025; Sun et al., 2025).

Our key distinction: dataset-level transition beyond token-wise lenses. Lens-style methods are typically token-wise, providing per-prompt, per-layer snapshots. Rather than tracking distribution-level dynamics, we *aggregate adjacent-layer logit*

updates into a dataset-averaged vector field and analyzing its stable structure (centroid coherence and transitions). This yields a reproducible **decision flow** for an entire dataset and supports diagnostics such as early-layer predictive.

3 Preliminaries

We consider scalar baselines for layer-wise change: *KL divergence* (distributional shift) and *cosine similarity* (directional alignment). Since our quantities are aggregated at the *dataset level*, they are **not directly comparable to token-wise lens readouts**. Logit-Trajectories are complementary: they summarize adjacent-layer logit updates as vectors (via centroids and coherence), exposing directional structure and mixture effects that scalar metrics collapse.

KL divergence (nostalgebraist, 2020). Let $p^{(\ell)} = \mathrm{softmax}(z^{(\ell)})$. We use $\mathrm{KL}(p^{(\ell)} \| p^{(\ell+1)})$ or $\mathrm{KL}(p^{(\ell)} \| p^{(L)})$.

Cosine similarity (Jiang et al., 2025a). We use logit cosine $\cos(z^{(\ell)}, z^{(\ell+1)})$ and hidden cosine $\cos(h^{(\ell)}, h^{(\ell+1)})$.

4 Logit-Trajectory Tracing

This section introduces Logit-Trajectory, our core tool for analyzing layer-wise behavior. Notation is summarized in Appendix B.

Core idea. (1) At layer ℓ, logits $z^{(\ell)}$ encode the model's output preferences. (2) The adjacent-layer difference $\Delta^{(\ell \to \ell+1)}$ captures how these preferences rotate as depth increases. (3) Keeping Δ as a vector (rather than a scalar) preserves both change magnitude and direction, indicating which tokens the model moves toward. Unlike snapshot-based probes that decode layers independently, we view inference as a sequence of updates and aggregate trajectories across the dataset for stable, comparable analysis.

4.1 Logit-Trajectory Definition

To capture layer-wise *directional* changes, we use adjacent-layer logit differences. For a model f_θ with L layers, let $h^{(\ell)}(x)$ be the hidden state at layer ℓ for input x, and let t^* denote the decision token (default: the last valid token). We define the vocab-space logit at t^* as

$$z^{(\ell)}(x) = \mathrm{LMHead}\Big(h^{(\ell)}(x)_{t^*}\Big) \in \mathbb{R}^{|V|}, \quad (1)$$

where V is the vocabulary. The vector update across an adjacent-layer transition is

$$\Delta^{(\ell \to \ell+1)}(x) = z^{(\ell+1)}(x) - z^{(\ell)}(x) \in \mathbb{R}^{|V|}. \quad (2)$$

If needed, we generalize to an arbitrary layer pair (ℓ_i, ℓ_j) as $\Delta^{(\ell_i \to \ell_j)}(x) = z^{(\ell_j)}(x) - z^{(\ell_i)}(x)$.

4.2 Extracting $z^{(\ell)}$: Architecture-agnostic Logit Projection

To trace Logit-Trajectory applicable to encoder-only, encoder–decoder, and decoder-only models, we compute $z^{(\ell)}$ by applying the LM head to the output of each layer block. By default, t^* is the *last valid token*; for encoder–decoder models, we apply the same definition using the logits at the first decoder step.

5 Dataset-level Logit-Trajectory Aggregation

This section aggregates transition vectors Δ at the dataset level to quantify and visualize representative depth-wise transition and consistency patterns as computation proceeds through depth.

5.1 Centroid Coherence and Angular Rotation

For a transition $t = \ell \to \ell + 1$, we define the dataset-level representative direction (centroid) as

$$c_t = \mathbb{E}_{x \sim \mathcal{D}}[\Delta^{(t)}(x)]. \quad (3)$$

We measure how well an individual sample's update aligns with the centroid via cosine similarity:

$$\mathrm{Agr}_t(x) = \cos\big(\Delta^{(t)}(x), c_t\big). \quad (4)$$

The centroid coherence $\mathbb{E}_x[\mathrm{Agr}_t(x)]$ yields a layer-wise consistency curve over depth.

Subset analysis. For any subset $\mathcal{D}' \subset \mathcal{D}$ (language/task/label/safety condition), we analogously compute $c_t(\mathcal{D}')$ and $\mathrm{Agr}_t(\mathcal{D}')$ to compare decision-flow patterns across conditions.

5.2 Dataset-level Layer×Layer Visualization

To visualize the structure of updates at the dataset level, we define a length-normalized mean Logit-Trajectory for a layer span (i, j). Given adjacent transitions $\Delta^{(\ell \to \ell+1)}(x)$, we define

$$\bar{\Delta}^{(i \to j)}(x) = \frac{1}{j-i} \sum_{\ell=i}^{j-1} \Delta^{(\ell \to \ell+1)}(x) \in \mathbb{R}^{|V|}, \quad (5)$$

which represents the average update trajectory pattern over the layer span $[i, j]$. The dataset-level layer-pair centroid is

$$C_{i,j} = \mathbb{E}_{x\sim\mathcal{D}}\left[\bar{\Delta}^{(i\to j)}(x)\right]. \quad (6)$$

We construct an upper-triangular heatmap by mapping $C_{i,j}$ to a scalar:

$$H^{\text{mag}}_{i,j} = \|C_{i,j}\|, \quad (7)$$

$$H^{\cos}_{i,j} = \cos(C_{i,j},\, C_{0,L-1})\,, \quad (8)$$

$$H^{\text{avg}}_{i,j} = \text{avg}(C_{i,j})\,. \quad (9)$$

Here, H^{mag} highlights spans with large average changes, $H^{\cos}$ measures alignment with the overall depth transition $C_{0,L-1}$, and H^{avg} captures the mean signed tendency of the centroid vector. To avoid span-length bias, we use mean aggregation.

For visualization of H^{avg}, we apply min–max scaling over all valid layer spans in the full layer range $(0 \sim L-1)$:

$$\tilde{H}^{\text{avg}}_{i,j} = \frac{H^{\text{avg}}_{i,j} - \min_{(p,q)} H^{\text{avg}}_{p,q}}{\max_{(p,q)} H^{\text{avg}}_{p,q} - \min_{(p,q)} H^{\text{avg}}_{p,q}}, \quad (10)$$

where the minimum and maximum are taken over all valid layer pairs (p, q) with $0 \le p < q \le L-1$.

We compute layerwise logits at a fixed "decision" position–last input token(encoder: `[CLS]` (Devlin et al., 2019), decoder: `[EOS]` (Radford et al., 2018))– to avoid dependence on input, output sentence lengths. See Appendix A.2 for further details.

6 Experimental Setup

We describe datasets, models, extraction settings, and baselines. Additional details are in Appendix A[1].

6.1 Dataset and Task Formulation

- **Multilingual multi-task (Aya) (Singh et al., 2024).**[2]
- **Languages.** English, German, Spanish, Hindi, Japanese, Korean.
- **Task focus.** We mainly report results on Paraphrase Identification, Event Linking, Question-Answering, Dialogue, Summarization (long-form generation), while the pipeline supports other Aya task types.
- **Downstream case study (binary safety).** A CSV-based dataset with *safe/unsafe* labels, split into train/test for predictive evaluation.

Detailed statistics are provided in Appendix A.1 and Appendix A.7.

6.2 Models

We include diverse architectures:

- **Encoder-only:** `bert-base-cased`, `bert-base-multilingual-cased` (Devlin et al., 2019).
- **Encoder–decoder:** `T5-large` (Raffel et al., 2020) , `flan-T5-large` (Chung et al., 2022) (encoder component).
- **Decoder-only:** `Qwen2.5`-(3B/7B/72B)-Instruct (Yang et al., 2024), `Llama`-(3.2-3B, 3.1-8B, 3.3-70B)-Instruct (Llama, 2024).

We discuss the main results for the decoder-only models Llama-3 and Qwen2.5. Results for the other models are reported in Appendix D.

6.3 Baselines and Statistical Testing

- **Baselines.** The **end-to-end difference** Δ_{end} $(= z^{(L)} - z^{(0)})$, which does not track intermediate layers. We also consider adjacent-layer **KL divergence** and **cosine similarity**. (For a detailed explanation, see Appendix A.5.)
- **Dataset-level stability (within vs. between).**
  - **Within:** same condition, different seeds.
  - **Between:** different conditions (e.g., tasks/languages) under the same model.
- **Uncertainty & significance.** Bootstrap CIs (default 300 resamples) and permutation tests (default 5,000 permutations).

7 Results

Throughout our analysis, we connect visual trajectory patterns to quantitative criteria: visually stable regions correspond to high centroid coherence (Eq. 4) and low angular rotation, while visually irregular or spiky patterns correspond to lower alignment and higher directional variance. This provides a consistent bridge between qualitative observations and measurable trajectory properties.

We show that vector-valued logit capture dataset-level structure that scalar summaries (KL-divergence/cosine similarity) often wash out.

[1] Code and data will be made publicly available upon acceptance.

[2] `CohereLabs/aya_collection_language_split` (`train`, field: `inputs`).

Model	Metric	Within	Between	δ
BERT -base-cased	KL	1.000	0.973	0.027†
	Cos	1.000	0.989	0.011†
	Δ_{end}	1.000	0.682	0.318†
	Δ	1.000	0.678	**0.322**†
BERT -base-multilingual-cased	KL	1.000	0.995	0.005†
	Cos	1.000	0.990	0.010†
	Δ_{end}	1.000	0.737	0.263†
	Δ	1.000	0.661	**0.339**†
T5 -large	KL	1.000	0.709	**0.291**†
	Cos	1.000	1.000	0.000†
	Δ_{end}	1.000	0.998	0.002†
	Δ	1.000	0.987	0.013†
Flan-T5 -large	KL	1.000	0.757	**0.243**†
	Cos	1.000	1.000	0.000†
	Δ	1.000	0.881	0.119†
	Δ_{end}	1.000	0.950	0.050†
Qwen2.5 -3B-Instruct	KL	1.000	0.751	0.249†
	Cos	1.000	0.999	0.001†
	Δ_{end}	1.000	0.870	0.130†
	Δ	1.000	0.470	**0.530**†
Qwen2.5 -7B-Instruct	KL	1.000	0.952	0.048†
	Cos	1.000	0.993	0.007†
	Δ_{end}	1.000	0.422	**0.578**†
	Δ	1.000	0.621	0.379†
Qwen2.5 -72B-Instruct	KL	0.933	0.758	**0.175**†
	Cos	1.000	0.997	0.003†
	Δ_{end}	0.034	0.149	-0.115
	Δ	0.004	0.267	-0.264†
Llama-3.2 -3B-Instruct	KL	1.000	0.909	0.091†
	Cos	1.000	0.996	0.004†
	Δ_{end}	1.000	0.609	0.390†
	Δ	1.000	0.364	**0.636**†
Llama-3.1 -8B-Instruct	KL	1.000	0.949	0.051†
	Cos	1.000	0.995	0.005†
	Δ_{end}	0.999	0.617	0.383†
	Δ	1.000	0.393	**0.607**†
Llama-3.3 -70B-Instruct	KL	0.998	0.985	**0.014**†
	Cos	1.000	1.000	0.000†
	Δ_{end}	0.025	0.080	-0.054
	Δ	0.011	0.163	-0.152†

Table 2: Trajectory generalization via within-/between-dataset similarity (higher is better). δ = mean(within − between); **bold** = max δ; †: $p < 0.001$ (within ≠ between). Baselines: KL-divergence(KL), Cosine Similarity(Cos), and $z^{(L)} - z^{(0)}$(Δ_{end}, it uses only the start and end representations without accounting for intermediate depth). Δ is generally more discriminative; for 70B+ models, higher within-dataset trajectory diversity may reduce within similarity, shrinking δ.

