SimLens: Two-Token Continuation for Early Exit in Single-Token Decision Tasks

Ming Ma[1,2,*], Bowen Zheng[3,*], Zhongqiao Lin[1], Tianming Yang[1,†]
[1]Institute of Neuroscience,
State Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology,
Center for Excellence in Brain Science and Intelligence Technology,
Chinese Academy of Sciences, Shanghai, 200031, China
[2]School of Future Technology, University of Chinese Academy of Sciences,
Beijing, 100049, China
[3]School of Data Science, The Chinese University of Hong Kong, Shenzhen
{mam2022, zqlin, tyang}@ion.ac.cn
bowenzheng3@link.cuhk.edu.cn
[*]These authors contributed equally to this work. [†]Corresponding author

Abstract

Intermediate-layer representations in decoder-only large language models (LLMs) contain task-relevant information, yet direct linear readouts often fail to recover the model's eventual choice. This gap limits both interpretability and early-exit methods for candidate-scoring workloads. We study *single-token decision* settings in which a small candidate set is supplied by the task format, such as multiple-choice and boolean QA. We propose SimLens, a training-free readout that, at a chosen intermediate layer, keeps only the start token and the candidate answer token and forwards this reduced two-token state through the model's remaining upper layers. To identify the layer at which the full sequence can be safely truncated, we distill Linear SimLens as a lightweight entropy monitor and combine the two in SimExit; unlike SimLens, this monitor is trained. Across five decoder-only backbones from 3B to 8B parameters, SimLens produces earlier and more faithful candidate readout than direct linear lenses, while SimExit delivers a modest but stable speed–accuracy trade-off on the two models used end-to-end under fair-compute accounting. Ablations show that the two retained tokens play complementary roles as global condition and semantic anchor.

1 Introduction

Large language models (LLMs) form their predictions gradually across depth. If we could accurately read out a model's latent prediction from an intermediate layer, this would help us understand how semantics emerge inside the network and could also reduce inference cost. Motivated by this, prior work studies hidden-state decoding with lens-style methods (nostalgebraist, 2020; Dar et al., 2023; Geva et al., 2023; Hendel et al., 2023; Belrose et al., 2023; Ghandeharioun et al., 2024; Skean et al., 2024; Skean et al., 2025), and many early-exit methods attempt to stop computation at intermediate layers to reduce cost (Schuster et al., 2022; Pal et al., 2023; Din et al., 2024; Valade, 2024; Elhoushi et al., 2024; Fan et al., 2024; Zhang et al., 2024a; Kavehzadeh et al., 2024; Jazbec et al., 2024; Miao et al., 2024; Mofakhami et al., 2024; Fan et al., 2025; Zarch et al., 2025).

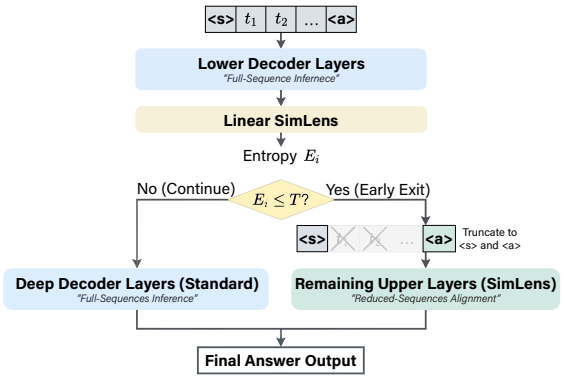

Figure 1: Overview of SimExit. At each monitored layer, Linear SimLens provides a low-cost entropy confidence score for exit decisions. Once the confidence threshold is met, full-sequence forwarding is truncated and SimLens continues with only two tokens—the start token and the candidate answer token—to decode the answer.

Existing approaches such as Logit Lens, Tuned Lens, and Future Lens read intermediate states with shared or learned linear projections. These methods are useful analysis tools, but decoding from earlier layers remains unreliable: even when answer-relevant information is already present, the decoded output can still deviate sharply from the

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 35725–35741
October 24-29, 2026 ©2026 Association for Computational Linguistics

model's final prediction. On ARC (Clark et al., 2018), for example, a linear probe recovers answer information much earlier than Logit Lens (nostalgebraist, 2020), suggesting that the bottleneck is not the absence of latent information, but the difficulty of reading it out faithfully. We provide the corresponding motivation visualizations in Appendix A.

We focus on a specific regime: *candidate-scoring* settings where the candidate label or answer token is provided by the task format. This includes multiple-choice QA, boolean verification, reranking, routing, moderation, and preference/reward scoring, and differs from unrestricted long-form generation. SimLens reads out intermediate candidate scores in this scoped setting; SimExit uses this readout for early exit. We do not claim SimLens as a general-purpose accelerator for long-form generation. For example, a RAG verifier may repeatedly decide whether a candidate answer is supported by retrieved evidence using a fixed `Yes`/`No` output; SimExit can stop full-sequence processing once that decision is confident and continue only the two retained tokens.

Within this scope we take a different route from stronger linear lenses. We ask whether a *minimal two-token continuation* can recover a much more faithful intermediate-layer prediction. We propose **SimLens**, a training-free readout that, at an intermediate layer l, retains only the start token and the candidate answer token—written $\langle s \rangle$ and $\langle a \rangle$ in equations—and forwards this two-token state through the remaining upper layers $T_{l+1}, \ldots, T_L$. The two tokens play complementary roles: the start token preserves global conditioning, while the candidate answer token serves as a semantic anchor for the candidate being scored.

This minimal continuation is already enough to outperform linear baselines at early layers and often moves much closer to the final model prediction. We evaluate five decoder-only 3–8B backbones—LLaMA-7B, Vicuna-7B, Llama-3.1-8B, Qwen2.5-7B-Instruct, and Qwen2.5-3B-Instruct—on ARC (Clark et al., 2018), BoolQ (Clark et al., 2019), and HeadQA (Vilares and Gómez-Rodríguez, 2019). In this setting the candidate answer token is given by the task format itself, so SimLens scores each candidate directly without needing a separate full-model decode to first discover the answer token.

More faithful intermediate readout naturally enables early exit. We therefore introduce **Linear SimLens**, a lightweight linear monitor distilled from SimLens for low-cost entropy estimation, and combine it with SimLens in **SimExit**, a hybrid early-exit framework (Figure 1). SimLens itself is training-free; Linear SimLens and SimExit use a small trained monitor. Across LLaMA-7B (Touvron et al., 2023) and Vicuna-7B (Chiang et al., 2023), SimLens improves iso-compute accuracy over matched-cost naive intermediate decoding in all six model-dataset settings, with an average gain of $+43.2$ pp after charging the extra two-token post-forward cost. SimExit yields an average $1.15\times$ speedup at the best-accuracy operating points and $1.40\times$ when allowing up to a 1 pp accuracy drop.

Contributions. We make three contributions: (1) we introduce SimLens, a training-free readout that uses a two-token continuation to elicit accurate candidate predictions from intermediate layers of LLMs; (2) we show that retaining the start token and the candidate answer token is sufficient to markedly improve early-layer readout, and we clarify their distinct roles as global condition and semantic anchor; and (3) we introduce Linear SimLens, a lightweight distilled monitor, and SimExit, a hybrid early-exit framework that combines low-cost entropy monitoring with the two-token SimLens readout.

