ON THE PERSISTENT EFFECTS OF LEXICALITY IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Representations extracted from large language models (LLMs) play an important role in many downstream applications. However, the structure of these representations is often influenced by lexical overlap rather than semantic content. Our understanding of the relationship between this lexical influence and semantic content, and its implications for downstream tasks, remains limited. In this work, we investigate representations using adversarial semantic stress tests to quantify the effect of lexical overlap relative to semantic content. We find that lexical influence extends across model depth and persists across training regimes and objective functions, including models trained for semantic similarity. Moreover, in autoregressive models we observe an intermediate regime in which semantic discrimination weakens alongside reduced linear accessibility of surface-form information, indicating that reduced lexical accessibility does not imply stronger semantic discrimination. We relate this regime to changes in representation geometry and find that its semantic weakness coincides with a plateau or partial contraction in intrinsic dimension. We further demonstrate the effect of lexical influence on downstream uses of LLMs using summarization and model editing as case studies.

1 INTRODUCTION

Natural Language Processing (NLP) has seen rapid advances with the emergence of LLMs, which now achieve strong performance across a wide variety of benchmarks and downstream tasks. Beyond generating text, these models are increasingly used as general-purpose embedding engines, providing vector representations that underpin retrieval, textual similarity, clustering, and evaluation pipelines.

The widespread reliance on LLM embeddings implicitly assumes that, as we progress through model depth, representations gradually move away from local lexical structure and converge toward increasingly abstract sequence-level semantics (Jawahar et al., 2019; Hewitt & Liang, 2019; Haider et al., 2024). In practice, this assumption is fragile, since it does not imply that lexical cues are eliminated as representations can remain highly similar whenever inputs share tokens, even when their meanings diverge (Li et al., 2020; Dumpala et al., 2024; Rizwan et al., 2025). Such lexical influence can distort similarity estimates, making it harder to retrieve or classify documents by meaning rather than wording and reducing the reliability of systems that depend on semantic representations.

Therefore, in this work, we aim to understand the evolution of representational geometry in the context of lexical and semantic information. We treat every layer of a model as a candidate embedding space and organize our study into four components. **First**, we quantify lexical influence using a triplet semantic equivalence stress test consisting of an anchor, a meaning-preserving paraphrase, and a meaning-changing distractor with substantial lexical overlap. We further test whether the resulting depth dependence is specific to the representation readout using learned token pooling, per-dimension scaling, and linear and nonlinear transformations. **Second**, we relate these failures to lexical decodability and semantic fidelity. Token-identity probes measure the accessibility of surface form across layers, while semantic fidelity is measured on diverse embedding tasks, allowing us to test whether reduced lexical accessibility is accompanied by stronger semantic structure. **Third**, we examine whether the observed depth-dependent behavior aligns with changes in representation geometry using prompt entropy and intrinsic dimension. **Fourth**, we examine the consequences of lexical influence in abstractive summarization evaluation and factual model editing. Our investigation leads to several central insights:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Lexical influence persists across depth and adaptation regimes (§ 3). Adversarial stress tests across multiple models reveal that token-overlap-driven similarity effects persist throughout model depth. Although attenuated in deeper layers, the effect is neither fully eliminated nor confined to shallow representations. Instruction tuning and semantic embedding objectives improve overall embedding quality, yet lexical influence remains across these adaptation regimes, indicating reduced but continued sensitivity to lexical overlap after semantic adaptation.

Lexical decodability is non-monotonic across depth (§4.1). Exact token identity is highly decodable in early layers, becomes least linearly accessible at intermediate depths, and partially recovers later. This non-monotonic pattern indicates that depth does not induce a clean lexical-to-semantic transition (Tenney et al., 2019; Jawahar et al., 2019; Li & Subramani, 2025).

An intermediate regime of reduced lexical accessibility and weak semantic fidelity (§ 4, § 5). Across the autoregressive models we study, lexical influence is strongest over an intermediate region where surface form is least linearly accessible, without a corresponding improvement in semantic fidelity. Using OLMo checkpoints from initialization through later training, we find that this depth pattern emerges early during pretraining and is largely preserved thereafter. Learned token pooling, per dimension scaling, and linear and nonlinear transformations improve absolute semantic discrimination, while the broad depth-dependent pattern remains. Geometric analysis further shows that the intermediate weakness coincides with stalled or partially contracting intrinsic dimension, while the corresponding prompt-entropy trajectory varies across models.

Practical implications (§ 6). In embedding-based pipelines, lexical overlap can miscalibrate similarity signals, degrading performance. In summarization evaluation, model-based metrics can favor lexically aligned non-summaries over gold summaries, showing that surface overlap can be rewarded over semantic preservation. In weight-space model editing, lexically similar locality prompts undergo larger unintended changes than lexically dissimilar controls, compromising edit locality.

More broadly, our findings show that semantic abstraction and robustness to lexical form should be treated as distinct properties of LLM representations. Accounting for this distinction is important both when selecting representations for downstream use and when evaluating systems whose behavior can be shaped by lexical overlap.

2 Experimental Setup and Preliminaries

We use a common set of datasets and model families to ensure consistency across experiments.

Datasets. To probe lexical influence, we use two adversarial benchmarks, CounterFact (Meng et al., 2022) and SugarCrepe++ (SCPP) (Dumpala et al., 2024). Relative to CounterFact, which primarily perturbs entities, SCPP offers a more targeted benchmark for lexical-influence tests, as it introduces systematic shifts in attributes and relations. We additionally use CounterFact for model editing. Dataset details and reference samples are provided in Appendix A. To measure lexical decodability, token-identity probes are trained on WikiText (Merity et al., 2017), and semantic fidelity across embedding tasks is evaluated using MTEB (Muennighoff et al., 2023).

Models. We consider pretrained, instruction-tuned, and semantic embedding models spanning multiple model families.[1] To study how the observed depth-dependent behavior develops, we additionally evaluate OLMo 3 7B checkpoints spanning initialization, pretraining, and post-training.

3 Measuring Lexical Influence

In the context of learned representations, we operationalize *lexical influence* using triplets $\mathcal{D} = \{(a, p, d)\}$, where a is an anchor, p a meaning-preserving paraphrase, and d a lexically overlapping but semantically mismatched distractor. We define a failure when the intended semantic ordering is violated, such that the anchor–paraphrase pair is not strictly closer than both pairs involving d. Dataset usage and details are provided in Appendix A.1.

[1] Llama 3.2 (Grattafiori et al., 2024) and Gemma 3 (Team et al., 2025) in both *pretrained* and *instruction-tuned* variants, along with *embedding* models Qwen 3 (Zhang et al., 2025) (Qwen3-Embedding-8B) and KaLM (Zhao et al., 2025) (KaLM-Embedding-Gemma3-12B-2511). KaLM modifies the Gemma 3 architecture to use full attention and both embedding models are trained on the last token in their general training.

For a given model M and input x consisting of tokens $t \in \{1, \ldots, T_x\}$, let $H_\ell^M(x) \in \mathbb{R}^{T_x \times d}$ denote the matrix of token-level hidden states at layer $\ell \in \mathcal{L}$, where the t-th row $H_\ell^M(x)_t \in \mathbb{R}^d$ corresponds to token position t. Let $h_\ell^M(x) \in \mathbb{R}^d$ denote the corresponding sentence-level embedding, obtained via one of two standard choices: mean pooling and last token.

$$h_\ell^{M,\text{mean}}(x) = \frac{1}{T_x} \sum_{t=1}^{T_x} H_\ell^M(x)_t, \quad h_\ell^{M,\text{last}}(x) = H_\ell^M(x)_{T_x}$$

We then ℓ_2-normalize $\tilde{h}_\ell^M(x) = h_\ell^M(x)/\|h_\ell^M(x)\|_2$ and measure euclidean distance,

$$d_\ell^M(x, y) = \left\|\tilde{h}_\ell^M(x) - \tilde{h}_\ell^M(y)\right\|_2.$$

For ℓ_2-normalized vectors, this is equivalent to cosine similarity, since $\left\|\tilde{u} - \tilde{v}\right\|_2^2 = 2 - 2\,\tilde{u}^\top \tilde{v}$.

We quantify lexical influence using triplet success rate, where success requires the meaning preserving pair to be closer than either pair involving the lexical distractor:

$$\text{SR}_\ell^M = \frac{1}{|\mathcal{D}|} \sum_{(a,p,d)\in\mathcal{D}} \mathbb{I}\left[d_\ell^M(a, p) < \min\left\{d_\ell^M(a, d), d_\ell^M(p, d)\right\}\right].$$

Lower success indicates greater lexical influence. Figure 1(a,b) reports layer-wise success rates for mean pooled embeddings on CounterFact and SCPP. Corresponding last-token results and task-wise SCPP breakdown is provided in Appendix B. Across pretrained and instruction-tuned models, mean pooling generally outperforms last-token representations, particularly on CounterFact. The gap in performance is partly explained by a control experiment showing that the last-token readout is sensitive to terminal-token identity, particularly in early layers (Appendix B). This pattern is consistent with prior work suggesting that pooling encourages more semantically abstract representations than relying on a single position (Reimers & Gurevych, 2019; Xing et al., 2024). For mean pooled representations, we observe a consistent nonmonotonic pattern across depth with a mid-layer valley, where performance briefly declines before recovering and stabilizing in the upper layers. By contrast, last-token representations typically stagnate over the same depth range or improve only gradually. We refer to the broader pattern associated with this valley as the *intermediate regime*.

Results for **pretrained and instruction-tuned models** show that post-training primarily changes stress-test performance at greater depth. The two variants behave similarly through much of the early network but diverge more strongly in later layers, where instruction-tuned models generally achieve larger gains. This suggests that post-training leaves early-layer behavior comparatively stable while modifying representations more strongly at greater depth, consistent with prior evidence of larger post-training representational changes in later layers (Zhao et al., 2024). We further test this hypothesis using OLMo-3-7B checkpoints spanning both base training and post-training.

Figure 1(c,d) applies the same stress test to OLMo 3 7B from initialization through Stage 1 pretraining, Stage 2 midtraining, Stage 3 long context training, SFT, DPO, and RLVR.[2] At initialization, performance shows little systematic variation across depth. The intermediate regime emerges early in Stage 1 and persists throughout subsequent training, indicating that susceptibility to lexical distractors is established during early pretraining. Later training primarily changes the magnitude of semantic discrimination, with larger changes at intermediate and later depths. SFT modifies these regions without removing the depth pattern, while DPO and successive RLVR checkpoints produce smaller additional changes.

Embedding models. Embedding models perform strongly on CounterFact but remain substantially weaker on SCPP. Even with the last-token readout used during embedding training, the strongest model reaches only 68% success. SCPP subtask analysis in Appendix B.2 further shows substantially higher performance on object changes than on attribute and relation changes. We conjecture that embedding objectives can exploit salient entity or object level lexical cues as shortcuts for semantic discrimination, as such cues are highly informative in CounterFact. This is consistent with prior work on shortcut solutions in representation learning (Robinson et al., 2021).

Learned semantic readouts. For the autoregressive models, mean pooling gives the strongest overall performance while exhibiting a clear nonmonotonic depth pattern. We therefore ask whether this depth

[2]The OLMo checkpoints are used to study the emergence of the depth pattern within a single training trajectory; we therefore compare qualitative depth trends rather than exact trajectories across model families.

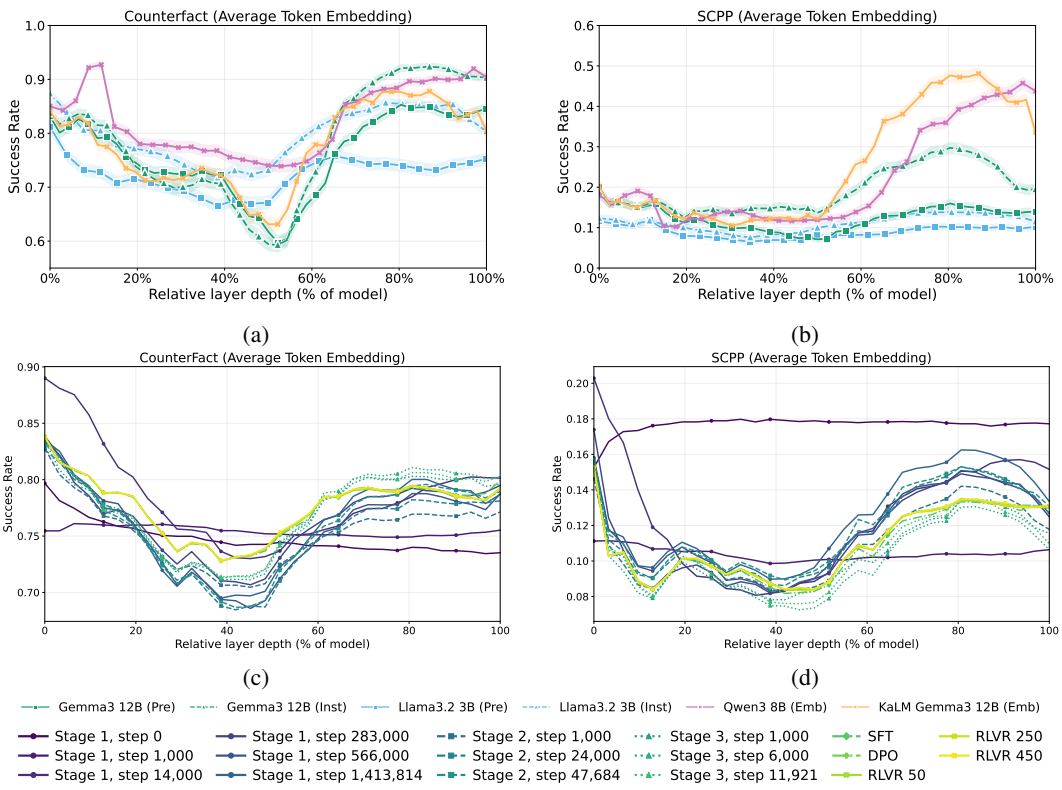


Figure 1: Layer-wise success rates on the lexical influence stress tests. Panels (a) and (b) compare mean pooled representations across pretrained, instruction-tuned, and embedding models on CounterFact and SCPP, respectively; shaded bands denote 95% bootstrap confidence intervals. Panels (c) and (d) track the same evaluation across OLMo 3 7B training checkpoints.

dependence is specific to the representation readout. We evaluate autoregressive models using learned token pooling, per dimension scaling, and full dimensional linear and nonlinear transformations trained on general semantic data, without using CounterFact or SCPP. Complete experimental details and corresponding results with each learned readout trained on last-token representations are provided in Appendix C.

For learned token pooling, we replace uniform mean pooling with a lightweight attention-based pooling readout, inspired by prior approaches that learn token importance for sentence-level representations (Lee et al., 2019; Qin et al., 2025). At each layer, a learned query assigns context-dependent weights to token representations. Figure 2 shows improvements across models and depths. On SCPP, for example, Gemma-3-12B-IT increases from roughly 0.15 to 0.32 at intermediate depths and from roughly 0.29 to 0.48 at later depths. Despite these gains, later layers remain stronger, preserving the intermediate weakness in the depth profile.