Δ_{end}—which does not account for layer-wise changes across depth—does not yield consistent results across models (Table 1). Our experiments establish three findings: 1) a reproducible *layer-depth wise trajectory pattern*, 2) summarizes its pattern with a layer×layer upper-triangular heatmap, and 3) reveals convergence and change-points through trajectory-aware diagnostics. Pattern variations across models, languages, and tasks are reported in Appendix D. Appendix D.5 presents stress tests for cross-condition confusability, and Appendix C compares computational costs between our method and the baselines.

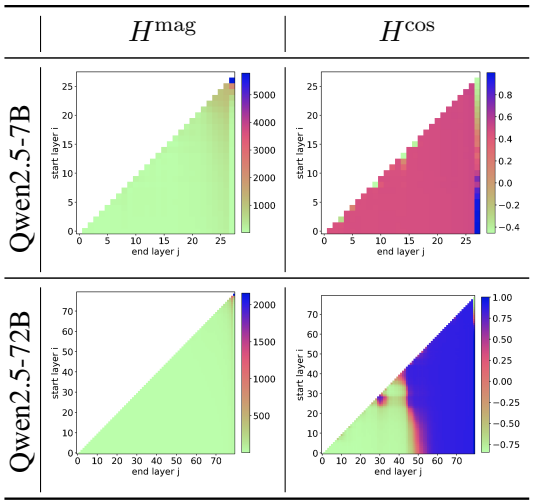


Figure 2: Layer×layer upper-triangular heatmaps of dataset-level logit updates for Qwen2.5-7B-Instruct, -72B-Instruct (brighter = larger change). H^{mag}: update magnitude; $H^{\cos}$: alignment to global update $C_{0,L-1}$. Early layers: larger/less-aligned; late layers: smaller/more-aligned—progressive stabilization.

7.1 Dataset-specific Logit-Trajectory Patterns

We test whether our method yields *reproducible trajectory patterns* at the dataset (condition) level, and whether preserving Δ as a vector provides an advantage over scalarized transition metrics.

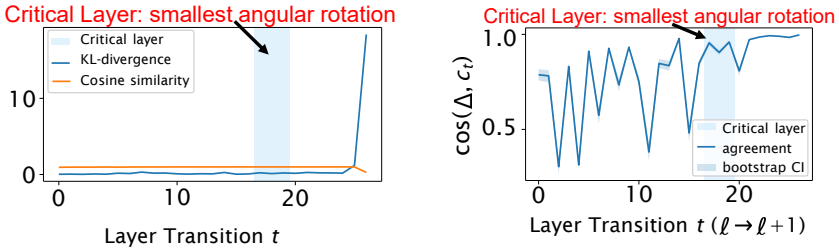
For each condition g, we compute transition centroids $\{c_t^{(g)}\}_{t=0}^{L-2}$ and define

$$\mathrm{Sim}(g, g') = \frac{1}{L-1} \sum_{t=0}^{L-2} \cos\big(c_t^{(g)}, c_t^{(g')}\big).$$

`within` measures coherence within condition g, while `between` measures similarity across conditions.

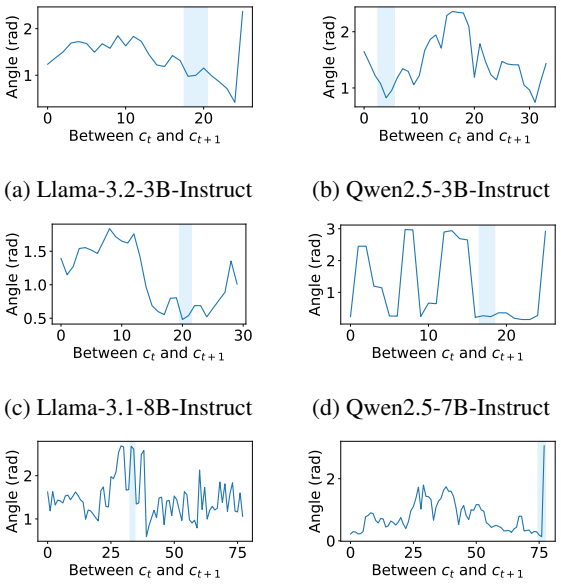
Ideally, a method should yield high `within` and low `between`. For models up to 8B parameters, Δ consistently exhibits the most desirable separation (Table 2). The full similarity distributions are provided in Appendix D.1.

In contrast, for very large models (70B+), Δ and Δ_{end} exhibit atypical behavior: both the `within` and `between` values are very low, while KL-divergence and cosine similarity are both high. We observe this pattern consistently across all tasks used in the experiments. We hypothesize that these models admit *more diverse trajectories even within a single condition*, which depresses `within` similarity. Moreover, this tendency appears to be overly compressed when compared against other metrics. Our goal is not to summarize behavior into a single scalar; rather, we aim to identify a dataset-level two-dimensional fingerprint. As shown in Sections 7.2–



(a) baselines (b) Logit-Trajectory

Figure 3: Transition-wise coherence on Qwen2.5-7B, English single turn dialogue task. For each transition t, we compute $\cos\big(\Delta^{(t)}(x), c_t\big)$ with $c_t = \mathbb{E}_x[\Delta^{(t)}(x)]$, and report the mean with confidence intervals. Higher values indicate more consistent update trajectories across samples. We excluded Δ_{end} since it yields no depth-wise result.



(a) Llama-3.2-3B-Instruct (b) Qwen2.5-3B-Instruct

(c) Llama-3.1-8B-Instruct (d) Qwen2.5-7B-Instruct

(e) Llama-3.3-70B-Instruct (f) Qwen2.5-72B-Instruct

Figure 4: Dialogue task results for Llama and Qwen. We quantify layer-wise angular rotation as $\angle(c_t, c_{t+1})$, where $c_t = \mathbb{E}_x[\Delta^{(t)}(x)]$. Larger angles indicate abrupt shifts in the average update trajectory. The Critical Layer region marks the lowest rotations, corresponding to layers that support stable task-wise decisions.

9, our methodology retains sufficient discriminative ability at the dataset level.

7.2 Trajectory Visualization Across Layer Depth: Layer×Layer Heatmaps

We summarize dataset-level layer×layer dynamics with an upper-triangular heatmap, visualizing 1) update magnitude $H^{\text{mag}}_{i,j}$ and 2) alignment with the overall transition $H^{\cos}_{i,j}$ from centroids $C_{i,j}$ (Method 5.2).

Figure 2 shows both maps: H^{mag} highlights *where* updates are large (often increasing toward later layers), while $H^{\cos}$ indicates *how* closely each update aligns with the global transition $C_{0,L-1}$ (typically low in early layers and increasing with depth). We quantitatively validate this visual pattern via the coherence and rotation analyses in Sec. 7.3, where increasing alignment and decreasing angular variation across depth confirm the observed stabilization trend.

Across models, H^{mag} exhibits broadly similar patterns, and because it captures only absolute changes, it suggests that input-specific information is rapidly filtered out as representations propagate through depth. Overall, models exhibit **larger rotate, less-aligned changes in early layers** and **small rotate, more-aligned updates in later layers**, consistent with progressive stabilization. Subset heatmaps by language/task reveal additional condition-specific structure (Appendix D.3).

7.3 Trajectory-aware Dynamics: Centroid Coherence and Angular Rotations

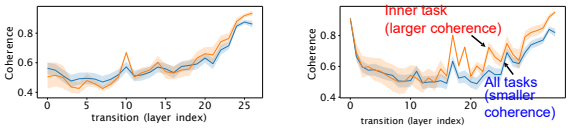
In this work, we interpret these quantities as operational criteria for trajectory structure: centroid coherence captures the consistency of update directions across samples, while angular rotation captures transitions in the dominant update direction across depth. Preserving Δ as a vector exposes directional structure that scalar summaries (KL divergence, cosine similarity) obscure. We measure (i) alignment between sample transitions and the dataset centroid (*centroid coherence*) and (ii) depth locations where the centroid direction turns (*angular rotations*).

Centroid coherence. Figure 3 shows centroid coherence $\mathbb{E}_x[\text{Agr}_t(x)]$. Unlike KL-divergence or cosine similarity, which wash out depth-wise directionality under scalar aggregation, **Logit-Transition preserves intermediate-layer directional dynamics**. Results for the other models are provided in Appendix D.2.

Angular rotations. Figure 4 plots the turning angle between continuous centroid directions across depth. Rotation profiles depend on architecture but

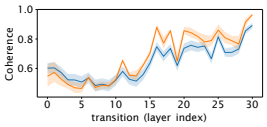
Model	Layer span (len=4)	Flip_{all}	Flip_{bd}	δ_{Margin} (mean)	δ_{Margin} (med)
Qwen2.5-3B-Instruct	Fragile (16–19)	0.248	0.500	+1.064	+1.051
	Critical (3–6)	**0.282**	**0.540**	+0.488	+0.508
	Random controls	0.420 ± 0.066	0.518 ± 0.023	–	–
Qwen2.5-7B-Instruct	Fragile (7–10)	0.448	0.510	+1.026	+1.152
	Critical (16–19)	**0.554**	**0.588**	+1.754	+1.848
	Random controls	0.481 ± 0.135	0.575 ± 0.074	–	–
Qwen2.5-72B-Instruct	Fragile (76–79)	**0.220**	**0.460**	+0.111	+0.099
	Critical (73–76)	0.020	0.087	−0.011	0.000
	Random controls	0.209 ± 0.259	0.344 ± 0.124	–	–
Llama-3.2-3B-Instruct	Fragile (24–27)	0.232	0.360	-0.483	-0.496
	Critical (17–20)	**0.256**	**0.450**	-0.383	-0.407
	Random controls	0.335 ± 0.220	0.446 ± 0.100	–	–
Llama-3.1-8B-Instruct	Fragile (8–11)	0.106	0.360	+0.031	+0.087
	Critical (20–23)	**0.210**	**0.400**	-0.335	-0.308
	Random controls	0.165 ± 0.207	0.342 ± 0.126	–	–
Llama-3.3-70B-Instruct	Fragile (29–32)	0.018	0.078	+0.259	+0.259
	Critical (33–36)	**0.020**	**0.087**	-0.011	0.000
	Random controls	0.016 ± 0.007	0.072 ± 0.037	–	–

Table 3: Layer-freezing (block-skip) ablation for decoder-only models. Freeze one contiguous 4-layer span: high-rotate (Fragile layer), low-rotate (Critical layer), or (5 repetitions). Report label-flip rates (all / borderline; bottom 20% by |baseline margin|) and decision-margin change (after−before). Random spans: mean ± std over 5 uniformly sampled controls. Flip↑ ⇒ freezing that span causes more final label reversals; δ_{Margin} ↑ ⇒ freezing that span perturbs confidence in the chosen label more. **Bold** marks the highest value among compared spans for each metric.

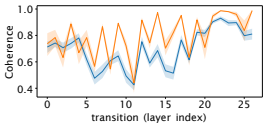


(a) Llama-3.2-3B-Instruct

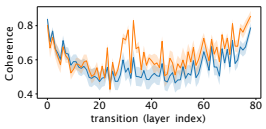
(b) Qwen2.5-3B-Instruct



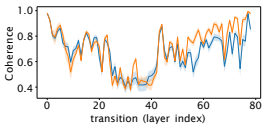
(c) Llama-3.1-8B-Instruct



(d) Qwen2.5-7B-Instruct



(e) Llama-3.3-70B-Instruct



(f) Qwen2.5-72B-Instruct

Figure 5: Centroid coherence (within-dialogue vs. overall) for Llama and Qwen. Coherence is consistently higher within-dialogue—especially in later layers—and decreases when tasks are mixed (signal dilution).

are broadly consistent across scales, suggesting recurring structural transitions. We call the minimum-rotation region the *Critical Layer*, where decision dynamics concentrate; we compute phase rotation with a stabilized normalized cosine (add ϵ, clamp to $[-1, 1]$).