2 Related Work

Hidden representation decoding. Intermediate layers in LLMs encode substantial task-relevant semantics (Hendel et al., 2023; Wang et al., 2023; Halawi et al., 2024; Din et al., 2024; Skean et al., 2024, 2025). Logit Lens reuses the output embedding to map each hidden state to logits (nostalgebraist, 2020); Tuned Lens (Belrose et al., 2023) and Future Lens (Pal et al., 2023) add learned linear projections; PatchScope is a training-free variant (Ghandeharioun et al., 2024). Activation patching shows that many tasks survive with most sequential context removed when intervention is performed in middle layers (Hendel et al., 2023; Wang et al., 2023; Pal et al., 2023; Zheng et al., 2024). Our work is closest to learned-lens methods but asks a different question: in single-token decision settings, can we replace a stronger linear projector with the model's own upper layers applied to a reduced two-token sequence? SimLens is therefore a sequence-axis alternative to direct linear readout rather than another lens variant.

Early exit. Early exit is one of several directions for reducing LLM inference cost, alongside

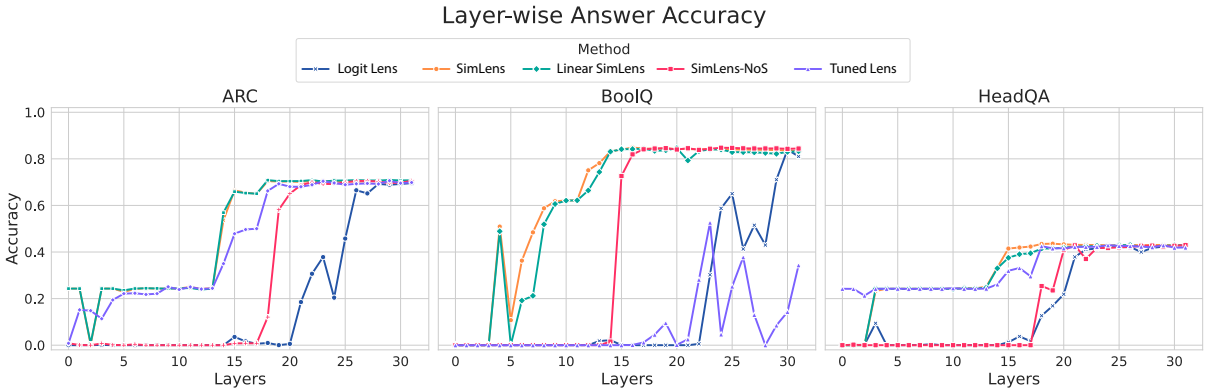

Figure 2: Layer-wise answer accuracy on ARC (left), BoolQ (middle), and HeadQA (right). We compare Logit Lens, Tuned Lens, SimLens, Linear SimLens, and SimLens-NoS (SimLens without the start token). Dashed lines show naive full-model accuracy. SimLens variants decode useful answers earlier than linear baselines, while removing the start token consistently reduces performance.

quantization (Zhang et al., 2024b; Hu et al., 2024; Hasan, 2024), pruning (Li et al., 2024; Fang et al., 2024), and distillation (Wang et al., 2024; Ko et al., 2024; Di Palo et al., 2024). Early-exit, layer-skip, and self-speculative methods—threshold-based exits (Schuster et al., 2022; Zhang et al., 2024a; Fan et al., 2024; Jazbec et al., 2024; Mofakhami et al., 2024), architecture-based exits (Pal et al., 2023; Din et al., 2024; Valade, 2024; Kavehzadeh et al., 2024), and LayerSkip (Elhoushi et al., 2024), D3 (Fan et al., 2025), and DEL (Zarch et al., 2025)—reduce computation along the *depth axis*: they cut the layers run per token to accelerate long-form generation while preserving full sequence length at every layer, with system support for the resulting KV-cache management (Miao et al., 2024). SimExit instead reduces along the *sequence axis*: after exiting, it forwards only the two-token reduced state, and it targets single-token decisions rather than long-form generation. The two families therefore optimize different axes for different workloads—depth-wise acceleration for generation, SimExit for candidate scoring.

3 Preliminaries

We consider a language model $\mathcal{M}$ with L layers, indexed by $l \in \{1, \ldots, L\}$. For an input sequence $S = \{x_1, \ldots, x_n\}$ the input hidden states are $H^0 \in \mathbb{R}^{n \times d}$, with rows $h_i^0 = E(x_i)$ given by the token embedding $E(\cdot)$. We write the l-th transformer block as T_l, a sequence-level operator combining causal self-attention and a feed-forward sublayer:

$$H^l = T_l(H^{l-1}), \quad l = 1, \ldots, L. \tag{1}$$

Let h_i^l denote the i-th row of H^l.

At the final layer L, a linear output projection maps each hidden state to vocabulary logits:

$$z_i^L = W h_i^L, \tag{2}$$

where $W \in \mathbb{R}^{V \times d}$ is the output projection matrix. The token distribution is then

$$p_i = \mathrm{softmax}(z_i^L) \tag{3}$$

with $p_i \in \mathbb{R}^V$.

4 SimLens

SimLens is a training-free readout that reuses the model's native forward dynamics to elicit a prediction from an intermediate layer, without any auxiliary training. In this paper we instantiate it on *single-token decision* tasks, in which the answer is a single token at a position fixed by the task format—an answer-choice letter in multiple-choice QA (e.g., ARC and HeadQA) or a yes/no token in boolean QA (e.g., BoolQ). Because this answer position is known from the prompt, SimLens reads out the model's next-token prediction there directly at an intermediate layer—its highest-probability token over the full vocabulary—without first running a full forward pass to obtain the answer.

For a chosen intermediate layer l, SimLens keeps only two tokens: the start token $\langle s \rangle$ and the candidate answer token $\langle a \rangle$ (the final prompt token, whose next-token prediction gives the answer). Let $h_{\langle s \rangle}^l$ and $h_{\langle a \rangle}^l$ be their layer-l hidden states, and let $T_{l+1:L} = T_L \circ T_{L-1} \circ \cdots \circ T_{l+1}$ denote the composition of the remaining upper blocks. SimLens forwards only these two hidden states through the upper blocks:

$$[\, h_{\langle s \rangle}^L,\ h_{\langle a \rangle}^L \,] = T_{l+1:L}\big([\, h_{\langle s \rangle}^l,\ h_{\langle a \rangle}^l \,]\big), \tag{4}$$

Model	Dataset	Best SimLens Layer	SimLens Acc. (%)	Matched-Layer Naive Acc. (%)	Δ vs. Naive (pp)
LLaMA-7B	ARC	L24	70.6	20.4	+50.2
LLaMA-7B	BoolQ	L16	84.9	0.0	+84.9
LLaMA-7B	HeadQA	L19	43.6	16.9	+26.7
Vicuna-7B	ARC	L26	75.4	72.9	+2.5
Vicuna-7B	BoolQ	L21	84.6	10.1	+74.5
Vicuna-7B	HeadQA	L20	51.2	30.6	+20.6
Mean	–	L21	68.4	25.2	+43.2

Table 1: Layer-wise readout summary at the SimLens best layer. For each model-dataset pair, we report the best SimLens layer, SimLens accuracy at that layer, the matched-layer naive intermediate decoding accuracy, and their difference in percentage points (pp).