Per-dimension scaling changes the relative contribution of existing coordinates, while the linear and nonlinear readouts learn full dimensional transformations of the representation. Semantic discrimination improves under all three readouts, although the magnitude of the gain varies across models, datasets, and depth. Per-dimension scaling produces comparatively modest changes, while the full dimensional readouts yield larger gains. On average, mean-pooled success rate on SCPP increases from 14.63% at baseline to 33.23% with the linear readout and 32.84% with the nonlinear readout. On CounterFact, the corresponding values increase from 78.56% to 82.38% and 80.08%. The nonlinear readout provides no systematic improvement over the linear readout. Possible causes and limitations of this result are discussed in Appendix C. Among all learned readouts, learned token pooling yields the strongest overall gains. Together, these results show that substantially different ways of extracting semantic information can improve absolute discrimination without eliminating lexical influence or the broader differences across depth. Thus, the observed depth profile is robust to changes in both token aggregation and the weighting or transformation of hidden dimensions.

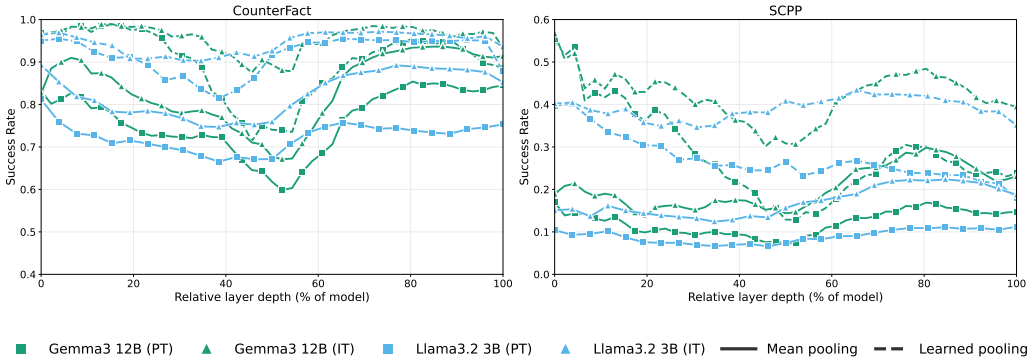


Figure 2: Layer-wise lexical-influence stress-test performance with learned semantic pooling. Panels report success rates on CounterFact and SCPP, respectively, comparing mean pooling with learned pooling for pretrained and instruction-tuned models.

Architectural controls. We further examine the depth pattern across architectures and within Transformer blocks. Bidirectional full attention encoders still show intermediate weakening under mean pooling, but not the same trajectory as the autoregressive models, while [CLS] representations show no consistent profile (Appendix D). Within the autoregressive models discussed above, no single block component consistently accounts for the pattern. Feedforward updates contribute more strongly to the decline and recovery in Gemma, whereas attention accounts for many of the larger changes in Llama (Appendix E). Additional CounterFact and SCPP evaluations on three autoregressive models, the reasoning oriented DeepSeek R1 Distill Llama 8B, the larger Gemma 3 27B IT, and Qwen3 14B, also show intermediate degradation (Appendix B.1).

4 Probing Lexical and Semantic Structure

In this section, we relate lexical-influence failures to the accessibility of lexical and semantic information across layers. We measure **lexical decodability** by predicting the input token from its hidden state, using this as an operational proxy for surface-form information. We measure **semantic fidelity** by evaluating layer-wise sentence embeddings with standard MTEB protocols. The base model is kept frozen throughout all evaluations.

4.1 Lexical Probes

For each layer ℓ and token position t, we test whether input-token identity is affinely recoverable from its hidden state $H_\ell(x)_t \in \mathbb{R}^d$. We use the model's frozen input embedding matrix $E_{\text{in}} \in \mathbb{R}^{V \times d}$ as a natural reference basis for lexical form, since it maps discrete token identities into the representation space. A layer-specific affine probe produces

$$z_{\ell,t} = W_\ell H_\ell(x)_t + b_\ell, \qquad p_\ell(v \mid x, t) = \frac{\exp(\langle z_{\ell,t}, e_v \rangle)}{\sum_{u=1}^{V} \exp(\langle z_{\ell,t}, e_u \rangle)}.$$

The probe is trained with cross-entropy to recover the original input token. The details regarding training and hyperparameters for the probes are provided in Appendix G

Figure 3 (a and b) reports top-1 accuracy and cross-entropy (CE) for token-identity probes across relative depth. Token identity is highly linearly decodable in early layers, becomes least accessible around mid-depth, coinciding with the intermediate regime identified in § 3, partially recovers in later layers, and declines again near the final layers. Thus, lexical influence remains strong precisely where linear accessibility of token identity is reduced. Prior work links intermediate layers to more syntactic abstractions (Jawahar et al., 2019), providing a complementary view of this region of the network. We conjecture that the final decline reflects increasing specialization toward the model's prediction objective. More broadly, this non-monotonic trajectory argues against a smooth progression from lexical to semantic.

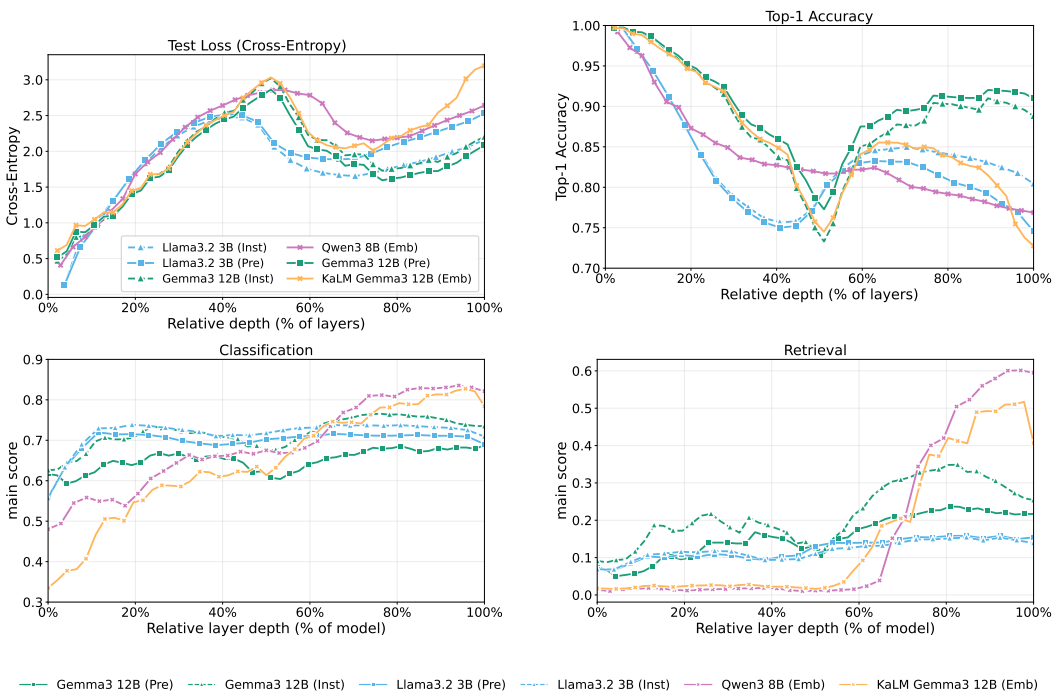


Figure 3: Layer-wise lexical decodability and semantic task performance across model depth. **(top panels)** Lexical probe performance on WikiText, measured by cross-entropy loss and Top-1 token accuracy. **(bottom panels)** MTEB classification and retrieval performance using sentence embeddings extracted at each layer, averaged across tasks within each category.

4.2 Semantic Fidelity Probes

We next ask whether the depth-dependent behavior observed in our stress tests also appears on standard semantic embedding tasks. We follow MTEB's two complementary evaluation settings to probe semantic fidelity in layerwise sentence embeddings. MTEB measures performance either directly from embedding similarities and induced rankings (no task-specific training) or by training a lightweight linear probe to test whether task-relevant information is linearly accessible at a given layer. Dataset and evaluation details are provided in Appendix H.

Because MTEB performance may itself benefit from lexical overlap, we assess this potential confound in Appendix H.2. In clustering and retrieval, positive pairs generally have greater lexical overlap than negatives. We further partition ArguAna examples by whether lexical overlap supports the gold relevance relation or instead favors a non-relevant document. Although lexical conflict lowers absolute performance, it largely preserves the layer-wise profile (distance correlation $dCor = 0.9890$ for Gemma and 0.9858 for Llama). We therefore interpret these results primarily through their layer-wise trajectories rather than absolute scores.

Figure 3 (c and d) reports classification and retrieval performance, averaged across datasets within each MTEB category. Results for clustering, pairwise classification, STS, and reranking are provided in Appendix H.1. Although these tasks differ substantially, several show a similar depth-dependent profile. For pretrained and instruction-tuned models, classification, pairwise classification, retrieval, and STS generally improve in early layers, weaken or stagnate over an intermediate region, and recover more clearly toward the top of the network. Clustering and reranking do not show the same profile. Reranking remains comparatively stable across depth, while clustering is noisier under the default MTEB setup, where small representation changes can trigger discrete cluster reassignments.

For embedding models, performance over intermediate depths either remains roughly constant or improves slowly, before increasing sharply later. Peak scores for all models typically occur in the last few layers, but not consistently at the final layer, consistent with prior layer-wise findings (Skean et al., 2025; Cheng et al., 2025). Together with § 3, these results further characterize the intermediate regime. Exact token identity becomes least linearly accessible over this regime, while semantic

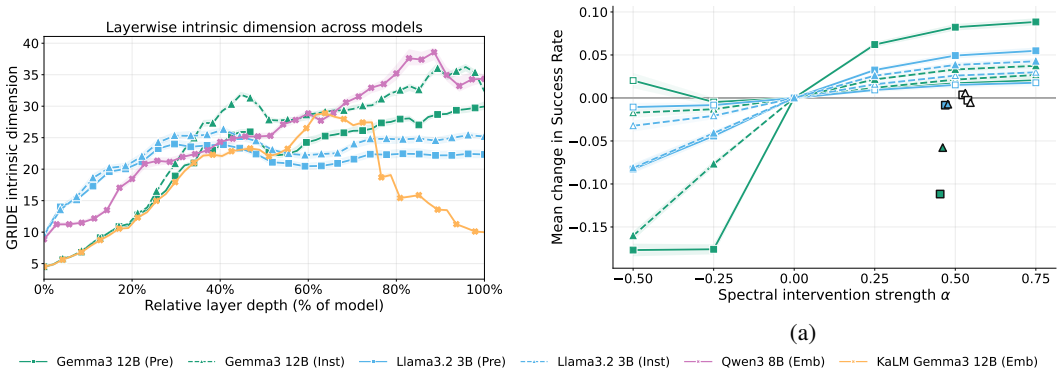


Figure 4: Layer-wise representation geometry and spectral intervention. Panel (left) reports GRIDE intrinsic dimension across depth (last token). Panel (right) reports the change in lexical stress-test success rate under spectral interventions relative to $\alpha = 0$. Dark outlined markers denote magnitude-matched random perturbation controls.

fidelity on several tasks weakens or stagnates and recovers only later. Consequently, the strongest representations typically emerge in mid-to-late layers.

5 Geometry of the Intermediate Regime

The lexical stress tests and probing results in §3 and §4 identify an intermediate regime in which semantic discrimination weakens alongside reduced linear accessibility of exact token identity. Prior work has shown that representation geometry changes systematically across Transformer depth (Skean et al., 2025; Valeriani et al., 2023; Cheng et al., 2025; Doimo et al., 2024). We therefore examine whether this regime aligns with changes in prompt entropy and intrinsic dimension. Intrinsic dimension estimates the local dimensionality of the representation space, while prompt entropy measures how broadly token representations are distributed within an input. We estimate intrinsic dimension with GRIDE (Denti et al., 2022) for both last-token and mean-pooled representations. Figure 4(left panel) reports the last-token results, providing a common comparison across all models. Intrinsic-dimension growth stalls or partially contracts over the intermediate regime before increasing again later. The corresponding mean-pooled analysis (Appendix I) shows a broader decline over the same region. Prompt entropy does not show the same consistent structure across models and is also reported in the appendix. The same broad intrinsic-dimension pattern also emerges across OLMo training checkpoints (Appendix I.1).

These geometric results motivate a controlled intervention on representation geometry. We rescale spectral directions of the mean-pooled representations used in our primary stress test to either flatten or concentrate variance spectrum. Figure 4(right panel) shows small but consistent effects. Flattening generally improves stress-test success, whereas concentrating the spectrum reduces it. Magnitude-matched random-direction perturbations do not show the same improvement, indicating that perturbation magnitude alone does not account for the effect. To test whether this effect is specific to spectral rescaling, Appendix I.2 compares it with BERT-flow (Li et al., 2020), an independently learned method for isotropizing sentence representations. Both methods yield improvements of similar magnitude, and their gains extend across depth rather than concentrating in the intermediate regime. Together, these results suggest that the intermediate regime is associated with a stalled or contracting intrinsic dimensionality. Modifying the overall spectrum can modestly change stress-test performance, but does not remove the characteristic depth-dependent pattern.

6 Practical Implications

In this section, we explore the impact of lexical influence on practical settings that rely on embedding geometry and representation-sensitive updates.

Table 1: Summarization ranking results (lexical distractor vs. gold). FR is the fraction of cases where the distractor is scored higher than the gold; ASD/ASG are mean scores for distractor/-gold (method-specific scales).

Method	FR ↓	ASD ↓	ASG ↑
BERTScore	0.640	0.691	0.649
BARTScore	0.200	-3.918	-2.945

Table 2: Mean distribution-shift metrics for lexically similar (LS) and lexically dissimilar (LD) prompts. Arrows indicate greater edit impact.

Metric	Top-K-20		Top-K-50	
	LS	LD	LS	LD
JSD ↑	0.290	0.242	0.293	0.245
TV ↑	0.561	0.493	0.565	0.499
Kendall τ	-0.016	0.030	-0.022	0.028
ETP ↑	0.023	0.021	0.085	0.043
Top-1 Flip Rate *(Top-K-indep.)*			0.566	0.493

6.1 SUMMARIZATION

Automatic summarization metrics aim to reward preservation of source content, but their scores can also be influenced by lexical overlap. Traditional overlap-based metrics such as ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002) explicitly reward n-gram matching, while model-based metrics such as BERTScore (Zhang et al., 2019) and BARTScore (Yuan et al., 2021) are intended to capture semantic similarity or conditional generation quality. We test whether lexical overlap is sufficient for model-based metrics to prefer a lexically aligned non-summary distractor over the gold summary.

Data. We use the Extreme Summarization benchmark Narayan et al. (2018). To probe lexical sensitivity, we augment each example with a non-summary distractor that has high lexical overlap with the reference and is generated by an LLM. Details for dataset construction and experimental methodology are provided in Appendix J.2.

Experiment. For each document, we score (i) the gold summary and (ii) an LLM-generated lexical distractor against the source document using *BERTScore* and document-conditioned *BARTScore*. We count a *failure* when the distractor scores higher than the gold summary. BERTScore fails on 64% of examples, consistent with a strong lexical signal in contextual representations, mirroring our earlier findings. BARTScore fails on 20%: despite being likelihood-based, lexical distractors can remain locally plausible and score competitively. This shows that lexical reliance is not limited to contextual representation similarity metrics, and the token-level generative process can reward lexically aligned text as well.