8 Ablation Studies

We conduct ablation studies to validate two questions: 1) How do directional variations affect the model's decisions? 2) Is the model's trajectory pattern over the full dataset always similar, or does it meaningfully differ across datasets?

8.1 How Angular Rotations Affect Model Decisions.

We measured how often predictions flip when layers are frozen, where a higher flip rate indicates greater variability and a larger change in (δ_{Margin}) indicates lower confidence.

Freezing the *Fragile Layer* (largest rotation) usually has more filps because the decision is not aligned. By contrast, freezing a *Critical Layer*(smallest rotation; stable update direction) turning region often causes large margin shifts and frequent label flips, especially on borderline inputs (Table 3). This indicates that rotation-defined layers are not just descriptive: they **provide functional leverage that stabilizes—or destabilizes—the final decision**.

8.2 Target Task vs. All Tasks: Layer-wise Coherence Trends

As an example, we compare the coherence trend for a canonical instruction-following task—a single-turn dialogue—against the coherence trend computed over all six tasks combined. As shown in Figure 5, the same qualitative result holds across all models: the overall pattern is similar, but coherence is **stronger for the target task** (dialogue) and tends to increase more sharply in later layers. We attribute this to a division of labor across layers: **early–mid layers determine the alignment direction, while later layers focus on output sta-**

bilization.

9 Case Study: Safety Classifier (Diagnostic)

As a **case study**, We use safety classification to validate that Logit-Trajectory features provide *learnable* diagnostic signals, and to illustrate their potential as a simple monitoring/pre-filter signal (not a claim of state-of-the-art safety filtering). We conduct a safety classification case study using logit-trajectory features on four public English safety datasets (Appendix A.7).

9.1 Logit-Trajectory Pattern of Safety Classification

Figure 6 shows qualitative separation: the upper-triangular $H^{\cos}$ patterns differ systematically between safe and harmful inputs, even in smaller models. For safe inputs, the safe samples exhibit generally simple and smoothly continuous variations, whereas harmful samples contain more frequent and sharper *spiky* points. These spikes suggest that, in **harmful cases, the model cautiously re-align layers** in order to understand the implied harmful intent and formulate an appropriate response.

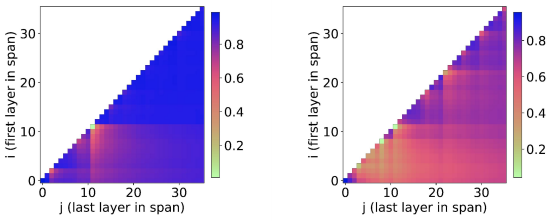
9.2 Diagnostic Pre-filter(Logistic regression)

To quantify discriminability and *prediction depth*, we train a logistic-regression probe on only the first k layer-to-layer transitions, asking how early the final label can be predicted.

Concretely, we feed Logit-Trajectory features (per-transition Δ, cumulative $\sum_t \Delta$) into a small regression model to predict binary labels (safe/harmful).

- Train **logistic regression** on trajectory-derived features; evaluate on held-out test data.
- Sweep the number of transitions k and report **AUC**; compare Logit-Trajectory vs. KL/cosine features under the same protocol.

Figure 7 shows that Logit-Trajectory features reach higher AUC with fewer layers than KL-divergence or cosine-similarity features;Across three different random seeds, only Δ exceeds an AUC of 0.78 (F1 = 0.715, precision = 0.70, recall = 0.73). Considering the performance ceiling (F1 = 0.80) reported in prior work (van Aken et al., 2018), these results indicate that **even a simple linear baseline (logistic regression) captures meaningful predictive signal**, suggesting that the task is already reasonably separable prior to applying more



(a) safe data (b) harmful data

Figure 6: Layer×Layer upper-triangular heatmaps visualizing(H^{avg}) dataset-level on Qwen2.5-3B-Instruct. Brighter regions indicate larger changes.

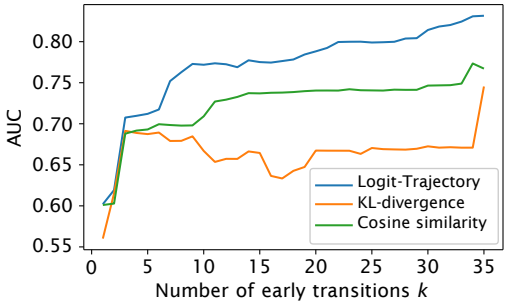


Figure 7: Early-exit prediction over early-transition count k (first k-layer only), Qwen2.5-3B-Instruct. Same classifier for Logit-Trajectory/KL-divergence/cosine similarity; higher AUC at smaller k indicates earlier decision formation.

sophisticated models. This, in turn, suggests that Logit-Trajectory captures decision stabilization, enabling early-exit prediction and safety-relevant monitoring (total results in Appendix D.4).

10 Discussion

Stable dataset-level signals live in intermediate structure, not endpoints. Although adjacent updates can be telescoped into an endpoint (e.g., $z^{(L)} - z^{(0)}$), the dataset-stable information is carried by the *intermediate structure* of trajectories. Centroid coherence and angular rotations stabilize under dataset aggregation and indicate where decisions consolidate across depth.

Why scalar baselines blur dataset-level discrimination. Scalar summaries such as KL-divergence and cosine similarity capture the *amount* of change, but tend to be high both within and across conditions, yielding small separability gaps at the dataset level (Table 1). By preserving updates as *vectors*, Logit-Trajectory Maps expose directional structure and yield repeatable, condition-specific signatures.

Angular rotations are important to decision.

Interventions around rotate-adjacent layers (e.g., freezing) substantially change margins and flip rates, suggesting these transitions have causal leverage over the final decision.

Practical implication: efficient probing from early layers. Beyond visualization, trajectory features are learnable with simple classifiers and support earlier prediction using fewer transitions than scalar features, enabling fast diagnostics (including safety pre-filtering as a use case).

Scalability / future directions. At present, both the trajectory-tracking method and the aggregation scheme are intentionally simple and intuitive. It would be valuable to adopt more sophisticated algorithms and to broaden the range of datasets and settings to which the approach is applied.

11 Conclusion

We introduced *Logit-Trajectory*, which represents layer-to-layer logit updates as directional vectors. We aggregate these vectors at the dataset level to reveal reproducible decision-dynamics trajectories across model architectures, scales, and families in multitask, multilingual settings. Empirically, these trajectories expose stage-wise structure over depth and provide signals that are useful beyond visualization, supporting downstream prediction in a practical case study.

Our findings highlight a simple principle: preserving layerwise *trajectories* enables stable dataset-level comparisons that scalar summaries often obscure. We hope this motivates future work that integrates dataset-level trajectory summaries with token-level lenses and targeted interventions to better understand and monitor Language Model decision formation.

Limitations

Due to page constraints and our intent to position this work as an initial exploration of dataset-level aggregation, further experimental expansion is warranted. In particular, while we conducted experiments across multiple models, languages, and tasks, the space limitations of the paper prevented us from including more detailed analyses and additional ablations.

First, we use Δ_{end}, KL divergence, and cosine similarity as reference metrics for our Δ-based representations primarily because they are training-free, enabling a lightweight initial comparison without introducing additional learned components. Because our quantities are defined after *dataset-level aggregation*, they are **not directly comparable to token-wise lens readouts**. Logit-Trajectories are instead complementary: they trade per-example detail for more stable distribution-level signals.

Second, although our core object is a vector-valued Logit-Trajectory, most of our heatmap visualizations still rely on scalar summaries for readability. Our key comparisons and diagnostics are nevertheless derived from the vector representation (e.g., coherence is defined via vector–centroid alignment, and the centroid itself is a vector). However, such scalarization can obscure multi-modal or heterogeneous trajectory patterns and may therefore be misleading. This motivates richer visualization formats that preserve more of the directional structure.

Third, while our analysis targets generalization across model architectures and languages, we fix a single decision position for logit extraction, consistent with the fixed extraction settings commonly used in prior lens-style studies. Broader evaluation across architectures and language groups, together with robustness checks under alternative decision positions, is needed to better establish generality.

Acknowledgments

This work was supported by the NRF grant funded by the Korea government (MSIT) (No. RS-2024-00334343, A System for Enhancing Language Model Reliability with High-Quality Data and Automated Quality Assessment) and the IITP grant funded by the Korea government (MSIT) (No. RS-2024-00445087, Enhancing AI Model Reliability Through Domain-Specific Automated Value Alignment Assessment), the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190004, Development of semi-supervised learning language intelligence technology and Korean tutoring service for foreigners), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-00556217314820926401 01), and a grant (26212MFDS008) from the Ministry of Food and Drug Safety in 2026.

References

Isha Agarwal, Saharsha Navani, and Fazl Barez. 2025. Context matters: Analyzing the generalizability of linear probing and steering across diverse scenarios.

In *Mechanistic Interpretability Workshop at NeurIPS 2025*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *Preprint*, arXiv:2303.08112.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, and 12 others. 2022. Scaling instruction-finetuned language models. *arXiv preprint*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. Explaining how transformers use context to build predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5513, Toronto, Canada. Association for Computational Linguistics.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Shaona Ghosh, Prasoon Varshney, Makesh Narsimhan Sreedhar, Aishwarya Padmakumar, Traian Rebedea, Jibin Rajan Varghese, and Christopher Parisien. 2025. AEGIS2.0: A diverse AI safety dataset and risks taxonomy for alignment of LLM guardrails. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5992–6026, Albuquerque, New Mexico. Association for Computational Linguistics.

Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Chi Zhang, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *arXiv preprint arXiv:2307.04657*.

Jiachen Jiang, Jinxin Zhou, and Zhihui Zhu. 2025a. Tracing representation progression: Analyzing and enhancing layer-wise similarity. In *International Conference on Representation Learning*, volume 2025, pages 1118–1143.

Jiachen Jiang, Jinxin Zhou, and Zhihui Zhu. 2025b. Tracing representation progression: Analyzing and enhancing layer-wise similarity. In *The Thirteenth International Conference on Learning Representations*.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. 2024. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. *Preprint*, arXiv:2406.18510.

Ryo Kishino, Yusuke Takase, Momose Oyama, Hiroaki Yamagiwa, and Hidetoshi Shimodaira. 2025. Revealing language model trajectories via kullback-leibler divergence. *Preprint*, arXiv:2505.15353.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR.

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. 2023. Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation. *Preprint*, arXiv:2310.17389.

Team Llama. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Ari Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. *Advances in neural information processing systems*, 31.

Thong Nguyen. Logit Prisms: Decomposing Transformer Outputs for Mechanistic Interpretability.

nostalgebraist. 2020. Interpreting gpt: the logit lens. LessWrong, posted 31 Aug 2020.

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 *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron Wallace, and David Bau. 2023. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, page 548–560. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Technical report (preprint). Accessed: 2026-01-04.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research (JMLR)*, 21(1).

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, and 14 others. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. *Preprint*, arXiv:2402.06619.

Yucheng Sun, Alessandro Stolfo, and Mrinmaya Sachan. 2025. Probing for arithmetic errors in language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8122–8139, Suzhou, China. Association for Computational Linguistics.

Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 33–42, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36:80079–80110.

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, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

A Experimental Setup

In this section, we precisely define the observational unit of the Logit-Trajectory Map and the procedure for aggregating it at the dataset level, and we describe the experimental design used to validate the approach in multilingual and multi-task settings.

A.1 Aya Dataset

To ensure broad coverage across multiple tasks and languages, we use the Aya dataset (Singh et al., 2024), which provides parallel translations of semantically equivalent content in multiple languages. The dataset statistics for the five languages and six tasks used in our experiments are summarized in Table 4.