Model	Dataset	Full-Depth Acc. (%)	Best Logit Lens	Best SimLens	Δ (pp)
Llama-3.1-8B	ARC	83.5	L31 / 83.9	L26 / 82.8	-1.1
Llama-3.1-8B	BoolQ	63.0	L29 / 71.9	L18 / 72.1	$+0.2$
Qwen2.5-7B-Instruct	ARC	92.8	L28 / 92.8	L27 / 92.9	$+0.1$
Qwen2.5-7B-Instruct	BoolQ	85.9	L27 / 86.3	L28 / 86.9	$+0.6$
Qwen2.5-3B-Instruct	ARC	90.8	L35 / 90.3	L31 / 90.9	$+0.6$
Qwen2.5-3B-Instruct	BoolQ	83.7	L34 / 83.5	L28 / 84.2	$+0.7$

Table 2: Best-layer readout on additional 3B–8B backbones under the candidate-scoring protocol. Best-layer differences against Logit Lens are small (between -1.1 and $+0.7$ pp); Qwen2.5-3B-Instruct was added during the rebuttal period.

where $[\cdot, \cdot]$ denotes the ordered two-token state. We then convert $h^L_{\langle a \rangle}$, the upper-layer representation of the candidate answer token, to a distribution over the full vocabulary via Eq. (2) and Eq. (3), and take its highest-probability token as the model's prediction.

The two retained tokens play complementary roles. The start token provides global conditioning and a stable context anchor, consistent with recent analyses of first-token attention and attention-sink behavior in LLMs (Gu et al., 2024; Barbero et al., 2025); the candidate answer token provides a semantic anchor for the answer prediction at that position. This two-token continuation is the core design of SimLens: it adds very little compute, yet preserves enough upper-layer non-linear processing to recover substantially more faithful latent predictions than a direct linear readout.

4.1 Linear SimLens: Approximation With Linear Projection

SimLens is accurate but still requires a two-token forward pass from layer l to layer L. For frequent confidence checks inside an early-exit loop we approximate SimLens with a lightweight linear mapping, termed *Linear SimLens*, which is trained.

For a monitored layer l with $0 < l \leq L$, we learn a linear map $\mathcal{F}_l$ that predicts the final-layer representation from the layer-l hidden state:

$$\hat{h}^L_i = \mathcal{F}_l(h^l_i), \tag{5}$$

yielding predicted logits $\hat{z}^L_i = W\hat{h}^L_i$. Unlike Tuned Lens (Belrose et al., 2023), the distillation targets are not produced by the original model: the target logits z^L_i are the SimLens logits from Eq. (4) at layer l. With temperature-scaled softmax distributions $p^{SL}_l = \mathrm{softmax}(z^L_i/\tau)$ (the SimLens target) and $\hat{p}_l = \mathrm{softmax}(\hat{z}^L_i/\tau)$ (the Linear SimLens prediction), where τ is a distillation temperature, we train $\mathcal{F}_l$ by minimising the cross-entropy between them

$$\mathcal{L}_l = -\sum_{v \in V} p^{SL}_{l,v} \log \hat{p}_{l,v}. \tag{6}$$

By distilling from SimLens rather than from the original model, Linear SimLens focuses on the representational shift introduced by the two-token continuation while avoiding any full-sequence propagation during training and inference-time monitoring. Full hyperparameters are given in Appendix B.3.

4.2 Setup

We test LLaMA-7B (Touvron et al., 2023), Vicuna-7B (Chiang et al., 2023), Llama-3.1-8B (Grattafiori et al., 2024), Qwen2.5-7B-Instruct, and Qwen2.5-3B-Instruct (Yang et al., 2024). Single-token decision benchmarks are ARC (Clark et al., 2018),

BoolQ (Clark et al., 2019), and HeadQA (Vilares and Gómez-Rodríguez, 2019). All five backbones use the same downstream data, code, and LoRA adaptation workflow (Hu et al., 2022); hence the cross-generation comparison is not confounded by fine-tuning status. We report layer-wise readout accuracy for all five models, while LLaMA-7B and Vicuna-7B drive the full fair-compute and SimExit evaluation.

4.3 Layer-wise Readout Accuracy

Figure 2 compares layer-wise accuracy on LLaMA-7B and Vicuna-7B across Logit Lens, Tuned Lens, SimLens, Linear SimLens, and SimLens-NoS on ARC, BoolQ, and HeadQA. Both SimLens and Linear SimLens decode useful answers at earlier layers than the linear baselines and reach the naive model's accuracy at shallower depths. On ARC, SimLens rises above the random baseline at layer 14 and reaches the naive LLM's accuracy by layer 18, while Logit Lens only starts to rise at layer 21 and remains below SimLens until layer 28; the same earlier-readout trend appears on BoolQ and HeadQA. Removing the start token (SimLens-NoS) substantially lowers ARC readout, which only reaches naive LLM accuracy after layer 20. Table 1 quantifies the gap at the SimLens best layer: across the six LLaMA-7B and Vicuna-7B settings, SimLens reaches mean accuracy 68.4% compared to 25.2% for naive intermediate decoding at the same layer—an average gain of $+43.2$ pp.

Tuned Lens, trained on ARC, is competitive in-domain (70.60% at layer 29) but remains slightly below Linear SimLens (70.91%); its cross-dataset transfer is less stable, with best accuracies of 52.63% on BoolQ and 42.92% on HeadQA, both well below SimLens (84.86% and 43.58%) and Linear SimLens (84.59%). To test cross-task transfer of Linear SimLens, we train it once on ARC and apply it directly to BoolQ, HeadQA, and WikiText-103 (Merity et al., 2016); the layer-wise cross-entropy curves (Figure 9 in Appendix B.1) show training-free SimLens attains the lowest CE across layers, while Linear SimLens remains competitive at much lower monitoring overhead. This supports the motivation that improving the linear readout alone is not enough to replace upper-layer non-linearity, especially under cross-dataset transfer.

The same earlier-readout pattern is reproduced on Llama-3.1-8B, Qwen2.5-7B-Instruct, and Qwen2.5-3B-Instruct under the candidate-scoring protocol (Table 2). On ARC, SimLens improves mid-depth readout by $+14.8$ pp at L16 on Llama-3.1-8B and by $+39.6$ pp at L21 on Qwen2.5-7B-Instruct; on BoolQ, it reaches full-depth candidate accuracy three to four layers earlier than Logit Lens. On Qwen2.5-3B-Instruct, the best SimLens readout appears four layers earlier on ARC and six layers earlier on BoolQ, with best-layer gains of $+0.6$ and $+0.7$ pp. Thus, across the evaluated 3B–8B models, the supported claim is earlier faithful readout rather than a universal scaling law or uniformly higher best-layer accuracy. Full 7B/8B layer-wise curves are in Appendix B.5.

4.4 Fair-Compute Analysis

The two-token continuation adds a small post-forward cost beyond the matched-layer naive decoding. To make the cost comparison explicit, we use two complementary criteria. *Iso-compute* compares methods at matched fair cost, while *iso-accuracy* reports the fair cost required by SimLens to reach the matched-layer target accuracy. In both cases, fair cost includes the two-token post-forward overhead. Table 3 and Figure 3 re-express this gain under fair-cost accounting. At matched compute, SimLens stays more accurate than naive decoding in all six LLaMA-7B/Vicuna-7B settings; this iso-compute advantage is the matched-layer gap already reported in Table 1 (mean $+43.2$ pp), now charging SimLens its two-token overhead (mean fair cost 0.693 vs. 0.688 for naive at matched compute). Under iso-accuracy, SimLens reaches the matched-layer target accuracy with mean fair cost 0.612, well below its best-layer cost.