6.2 MODEL EDITING

Model editing enables targeted model updates while aiming to preserve behavior on unrelated inputs Meng et al. (2022); Mitchell et al. (2021), avoiding the cost of retraining and the forgetting associated with fine-tuning (Hartvigsen et al., 2023; Rizwan et al., 2025; Luo et al., 2025). Given an edit request $(x_e, y_e^\star)$, a successful edit should change the requested fact while preserving behavior on unrelated prompts in a locality set $\mathcal{N}$. We focus on weight-modifying editing, where this corresponds to updating θ to θ' such that $M_{\theta'}(x_e) = y_e^\star$ while $M_{\theta'}(\cdot \mid x)$ remains close to $M_\theta(\cdot \mid x)\ \forall x \in \mathcal{N}$.

Rizwan et al. (2025) showed that semantic similarity based scoping systems can fail in weight preserving editing due to lexical overlap. We conjecture that because lexically similar prompts produce similar internal representations, a parameter update targeted at one prompt can disproportionately affect semantically unrelated prompts with similar surface form. We test this by comparing unintended changes on lexically similar and lexically dissimilar locality prompts.

Method and dataset. We use AlphaEdit (Fang et al., 2025) and evaluate on a modified version of the CounterFact benchmark. For each edit, we augment CounterFact with two prompt sets: (*i*) *lexically similar* variants created by substituting distractor entities into the edit prompt, and (*ii*) *lexically dissimilar* variants that keep the distractor's semantics but substantially change its surface form.

Metrics. We measure post-edit distribution shift at the answer position using Jensen–Shannon divergence (JSD), total variation (TV), Kendall's τ for pre-/post-edit rank agreement over merged top-K support, and Top-1 Flip Rate (fraction of prompts whose argmax token changes after editing). We additionally report EDITED-TARGET PRESENCE (ETP), the fraction of locality prompts whose

edited-model top-K contains a relation-specific edited target token. Full methodological details and top-K $= 100$ results are provided in Appendix K.

Table 2 shows that lexically similar (LS) locality prompts are more affected by the edit than lexically dissimilar (LD) controls, despite both querying facts unrelated to the edited target. LS yields consistently higher JSD and TV across K, indicating larger changes in the post-edit probability distribution, and lower Kendall's τ for LS indicates greater post-edit rank disruption. Top-1 Flip Rate, which is independent of K, is likewise higher for LS. Finally, EDITED-TARGET PRESENCE (ETP) is more frequent for LS, with the separation becoming clearer as the evaluated support expands. Together, these results show that lexically similar but semantically different locality prompts are disproportionately perturbed, producing systematic skew in edit locality.

Implication. Surface-form overlap increases susceptibility to unintended changes, motivating scope evaluations of representation and parameter-modifying interventions (e.g., steering, activation methods, adapters, fine-tuning) with lexically similar control prompts that request different information.

7 RELATED WORK

Layer-wise probing studies show that linguistic properties are not uniformly encoded across depth, but peak at different layers (Liu et al., 2019; Tenney et al., 2019; Jawahar et al., 2019); later work strengthens these analyses with selectivity and control tasks (Hewitt & Liang, 2019). Complementary geometric and similarity-based approaches study how representation spaces evolve across layers (Ethayarajh, 2019; Kornblith et al., 2019), including systematic changes in intrinsic dimension and neighborhood structure (Valeriani et al., 2023). Recent work further identifies informative intermediate-layer features (Skean et al., 2025), high-dimensional abstraction phases (Cheng et al., 2025), staged inference dynamics (Lad et al., 2026), and representation changes induced by few-shot learning and fine-tuning (Doimo et al., 2024). Related depth-centric work traces the emergence of concepts and knowledge across layers (Haider et al., 2024; Jin et al., 2025) and model predictions can stabilize well before the final layers (Sajjad et al., 2023; Fan et al., 2025). These findings motivate examining how lexical and semantic signals interact as representations change across depth.

Lexical influence is closely related to shortcut learning, where models exploit surface cues that correlate with labels rather than semantics. In NLI, annotation artifacts introduce exploitable lexical patterns (Gururangan et al., 2018; He et al., 2019; Poliak et al., 2018), while controlled diagnostics such as HANS expose failures driven by lexical overlap and subsequence heuristics (McCoy et al., 2019). Large-scale evaluations likewise document vulnerability to lexical shortcuts (Yuan et al., 2024), which can persist after mitigation (Serrano et al., 2023; Utama et al., 2020; Zhou & Bansal, 2020). A complementary line of work modifies sentence-embedding geometry to improve semantic similarity, including BERT-flow and whitening (Li et al., 2020; Su et al., 2021). Our work connects these directions by tracking lexical influence across depth and training, relating it to lexical accessibility, semantic fidelity, and representation geometry, and showing how lexical sensitivity affects summarization evaluation and model-editing locality.

8 CONCLUSION

We study how lexical influence evolves across model depth and relates to semantic structure. Across pretrained, instruction-tuned, and embedding models, semantic adaptation improves representation quality without eliminating overlap-driven similarity failures. In the autoregressive models, semantic discrimination weakens over an intermediate regime alongside reduced linear accessibility of exact token identity, arguing against a simple lexical-to-semantic progression with depth. OLMo checkpoints show that this pattern emerges early in pretraining and persists through later adaptation, while learned pooling and semantic transformations improve absolute discrimination without removing the broader depth-dependent organization. Geometric analysis further shows that this intermediate weakness aligns with stalled or contracting intrinsic dimension. Finally, we show that lexical sensitivity extends to downstream LLM use, affecting both automatic summarization evaluation and the locality of model editing. Together, these findings motivate lexically controlled stress tests for representation learning, automatic evaluation, and model interventions. Future work should further characterize the mechanisms underlying the intermediate regime and develop targeted interventions to reduce overlap-driven semantic errors.

REFERENCES

Emily Cheng, Diego Doimo, Corentin Kervadec, Iuri Macocco, Lei Yu, Alessandro Laio, and Marco Baroni. Emergence of a high-dimensional abstraction phase in language transformers. In *The Thirteenth International Conference on Learning Representations*, 2025.

Francesco Denti, Diego Doimo, Alessandro Laio, and Antonietta Mira. The generalized ratios intrinsic dimension estimator. *Scientific Reports*, 12(1):20005, 2022.

Diego Doimo, Alessandro Serra, Alessio Ansuini, and Alberto Cazzaniga. The representation landscape of few-shot learning and fine-tuning in large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/206018a258033def63607fbdf364bd2d-Abstract-Conference.html.

Sri Harsha Dumpala, Aman Jaiswal, Chandramouli Shama Sastry, Evangelos E. Milios, Sageev Oore, and Hassan Sajjad. SUGARCREPE++ dataset: Vision-language model sensitivity to semantic and lexical alterations. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 55–65. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-1006.

Siqi Fan, Xin Jiang, Xiang Li, Xuying Meng, Peng Han, Shuo Shang, Aixin Sun, and Yequan Wang. Not all layers of llms are necessary during inference. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 5083–5091. ijcai.org, 2025.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=HvSytvg3Jh.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pp. 107–112. Association for Computational Linguistics, 2018.

Muhammad Umair Haider, Umar Farooq, AB Siddique, and Mark Marron. Looking into black box code language models. *arXiv preprint arXiv:2407.04868*, 2024.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36:47934–47959, 2023.

He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. In Colin Cherry, Greg Durrett, George F. Foster, Reza Haffari, Shahram Khadivi,

Nanyun Peng, Xiang Ren, and Swabha Swayamdipta (eds.), *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP, DeepLo@EMNLP-IJCNLP 2019, Hong Kong, China, November 3, 2019*, pp. 132–142. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-6115. URL https://doi.org/10.18653/v1/D19-6115.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1275. URL https://doi.org/10.18653/v1/D19-1275.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 3651–3657. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1356. URL https://doi.org/10.18653/v1/p19-1356.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, Fan Yang, Mengnan Du, and Yongfeng Zhang. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pp. 558–573. Association for Computational Linguistics, 2025.

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

Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. Remarkable robustness of llms: Stages of inference? *Advances in Neural Information Processing Systems*, 38:130050–130083, 2026.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 9119–9130, 2020.

Michael Li and Nishant Subramani. Model internal sleuthing: Finding lexical identity and inflectional morphology in modern language models. *arXiv preprint arXiv:2506.02132*, 2025.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 1073–1094. Association for Computational Linguistics, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 3428–3448. Association for Computational Linguistics, 2019.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: massive text embedding benchmark. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pp. 2006–2029. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EACL-MAIN.148. URL https://doi.org/10.18653/v1/2023.eacl-main.148.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, 2018.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In Malvina Nissim, Jonathan Berant, and Alessandro Lenci (eds.), *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, *SEM@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pp. 180–191. Association for Computational Linguistics, 2018. doi: 10.18653/V1/S18-2023. URL https://doi.org/10.18653/v1/s18-2023.

Jin Qin, Zihan Liao, Ziyin Zhang, Hang Yu, Peng Di, and Rui Wang. C2llm technical report: A new frontier in code retrieval via adaptive cross-attention pooling. *arXiv preprint arXiv:2512.21332*, 2025.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Hammad Rizwan, Domenic Rosati, Ga Wu, and Hassan Sajjad. Resolving lexical bias in model editing. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=aPm6SfcMWQ.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 4974–4986, 2021.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the effect of dropping layers of pre-trained transformer models. *Comput. Speech Lang.*, 77:101429, 2023.

Sofia Serrano, Jesse Dodge, and Noah A. Smith. Stubborn lexical bias in data and models. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 8131–8146. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.516. URL https://doi.org/10.18653/v1/2023.findings-acl.516.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=WGXb7UdvTX.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4593–4601. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/p19-1452.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Towards debiasing NLU models from unknown biases. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 7597–7610. Association for Computational Linguistics, 2020.

Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. The geometry of hidden representations of large transformer models. *Advances in Neural Information Processing Systems*, 36:51234–51252, 2023.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. EasyEdit: An easy-to-use knowledge editing framework for large language models. In Yixin Cao, Yang Feng, and Deyi Xiong (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 82–93, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-demos.9.

Jinming Xing, Dongwen Luo, Chang Xue, and Ruilin Xing. Comparative analysis of pooling mechanisms in llms: A sentiment analysis perspective. *arXiv preprint arXiv:2411.14654*, 2024.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: evaluating generated text as text generation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp. 27263–27277, 2021.

Yu Yuan, Lili Zhao, Kai Zhang, Guangting Zheng, and Qi Liu. Do llms overcome shortcut learning? an evaluation of shortcut challenges in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 12188–12200. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.679. URL https://doi.org/10.18653/v1/2024.emnlp-main.679.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Xinping Zhao, Xinshuo Hu, Zifei Shan, Shouzheng Huang, Yao Zhou, Xin Zhang, Zetian Sun, Zhenyu Liu, Dongfang Li, Xinyuan Wei, Youcheng Pan, Yang Xiang, Meishan Zhang, Haofen Wang, Jun Yu, Baotian Hu, and Min Zhang. Kalm-embedding-v2: Superior training techniques and data inspire a versatile embedding model, 2025. URL https://arxiv.org/abs/2506.20923.

Zheng Zhao, Yftah Ziser, and Shay B Cohen. Layer by layer: Uncovering where multi-task learning happens in instruction-tuned large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

Xiang Zhou and Mohit Bansal. Towards robustifying nli models against lexical dataset biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8759–8771, 2020.

LIMITATIONS

We compare models using relative layer depth to account for differences in network depth. This enables comparison of broad trajectories across architectures, but does not imply that layers at the same relative depth are functionally equivalent.

IMPACT STATEMENT

This work characterizes lexical influence in representation spaces, showing that surface-form overlap can distort embedding similarity and miscalibrate similarity-based evaluation under lexical confounds. It also introduces diagnostics that quantify when and where semantic structure degrades. We expect the primary impact to be positive: improving evaluation rigor and informing more robust use of embeddings in retrieval, ranking, and monitoring pipelines. While the findings could in principle be used to stress similarity-based systems via lexically similar distractors, such misuse is likely limited in practice because it generally requires model and pipeline-specific knowledge and sustained control over inputs; more often, the results will help identify and mitigate unintended brittleness.

AI USE STATEMENT

LLMs were used to improve grammar, wording, and prose clarity, and to generate the adversarial summarization distractors and model-editing prompt variants described in Appendices J.2 and K. All experimental design, analysis, and conclusions were developed and verified by the authors.

REPRODUCIBILITY STATEMENT

The appendices provide training details for lexical probes and learned readouts, semantic-fidelity evaluations, representation-geometry methods, downstream evaluation details, and compute resources.

A DATASETS

For all experiments in the paper, we prioritize large evaluation sets and aggregate layer-wise trends over repeated split-based reruns, given the breadth of the study across models, tasks, and layers.

Stress Tests and Model Editing. Table 3 shows representative examples from CounterFact and SugarCrepe++ (SCPP). Each instance specifies a factual association of the form (subject, relation, object) together with a set of prompts designed to elicit that fact. It includes (i) *anchor* prompts that directly query the target relation for the subject, (ii) *paraphrase* prompts that ask for the same fact using alternative surface forms to test generalization beyond a single template, and (iii) *distractor* prompts that query the same relation for semantically related subjects. For model editing, distractors are used to test specificity/locality (i.e., whether the edit remains localized). After an edit, a successful model should produce the new target object for the anchor/paraphrase prompts while leaving predictions on distractor prompts largely unchanged.

SCPP provides image–caption triplets with two semantically equivalent but lexically different positives and a hard negative that is lexically confusable yet semantically incorrect for the image. We utilize only the texts for our paper. It partitions negatives into five subsets based on the perturbation used to create the distractor: *Swap Object* exchanges two object nouns (or noun phrases) in the caption; *Swap Attribute* swaps attributes between objects (e.g., colors/sizes); *Replace Object* substitutes an object noun with another; *Replace Attribute* substitutes an attribute term with another; and *Replace Relation* substitutes the relation predicate (e.g., action or spatial preposition), altering who-does-what-to-whom or the spatial configuration.

A.1 Stress Test Sample Details

For the adversarial stress tests, we use the full SCPP dataset, consisting of 4,757 samples, together with 5,000 samples from CounterFact. CounterFact exists in several variants, including the original version released with ROME (Meng et al., 2022), as well as later sampled or updated versions from EasyEdit (Wang et al., 2024) and PENME (Rizwan et al., 2025). The original dataset contains two paraphrase prompts per sample, whereas the EasyEdit version contains one paraphrase prompt and one locality/neighborhood prompt per sample. The generation prompts contained in the dataset are not exact paraphrases, but rather prompts that may lead to the edited answer. For example, for the edit prompt "Autonomous University of Madrid, which is located in," one generation prompt is "The best restaurants around Autonomous University of Madrid include." Since this is not an exact paraphrase, we cannot use these prompts.

We therefore use the version released by PENME (Rizwan et al., 2025), which provides the two original paraphrases along with several paraphrases generated via LLM (3 minimum per sample). There are 10 locality/neighborhood prompts for each sample. For this dataset, we used token-level Jaccard overlap to rank locality prompts by lexical similarity to the edit prompt to obtain a clear separation between five lexically similar samples and five dissimilar ones. Using this ranking, we sampled 5,000 samples from the dataset. For the model editing experiments, we used this sampled dataset. For the stress tests, we restricted evaluation to the lexically similar locality prompts, resulting in 25 triplet evaluations per sample and $125,000$ evaluations in total.

For both stress-test datasets, we compute pointwise confidence intervals using a paired nonparametric bootstrap over evaluation examples. For each of 10,000 replicates, we sample with replacement the full number of examples in the corresponding dataset, 5,000 for CounterFact and 4,757 for SCPP, and reuse the same sampled indices across all models and layers. We report the 2.5th and 97.5th percentiles as 95% confidence intervals, using random seed 42.