A.2 Logit-TrajectoryΔ Extraction Details

- **Decision position.** We compute layerwise logits at a fixed "decision" position to avoid dependence on input, output sentence lengths:
  - **Encoder (encoder-only models and the T5 encoder):** We use the **last input token** (`[CLS]` / the last non-padding token) as the sequence representation. Following Devlin et al. (2019), the final hidden state corresponding to `[CLS]` is commonly treated as an aggregate representation of the input sequence and is used for downstream classification.
  - **Decoder (decoder-only models and the T5 decoder):** We use the **last input token** (`[EOS]` / the last non-padding token) as the sequence representation. Following Radford et al. (2018), the hidden state at the final position naturally summarizes the entire left context, making it the most direct representation after the model has consumed the full prompt.
- **Subsets.** We run the same pipeline on dataset-wide results and on subsets.

language	task	N_total	avg_chars	avg_tokens
english	dialogue	7,844,049	208.7	36.7
	event-linking	2,092,448	343.9	56.8
	paraphrase-identification	49,401	329.6	60.8
	question-answering	651,496	136.9	23.4
	summarization	851,307	5,369.1	873.7
german	dialogue	1,191,582	95.7	15.0
	event-linking	1,194,092	397.2	58.0
	paraphrase-identification	49,401	318.8	51.4
	question-answering	635,428	143.8	21.2
	summarization	517,349	4,597.2	645.5
hindi	dialogue	1,191,582	113.6	21.1
	event-linking	527,052	309.1	57.3
	paraphrase-identification	49,401	318.1	62.3
	question-answering	635,428	129.0	24.3
	summarization	169,671	2,355.7	465.2
japanese	dialogue	1,191,582	37.5	2.36
	event-linking	642,740	151.1	7.82
	paraphrase-identification	49,401	149.4	11.1
	question-answering	3,099,052	62.2	3.14
	summarization	127,147	844.1	75.6
korean	dialogue	1,191,582	49.1	11.5
	event-linking	555,965	164.4	37.9
	paraphrase-identification	49,401	182.4	46.1
	question-answering	579,428	68.5	17.2
	summarization	127,147	1,021.4	247.2
spanish	dialogue	1,191,582	101.7	17.6
	event-linking	748,931	336.3	56.8
	paraphrase-identification	49,401	343.8	63.0
	question-answering	649,428	151.1	25.5
	summarization	127,147	2,188.2	369.7

Table 4: Aya dataset overview for all language×task settings used in our experiments. avg_tokens is a whitespace-token proxy.

A.3 Outputs: Vector and Visual Views

Vector outputs. We report per-transition magnitudes $\|\Delta^{(t)}\|$, coherence Agr_t, and the top dimensions (token directions) of each centroid c_t.

Visual outputs. We provide: (i) layer×layer heatmaps, and (ii) trajectory plots over layer depth (e.g., centroid coherence curves).

A.4 Layer Partitioning and Dataset-level Aggregation

Logit updates. We define the layer-to-layer logit update as $\Delta^{(\ell)}(x) = z^{(\ell+1)}(x) - z^{(\ell)}(x) \in \mathbb{R}^{|V|}$.

Dataset-level aggregation. For a dataset (or language–task slice) $\mathcal{D}$, we compute the centroid $\bar{\Delta}^{(\ell)} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \Delta^{(\ell)}(x)$, and the cumulative trajectory $\bar{S}^{(k)} = \sum_{\ell=1}^{k-1} \bar{\Delta}^{(\ell)}$.

Vocabulary restriction. For efficiency, we evaluate either the full vocabulary or a fixed token set of size K (e.g., top-K by frequency or a predefined token set); we keep the same K and construction procedure across all comparisons.

A.5 Baselines

To validate the benefit of preserving directionality in Logit-Trajectory Maps, we compare against the following baselines:

- **Divergence baseline (KL):** For layer-wise distributions $p^{(\ell)} = \mathrm{softmax}(z^{(\ell)})$, we measure either the adjacent-layer divergence $D_{\mathrm{KL}}(p^{(\ell)} \| p^{(\ell+1)})$ or the divergence to the final layer $D_{\mathrm{KL}}(p^{(\ell)} \| p^{(L)})$. These reduce each layer transition to a scalar summary of the *magnitude* of change.
- **Similarity baseline (cosine):** We measure the cosine similarity between adjacent-layer hidden states $h^{(\ell)}$ and $h^{(\ell+1)}$.
- **End-to-end baseline (Δ_{end}):** To motivate the need to examine adjacent-layer transitions, we also include an end-to-end baseline that only

compares the first and last layers: Δ_{end} ($= z^{(L)} - z^{(0)}$).

A.6 Implementation details.

All experiments were run on a single NVIDIA H100 GPU. For models up to 8B parameters, we loaded the full model on-device and performed inference in a single pass. For the 70B model, which does not fit in memory under our setup, we used reduced-precision inference by loading weights in `bfloat16` (`bf16`).

Across all models, we use a fixed experimental configuration:

- `seeds`: {500, 1000, 1500}
- `batch_size`: 8
- `max_length`: 256
- `max_sample (per dataset)`: 10,000
- `bootstrap_resamples`: 300

All other hyperparameters follow the recommended default configurations of the respective model implementations.

A.7 Safety Dataset

We use a curated safety dataset released on Hugging Face.[3] It comprises train/validation/test splits constructed from four underlying sources. In the main paper, we report results on the `train` split and use it to train the pre-filter classifier; unless otherwise noted, all evaluation metrics are computed on the `test` split. Table 5 shows the statistics of the dataset used.

The included sources are:

- `nvidia/Aegis-AI-Content-Safety-Dataset-2.0` (Ghosh et al., 2025)
- `allenai/wildjailbreak-r1-v2-format-filtered` (Jiang et al., 2024)
- `PKU-Alignment/BeaverTails` (Ji et al., 2023)
- `lmsys/toxic-chat` (Lin et al., 2023)

[3] https://huggingface.co/datasets/agentlans/prompt-safety-classification

Label	N_total	avg_chars	avg_tokens
Safe	31,747	171.8	28.7
Harmful	25,899	179.7	30.9

Table 5: Safety CSV datasets used for early-exit prediction experiments. avg_tokens is a whitespace-token proxy.

B Notation and Formula Reference

B.1 Core equations

Let f_θ have L layer blocks. For input x, let $h^{(\ell)}(x)$ be the hidden state after layer ℓ and let t^* denote the decision-token position (default: last valid token). The layer-wise vocab-space logits are

$$z^{(\ell)}(x) = \mathrm{LMHead}\Big(h^{(\ell)}(x)_{t^*}\Big) \in \mathbb{R}^{|V|}. \quad (11)$$

Layer-wise transition vectors. The adjacent-layer logit update (our basic trajectory signal) is

$$\Delta^{(\ell\to\ell+1)}(x) = z^{(\ell+1)}(x) - z^{(\ell)}(x) \in \mathbb{R}^{|V|}. \quad (12)$$

More generally, for any layer pair (i, j),

$$\Delta^{(i\to j)}(x) = z^{(j)}(x) - z^{(i)}(x). \quad (13)$$

Dataset-level centroid and its coherence. For transition $t = \ell \to \ell+1$, define the dataset centroid direction

$$c_t = \mathbb{E}_{x\sim\mathcal{D}}\Big[\Delta^{(t)}(x)\Big], \quad (14)$$

and the per-sample alignment to this representative direction

$$\mathrm{Agr}_t(x) = \cos\Big(\Delta^{(t)}(x),\, c_t\Big). \quad (15)$$

The centroid coherence $\mathbb{E}_{x\sim\mathcal{D}}[\mathrm{Agr}_t(x)]$ yields a depth-wise consistency curve.

Layer×Layer span aggregation. For a span (i, j) with $0 \le i < j \le L-1$, define the length-normalized mean update over the span

$$\bar{\Delta}^{(i\to j)}(x) = \frac{1}{j-i}\sum_{\ell=i}^{j-1} \Delta^{(\ell\to\ell+1)}(x), \quad (16)$$

and its dataset-level centroid

$$C_{i,j} = \mathbb{E}_{x\sim\mathcal{D}}\Big[\bar{\Delta}^{(i\to j)}(x)\Big]. \quad (17)$$

We map each $C_{i,j}$ to scalars for visualization:

$$H^{\mathrm{mag}}_{i,j} = \|C_{i,j}\|, \quad (18)$$

$$H^{\cos}_{i,j} = \cos(C_{i,j},\, C_{0,L-1}). \quad (19)$$

Token vocabulary (optional). For efficiency, we may restrict logits to an token set $A \subset V$: $z_A^{(\ell)}(x) \in \mathbb{R}^{|A|}$ and $\Delta_A^{(\cdot)}(x) = z_A^{(\cdot)}(x)$ differences. All quantities above (c_t, Agr_t, $C_{i,j}$, H^{mag}, $H^{\cos}$) are computed identically in the vocab space.

B.2 Notation table

Table 6 shows the total notation described on the paper.

C Computational Cost and Information Loss (Logit-Space, Vector-Op Cost Model)

We compare four aggregation baselines—*KL divergence*, *cosine similarity*, *end-to-end delta* (Δ_{end}), and *our method(Δ)*—under a simplified, logit-space computational model. The goal is to quantify (i) the per-sample metric cost, (ii) the additional aggregation cost when averaging over a dataset of $N = 10{,}000$ examples, (iii) the total cost, and (iv) the information loss induced by aggregation.

Setup. Let L denote the number of layers and V the vocabulary size. For each example, we assume access to per-layer logits $\{\mathbf{z}^{(\ell)}\}_{\ell=1}^{L}$ where $\mathbf{z}^{(\ell)} \in \mathbb{R}^V$. We explicitly **exclude** the cost of generating logits (e.g., forward pass and projection to vocabulary); we only count subsequent vector operations performed *on* logits.

Vector-operation cost model. Any length-V vector operation (e.g., elementwise multiply, elementwise add/subtract, reduction/sum over V, dot-product over V) is counted as **one unit cost**. Scalar operations are negligible in comparison and are not separately modeled. Aggregation is performed as a running sum over examples followed by a single division to obtain the mean.

Definitions of compared metrics. We use layer-wise metrics indexed by ℓ and then average across the dataset:

- **KL divergence:** computed per layer (layer-wise comparison), with an effective cost of 4 vector-ops per layer.
- **Cosine similarity:** computed per layer, with an effective cost of 2 vector-ops per layer.
- **Δ_{end}:** single end-to-end difference between final and initial logits, $\mathbf{z}^{(L)} - \mathbf{z}^{(1)}$, cost 1 vector-op per example.
- **Our method:** per-layer aggregation of logits using one vector-sum per layer (i.e., maintaining the mean logits at each layer), cost 1 vector-op per layer.

Instantiated values ($L = 32$, $N = 10{,}000$). Substituting $L = 32$ and $N = 10{,}000$ into Table 7 yields:

- KL divergence: per-sample $4L = 128$, aggregation $NL + L = 320{,}032$, total $1{,}600{,}032$.
- Cosine similarity: per-sample $2L = 64$, aggregation $320{,}032$, total $960{,}032$.
- Δ_{end}: per-sample 1, aggregation $N + 1 = 10{,}001$, total $20{,}001$.
- Our method: per-sample $L = 32$, aggregation $320{,}032$, total $640{,}032$.

D Experimental Results

D.1 Total Within-Between Analysis

Figure 8, 9 and Figure 10 compare centroid similarity measured (i) within the same condition under different random seeds/sampling (`within`) and (ii) between different conditions (`between`).

D.2 Total Coherence Curves

Figure 12, 13 shows the coherence curves and comparison with the baseline, KL-divergence and cosine similarity.

D.3 Task-wise Layer×Layer Visualization

Computing the same heatmaps over language/task subsets further reveals condition-specific structure (Figures 14–17): some conditions reach strong alignment earlier, whereas others only form and stabilize later in depth. We report $H^{\cos}$ for BERT-base-multilingual-cased, T5-large, Qwen2.5-7B, Qwen2.5-72B, Llama-3.1-8B, and Llama-3.3-70B. Thus, Logit-Trajectory Maps enable one-shot comparisons of where updates concentrate and when convergence emerges across data conditions, going beyond single-example explanations.

Visually, the heatmap shapes are similar according to linguistic similarity (e.g., geographic proximity or grammatical relatedness). For example, Korean exhibits a trajectory pattern with late-layer variation that is similar to Hindi, whereas its pattern differs from Spanish, which shows little variation in later layers.