5 SimExit: Accelerating LLMs with SimLens

Linear SimLens offers cheap decoding with a single linear map; SimLens remains more accurate, especially under cross-task transfer. We combine the two in SimExit, a hybrid early-exit framework that uses Linear SimLens as a fast confidence monitor and SimLens as the final readout.

5.1 Entropy-Based Confidence with Linear SimLens

Because confidence must be re-evaluated repeatedly during the forward pass, we use Linear SimLens for monitoring. We use the entropy of the Linear SimLens distribution as the exit signal:

$$E^{(l)} = -\sum_{v=1}^{V} p_v^{(l)} \log p_v^{(l)}, \tag{7}$$

Model	Dataset	SimLens Fair Cost	Naive Cost at Iso-Compute	Δ Acc. at Iso-Compute (pp)	SimLens Cost at Iso-Accuracy
LLaMA-7B	ARC	0.788	0.781	+50.2	0.606
LLaMA-7B	BoolQ	0.537	0.531	+84.9	0.506
LLaMA-7B	HeadQA	0.633	0.625	+26.7	0.602
Vicuna-7B	ARC	0.849	0.844	+2.5	0.849
Vicuna-7B	BoolQ	0.691	0.688	+74.5	0.506
Vicuna-7B	HeadQA	0.663	0.656	+20.6	0.602
Mean	–	0.693	0.688	+43.2	0.612

Table 3: Iso-compute and iso-accuracy alignment under fair-cost accounting. Fair cost includes the extra two-token post-forward overhead of SimLens. SimLens improves iso-compute accuracy in all six settings and reaches the matched-layer target accuracy with substantially lower fair cost.

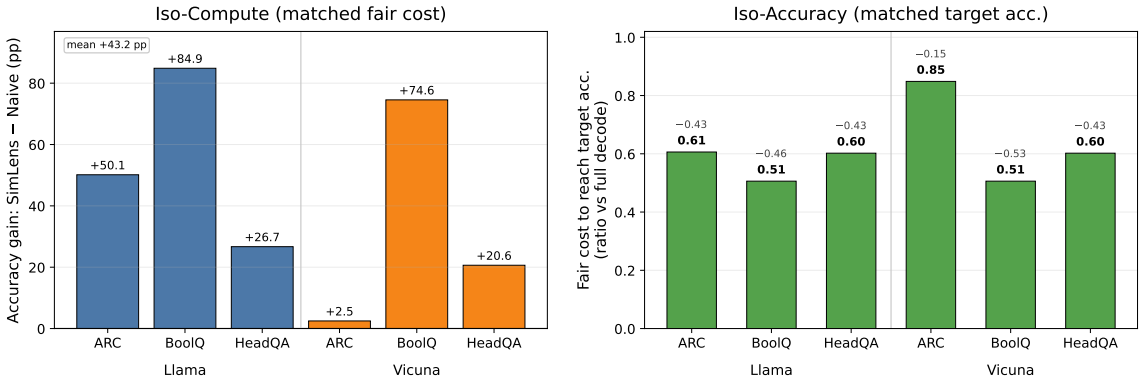

Figure 3: Iso-compute and iso-accuracy alignment under fair-cost accounting. **Left:** accuracy gain of SimLens over matched-cost naive decoding at iso-compute (pp). **Right:** SimLens's fair cost to reach the matched-layer target accuracy, as a ratio to full-depth decoding; the gray number above each bar is the fair-cost difference vs. naive decoding.

where $p^{(l)} \in \mathbb{R}^V$ is the Linear SimLens softmax distribution at layer l. Lower $E^{(l)}$ indicates more concentrated probability mass and higher confidence. Entropy is less sensitive than the top-2 gap (Fan et al., 2024; Din et al., 2024; Jazbec et al., 2024; Mofakhami et al., 2024) to ties among several plausible tokens, especially in settings where multiple candidates remain competitive.

5.2 Adaptive Early-Exit Mechanism

When entropy falls below a fixed threshold τ_{exit}, the full-sequence forward pass is truncated and SimLens completes decoding with only the start and candidate answer tokens, using the upper transformer blocks $T_{l+1:L}$ in Eq. (4). Similar thresholding mechanisms appear in biological decision-making (Gold and Shadlen, 2002, 2007; Kira et al., 2015); tuning τ_{exit} trades accuracy for speed.

To control monitoring overhead, entropy is checked every N layers ($l \bmod N = 0$). A smaller N checks confidence more frequently and can trigger earlier exits, but adds more Linear SimLens evaluations; a larger N reduces monitoring cost but may delay truncation. We select N on a validation set by maximizing speedup under an accuracy-drop constraint δ:

$$N^\star = \arg\max_N \text{Speedup}(N, \tau_{\text{exit}}) \quad \text{s.t. } \Delta\text{Acc}(N, \tau_{\text{exit}}) \leq \delta. \tag{8}$$

With the original LLM, a full forward pass over sequence length n has $\mathcal{O}(n^2)$ attention cost. After exit, SimExit forwards only two tokens, so the post-exit cost is effectively independent of n. The complete procedure is given in Algorithm 1.

5.3 Performance

We evaluate SimExit on ARC, BoolQ, and HeadQA with LLaMA-7B (Figure 4); Vicuna-7B curves are in Appendix C.3. Sweeping the entropy threshold τ_{exit} traces a speed–accuracy frontier within the single-token decision setting. Because Linear SimLens transfers across tasks, an SimExit monitor trained on one task remains competitive on other tasks (Figure 5).

5.4 Threshold Sensitivity

We quantify the threshold trade-off using fair compute, which includes the extra two-token post-forward cost. Averaged over the six set-

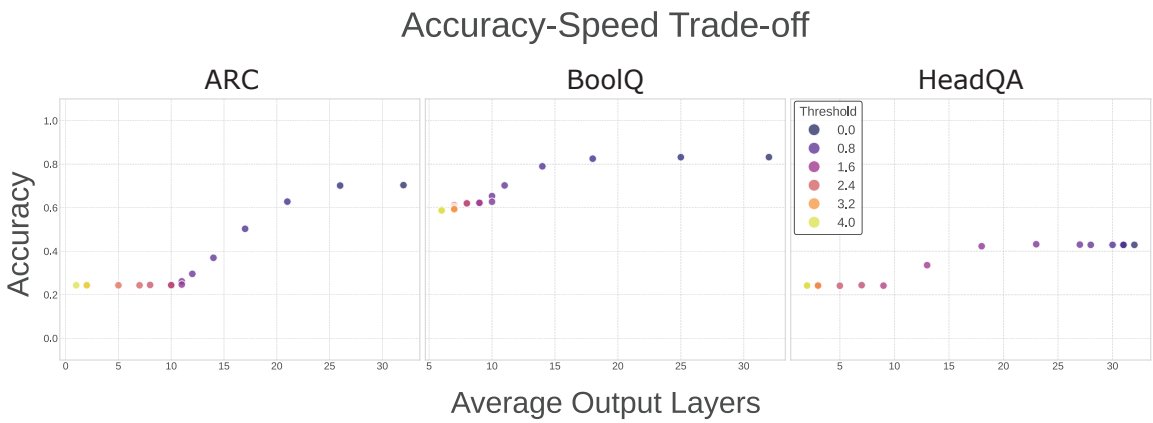

Figure 4: SimExit performance on ARC (left), BoolQ (middle), and HeadQA (right). Colors indicate different entropy thresholds τ_{exit}. Lower thresholds delay exit and preserve accuracy; higher thresholds exit earlier and trade accuracy for speed.