B Further Results Lexical Influence

The last-token results are shown in Figure 5 (a) and (b). Overall, for the pretrained and instruction-tuned models, last-token representations perform worse than mean-pooled representations, particularly on CounterFact. Outside the first few layers, success rates generally increase with depth, suggesting that the final-token representation becomes more effective at capturing sequence-level context as information is integrated across the network. This improvement does not extend to the final layers, where performance decreases slightly. We conjecture that this late decline reflects increasing specialization toward next-token prediction, making the final-token state less suitable as a general semantic representation.

To test whether the weaker performance of last-token representations is partly driven by terminal-token identity, we partition CounterFact and SCPP examples according to whether the anchor and distractor

Table 3: The table shows samples from CounterFact and SugarCrepe++ (SCPP) datasets.

Dataset	Prompt/Anchor	Paraphrase	Distractor
CounterFact	Justin Fleming was originally from	Justin Fleming was native to	Andrew Durante is originally from
SCPP++ replace attribute	People standing near a table with open pizza boxes	Individuals are positioned close to a table with open pizza boxes.	People standing near a table with closed pizza boxes.
SCPP++ replace relation	A motorcycle is parked inside of a building.	The building contains the parked motorcycle.	A motorcycle is parked outside of a building.
SCPP++ swap attribute	A person opens a fire hydrant and brown colored water is flowing out.	Brown-colored water flows out of the fire hydrant opened by a person.	A person opens a brown colored hydrant and water is flowing out.
SCPP++ replace object	Multiple time pieces displayed on the wall and on a flat surface.	Several timepieces are exhibited on a surface which is flat, and on the wall.	Multiple time pieces displayed on the ceiling and on a flat surface.
SCPP++ swap object	A child holds a spoon and looks at a cupcake.	The child looks at a cupcake while holding a spoon.	A child looks at a spoon and holds a cupcake.

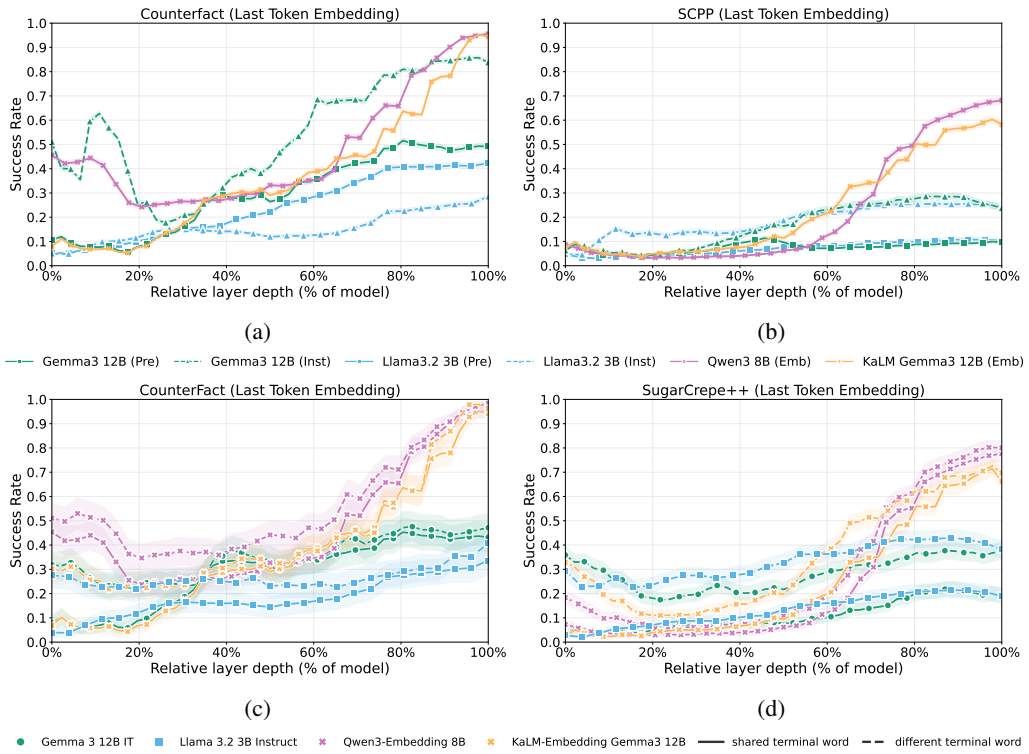


Figure 5: Success rates for lexical-influence stress tests on CounterFact and SCPP using last-token representations. Panels (a) and (b) report the standard layer-wise results on CounterFact and SCPP, respectively. Panels (c) and (d) partition examples by whether the anchor and distractor share the same terminal word, showing the effect of terminal-word overlap on CounterFact and SCPP, respectively.

share the same terminal word. CounterFact contains 4,759 shared-terminal and 241 different-terminal examples, while SCPP contains 4,127 and 630, respectively. Because these partitions are highly imbalanced, we bootstrap both groups at the size of the smaller partition using 1,000 prompt-level resamples with replacement. Figure 5 (c) and (d) report the resulting mean success rates and 95% confidence intervals for the instruction-tuned and embedding models.

Across all four models, semantic discrimination is consistently weaker when the anchor and distractor share their terminal word. The separation is visible on CounterFact and becomes particularly pronounced on SCPP, where shared-terminal examples remain substantially harder across much of the network. Notably, the same effect appears in the embedding models despite their explicit training for semantic representation. Thus, sensitivity to terminal-token identity persists even when the last-token representation provides stronger overall semantic discrimination.

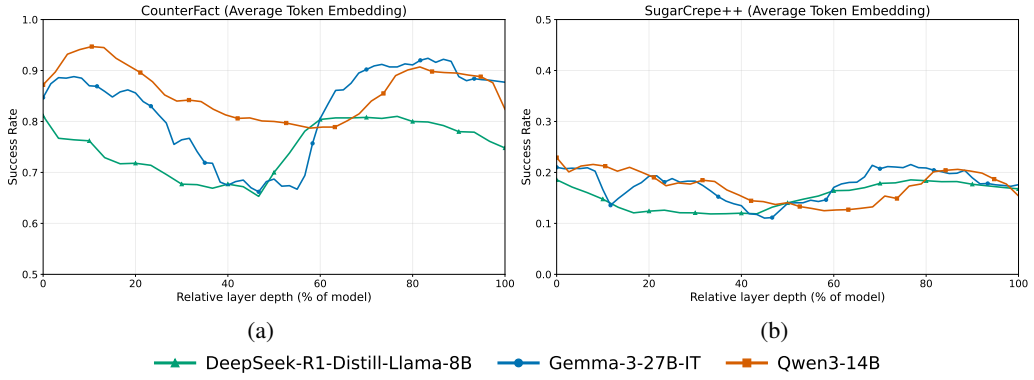


Figure 6: Layer-wise lexical-influence stress-test performance for additional autoregressive models. Panels (a) and (b) report mean-pooled success rates on CounterFact and SCPP, respectively, across a reasoning-oriented model, a larger-scale variant, and a model from a different family.

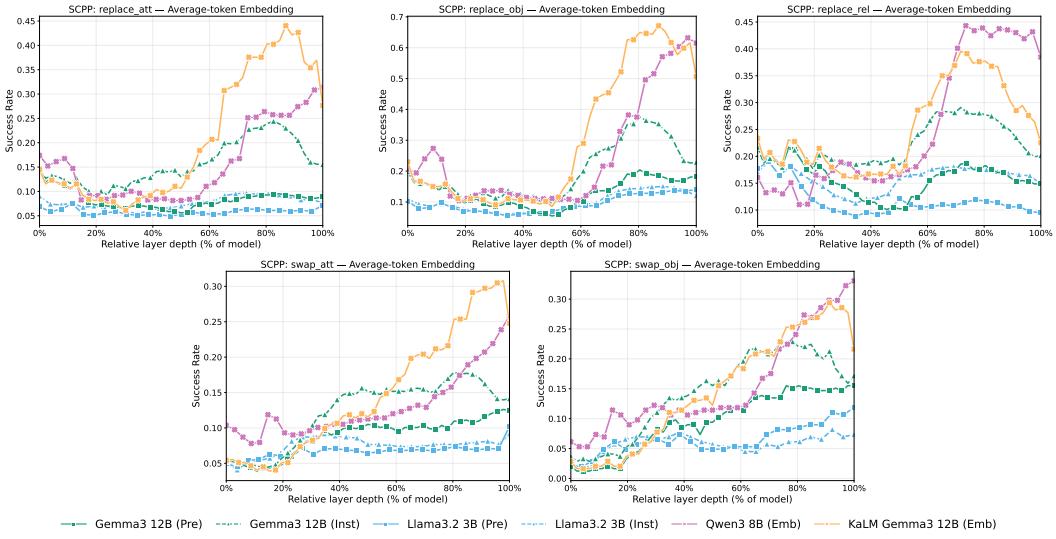


Figure 7: Layer-wise lexical-influence stress-test performance on the SCPP subtasks using mean-pooled representations. Results are shown separately for attribute, object, and relation replacements and swaps.

B.1 Additional autoregressive models.

To test whether the observed depth-dependent behavior is specific to the models used in the main analysis, we repeat the lexical-influence stress tests with DeepSeek-R1-Distill-Llama-8B, Gemma 3 27B-IT, and Qwen3-14B. These models broaden the evaluation across model families, scales, and training regimes, including a reasoning-oriented model. Figure 6 shows that the same broad intermediate weakness extends beyond the models in the main analysis. On CounterFact, DeepSeek and Gemma exhibit a pronounced decline through intermediate layers followed by substantial recovery at greater depth, while Qwen shows a weaker but related non-monotonic profile. SCPP shows a similar overall tendency, although the curves are flatter and success remains lower. These results further support that the mid-depth weakness is not specific to a single model family, scale, or training regime, while its precise shape and magnitude remain model dependent.

B.2 Results for Lexical Influence Test on SCPP (subtasks)

Figures 7, and 8 show performance for the selected mean pooled and last token embeddings, respectively, for models referenced in § 2.

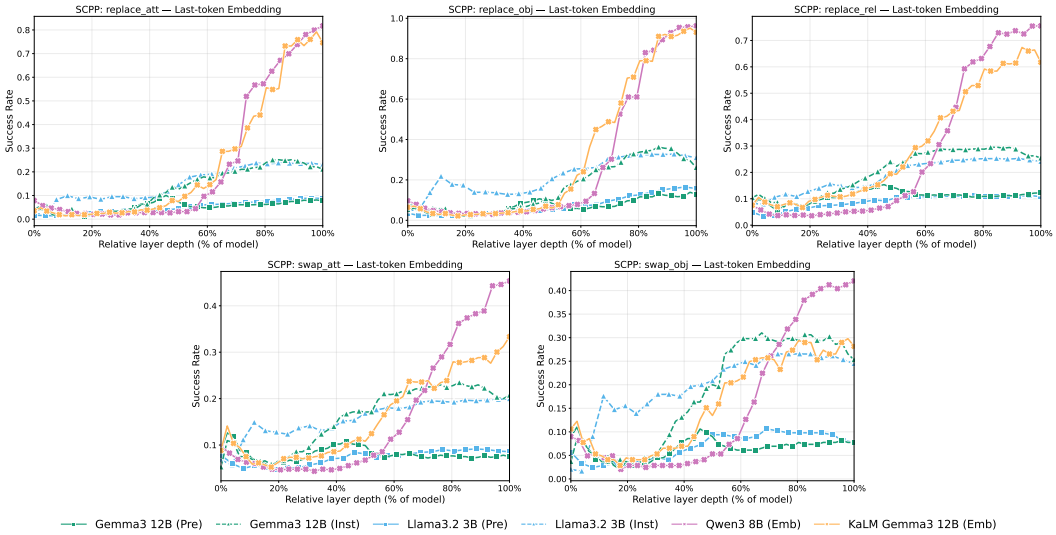


Figure 8: Layer-wise lexical-influence stress-test performance on the SCPP subtasks using last-token representations. Results are shown separately for attribute, object, and relation replacements and swaps.

C Learned Semantic Transformation and Learned Pooling

Linear and Nonlinear Semantic Readouts. We first ask whether nonlinear transformation provides additional semantic discrimination beyond a full dimensional linear readout. We use the sentence level representations defined in the main text, considering the mean pooled representation $\mathbf{h}_\ell^{M,\mathrm{mean}}(x)$ and the last token representation $\mathbf{h}_\ell^{M,\mathrm{last}}(x)$ in separate experiments. The language model remains frozen throughout. For notational simplicity, we write $\mathbf{h}_\ell^{M}(x)$ below for either representation type. At each layer, we train a linear affine readout

$$\mathbf{z}_\ell^{M,\mathrm{lin}}(x) = \mathrm{norm}_2\left(\mathbf{W}_\ell^{\mathrm{lin}}\mathbf{h}_\ell^{M}(x) + \mathbf{b}_\ell^{\mathrm{lin}}\right), \tag{1}$$

or a nonlinear readout

$$\mathbf{z}_\ell^{M,\mathrm{nonlin}}(x) = \mathrm{norm}_2\left(\phi\left(\mathbf{W}_\ell^{\mathrm{nonlin}}\mathbf{h}_\ell^{M}(x) + \mathbf{b}_\ell^{\mathrm{nonlin}}\right)\right), \tag{2}$$

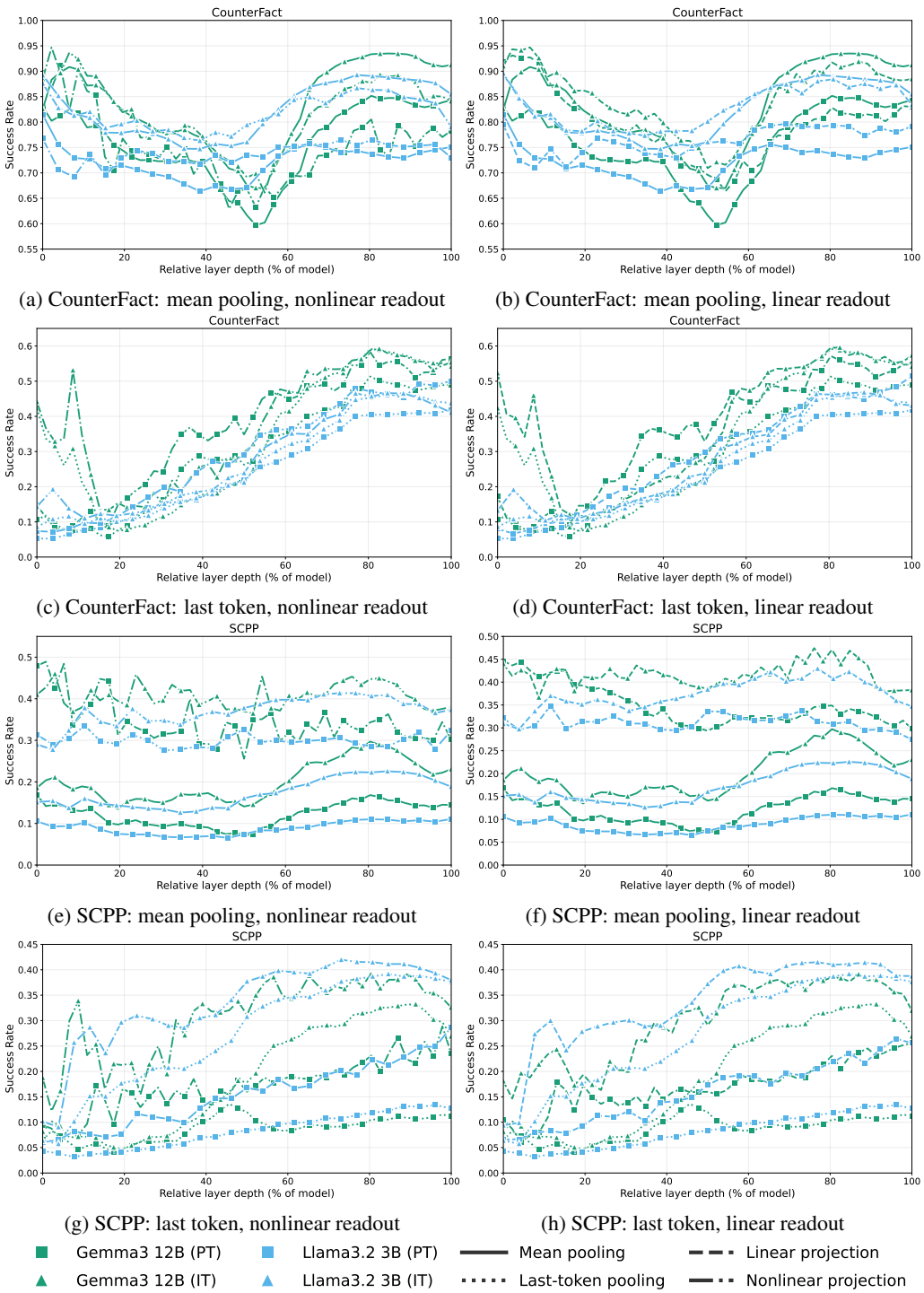
where ϕ is GELU in the primary nonlinear experiment. We additionally evaluate ReLU, Leaky ReLU, and Tanh followed by RMSNorm; the qualitative outcome remains unchanged across nonlinearities. The transformation matrices are Xavier initialized, biases are initialized to zero, and only the readout parameters are optimized.