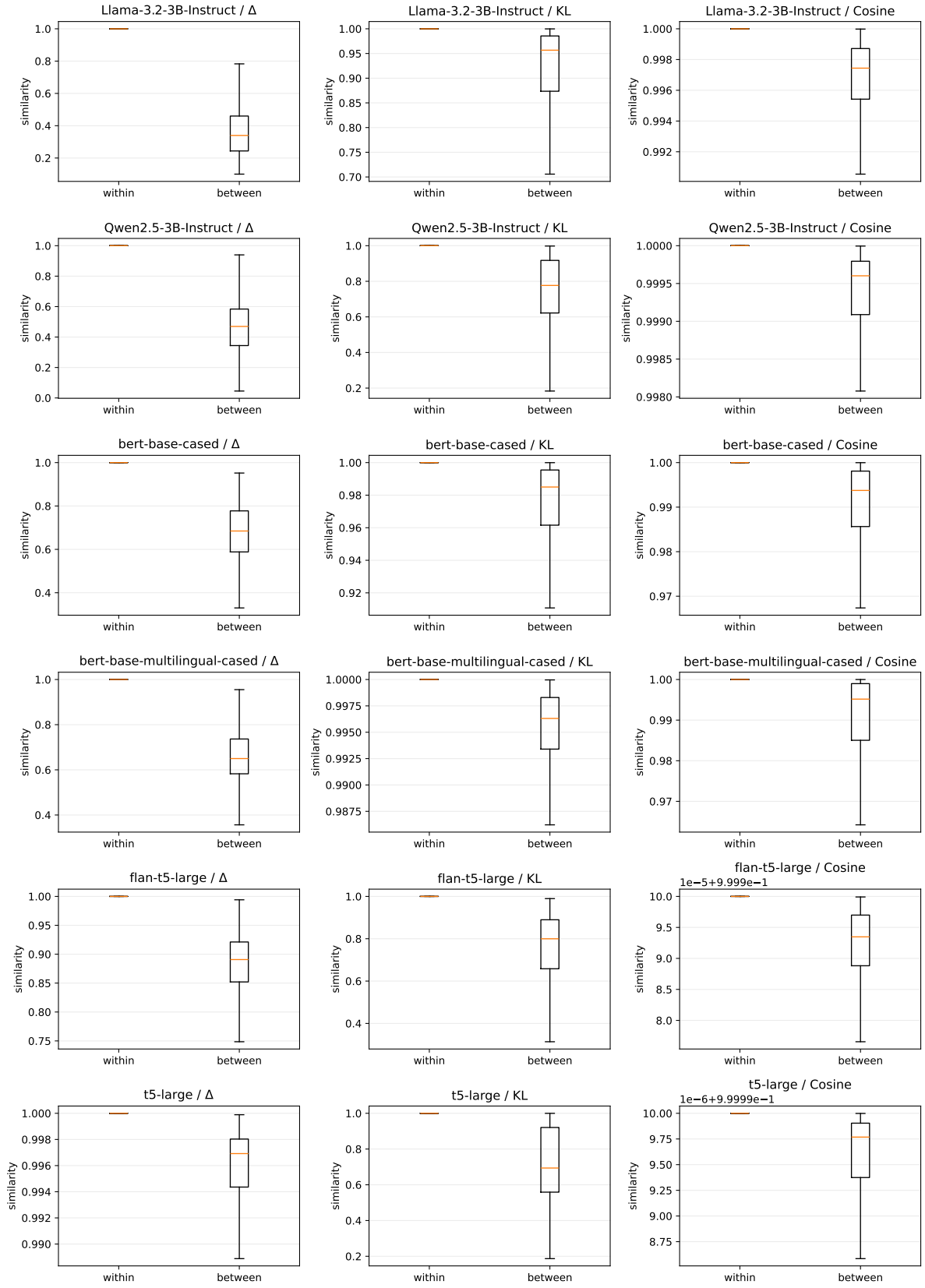


Figure 8: Within- vs. between-condition centroid similarity for the $\leq$3B encoder and decoder model (with statistical significance). We excluded Δ_{end} because it does not provide a layer-depth–wise distribution. Within-condition similarity remains high across seeds/sampling, while between-condition separation is strongest when preserving directional trajectories.

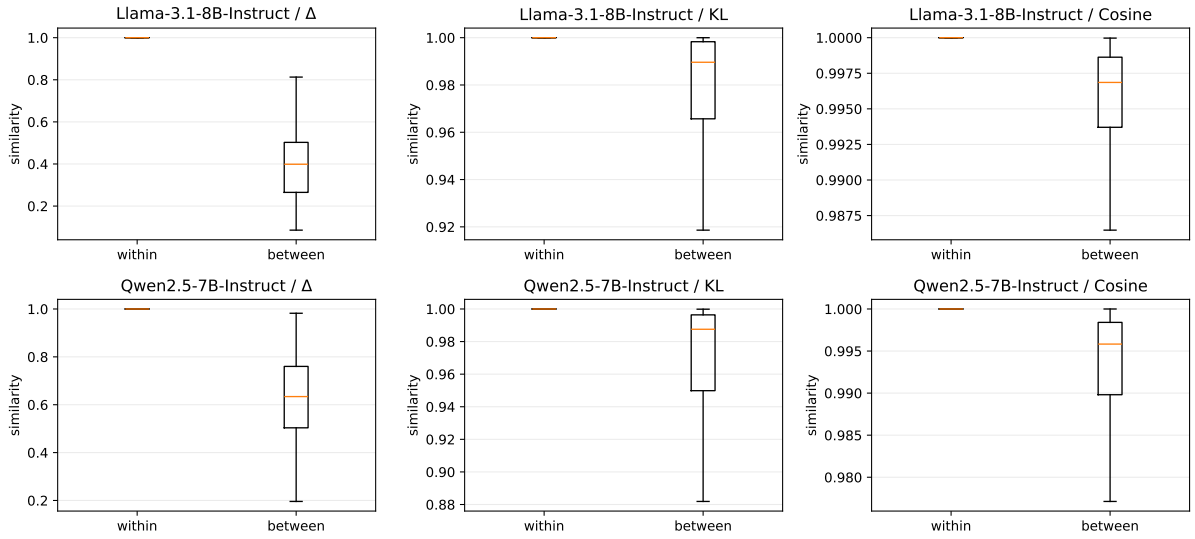


Figure 9: Within- vs. between-condition centroid similarity for the 7B-8B model (with statistical significance). We excluded Δ_{end} because it does not provide a layer-depth–wise distribution. Within-condition similarity remains high across seeds/sampling, while between-condition separation is strongest when preserving directional trajectories.

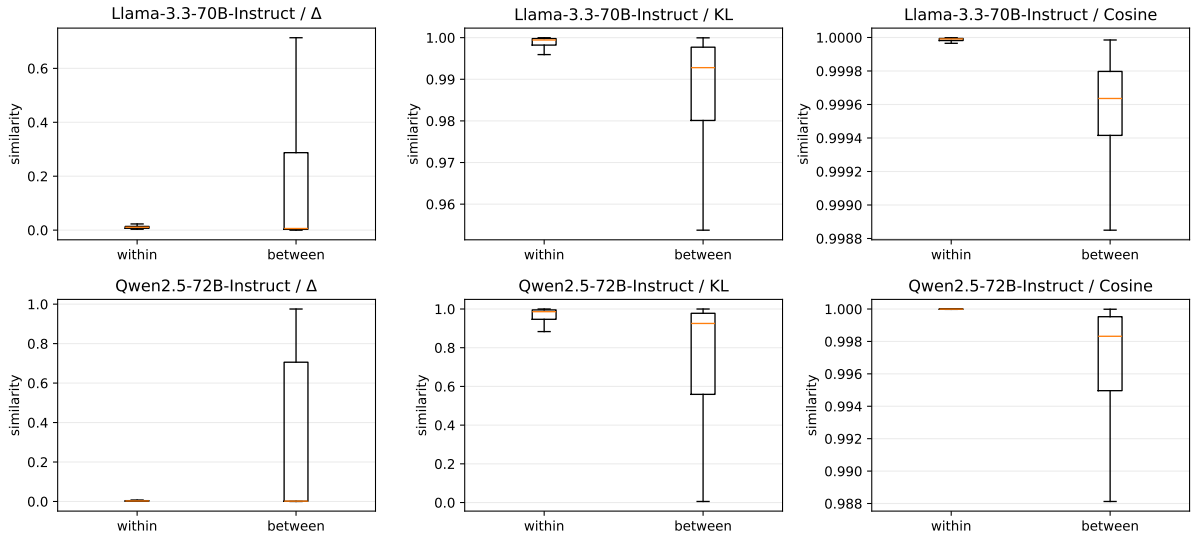


Figure 10: Within- vs. between-condition centroid similarity for the $\geq$70B model (with significance tests). We excluded Δ_{end} because it does not provide a layer-depth–wise distribution. Directional-trajectory similarity is weaker in both cases, consistent with higher trajectory diversity in larger models.

Symbol	Type / Shape	Meaning
f_θ	model	Transformer language model with parameters θ.
L	integer	Number of layer blocks.
x	sequence	Input token sequence.
V	set	Vocabulary; $\lvert V\rvert$ is vocabulary size.
A	set	Token vocabulary subset.
ℓ	index	Layer index.
i, j	indices	Span boundaries over layers ($i \le j$).
t^*	index	Decision token position (default: last valid token).
y^*	index	Target token id at t^* (e.g., rotated next-token label).
$h^{(\ell)}(x)$	tensor	Hidden representation at layer ℓ for input x.
$z^{(\ell)}(x)$	$\mathbb{R}^{\lvert V\rvert}$	Layer-ℓ logits (at position t^*).
$z_A^{(\ell)}(x)$	$\mathbb{R}^{\lvert A\rvert}$	Token-restricted logits on A (optional).
$\Delta^{(\ell\to\ell+1)}(x)$	$\mathbb{R}^{\lvert V\rvert}$	Vector logit delta between adjacent layers.
c_ℓ	$\mathbb{R}^{\lvert V\rvert}$	Transition centroid: $\mathbb{E}_x[\Delta^{(\ell\to\ell+1)}(x)]$.
$\mathrm{Agr}_\ell(x)$	scalar	Coherence: $\cos(\Delta^{(\ell\to\ell+1)}(x), c_\ell)$.
$\mathcal{D}$	dataset	Full dataset (or evaluation set).
$\mathcal{D}'$	subset	Subset of $\mathcal{D}$ (language/label/safety stratification).
$M^{(\ell)}(x)$	scalar	Target logit: $z^{(\ell)}(x)_{y^*}$.
$D^{(\ell)}(x)$	scalar	Relative delta over layers.
ε	scalar	Small constant for numerical stability.
$G(x)$	$\mathbb{R}^{L\times L}$	Span-grid (upper-triangular).
$G_{i,j}(x)$	scalar	Mean-reduced span score over $D^{(\ell)}(x)$ for $\ell \in [i, j]$.
C	$\mathbb{R}^{L\times L}$	Span centroid heatmap: $C_{i,j} = \mathbb{E}_x[G_{i,j}(x)]$.
Var	$\mathbb{R}^{L\times L}$	Span dispersion heatmap: $\mathrm{Var}_{i,j} = \mathrm{Var}_x(G_{i,j}(x))$.

Table 6: Notation used in Logit-Trajectory Map and the standard span-grid feature pipeline.

Method	Per-sample ops	Aggregation ops	Total ops	Information loss (dims)
KL divergence	$4L$	$NL + L$	$N(4L) + (NL + L)$	$LV - L$
Cosine similarity	$2L$	$NL + L$	$N(2L) + (NL + L)$	$LV - L$
Δ_{end}	1	$N + 1$	$N + (N + 1)$	$LV - V = (L - 1)V$
Our method	L	$NL + L$	$NL + (NL + L)$	$LV - LV \simeq 0$

Table 7: Computational cost and information loss under the logit-space vector-operation model. L is the number of layers, V the vocabulary size, and N the dataset size. Logit generation cost is excluded; only vector operations on logits are counted. Aggregation computes dataset means via running sums and a final division.