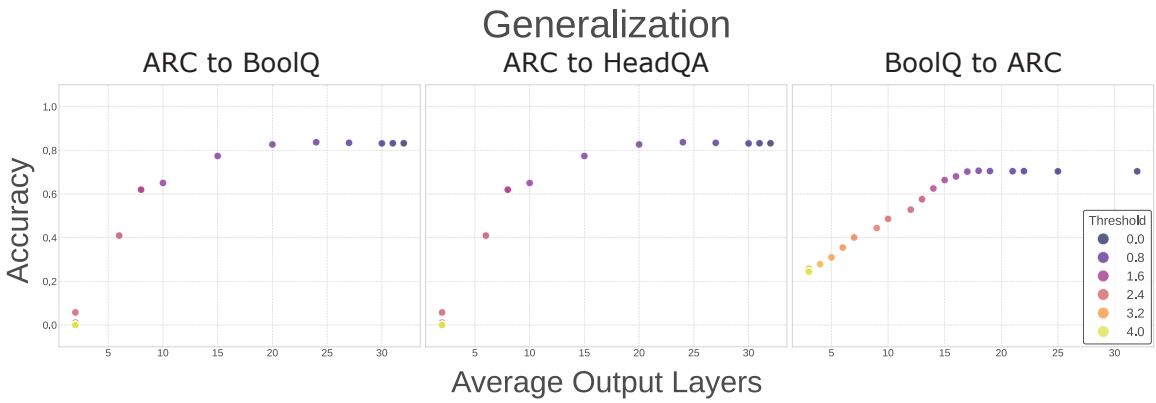

Figure 5: Cross-task transfer for SimExit. Left/middle: evaluate on BoolQ and HeadQA using Linear SimLens trained on ARC. Right: evaluate on ARC using Linear SimLens trained on BoolQ. The method preserves a favorable speed–accuracy trade-off without per-task retraining.

tings, SimExit obtains $1.15\times$ speedup at the best-accuracy operating points and $1.40\times$ when allowing up to a 1 pp accuracy drop. These gains are modest but are reported under fair accounting and without modifying the frozen LLM. Figure 6 shows the LLaMA-7B fair-compute frontier across the three tasks; the per-setting operating points underlying these speedups are listed in Table 7, and the Vicuna-7B frontier in Appendix C.1. For deployment, a simple policy is to first choose an allowable accuracy drop on a validation set (e.g., at most 1 pp), and then select the threshold with the lowest fair cost under that constraint.

6 Ablation Studies

6.1 Why the Candidate Answer Token Matters: Semantic Anchor

We run a wrong-anchor ablation on ARC: keeping the start token fixed, we replace only the candidate answer token $\langle a \rangle$ in the reduced two-token state (Table 4).

Variant	Acc. (%)	Δ vs. Correct-A (pp)
Correct-A	61.4	0.0
Wrong-A	48.2	−13.2
Random-A	24.9	−36.5
Zero-A	25.2	−36.2
Neutral-A	25.9	−35.5

Table 4: Wrong-anchor ablation on ARC. $\langle a \rangle$ is replaced by the correct option token (*Correct-A*), a wrong option token (*Wrong-A*), a random Gaussian vector (*Random-A*), an all-zero vector (*Zero-A*), or a neutral padding token (*Neutral-A*). Non-semantic anchors cause large accuracy drops.

Replacing Correct-A with Wrong-A causes a 13.2 pp drop, and the non-semantic Random-A/Zero-A/Neutral-A cause even larger drops (35.5–36.5 pp), confirming that $\langle a \rangle$ is not a formatting placeholder but provides essential semantic anchoring for alignment with the candidate.

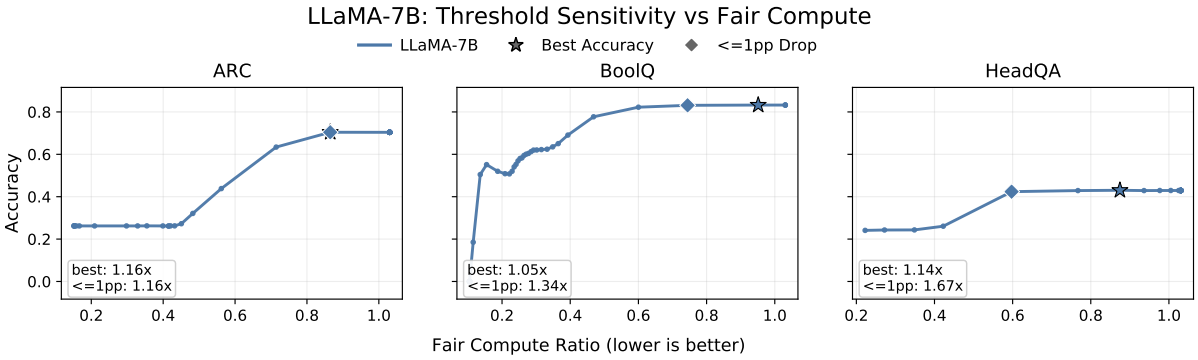

Figure 6: Accuracy vs. fair compute under threshold sweeping for LLaMA-7B. Fair compute includes the $+2$-token post-forward overhead after truncation. Star markers indicate best-accuracy operating points; diamond markers indicate operating points within 1 pp accuracy drop.

Algorithm 1 SimExit (single-token decision)

Require: Frozen LLM with layers $T_1, \ldots, T_L$; pre-trained Linear SimLens maps $\{\mathcal{F}_l\}$; entropy threshold τ_{exit}; monitoring interval N

Require: Input embeddings H^0 containing the start token $\langle s \rangle$ and the candidate answer token $\langle a \rangle$

for $l = 1, \ldots, L$ **do**
  $H^l \leftarrow T_l(H^{l-1})$ {full-sequence forward}
  if $l \bmod N = 0$ **then**
    $\hat{z}^L \leftarrow W\mathcal{F}_l(h^l_{\langle a \rangle})$ {Linear SimLens at $\langle a \rangle$}
    $E^{(l)} \leftarrow \text{Entropy}(\text{softmax}(\hat{z}^L))$
    if $E^{(l)} \leq \tau_{\text{exit}}$ **then**
      $[\,h^L_{\langle s \rangle},\ h^L_{\langle a \rangle}\,] \leftarrow T_{l+1:L}([\,h^l_{\langle s \rangle},\ h^l_{\langle a \rangle}\,])$ {two-token continuation}
      return prediction from $h^L_{\langle a \rangle}$ {SimLens readout}
    end if
  end if
end for
return prediction from $h^L_{\langle a \rangle}$ {no early exit: full-depth readout}

6.2 Why the Start Token Matters: Global Condition

We further test SimLens-NoS, which removes the start token from the reduced state. Removing the start token degrades all six settings with an average drop of 10.7 pp at SimLens's best layer; the effect is especially strong on HeadQA (-20.1 pp on LLaMA-7B and -38.0 pp on Vicuna-7B), where the answer space is more complex, and smallest on ARC (-1.4 and -0.4 pp). The full per-setting results are reported in Table 5. Together with the wrong-anchor results in Table 4, this indicates that robust SimLens decoding depends on both semantic anchoring from the candidate answer token and global conditioning from the start token.