We train the readouts jointly on SNLI, MultiNLI, STS-B, SICK-R, and PAWS, providing supervision for entailment, semantic similarity, paraphrase, and compositional meaning. To prevent lexical overlap alone from solving the objective, positive and negative pairs are matched by token-level Jaccard overlap. Text is lowercased and tokenized with `[A-Za-z0-9]+`; $[0, 1]$ is divided into ten equal-width Jaccard bins, and positives are matched without replacement only to negatives from the same bin within each dataset and split. Unmatched examples are discarded. The resulting data contain 4,057 training and 467 validation comparisons (SNLI: 1,000/100; MultiNLI: 1,000/100; STS-B: 667/87; SICK-R: 390/80; PAWS: 1,000/100).

For each layer, we optimize the margin-ranking loss

$$\mathcal{L}_\ell = \left[d_\ell(x_p, x_p') - d_\ell(x_n, x_n') + \gamma\right]_+, \qquad d_\ell(x, y) = \left\|\mathbf{z}_\ell^{M}(x) - \mathbf{z}_\ell^{M}(y)\right\|_2, \tag{3}$$

with $\gamma = 0.1$. We use AdamW with learning rate 10^{-3}, weight decay 10^{-4}, batch size 32 comparisons, no learning-rate scheduler, and gradient clipping at norm 1.0. A separate readout is trained at each layer for a maximum of 1000 epochs with early stopping patience set to 3 epochs using layer-specific validation accuracy, which is defined as the fraction of comparisons for which the transformed negative pair is at least as distant as the corresponding positive pair.



(a) CounterFact: mean pooling, nonlinear readout

(b) CounterFact: mean pooling, linear readout

(c) CounterFact: last token, nonlinear readout

(d) CounterFact: last token, linear readout

(e) SCPP: mean pooling, nonlinear readout

(f) SCPP: mean pooling, linear readout

(g) SCPP: last token, nonlinear readout

(h) SCPP: last token, linear readout

Figure 9: Layer-wise lexical-influence stress-test performance with linear and nonlinear semantic readouts. Panels (a–d) report CounterFact results and panels (e–h) report SCPP results. Mean-pooled representations are shown in (a–b, e–f), while last-token representations are shown in (c–d, g–h).

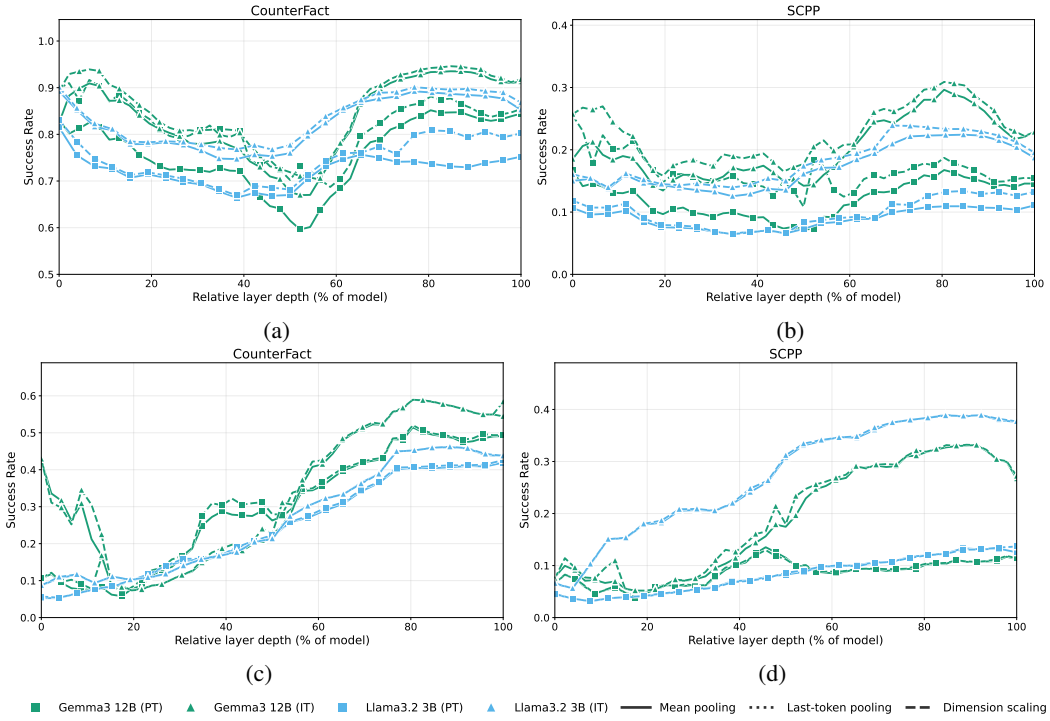


Figure 10: Layer-wise lexical-influence stress-test performance after applying the learned dimension scaling transformation to mean-pooled and last-token representations. Panels (a) and (b) report success rates for mean-pooled representations on CounterFact and SCPP, respectively, while panels (c) and (d) report the corresponding results for last-token representations.

Results. Figure 9 shows the layerwise behavior of the linear and GELU readouts for both mean pooled and last token representations. For mean pooled representations, the untransformed baseline obtains 78.56% on CounterFact and 14.63% on SugarCrepe++. The linear readout increases these scores to 82.38% and 33.23%, respectively, while the GELU readout reaches 80.08% and 32.84%. For last token representations, the baseline obtains 28.24% on CounterFact and 15.76% on SugarCrepe++. The linear readout reaches 30.22% and 22.46%, while the GELU readout reaches 30.34% and 21.98%. These values are obtained by first averaging over layers within each model and then macro averaging across the four models.

The nonlinear readout provides no systematic advantage over the parameter-matched linear transformation. This conclusion is not specific to GELU: replacing GELU with ReLU, LeakyReLU ($\alpha = 0.2$), or RMS-normalized Tanh leads to the same qualitative result, with no activation producing consistent improvements across models, layers, pooling strategies, and evaluation datasets. One plausible explanation is that the linear transformation can reweight and rotate the directional geometry already present in the frozen representations, whereas a coordinate-wise nonlinearity additionally gates or compresses signed components of that geometry. This additional nonlinear warping can improve the fit to the training relations without improving transfer to CounterFact and SugarCrepe++.

We do not interpret this result as evidence that the underlying semantic structure is intrinsically linear. Under this training protocol, increasing functional flexibility beyond a full-rank linear transformation does not yield a corresponding improvement in held-out semantic discrimination. Nevertheless, both linear and nonlinear readouts improve absolute performance while largely preserving the broad depth-dependent structure visible in Figure 9.

Per Dimension Scaling. We next ask whether semantic discrimination can be improved by changing the relative contribution of individual hidden dimensions without mixing information across them. For each layer ℓ and representation type $r \in \{\text{mean}, \text{last}\}$, we learn a scaling vector $\mathbf{s}_\ell^r \in \mathbb{R}^d$ and

compute

$$\mathbf{z}_\ell^r(x) = \mathrm{norm}_2\left(\mathbf{s}_\ell^r \odot \mathbf{h}_\ell^r(x)\right), \tag{4}$$

where $\odot$ denotes elementwise multiplication and $\mathbf{h}_\ell^r(x)$ denotes the corresponding mean pooled or last token representation. Each scaling vector is initialized to ones, exactly recovering the normalized original representation at initialization. The transformation therefore changes the contribution of existing coordinates while preserving the model's original coordinate system. Training is performed using the same semantic supervision and optimization procedure used for the linear and nonlinear readouts above.

Results. Figure 10 shows that per dimension scaling produces modest and model dependent changes while largely preserving the original trajectories across depth. The effect is most visible for mean pooled representations on SugarCrepe++. For Gemma-3-12B-IT, for example, success increases from about 0.19 to 0.26 in the earliest layers and from a later peak of about 0.29 to 0.31. The scaled and untransformed curves nevertheless follow the same broader trajectory across the network. On CounterFact, scaling similarly changes absolute performance at individual depths without altering the overall decline and subsequent recovery of the mean pooled representations. For last token representations, the scaled trajectories remain close to the untransformed trajectories on both datasets across most of the network. Overall, coordinate wise reweighting can improve semantic discrimination at some depths, but its effect is limited compared with the full dimensional linear and nonlinear readouts.

Learned Semantic Token Pooling. Mean pooling assigns the same weight to every content token, even though the semantic information relevant to the stress tests may be distributed unevenly across token positions. We therefore replace uniform pooling with a lightweight learned query that determines how strongly each token contributes to the sentence representation, while keeping the language model frozen.

For each layer ℓ, we learn a query $\mathbf{q}_\ell \in \mathbb{R}^d$ and compute

$$\alpha_{\ell,t} = \mathrm{softmax}_{t\in C}\left(\frac{q_\ell^\top \mathrm{LN}(H_\ell(x)_t)}{\sqrt{d}}\right), \qquad z_\ell = \mathrm{norm}_2\left(\sum_{t\in C} \alpha_{\ell,t} H_\ell(x)_t\right). \tag{5}$$

where $\mathcal{C}$ denotes the set of input content tokens. Layer normalization is used only to compute the attention scores, reducing the influence of token specific differences in hidden state magnitude. The weighted sum is formed from the original hidden states. We learn a separate query at each layer, allowing the readout to adapt as the information carried by token states changes across depth.

Each query is initialized to zero. At initialization, all attention logits are therefore equal and

$$\alpha_{\ell,t}^{(0)} = \frac{1}{|\mathcal{C}|}. \tag{6}$$

The readout thus begins exactly as uniform pooling over the content tokens and learns only how their relative contributions should change. This construction directly tests whether semantic information is already present in the token states but is obscured when those states are aggregated uniformly.

We train one query independently at each layer using the same semantic supervision, lexical overlap matching, and margin ranking objective as the readouts above. Only the query parameters are optimized using the same training formulation as linear and non-linear projections.

Results. Learned pooling yields the strongest overall improvements among the readout controls. Figure 2 shows that the gains occur across the depth of the network on both CounterFact and SugarCrepe++. On CounterFact, for example, the minimum success rate for Gemma-3-12B-PT increases from roughly 0.60 under mean pooling to about 0.74 with learned pooling, while success around the later layers increases from approximately 0.85 to 0.95. The effect is more pronounced on SugarCrepe++. For Llama-3.2-3B-IT, success increases from roughly 0.15 to 0.40 around the middle of the network and from approximately 0.22 to 0.42 around 80% depth. Thus, the improvement is not confined to the weakest region of the original curve but extends across depth.

Learned pooling improves semantic discrimination across both evaluations, showing that nonuniform token weighting provides a stronger readout than mean pooling. However, the same depth dependent

structure remains, with intermediate representations performing worse than later ones. This suggests that the intermediate regime cannot be explained by uniform pooling alone.

D Bidirectional Full-Attention Models

To test whether the intermediate regime is specific to autoregressive models, we repeat the lexical-influence evaluation from § 3 on two bidirectional full-attention encoders, DeBERTa-v3-large and RoBERTa-large. We evaluate both mean pooled representations and the models' `[CLS]` representations.

Figure 11 shows that full attention does not eliminate the depth-dependent degradation. Average-pooled representations still weaken substantially at intermediate depth, although the exact trajectory differs across models and datasets rather than forming the same valley observed in autoregressive models. In contrast, `[CLS]` representations exhibit highly irregular layer-wise behavior with no consistent depth profile. These results show that the intermediate degradation is not specific to causal attention, while its precise form remains dependent on architecture and representation readout.

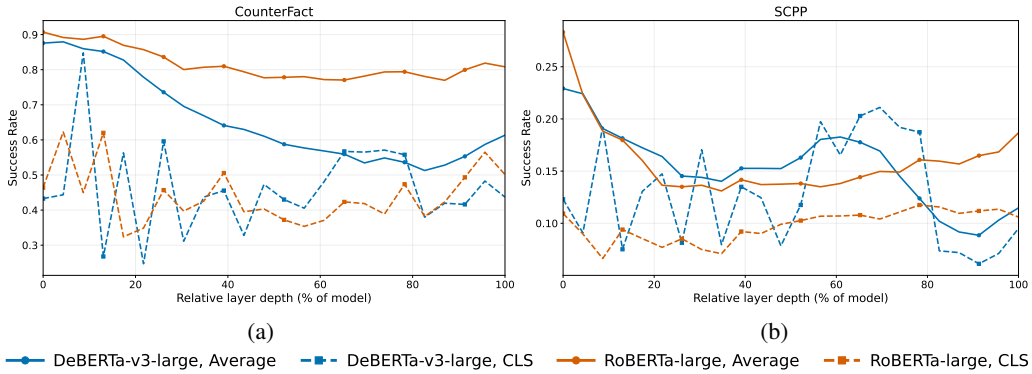


Figure 11: Layer-wise lexical-influence stress-test performance for bidirectional full-attention encoders. Panels (a) and (b) report CounterFact and SCPP, respectively, comparing mean-pooled and `[CLS]` representations for DeBERTa-v3-large and RoBERTa-large.

E Attention and Feedforward Contributions

Our main analysis measures lexical influence only at the output of each Transformer block. We next ask whether the changes across depth can be localized more precisely to the attention or feedforward update. For both CounterFact and SCPP, we evaluate the same triplet criterion before attention, after the attention residual update, and after the feedforward residual update in every block. We use mean pooling for the pretrained and instruction tuned models and the last token representation for the embedding models. We then measure the change in success induced by each component,

$$\Delta_\ell^{\text{attn}} = S_\ell^{\text{attn}} - S_\ell^{\text{in}}, \qquad \Delta_\ell^{\text{ffn}} = S_\ell^{\text{out}} - S_\ell^{\text{attn}}.$$

Positive values indicate that the update improves separation of paraphrases from lexically similar distractors, while negative values indicate increased lexical interference.