D.4 Safety Pre-filter

Figure 11 reports training performance across decoder-based models of different sizes. Across all models, performance as a function of layer depth is highest when using our Logit-Trajectory features. We report AUC/F1 over three different random seeds (Table 8).

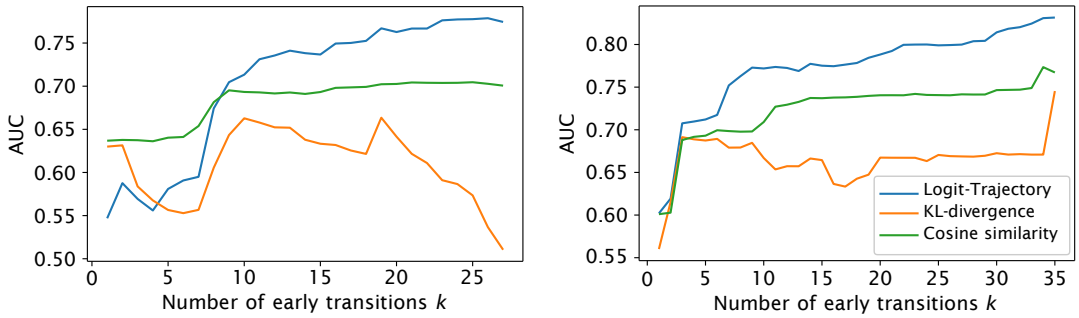
D.5 Stress tests: when do directional signatures weaken?

Finally, we run negative-control randomizations (dimension permutation and random sign flips) to verify that the observed centroid coherence is not a trivial artifact of the pipeline.

Negative-control randomization (sanity check). To test whether directional coherence reflects non-trivial structure, we apply two randomizations to the per-transition vectors $\Delta_t(x)$ and re-evaluate coherence w.r.t. the original centroids $\{c_t\}$: (1) *dimension permutation*, which shuffles the vocabulary dimensions of $\Delta_t(x)$, and (2) *random sign flips*, which multiplies each $\Delta_t(x)$ by a random ± 1. Table 11 shows that coherence drops substantially under dimension permutation and collapses to nearly zero under sign flips. This confirms that the directional signature is not explained by update magnitude alone, but relies on coherent directional alignment across samples and transitions.

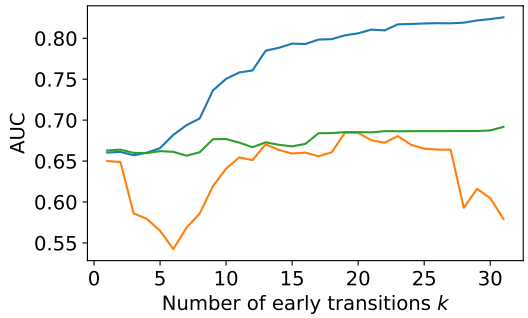
model	AUC_{Δ}	AUC_{KL}	AUC_{cos}	F1_{Δ}	F1_{KL}	F1_{cos}
Llama-3.2-3B-Instruct	**0.779** ± 0.008	0.670 ± 0.022	0.703 ± 0.018	**0.720** ± 0.012	0.547 ± 0.136	0.664 ± 0.019
Llama-3.1-8B-Instruc	**0.824** ± 0.005	0.691 ± 0.007	0.706 ± 0.017	**0.765** ± 0.008	0.616 ± 0.031	0.680 ± 0.019
Llama-3.3-70B-Instruct	**0.874** ± 0.014	0.761 ± 0.013	0.770 ± 0.008	**0.803** ± 0.013	0.516 ± 0.139	0.702 ± 0.021
Qwen2.5-3B-Instruct	**0.824** ± 0.014	0.745 ± 0.012	0.755 ± 0.025	**0.765** ± 0.009	0.686 ± 0.007	0.688 ± 0.029
Qwen2.5-7B-Instruct	**0.798** ± 0.029	0.734 ± 0.022	0.698 ± 0.009	**0.726** ± 0.033	0.674 ± 0.027	0.623 ± 0.059
Qwen2.5-72B-Instruct	**0.868** ± 0.018	0.774 ± 0.035	0.773 ± 0.012	**0.804** ± 0.008	0.708 ± 0.037	0.679 ± 0.012

Table 8: Prediction performance across early-transition budgets k (mean ± std over seeds). **Bold** shows the best performance of each model. Δ denotes the Logit-Trajectory, KL denotes KL-divergence, and cos denotes cosine similarity. For larger models, Logit-Trajectory appears more fine-grained, showing a trend of increasing performance.

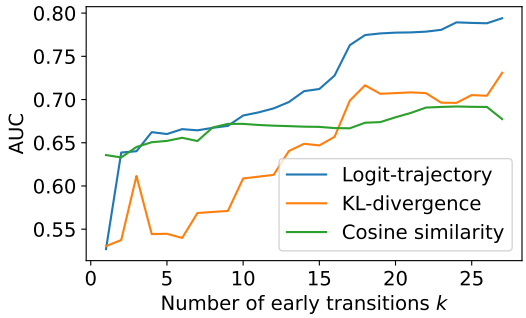


(a) Llama-3.2-3B-Instruct

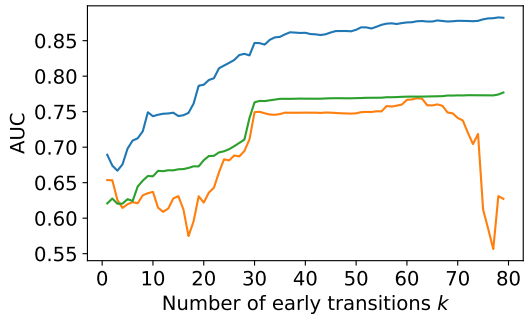
(b) Qwen2.5-3B-Instruct



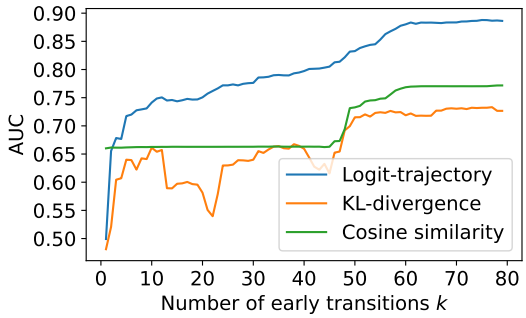
(c) Llama-3.1-8B-Instruct



(d) Qwen2.5-7B-Instruct



(e) Llama-3.3-70B-Instruct



(f) Qwen2.5-72B-Instruct

Figure 11: Early-exit prediction vs. the number of early transitions k (using only the first k). We compare Logit-Trajectory features to KL-divergence and cosine similarity features with the same classifier; higher AUC/F1 at smaller k indicates earlier decision formation.

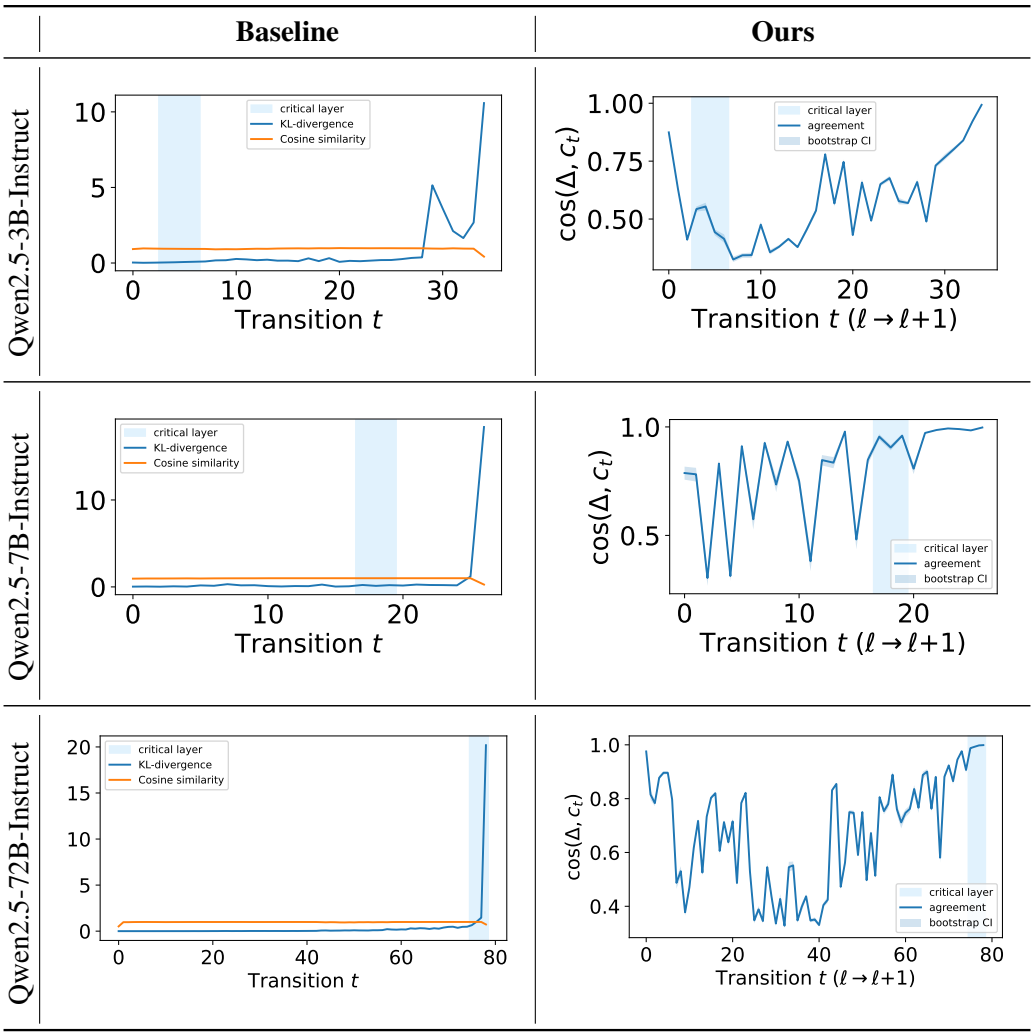


Figure 12: Transition-wise coherence on Qwen2.5-(3B, 7B, 72B)-Instruct on English dialogue task. For each transition t, we compute $\cos\big(\Delta^{(t)}(x), c_t\big)$ with $c_t = \mathbb{E}_x[\Delta^{(t)}(x)]$, and report the mean with confidence intervals. Higher values indicate more consistent update directions across samples.

Model	Qwen2.5-3B-Instruct	Llama-3.2-3B-Instruct
orig	0.5780	0.5931
perm	0.0656	0.1980
flip	-0.0010	-0.0024
δ_{perm}	0.5124	0.3951
δ_{flip}	0.5790	0.5955

Table 9: Sanity randomization stress test for centroid coherence on dialogue task (3B models). We report the centroid coherence averaged over transitions. **orig** is computed on the original Δ, **perm** permutes vocabulary dimensions of Δ, and **flip** applies random ± 1 sign flips. δ_{perm}/δ_{flip} report the drop relative to orig. A large drop under randomization indicates that the observed coherence reflects a non-trivial centroid rather than a degenerate artifact.

Model	Qwen2.5-7B-Instruct	Llama-3.1-8B-Instruct
orig	0.8062	0.6417
perm	0.4088	0.0722
flip	0.0030	0.0019
δ_{perm}	0.3974	0.5696
δ_{flip}	0.8031	0.6398

Table 10: Sanity randomization stress test for centroid coherence on dialogue task (7-8B models). We report the centroid coherence averaged over transitions. **orig** is computed on the original Δ, **perm** permutes vocabulary dimensions of Δ, and **flip** applies random ± 1 sign flips. δ_{perm}/δ_{flip} report the drop relative to orig. A large drop under randomization indicates that the observed coherence reflects a non-trivial centroid rather than a degenerate artifact.