Model	Dataset	SimLens	NoS	Drop	Layer
LLaMA-7B	ARC	70.6	69.2	-1.4	24
LLaMA-7B	BoolQ	84.9	81.9	-3.0	16
LLaMA-7B	HeadQA	43.6	23.5	-20.1	19
Vicuna-7B	ARC	75.4	75.0	-0.4	26
Vicuna-7B	BoolQ	84.6	83.3	-1.3	21
Vicuna-7B	HeadQA	51.2	13.2	-38.0	20
Mean	–	68.4	57.7	-10.7	21

Table 5: SimLens-NoS ablation (removing $\langle s \rangle$) at SimLens's best layer for each setting. SimLens / NoS columns are accuracies (%), Drop is in pp, and Layer is the SimLens best layer.

We read these ablations through two non-exclusive lenses: *space alignment*—forwarding $\langle s \rangle$ and $\langle a \rangle$ through the upper layers may better align intermediate states with the output space—and *extra non-linearity*—the extra forward step may recover transformations a linear map cannot express. We treat both as analysis rather than mechanistic proof; a supporting attention-allocation diagnostic is provided in Appendix B.6.

7 Conclusion

We introduce SimLens, a training-free readout that forwards a reduced two-token continuation through the model's upper layers for single-token decision tasks, and Linear SimLens, a lightweight distilled monitor used by the hybrid early-exit framework SimExit. Across different models, SimLens improves fair-compute candidate readout over direct linear lenses, and SimExit produces a modest but stable speed–accuracy frontier on the two models

used end-to-end. Ablations indicate that the start token contributes global conditioning and the candidate answer token contributes semantic anchoring; we treat the underlying mechanism as analysis rather than a single hard claim.

8 Limitations

Our main experiments focus on single-token decision tasks in which candidate labels are supplied by the task format. While the same candidate-scoring formulation also underlies several production inference pipelines—such as RAG reranking, answer verification and routing, content moderation, and preference/reward scoring—our evaluation is limited to benchmark multiple-choice datasets, and we leave a study of SimExit in these deployed settings to future work. We also do not position SimLens as a general-purpose accelerator for unrestricted long-form generation. Within our current setting, extending SimExit to multi-token autoregressive decoding requires skipped-layer KV-cache backfilling, which can erode or eliminate the ideal early-exit gain as generation length grows. A KV-consistent implementation can always fall back to full-depth decoding, so its normalized cost is bounded by the full-model cost rather than allowed to exceed it (Appendix C.2). This accounting issue is isolated to the auxiliary multi-token limitation analysis and does not affect the main single-token readout, fair-compute, or threshold-sensitivity results. Finally, we compare against representative linear lenses but not every recent variant, and we leave a broader head-to-head against increasingly strong learned linear decoders to future work.

Acknowledgments

This work was supported by the Brain Science and Brain-like Intelligence Technology-National Science and Technology Major Project (grant no. 2021ZD0203701) to T.Y.

References

Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. 2025. Why do llms attend to the first token? *arXiv preprint arXiv:2504.02732*.

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

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *Preprint*, arXiv:1905.10044.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. Analyzing transformers in embedding space. *Preprint*, arXiv:2209.02535.

Flavio Di Palo, Prateek Singhi, and Bilal Fadlallah. 2024. Performance-guided llm knowledge distillation for efficient text classification at scale. *arXiv preprint arXiv:2411.05045*.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2024. Jump to conclusions: Short-cutting transformers with linear transformations. *Preprint*, arXiv:2303.09435.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed A. Aly, Beidi Chen, and Carole-Jean Wu. 2024. Layerskip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12622–12642.

Siqi Fan, Xuezhi Fang, Xingrun Xing, Peng Han, Shuo Shang, and Yequan Wang. 2025. Position-aware depth decay decoding (d3): Boosting large language model inference efficiency. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2990–3001. Association for Computational Linguistics.

Siqi Fan, Xin Jiang, Xiang Li, Xuying Meng, Peng Han, Shuo Shang, Aixin Sun, Yequan Wang, and Zhongyuan Wang. 2024. Not all layers of llms are necessary during inference. *Preprint*, arXiv:2403.02181.

Gongfan Fang, Hongxu Yin, Saurav Muralidharan, Greg Heinrich, Jeff Pool, Jan Kautz, Pavlo Molchanov, and Xinchao Wang. 2024. Maskllm: Learnable semi-structured sparsity for large language models. *arXiv preprint arXiv:2409.17481*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *Preprint*, arXiv:2304.14767.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. *Preprint*, arXiv:2401.06102.

Joshua I Gold and Michael N Shadlen. 2002. Banburismus and the brain: decoding the relationship between sensory stimuli, decisions, and reward. *Neuron*, 36(2):299–308.

Joshua I Gold and Michael N Shadlen. 2007. The neural basis of decision making. *Annu. Rev. Neurosci.*, 30(1):535–574.

Aaron Grattafiori and 1 others. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2024. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*.

Danny Halawi, Jean-Stanislas Denain, and Jacob Steinhardt. 2024. Overthinking the truth: Understanding how language models process false demonstrations. *Preprint*, arXiv:2307.09476.

Jahid Hasan. 2024. Optimizing large language models through quantization: A comparative analysis of ptq and qat techniques. *arXiv preprint arXiv:2411.06084*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. *Preprint*, arXiv:2310.15916.

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

Xing Hu, Yuan Cheng, Dawei Yang, Zhihang Yuan, Jiangyong Yu, Chen Xu, and Sifan Zhou. 2024. I-llm: Efficient integer-only inference for fully-quantized low-bit large language models. *arXiv preprint arXiv:2405.17849*.

Metod Jazbec, Alexander Timans, Tin Hadži Veljković, Kaspar Sakmann, Dan Zhang, Christian A. Naesseth, and Eric Nalisnick. 2024. Fast yet safe: Early-exiting with risk control. *Preprint*, arXiv:2405.20915.

Parsa Kavehzadeh, Mojtaba Valipour, Marzieh Tahaei, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. 2024. Sorted llama: Unlocking the potential of intermediate layers of large language models for dynamic inference. *Preprint*, arXiv:2309.08968.

Shinichiro Kira, Tianming Yang, and Michael N Shadlen. 2015. A neural implementation of wald's sequential probability ratio test. *Neuron*, 85(4):861–873.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*.

Jianwei Li, Yijun Dong, and Qi Lei. 2024. Greedy output approximation: Towards efficient structured pruning for llms without retraining. *arXiv preprint arXiv:2407.19126*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Ruijie Miao, Yihan Yan, Xinshuo Yao, and Tong Yang. 2024. An efficient inference framework for early-exit large language models. *Preprint*, arXiv:2407.20272.

Mehrnaz Mofakhami, Reza Bayat, Ioannis Mitliagkas, Joao Monteiro, and Valentina Zantedeschi. 2024. Performance control in early exiting to deploy large models at the same cost of smaller ones. *Preprint*, arXiv:2412.19325.

nostalgebraist. 2020. Interpreting gpt: The logit lens. LessWrong blog post.

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

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. *Preprint*, arXiv:2207.07061.

Oscar Skean, Md Rifat Arefin, Yann LeCun, and Ravid Shwartz-Ziv. 2024. Does representation matter? exploring intermediate layers in large language models. *Preprint*, arXiv:2412.09563.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. *Preprint*, arXiv:2502.02013.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Florian Valade. 2024. Accelerating large language model inference with self-supervised early exits. *Preprint*, arXiv:2407.21082.