Figure 12 shows that the source of these changes is not consistent across architectures. In Llama, attention accounts for many of the larger changes across depth. In Gemma, the decline through intermediate layers is more strongly associated with feedforward updates, which also contribute to the later recovery. Qwen3 and KaLM show a different pattern, with most of the larger changes concentrated in attention and comparatively small feedforward effects. Because the embedding models use a different sentence readout, we interpret these cross-model differences qualitatively rather than comparing their magnitudes directly. Across both CounterFact and SCPP, no single component consistently tracks the depth dependent degradation.

Gemma 3 provides an additional architectural control because it interleaves sliding window and global attention blocks, a pattern also inherited by the Gemma based KaLM model. We find no

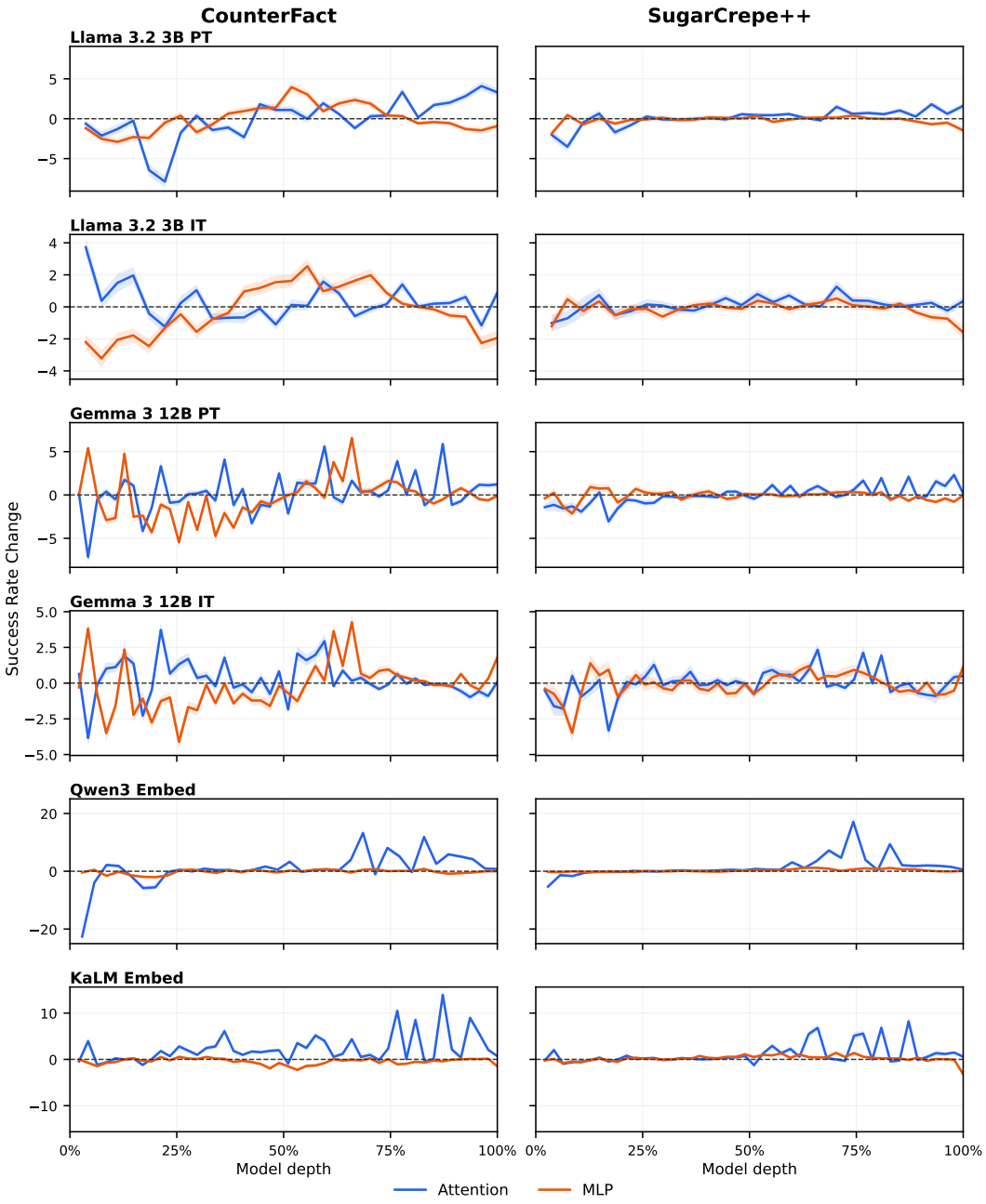


Figure 12: Layer-wise changes in lexical-influence stress-test success induced by attention and feedforward updates. Positive values indicate improved separation of paraphrases from lexical distractors, while negative values indicate increased lexical interference. Results are shown for combined CounterFact and SCPP evaluation.

systematic change in the sign or magnitude of the attention contribution at global blocks relative to sliding window blocks. CounterFact and SCPP sequences are shorter than Gemma 3's 1024 token sliding window, however, so both attention types can access the full preceding sequence in our setting. The comparison therefore does not isolate receptive field size. It instead shows that, for the contexts studied here, the changes in lexical influence do not systematically align with Gemma's alternation between sliding window and global attention.

F PROMPTING VARIATION

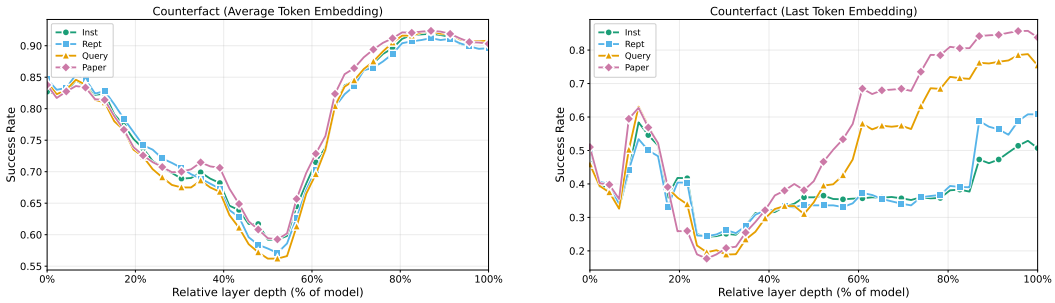


Figure 13: Layer-wise lexical-influence stress-test performance under prompt variation for Gemma 3 12B-IT on CounterFact. Results compare the evaluated instruction templates for mean-pooled and last-token representations.

To gauge the impact of baseline prompt variation on the instruction-tuned model, we evaluate the Gemma-3-12b-it on the CounterFact model under multiple instruction prompts. The list of prompts evaluated includes the following: ("query: " (query), "Represent this sentence for searching relevant passages: " (Rept), "Instruct: Represent this text for retrieval. Query: " (Inst), (Paper) contains no instruction other than the base structure for the instruction models.

The results are shown in Figure 13. The results indicate that the overall pattern is preserved across prompt variations. To measure the correlation between the resulting curves, we compute the pairwise average distance correlation (dCor). For mean-pooled embeddings, both the pattern and performance are nearly unchanged, with dCor = 0.993. For last-token embeddings, variation is larger in the second half of the network, where the model moves closer to its generation objective, but the overall trend remains the same; the dCor is 0.78 in this case.

G LEXICAL AND SEMANTIC PROBES

G.1 LEXICAL PROBES

Data. We train the probe on raw WikiText (WikiText-103-raw). We tokenize the corpus with the same tokenizer as the base model, discard empty/whitespace-only lines, and concatenate text segments with an end-of-sequence boundary token to avoid creating spurious cross-document contexts. From this token stream, we construct fixed-length sequences of $L = 50$ tokens and use the token at each position as the prediction target for its corresponding hidden state. Thus, the probe reconstructs the token occupying the same input position rather than predicting the next token.

To promote diversity in the training signal, we sample sequences from a shuffled stream of documents (and, in the streaming setting, shuffle with a buffer) before chunking into length-L windows. We additionally cap the number of extracted sequences per split at approximately 3,000 for compute efficiency while maintaining broad topical coverage. Training uses the WikiText-103-raw-v1 `train` split, while evaluation uses the separate `test` split, processed identically.

Training and Hyperparameters. We learn a separate affine map and bias for each selected layer while keeping the backbone and vocabulary embeddings frozen. For layer ℓ, the probe applies the affine transformation:

$$z_{\ell,t} = W_\ell H_\ell(x)_t + b_\ell.$$

Token probabilities are obtained by scoring the transformed representation against the frozen input embedding matrix using unnormalized dot products.

$$p_\ell(v \mid x, t) = \frac{\exp(\langle z_{\ell,t}, e_v \rangle)}{\sum_{u=1}^{V} \exp(\langle z_{\ell,t}, e_u \rangle)}.$$

We train the probe by minimizing the token-level cross-entropy loss $\mathcal{L}_{\mathrm{CE}}^{(\ell)}$ between the probe's scores and the ground-truth token IDs at the same positions, ignoring padded positions. For causal models,

Table 4: MTEB tasks used for the layer-wise semantic-fidelity analysis.

Category	Tasks
Classification	Banking77Classification, ToxicConversationsClassification, MTOPDomainClassification
Clustering	ArXivHierarchicalClusteringS2S, MedrxivClusteringS2S.v2, StackExchangeClustering.v2
Pair classification	SprintDuplicateQuestions, TwitterSemEval2015
Reranking	AskUbuntuDupQuestions, MindSmallReranking
Retrieval	ArguAna, HotpotQAHardNegatives, SCIDOCS, TRECCOVID
Semantic textual similarity	BIOSSES, SICK-R, STSBenchmark, STS22.v2

we use the EOS token as the pad token. We also apply a weak ℓ_2 penalty that encourages W_ℓ to remain close to the identity. This regularizer primarily stabilizes training and provides a mild preference for identity-like mappings, discouraging large remappings that can confound probe-based interpretation (Hewitt & Liang, 2019). Consequently, probe accuracy reflects the linear accessibility of token identity in a space aligned with the model's input embedding geometry, which we use as an operational reference for surface form.

The resulting objective at layer ℓ is

$$\mathcal{L}_\ell = \mathcal{L}_{\mathrm{CE},\ell} + \lambda \|W_\ell - I\|_F^2,$$

where $\lambda > 0$ is the identity-regularization coefficient. We optimize the probe parameters with AdamW (Loshchilov & Hutter, 2017), using the default learning rate 10^{-3}, weight decay 10^{-4}, identity regularization $\lambda = 10^{-6}$, and global gradient-norm clipping at 1.0. Each batch contains four sequences of 50 tokens, corresponding to approximately 200 token positions per update. We train for at most five epochs without gradient accumulation or a learning-rate scheduler.

H Semantic Fidelity Probe

Following the findings from the stress test conducted in the main paper, we use the mean pooled representations for pretrained and instruction-tuned models and the last-token representation for embedding models.

We evaluate on a suite of English tasks from the Massive Text Embedding Benchmark (MTEB) and cache each task locally to ensure deterministic reruns. We perform subsampling for the few *massive* cases to keep runtime bounded. For `MedRxivClusteringS2S.v2` (37,500 test instances), we uniformly sample 10,000 test points without replacement using a fixed seed and cache the resulting subset. For extremely large reranking data such as `MindSmallReranking`, we cap the cached data to 10,000 queries and 200,000 corpus documents; we then filter `relevant_docs` and `top_ranked` to remove any entries that reference dropped query or document IDs, ensuring internal consistency. The datasets used and their categorization are provided in Table 4.

H.1 MTEB Results

Figure 15 reports results for STS, pairwise classification, clustering, and reranking. STS and pairwise classification closely follow the depth-dependent behavior discussed in the main text. Clustering and reranking show more task-specific trajectories. Clustering varies more across layers, while reranking remains comparatively stable for the pretrained and instruction-tuned models and improves more clearly at later depths for the embedding models.

H.2

Lexical influence in semantic-fidelity tasks. The MTEB evaluations in §4 measure performance on semantic tasks, but these tasks are not lexically controlled. Table 5 shows that gold-positive pairs

Task	Negative pairs (all tokens)	Negative pairs (stop words removed)	Positive pairs (all tokens)	Positive pairs (stop words removed)
MedRxiv clustering	5.32%	1.11%	9.27%	5.28%
ArXiv clustering (L0)	3.06%	0.27%	3.65%	0.87%
ArXiv clustering (L1)	3.35%	0.28%	4.47%	1.23%
StackExchange clustering	2.30%	0.12%	3.67%	1.05%
ArguAna retrieval	9.64%	2.19%	16.95%	10.14%
HotpotQA retrieval	4.85%	0.45%	14.25%	11.97%
SCIDOCS retrieval	1.94%	0.47%	3.66%	2.56%
TRECCOVID retrieval	3.55%	1.93%	5.93%	4.84%

Table 5: Lexical overlap in gold-positive and gold-negative samples from the clustering and retrieval tasks.

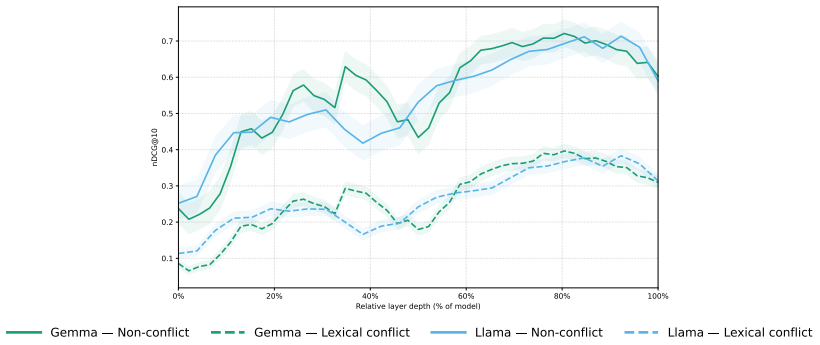


Figure 14: Layer-wise ArguAna retrieval performance for lexical-conflict and non-conflict examples. Lexical conflict lowers absolute performance while largely preserving the depth-dependent trajectory.

generally have greater lexical overlap than gold-negative pairs across the clustering and retrieval datasets, including after stop words are removed. Lexical overlap can therefore provide a useful signal for task performance rather than serving only as a confound in our stress tests.

We examine whether this signal explains the layer-wise trends using ArguAna, separating examples with lexical cues that agree with the relevance relation from those where they conflict. The data is partitioned based on whether lexical overlap agrees with or conflicts with the gold relevance relation. For each query q with gold document d^+, we select the non-gold corpus document $d_\star^-$ with the highest content-word Jaccard similarity to q. We define a lexical-conflict example as one satisfying

$$J(q, d_\star^-) \geq J(q, d^+),$$

so that the strongest non-relevant lexical distractor has at least as much content-word overlap with the query as the gold document. Equality is included in the lexical-conflict subset. Figure 14 shows that lexical-conflict examples are consistently harder for both Gemma and Llama, confirming that lexical overlap affects absolute retrieval performance. Despite this separation, the two partitions follow highly similar trajectories across depth, with distance correlations of 0.9890 for Gemma and 0.9858 for Llama. Both exhibit the same broad early improvement, intermediate weakening, and later recovery. Thus, on ArguAna, lexical overlap affects absolute task difficulty but does not account for the observed depth-dependent trajectory.