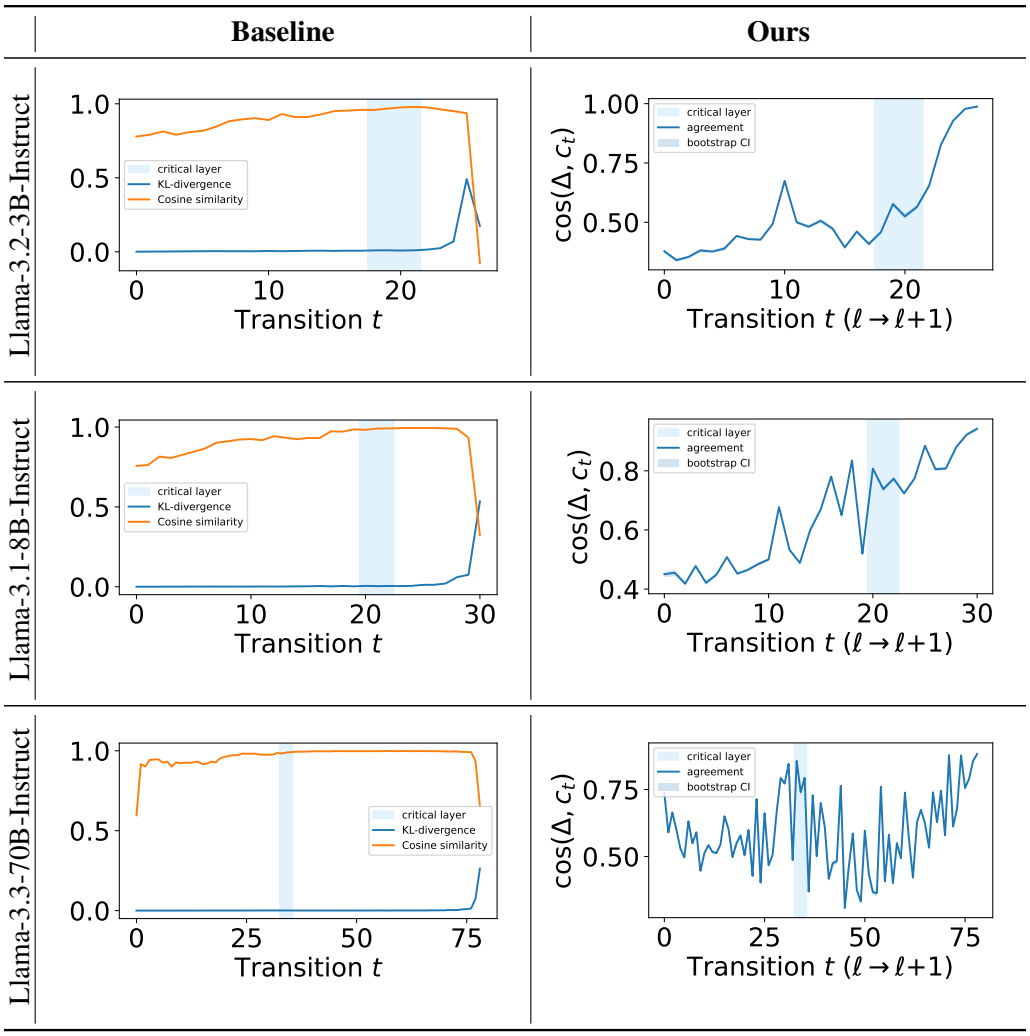
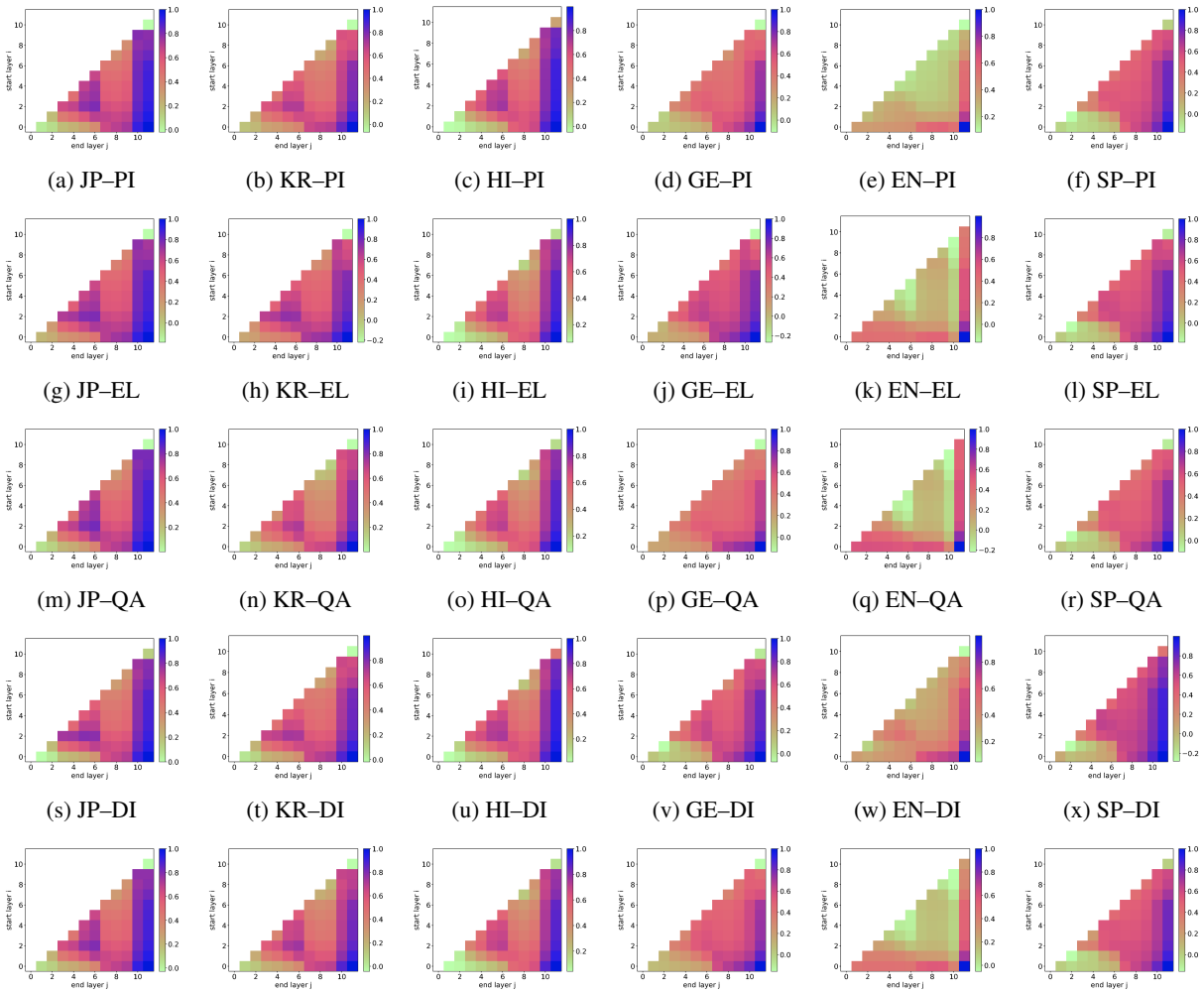


Figure 13: Transition-wise coherence on Llama-(3.2-3B, 3.1-8B, 3.3-70B)-Instruct on English dialogue task. For each transition t, we compute $\cos\big(\Delta^{(t)}(x), c_t\big)$ with $c_t = \mathbb{E}_x[\Delta^{(t)}(x)]$, and report the mean with confidence intervals. Higher values indicate more consistent update directions across samples.

Model	Qwen2.5-72B-Instruct	Llama-3.3-70B-Instruct
orig	0.6677	0.5931
perm	0.3638	0.1980
flip	0.0009	-0.0024
δ_{perm}	0.3039	0.3951
δ_{flip}	0.6668	0.5955

Table 11: Sanity randomization stress test for centroid coherence on dialogue task ($\geq$70B models). We report the centroid coherence averaged over transitions. **orig** is computed on the original Δ, **perm** permutes vocabulary dimensions of Δ, and **flip** applies random ± 1 sign flips. δ_{perm}/δ_{flip} report the drop relative to orig. A large drop under randomization indicates that the observed coherence reflects a non-trivial centroid rather than a degenerate artifact.



(a) JP–PI (b) KR–PI (c) HI–PI (d) GE–PI (e) EN–PI (f) SP–PI

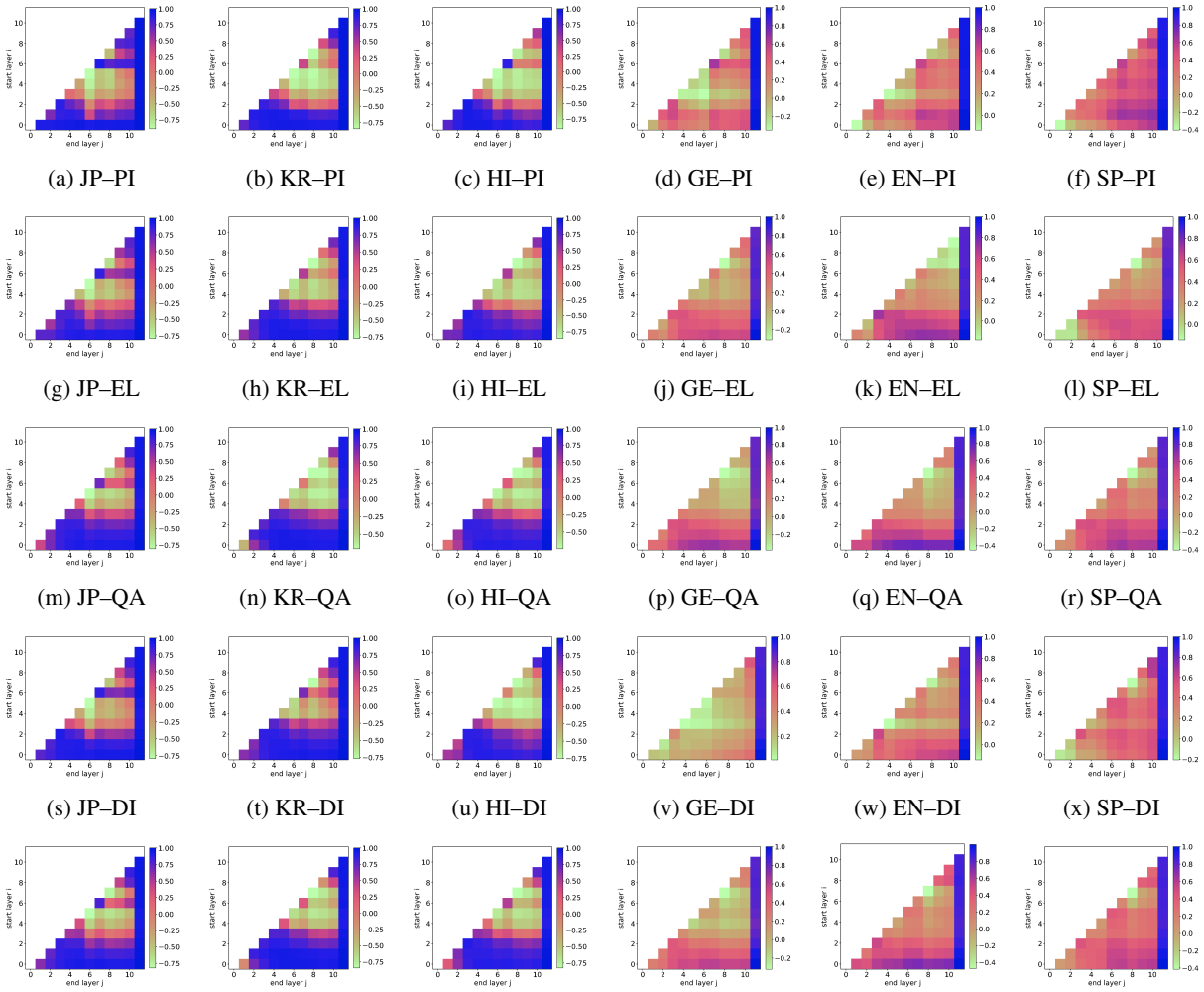
(g) JP–EL (h) KR–EL (i) HI–EL (j) GE–EL (k) EN–EL (l) SP–EL

(m) JP–QA (n) KR–QA (o) HI–QA (p) GE–QA (q) EN–QA (r) SP–QA

(s) JP–DI (t) KR–DI (u) HI–DI (v) GE–DI (w) EN–DI (x) SP–DI

(y) JP–SUM (z) KR–SUM (aa) HI–SUM (ab) GE–SUM (ac) EN–SUM (ad) SP–SUM

Figure 14: Heatmap multiples across language and task conditions on BERT-base-cased. The languages include Japanese (JP), Korean (KR), Hindi (HI), German (GE), English (EN), and Spanish (SP), and are sorted by linguistic similarity. The tasks include Paraphrase Identification (PI), Event Linking (EL), Question Answering (QA), Dialogue (DI), and Summarization (SUM), and are sorted in ascending order of output length.