David Vilares and Carlos Gómez-Rodríguez. 2019. Head-qa: A healthcare dataset for complex reasoning. *Preprint*, arXiv:1906.04701.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. *Preprint*, arXiv:2305.14160.

Shuo Wang, Chihang Wang, Jia Gao, Zhen Qi, Hongye Zheng, and Xiaoxuan Liao. 2024. Feature alignment-based knowledge distillation for efficient compression of large language models. *arXiv preprint arXiv:2412.19449*.

An Yang and 1 others. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Hossein Entezari Zarch, Lei Gao, Chaoyi Jiang, and Murali Annavaram. 2025. Del: Context-aware dynamic exit layer for efficient self-speculative decoding. *arXiv preprint arXiv:2504.05598*.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2024a. Draft & verify: Lossless large language model acceleration via self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11263–11282.

Yi Zhang, Fei Yang, Shuang Peng, Fangyu Wang, and Aimin Pan. 2024b. Flattenquant: Breaking through the inference compute-bound for large language models with per-tensor quantization. *arXiv preprint arXiv:2402.17985*.

Bowen Zheng, Ming Ma, Zhongqiao Lin, and Tianming Yang. 2024. Label words as local task vectors in in-context learning. *arXiv preprint arXiv:2406.16007*.

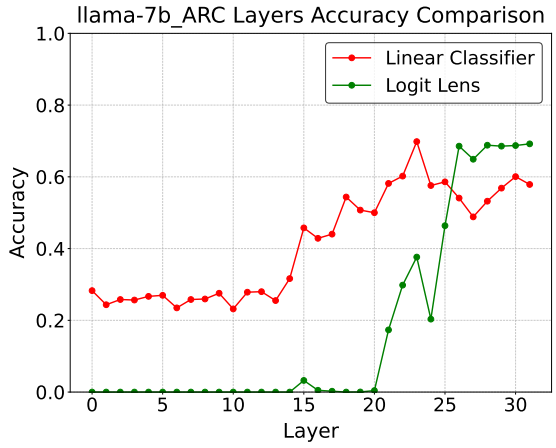

Figure 7: Motivation from representation-output mismatch on ARC. A linear probe recovers answer information much earlier than Logit Lens, indicating that useful task information emerges before final-layer alignment.

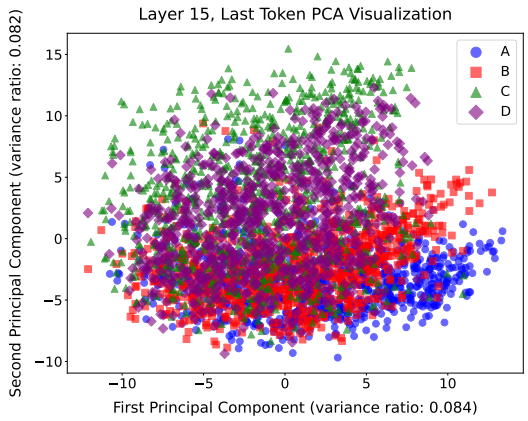

Figure 8: PCA snapshot of layer-15 last-token representations on ARC. Answer choices already form partially separable clusters, consistent with the early linear-probe advantage discussed in the main text.

A Additional Motivation Visualizations

Figure 7 provides the main quantitative motivation: a linear probe recovers answer information much earlier than Logit Lens on ARC, indicating that useful task information is present before final-layer alignment. Figure 8 adds a qualitative snapshot of this phenomenon at layer 15.

B Supplementary Experiments for SimLens

B.1 Cross-Dataset CE / Perplexity

Figure 9 reports layer-wise cross-entropy / perplexity when Linear SimLens and Tuned Lens are trained once on ARC and directly evaluated on BoolQ, HeadQA, and WikiText-103 (Merity et al., 2016). SimLens (training-free) consistently attains the lowest CE across layers; Linear SimLens remains competitive at much lower monitoring overhead.

B.2 Vicuna-7B Layer-wise Curves

This appendix presents SimLens and Linear SimLens results on the Vicuna-7B model. All hyperparameters and evaluation protocols strictly follow those used for LLaMA-7B in the main text.

Figure 10 shows SimLens-based decoding on Vicuna-7B for ARC, BoolQ, and HeadQA. Trends are consistent with LLaMA-7B: SimLens and Linear SimLens outperform Logit Lens, and removing the start token causes a clear performance drop. This supports SimLens' effectiveness on instruction-tuned models.

All five backbones in the main layer-wise comparison use the same downstream data, code, and LoRA adaptation workflow (Hu et al., 2022). For LLaMA-7B and Vicuna-7B, we additionally evaluate the untuned checkpoints. Although their overall perplexity is higher without fine-tuning, SimLens still outperforms the other readout baselines across most layers in Figure 14.

B.3 Linear SimLens Reproducibility

We train one linear map per monitored layer with the underlying LLM—including its embedding and `lm_head`—kept frozen, distilling from the full-vocab SimLens logits of Eq. (4) on the ARC training split of each model/dataset setting. Optimization uses AdamW with a learning rate of 1×10^{-3}, batch size 4, and distillation temperature $\tau = 1.0$, run for 10 epochs with checkpoint selection by validation loss; all runs use fp16 on a single A100. Exit thresholds are tuned on a held-out monitoring split.

B.4 Position-id Implementation and Ablation

For every SimLens experiment in the main text, the two-token state $[\langle s \rangle, \langle a \rangle]$ is forwarded with freshly indexed position ids $\tilde{\text{pos}} = [0, 1]$ rather than preserving $\langle a \rangle$'s original prompt position. This follows the default behavior of our transformer wrapper when `inputs_embeds` is passed without explicit `position_ids`.

To check whether this choice is load-bearing, we ran a small ablation on 200 cached LLaMA-7B + ARC LoRA examples, comparing compact $[0, 1]$ against original $[0, L-1]$ positions for $\langle a \rangle$. At the best SimLens layer (L21), the two schemes are identical at 79.5% accuracy. On any layer where

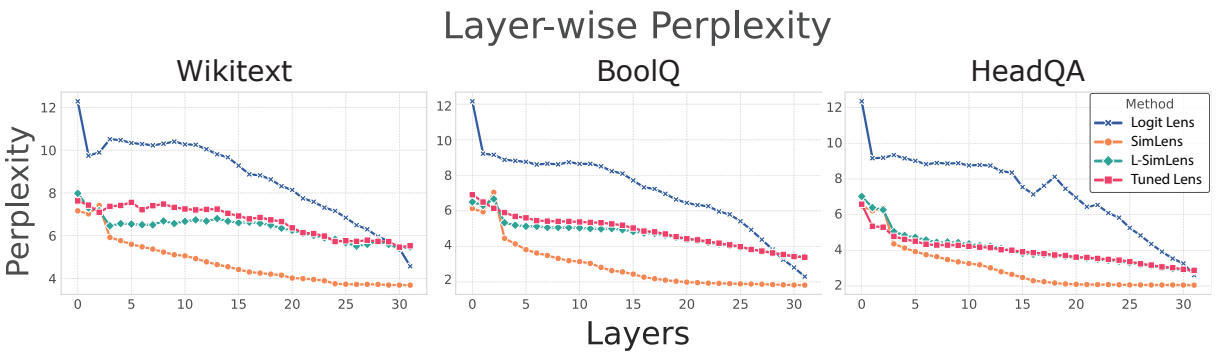

Figure 9: Layer-wise cross-entropy / perplexity on WikiText-103 (left), BoolQ (middle), and HeadQA (right). Linear SimLens and Tuned Lens mappings are trained once on ARC and directly evaluated on target datasets. Lower is better.