I Information Geometry

Input/Prompt Entropy. Extracted from (Skean et al., 2025). For an input (prompt) p with T_p tokens, let $Z_\ell(p) \in \mathbb{R}^{T_p \times d}$ be the matrix of layer-ℓ token embeddings (rows correspond to token positions). Define the Gram matrix

$$K_\ell(p) = Z_\ell(p) Z_\ell(p)^\top, \qquad \widetilde{K}_\ell(p) = \frac{K_\ell(p)}{\mathrm{tr}(K_\ell(p))}.$$

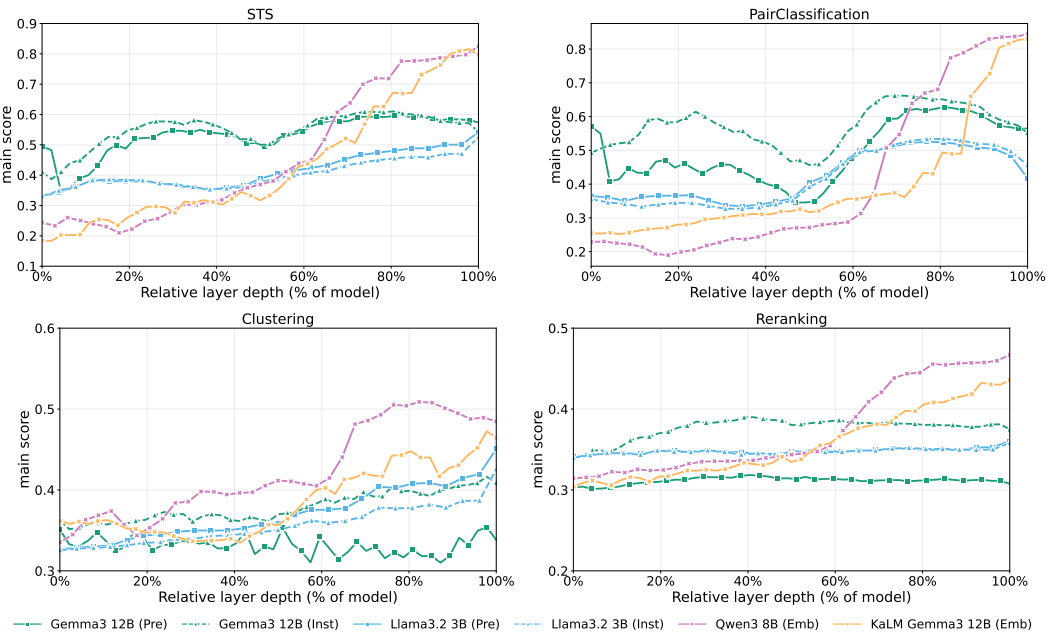


Figure 15: Layer-wise MTEB performance on additional semantic-fidelity tasks. Panels (a), (b), (c) and (d) report STS, pairwise classification, clustering, and reranking, respectively, averaged across datasets within each category.

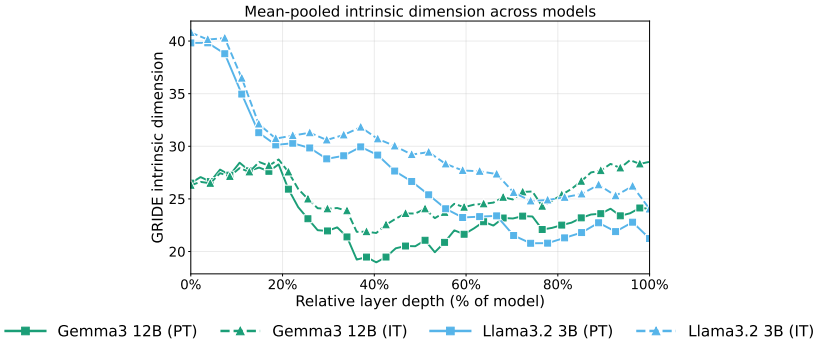


Figure 16: Layer-wise GRIDE intrinsic dimension for mean-pooled representations. Intrinsic dimension generally declines across early-to-intermediate depths, with later recovery varying across models.

Let $\{\lambda_i\}_{i=1}^{T_p}$ be the eigenvalues of $\widetilde{K}_\ell(p)$, so that $\sum_i \lambda_i = 1$. The input (prompt) entropy is the matrix-based Rényi entropy

$$H_{\text{in}}^{(\ell)}(p) = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{T_p} \lambda_i^\alpha \right), \qquad \alpha > 0,\ \alpha \neq 1.$$

For all reported results, we use the $\alpha \to 1$ limit, corresponding to the von Neumann entropy,

$$H_{\text{in}}^{(\ell)}(p) = -\sum_{i=1}^{T_p} \lambda_i \log \lambda_i.$$

Intrinsic Dimension. Following prior work on the geometry of Transformer representations (Cheng et al., 2025), we estimate layer-wise intrinsic dimension using the Generalized Ratios Intrinsic

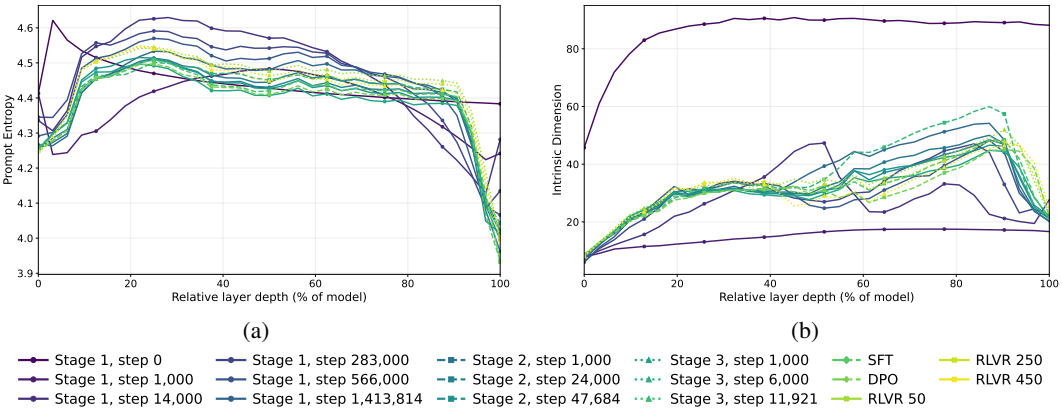


Figure 17: Layer-wise representation geometry across OLMo 3 7B training checkpoints. (a) Prompt entropy. (b) Intrinsic dimension.

Dimension Estimator (GRIDE). Given layer-ℓ representations $\{h_i^{(\ell)}\}_{i=1}^N$, let $r_{i,j}^{(\ell)}$ denote the Euclidean distance from $h_i^{(\ell)}$ to its j-th nearest neighbor. GRIDE uses

$$\mu_i^{(\ell)} = \frac{r_{i,n_2}^{(\ell)}}{r_{i,n_1}^{(\ell)}},$$

with likelihood

$$p(\mu \mid d_{\text{int}}) = \frac{d_{\text{int}} \left(\mu^{d_{\text{int}}} - 1\right)^{n_2 - n_1 - 1}}{\mu^{(n_2-1)d_{\text{int}}+1} B(n_2 - n_1, n_1)}, \qquad \mu > 1,$$

where d_{int} is the intrinsic dimension. We estimate d_{int} independently at each layer by maximum likelihood.

We construct $N = 5{,}000$ unique, non-overlapping 128-token sequences from WikiText-2 using each model's tokenizer. At each Transformer block, we extract both the final non-padding token representation and the mean-pooled representation, and estimate intrinsic dimension separately for each readout. Distances are computed in the original hidden-state space using Euclidean distance with FP64 accumulation, without centering, normalization, PCA, or other dimensionality reduction. We evaluate GRIDE across multiple neighborhood sizes with $n_2 = 2n_1$ and report $n_1 = 8$, where the intrinsic-dimension estimates had stabilized.

Uncertainty is estimated from 20 repeated subsamples containing 80% of the representations. The resulting intrinsic-dimension estimates characterize the local degrees of freedom occupied by the representations, rather than the ambient hidden-state dimensionality.

Figure 16 reports the corresponding results for mean-pooled representations. Compared with the final-token analysis, intrinsic dimension shows a broader decline across depth, with the strongest contraction occurring over early-to-intermediate layers for several models. Recovery at later depths is model dependent, with some trajectories increasing again while others remain comparatively low.

I.1 OLMo 3

We track prompt entropy and intrinsic dimension across OLMo 3 7B training checkpoints to examine how the geometry associated with the intermediate regime develops during pretraining. Figure 17 shows that prompt entropy does not exhibit the same clear depth-dependent correspondence with the valley observed for Llama and Gemma. Intrinsic dimension, however, shows the same relationship identified in the main analysis. Once the valley emerges, its weakest intermediate region coincides with a plateau or partial contraction in intrinsic dimension, while the later recovery in semantic discrimination accompanies renewed growth in intrinsic dimension.

This relationship develops early in Stage 1 and remains visible through later Stage 1 and Stages 2–3. Thus, OLMo provides training-time evidence that the lexical-influence valley is associated with an

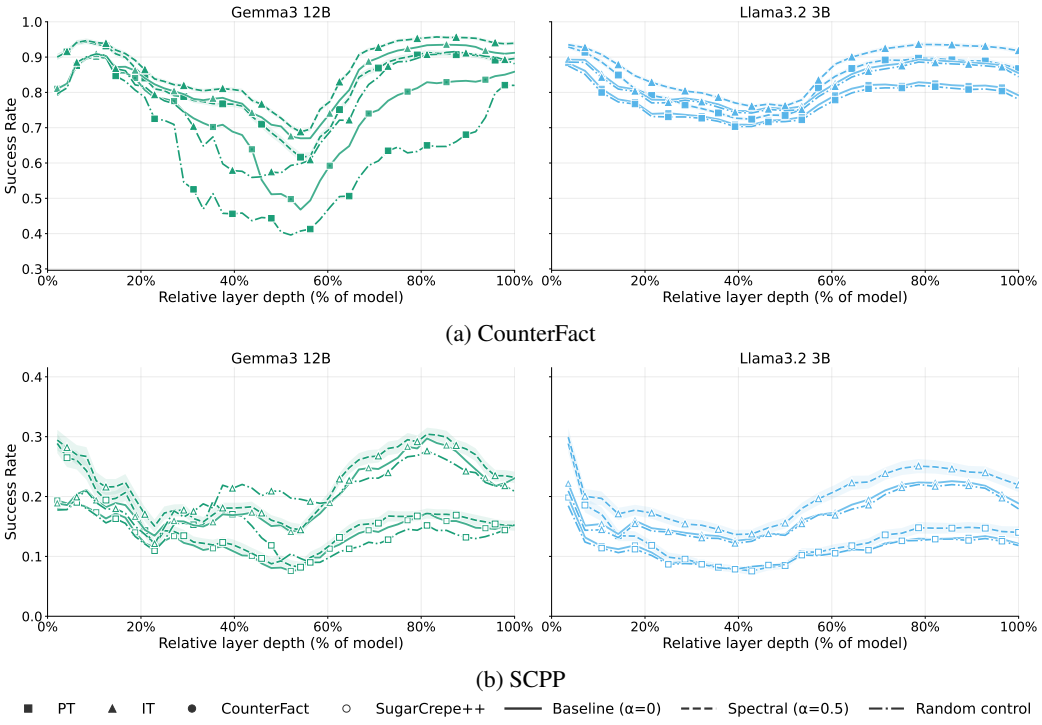


Figure 18: Layer-wise stress-test success before and after spectral intervention. Solid lines show the original representations, dashed lines show spectral flattening at $\alpha = 0.5$, and dash-dotted lines show magnitude-matched random controls.

intermediate interruption in the growth of local representational degrees of freedom, even when the prompt-entropy trajectory differs across architectures.

I.2 Spectral Intervention

We test whether directly modifying representation geometry affects lexical stress-test performance. At each layer, we fit PCA to 5,000 mean-pooled WikiText-2 representations, separately for each model and checkpoint. No CounterFact or SCPP examples are used to fit the transformation.

For a representation h, with PCA mean μ, basis U, and component variances λ_i, we compute

$$z = U^\top (h - \mu), \qquad z'_i = g_i(\alpha) z_i,$$

where

$$g_i(\alpha) = \mathrm{clip}\left(\frac{(\lambda_i + \epsilon)^{-\alpha/2}}{\mathrm{median}_j (\lambda_j + \epsilon)^{-\alpha/2}}, 0.1, 10 \right),$$

and $\epsilon = 10^{-3} \,\mathrm{median}_{\lambda_i > 0} \lambda_i$. We reconstruct $h' = \mu + U z'$ and ℓ_2-normalize it before applying the same stress-test criterion. We evaluate $\alpha \in \{-0.5, -0.25, 0, 0.25, 0.5, 0.75\}$, where $\alpha = 0$ recovers the original representation, positive values flatten the spectrum, and negative values concentrate it.

Spectral flattening consistently improves stress-test performance. At $\alpha = 0.5$, the layer-averaged gain is approximately 2–8 percentage points across the eight model–dataset combinations, with the largest improvements on CounterFact. Negative α generally produces the opposite effect. To test whether these gains arise simply from perturbing the representations, we compare against magnitude-matched random controls obtained by permuting and sign-flipping the spectral displacement in PCA coordinates. Across 10 random seeds, spectral flattening outperforms the random control in all comparisons shown in Figure 18. The layer-wise results further show that the gains extend across depth rather than being concentrated in the intermediate regime, while the broader depth profile remains visible after intervention.

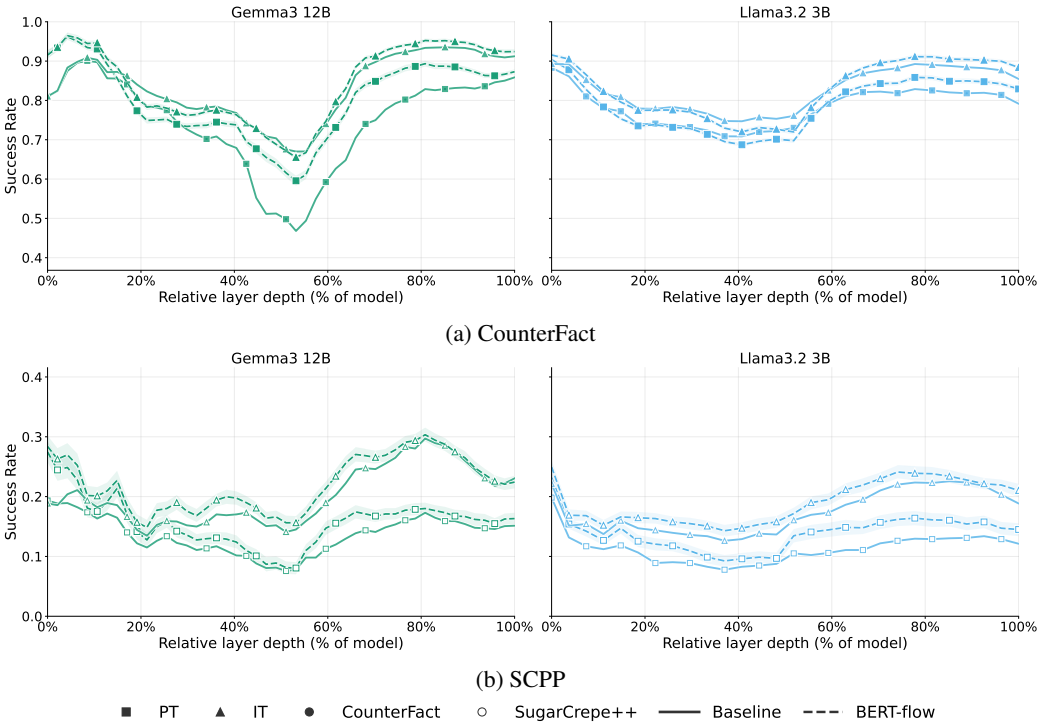


Figure 19: Layer-wise stress-test success before and after BERT-flow. Solid lines show the original representations and dashed lines show the BERT-flow transformed representations.