(a) JP–PI (b) KR–PI (c) HI–PI (d) GE–PI (e) EN–PI (f) SP–PI

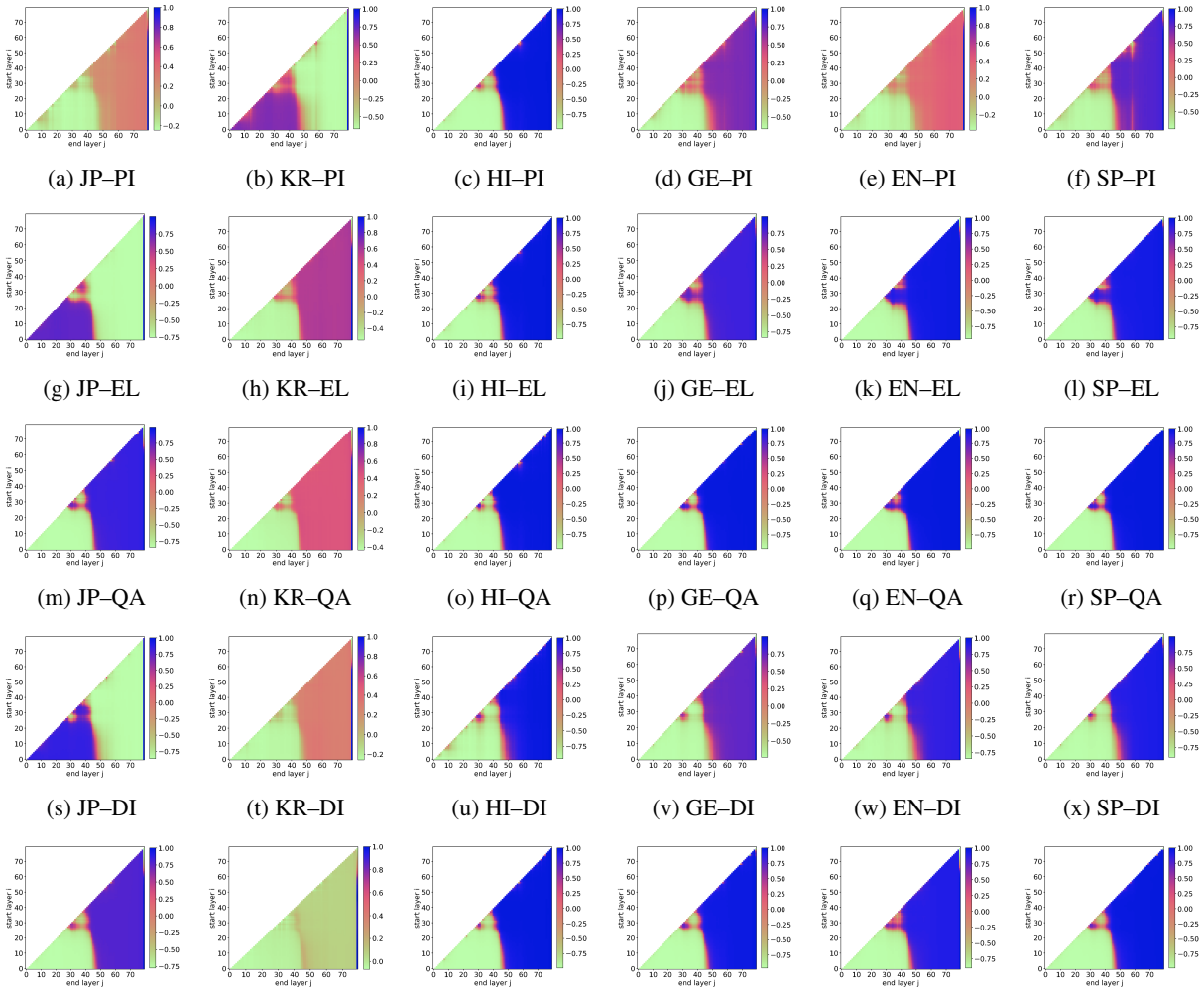
(g) JP–EL (h) KR–EL (i) HI–EL (j) GE–EL (k) EN–EL (l) SP–EL

(m) JP–QA (n) KR–QA (o) HI–QA (p) GE–QA (q) EN–QA (r) SP–QA

(s) JP–DI (t) KR–DI (u) HI–DI (v) GE–DI (w) EN–DI (x) SP–DI

(y) JP–SUM (z) KR–SUM (aa) HI–SUM (ab) GE–SUM (ac) EN–SUM (ad) SP–SUM

Figure 15: Heatmap multiples across language and task conditions on BERT-base-multilingual-cased. The languages include Japanese (JP), Korean (KR), Hindi (HI), German (GE), English (EN), and Spanish (SP), and are sorted by linguistic similarity. The tasks include Paraphrase Identification (PI), Event Linking (EL), Question Answering (QA), Dialogue (DI), and Summarization (SUM), and are sorted in ascending order of output length.



(a) JP–PI (b) KR–PI (c) HI–PI (d) GE–PI (e) EN–PI (f) SP–PI

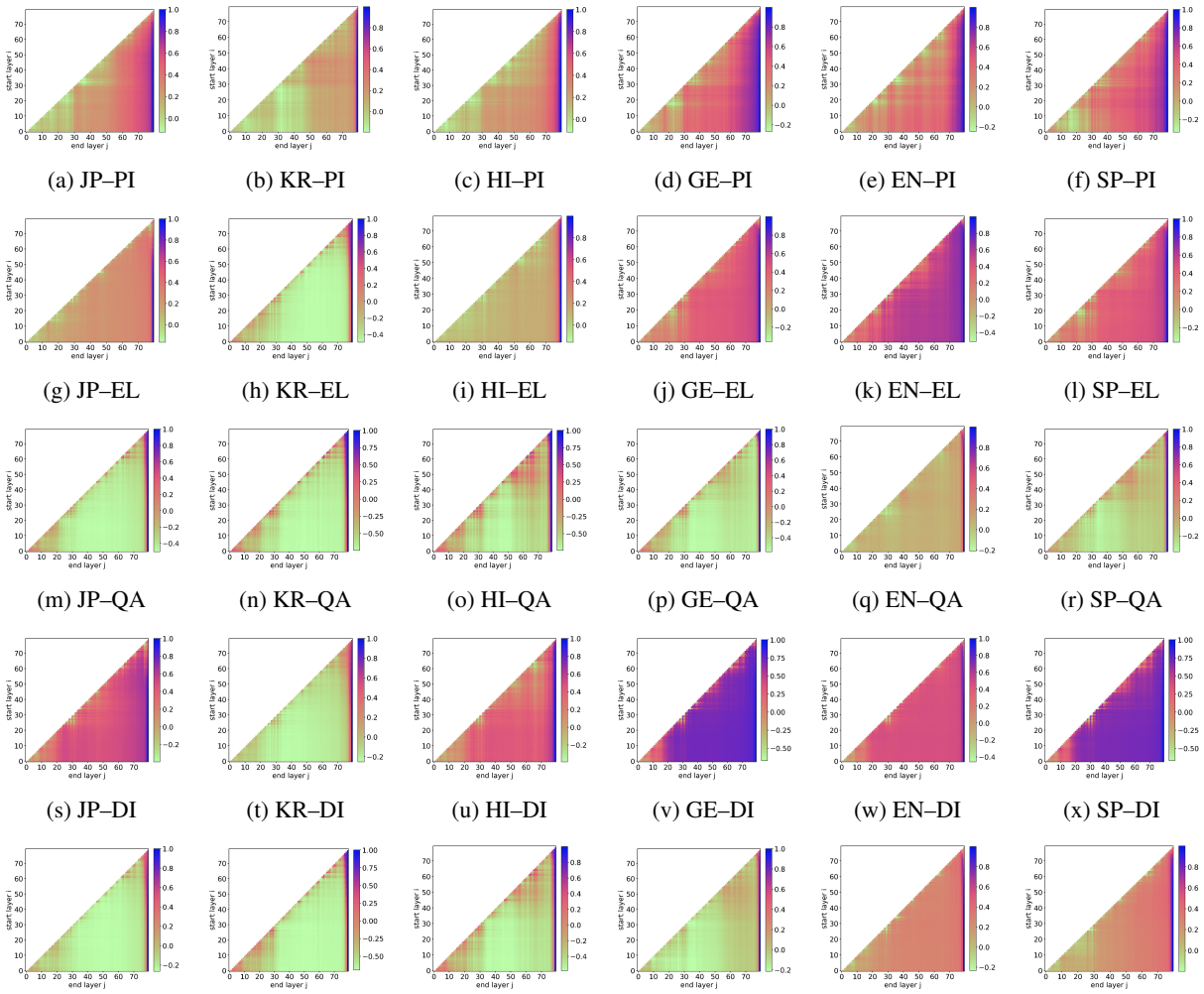
(g) JP–EL (h) KR–EL (i) HI–EL (j) GE–EL (k) EN–EL (l) SP–EL

(m) JP–QA (n) KR–QA (o) HI–QA (p) GE–QA (q) EN–QA (r) SP–QA

(s) JP–DI (t) KR–DI (u) HI–DI (v) GE–DI (w) EN–DI (x) SP–DI

(y) JP–SUM (z) KR–SUM (aa) HI–SUM (ab) GE–SUM (ac) EN–SUM (ad) SP–SUM

Figure 16: Heatmap multiples across language and task conditions on Qwen2.5-72B-Instruct. The languages include Japanese (JP), Korean (KR), Hindi (HI), German (GE), English (EN), and Spanish (SP), and are sorted by linguistic similarity. The tasks include Paraphrase Identification (PI), Event Linking (EL), Question Answering (QA), Dialogue (DI), and Summarization (SUM), and are sorted in ascending order of output length.



(a) JP–PI (b) KR–PI (c) HI–PI (d) GE–PI (e) EN–PI (f) SP–PI

(g) JP–EL (h) KR–EL (i) HI–EL (j) GE–EL (k) EN–EL (l) SP–EL

(m) JP–QA (n) KR–QA (o) HI–QA (p) GE–QA (q) EN–QA (r) SP–QA

(s) JP–DI (t) KR–DI (u) HI–DI (v) GE–DI (w) EN–DI (x) SP–DI

(y) JP–SUM (z) KR–SUM (aa) HI–SUM (ab) GE–SUM (ac) EN–SUM (ad) SP–SUM

Figure 17: Heatmap multiples across language and task conditions on Llama-3.3-70B-Instruct. The languages include Japanese (JP), Korean (KR), Hindi (HI), German (GE), English (EN), and Spanish (SP), and are sorted by linguistic similarity. The tasks include Paraphrase Identification (PI), Event Linking (EL), Question Answering (QA), Dialogue (DI), and Summarization (SUM), and are sorted in ascending order of output length.