We additionally compare with BERT-flow (Li et al., 2020), an independently learned transformation that makes sentence representations more isotropic. BERT-flow is fit separately at each layer on 15,455 unlabeled STS-B sentences and evaluated with the same stress test (Figure 19). It also improves performance, with the largest changes appearing on CounterFact; for example, Gemma-PT improves by roughly 10 percentage points at its weakest intermediate layers, while changes on SCPP are generally only a few points. Overall, BERT-flow and spectral flattening produce improvements of similar scale, supporting the broader finding that reducing spectral concentration can improve robustness to lexical distractors.

J SUMMARIZATION

J.1 DETAILS OF METRICS

BERTScore. BERTScore leverages contextual embeddings from a pretrained Transformer (e.g BERT) to compute token level similarity between two texts using cosine similarity and greedy matching. In our experiment, we compare both the gold summary and the lexical distractor summary against the source document.

Given a source document $x = \langle x_1, \ldots, x_{T_x} \rangle$ and a candidate summary $\hat{x} = \langle \hat{x}_1, \ldots, \hat{x}_{T_{\hat{x}}} \rangle$, let $\mathbf{v}_i \in \mathbb{R}^d$ and $\hat{\mathbf{v}}_j \in \mathbb{R}^d$ denote their contextual token embeddings. Define the cosine similarity

$$s_{ij} = \frac{\mathbf{v}_i^\top \hat{\mathbf{v}}_j}{\|\mathbf{v}_i\|_2 \, \|\hat{\mathbf{v}}_j\|_2}.$$

The metrics recall (R_{BERT}), precision (P_{BERT}), and F1 (F_{BERT}) are defined as:

$$R_{\text{BERT}} = \frac{1}{T_x} \sum_{i=1}^{T_x} \max_{1 \le j \le T_{\hat{x}}} s_{ij}, \tag{7}$$

$$P_{\text{BERT}} = \frac{1}{T_{\hat{x}}} \sum_{j=1}^{T_{\hat{x}}} \max_{1 \le i \le T_x} s_{ij}, \tag{8}$$

$$F_{\text{BERT}} = 2 \cdot \frac{P_{\text{BERT}}\, R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}. \tag{9}$$

We report the F1 score (F_{BERT}) for our experiments while following the official BERTScore configuration of using the DeBERTa layer (40), which is tuned for WMT16 human-correlation.

BARTScore. BARTScore evaluates generated text using the conditional log-likelihood of a candidate under a sequence-to-sequence model. In our experiment, the source document is used as the conditioning sequence, and either the gold summary or the lexical distractor summary is used as the target sequence.

The score is defined as the average log-likelihood of the target sequence y given the source sequence x:

$$\text{BARTScore} = \frac{1}{m} \sum_{t=1}^{m} \log P(y_t | y_{<t}, x, \theta) \tag{10}$$

where θ represents the parameters of the pre-trained BART model, and m is the length of the target sequence.

J.2 Data and Models

Data. We use the standard extreme summarization(Narayan et al., 2018) benchmark. We utilize 500 random samples from the dataset. To probe lexical sensitivity, we augment each example with a highly lexicalized non-summary distractor generated by an LLM(gpt-5.2). The prompt used for the generation of the distractor is as follows:

```
prompt = f"""
    You are generating adversarial evaluation data to test lexical bias
    in summarization systems.

    Source Document:
    {document}

    Target Length: {target_length} characters (+/-5%)

    TASK:
    Generate a text that has HIGH lexical overlap with the source
    document but NO semantic overlap with it.

    CRITICAL DEFINITION:
    "NO semantic overlap" means that the text must not convey the topic,
    purpose, claims, events, or conclusions of the source in any form.
    A reader should be unable to infer what the source document is about
    from your text.

    CONSTRAINTS (hard requirements):

    1. Lexical Overlap:
    - Reuse distinctive words, phrases, and named entities from the
    source.
    - Do not introduce new terminology.
    - Try to use all words from the source.

    2. Semantic Orthogonality:
    - Do NOT describe, summarize, or restate any part of the source's
    meaning.
    - Do NOT preserve relations, causality, chronology, or argumentation.
    - The text must be about something else entirely, despite using the
    same words.
```

```
    3. Length:
    - Approximately {target_length} characters.

    VALIDATION CHECK (internal):
    If the output allows a reader to identify the subject or purpose of
    the source document, it fails.

    OUTPUT FORMAT:
    Return a JSON object with exactly one field:

    {{
    "lexical_non_summary": "<text>"
    }}
```

Listing 1: Prompt for Lexical Distractor Non-Summary Generation

Models. For BERTScore, DeBERTa-XLarge He et al. (2021) is used. For BARTScore, we utilize the *bart-large* Lewis et al. (2019) model fine-tuned on the CNN/Daily Mail summarization dataset.

K Model Editing

K.1 Method and dataset.

AlphaEdit (null-space constrained editing). AlphaEdit is a simple refinement that can be applied on top of a base locate–then–edit method (e.g., ROME/MEMIT). Suppose the base editor proposes an update Δ_{base} to a weight matrix W to enforce the desired edit. AlphaEdit additionally builds a *preservation set* of inputs whose behavior should remain unchanged, and extracts their corresponding key vectors $\{k_i\}_{i=1}^{m}$ at the edited layer. Let $K_0 \in \mathbb{R}^{d\times m}$ be the matrix with columns k_i. AlphaEdit then projects the base update onto the subspace orthogonal to $\mathrm{span}(K_0)$:

$$\Delta_\alpha \;=\; \Pi_\perp \, \Delta_{\text{base}}, \qquad \text{where } \Pi_\perp \text{ projects onto } \mathrm{span}(K_0)^\perp.$$

This guarantees

$$\Delta_\alpha K_0 = 0,$$

so the refined update leaves the model's mapping on the preservation keys unchanged, while retaining the base editor's effect on the target edit.

CounterFact dataset. CounterFact's distractor prompts are already partially lexically aligned with the anchor/edit prompt. To more directly quantify the effect of model editing under controlled lexical overlap, we augment CounterFact with two additional prompt sets: (*i*) *lexically similar* prompts, obtained by substituting the subject entity in the original anchor/edit template with distractor entities; and (*ii*) *lexically dissimilar* prompts, which preserve the distractor's semantics while substantially rewriting its surface form. We generate the lexically dissimilar prompts using an LLM (gpt-5.2-mini) with the following prompt:

```
prompt = f"""
You are a language expert tasked with rewriting prompts.

Input fields:
- original_locality_prompt: the prompt whose meaning must be preserved
    exactly
- edit_prompt: a reference prompt whose wording you must avoid copying

Task:
Produce THREE rewritten variants of original_locality_prompt.

STRICT meaning preservation (each variant):
1) Ask for the exact same fact as original_locality_prompt: same entity
    name(s) + same attribute/relation.
2) Copy entity name(s) EXACTLY as in original_locality_prompt (same
    spelling/casing). Do not replace entities with descriptions.
```

```
3) Do NOT add/remove qualifiers (time, certainty, official/current, "
    capital", "city of", etc.). No hints, no assumptions.

Cloze formatting (each variant):
4) Each prompt must be a PREFIX such that the correct answer should be
    the next text generated immediately after the final space.
5) End with exactly ONE trailing ASCII space and NO trailing punctuation
    (no ., ?, !, :, ;, ,).
6) Do NOT end with the entity name(s).
7) No interrogatives: do NOT use WH-words (what/which/where/who/when/how)
    . Use declarative stems only.
8) Validity check: appending the correct answer immediately after the
    final space must yield a complete grammatical result without adding
    any extra words.

Anti-copy / maximize lexical difference from edit_prompt (each variant):
9) Hard ban: Do NOT reuse any contiguous 2+ word sequence found in
    edit_prompt (exception: entity tokens).
10) Content-word taboo: Avoid reusing ANY non-entity content word from
    edit_prompt (verbs/nouns/adjectives/adverbs).
    - Allowed function-word set (may repeat): {the, a, an, of, to, in, on
    , at, from, for, with, by, is, are, was, were, be, been, being, as}
    - If a relation word from original_locality_prompt also appears in
    edit_prompt, you should paraphrase it using a synonym NOT present in
    edit_prompt, while preserving meaning.
11) Target: minimize overlap of non-entity content words with edit_prompt
     to 0 whenever possible.

MANDATORY structural diversity (across the 3 variants):
Generate exactly one from each frame:

A) ENTITY-SUBJECT frame:
   "<ENTITY> <connector> "
B) POSSESSIVE / ownership frame:
   "<ENTITY>'s <attribute phrase> <connector>"  OR  "<attribute phrase>
    of <ENTITY> <connector>"
C) ATTRIBUTE-SUBJECT frame:
   "The <attribute phrase> for <ENTITY> <connector>"

Additional diversity constraints:
12) No shared opening 3 tokens across variants.
13) Use three different connector tails (the last 1 to 4 words before the
     final space must differ across variants).
14) Across the three outputs, do not reuse the same key content words (
    besides the entity and unavoidable attribute noun). Prefer different
    synonym sets.

Search-and-select requirement (important):
15) For EACH frame (A/B/C), internally draft at least 5 candidate
    rewrites, then choose the best one that:
    - satisfies all constraints,
    - has the fewest non-entity content words overlapping with
    edit_prompt,
    - and differs most from the other selected variants.

Forbidden:
- Do not include the literal strings "original_locality_prompt" or "
    edit_prompt".
- Output JSON only; no extra text.

Output JSON only (exactly these keys, no extras):
{"rewritten_prompts":["... ","... ","... "]}
"""
```

Listing 2: Data generation prompt for lexically dissimilar distractor/locality

We perform editing on 2,000 sequential edit samples using the Llama-3-8B model, which is directly supported by the AlphaEdit codebase and evaluated with the default base settings. Note that we utilize the default batching of 100 samples for a single edit for a total of 20 edit batches.

Lexical-overlap validation. We verify that the lexically similar and lexically dissimilar prompt sets produce the intended separation in surface overlap. Across 2000 comparisons, mean unigram Jaccard overlap with the edit prompt is 0.4085 ± 0.1535 for lexically similar (LS) and 0.1036 ± 0.0935 for lexically dissimilar (LD), with a mean difference of 0.3049 (95% CI $[0.2976, 0.3122]$). LS has higher overlap than LD in 96.56% of comparisons. The prompts were tested for generation continuity: whether the generation led to the same answers.

K.2 METRICS

Let M_0 denote the base model and M_1 the edited model. For each locality prompt $x \in \mathcal{L}$, let τ_x denote the answer position and let

$$p_x^M(v) = P_M(v \mid x, \tau_x), \qquad v \in \mathcal{V},$$

be the model's next-token distribution at that position.

Merged top-K support. For $K \in \{20, 50, 100\}$, we define

$$S_K(x) = \mathrm{TopK}\big(p_x^{M_0}\big) \cup \mathrm{TopK}\big(p_x^{M_1}\big)\,.$$

For $M \in \{M_0, M_1\}$, we renormalize the distribution on this support as

$$\widetilde{p}_{x,K}^{M}(v) = \frac{p_x^M(v)}{\sum_{u \in S_K(x)} p_x^M(u)}, \qquad v \in S_K(x).$$

All $@K$ distributional metrics are computed on these restricted distributions for the full locality set $\mathcal{L}$.

Jensen–Shannon divergence. For each x, let

$$m_{x,K} = \frac{1}{2}\left(\widetilde{p}_{x,K}^{M_0} + \widetilde{p}_{x,K}^{M_1}\right).$$

We compute

$$\mathrm{JSD}_K(x) = \frac{1}{2} D_{\mathrm{KL}}\left(\widetilde{p}_{x,K}^{M_0} \,\|\, m_{x,K}\right) + \frac{1}{2} D_{\mathrm{KL}}\left(\widetilde{p}_{x,K}^{M_1} \,\|\, m_{x,K}\right),$$

where

$$D_{\mathrm{KL}}(p\|q) = \sum_{v \in S_K(x)} p(v) \log \frac{p(v)}{q(v)}.$$

Total variation distance. We compute

$$\mathrm{TV}_K(x) = \frac{1}{2} \sum_{v \in S_K(x)} \left|\widetilde{p}_{x,K}^{M_0}(v) - \widetilde{p}_{x,K}^{M_1}(v)\right|.$$

Kendall's τ. Let $r_{x,K}^M(v)$ denote the rank of token $v \in S_K(x)$ under p_x^M. For token pairs $\{u, v\} \subset S_K(x)$, we compare the signs of

$$r_{x,K}^{M_0}(u) - r_{x,K}^{M_0}(v) \qquad \text{and} \qquad r_{x,K}^{M_1}(u) - r_{x,K}^{M_1}(v).$$

Let C_x and D_x denote the numbers of concordant and discordant pairs, respectively. We define

$$\tau_K(x) = \frac{C_x - D_x}{C_x + D_x},$$

with $\tau_K(x) = 0$ when $C_x + D_x = 0$.

Table 6: Distribution-shift metrics on *lexically similar and dissimilar* prompts (Top-K-100).

Metric	**LS**	**LD**
JSD	0.294	0.246
TV	0.568	0.502
Kendall τ	0.001	0.039
Edited-Target Presence	0.142	0.072
Top-1 Flip Rate *(Top-K-independent)*	0.566	0.493

Top-1 Flip Rate. A flip occurs when the highest-probability token changes after editing:

$$F(x) = \mathbf{1}\left[\arg\max_{v\in\mathcal{V}} p_x^{M_0}(v) \neq \arg\max_{v\in\mathcal{V}} p_x^{M_1}(v)\right].$$

The Top-1 Flip Rate is the fraction of prompts in $\mathcal{L}$ for which $F(x) = 1$.

Edited-Target Presence@K. For an edit e with edited target y_e^*, let $T_e \subseteq \mathcal{V}$ denote the token IDs obtained by tokenizing y_e^*. For a locality prompt x associated with edit e, define

$$\mathrm{ETP}_K(x) = \mathbf{1}\left[\mathrm{TopK}\big(p_x^{M_1}\big) \cap T_e \neq \emptyset\right].$$

ETP@K is the fraction of locality prompts in $\mathcal{L}$ for which this condition holds.

We compute all reported metrics over the complete locality set $\mathcal{L}$ for the corresponding LS or LD condition.

L Compute Resources

All experiments are conducted on an NVIDIA H100 GPU with 96 GB of VRAM. Several experiments, including the semantic probing evaluations, require storing intermediate embeddings on disk to avoid the compute overhead of running large models on GPU. Due to storage requirements, the evaluations runs use upto 512 GB of storage and 128 GB of system RAM.