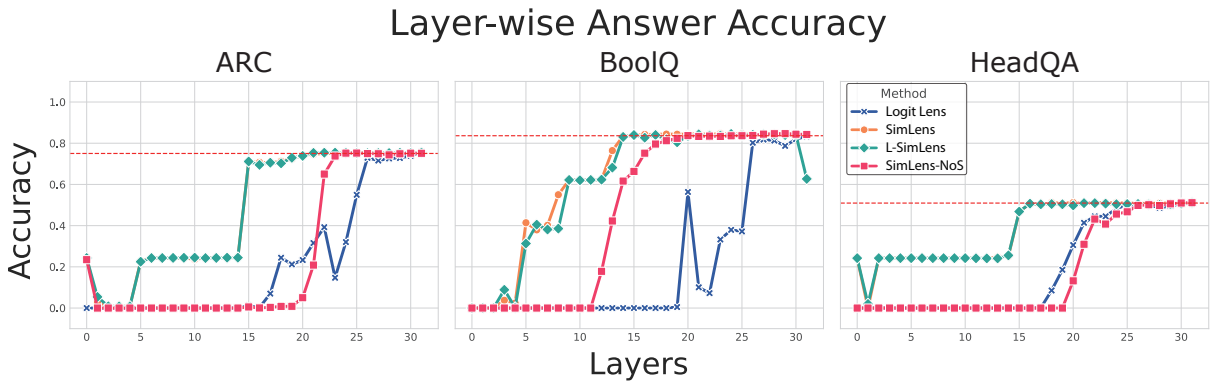

Figure 10: Vicuna-7B layer-wise decoding performance on ARC, BoolQ, and HeadQA for Logit Lens, SimLens, Linear SimLens, and SimLens-NoS.

SimLens reaches above the random baseline (from L14 onwards), the two schemes agree within 1 pp; the only large differences appear at L0–L1, where SimLens is at chance under either scheme. We therefore retain compact positions throughout the main experiments. The per-layer numbers and the script that produces them are released with the paper.

B.5 Layer-wise Readout Curves on Llama-3.1-8B and Qwen2.5-7B-Instruct

Figure 15 shows the full layer-wise Logit Lens vs. SimLens curves on Llama-3.1-8B and Qwen2.5-7B-Instruct for ARC and BoolQ. These curves are the source of the summary numbers in Table 2. The pattern of small best-layer gaps with a visible mid-depth advantage is consistent across all four settings.

B.6 Attention Diagnostic

We report two complementary attention diagnostics. Both should be read as analysis evidence supporting the global-conditioning interpretation of $\langle s \rangle$; neither is meant as an exit-confidence signal.

Attention from the readout position to prompt spans. On 500 LLaMA-7B + ARC examples we measure, at selected layers, the attention mass that the readout (final prompt) position allocates to four reconstructed input spans of the original full prompt: the start token, the question, the options, and the answer prefix. Selected layers are reported in Table 6. Attention to $\langle s \rangle$ grows quickly after the first few layers (e.g., 0.80 at L4, 0.72 at L28) while SimLens accuracy only rises sharply around L16–L24. The two are therefore not linearly tied.

Reduced-sequence $\langle a \rangle \to \langle s \rangle$ attention. On Llama-3.1-8B ARC / BoolQ we additionally measure, inside the reduced two-token continuation, the layer-mean attention mass from $\langle a \rangle$ to $\langle s \rangle$. Across 128 ARC and 128 BoolQ examples, this mass is 0.85 on ARC and 0.85 on BoolQ (means over layers, excluding L0). The correct-vs-incorrect gap is below 0.002 at every layer, which is consistent with $\langle s \rangle$ acting as a stable global-conditioning anchor but rules out using this attention as an exit signal.

Layer	$\langle s \rangle$	Question	Options	Ans. prefix	SimLens acc (%)
0	0.057	0.230	0.519	0.194	24.3
4	0.803	0.010	0.049	0.138	24.3
8	0.745	0.058	0.097	0.100	24.2
12	0.509	0.125	0.237	0.129	24.3
16	0.577	0.079	0.162	0.182	65.5
20	0.440	0.064	0.203	0.293	70.3
24	0.455	0.058	0.200	0.287	70.6
28	0.719	0.010	0.049	0.223	70.4
31	0.652	0.089	0.102	0.157	70.4

Table 6: Attention allocation from the readout position to reconstructed input spans of the original full ARC prompt, and matched-layer SimLens accuracy.

C Supplementary Experiments for SimExit

C.1 Detailed Threshold Tables

Table 7 reports the exact operating points underlying the threshold-sensitivity summary in the main text. Figure 12 provides the corresponding fair-compute frontier for Vicuna-7B (the LLaMA-7B frontier is shown in the main text, Figure 6).

C.2 Multi-Token KV-Cache Backfill Proxy

For multi-token generation, an early-exit decoder must either backfill cache entries for skipped layers or revert to full-depth computation. The latter provides a strict fallback, so normalized cost is capped at 1.0 (the full model) rather than allowed to exceed it. Backfilling can nevertheless consume the entire ideal saving as the output grows; our present experiments therefore do not establish acceleration for multi-token generation. Figure 11 visualizes this corrected accounting: the no-backfill curve is only an approximate, KV-inconsistent lower-bound proxy, whereas KV-consistent decoding is bounded by falling back to the full model. This accounting issue is isolated to this auxiliary limitation analysis and does not affect the single-token layer-wise readout, fair-compute, or threshold-sensitivity results.

C.3 Additional Performance on Other Datasets and Models

This appendix reports SimExit results with LLaMA-7B and Vicuna-7B on ARC, BoolQ, and HeadQA using the same adaptive-threshold protocol as in the main text.

Figure 13 shows that SimExit consistently achieves competitive accuracy while enabling early termination across layers. These results confirm the general effectiveness of SimExit in a variety of QA tasks and models.

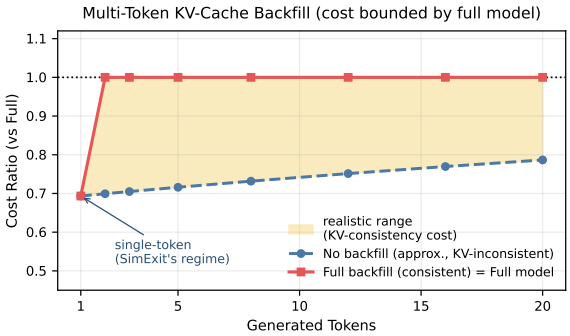

Figure 11: Multi-token limitation under KV-cache backfill assumptions. KV-consistent decoding is capped at the full-model cost, while the no-backfill curve is an approximate, KV-inconsistent lower-bound proxy. The shaded region marks the single-token setting evaluated by SimExit.

C.4 Additional Generalization Performance on Other Datasets and Models

This section expands on the generalization analysis, presenting full results where SimExit is applied to unseen datasets using Linear SimLens mappings trained on other datasets. Results are shown for both LLaMA-7B and Vicuna-7B across all dataset permutations (ARC, BoolQ, HeadQA).

The performance curves in Figures 16 and 17 demonstrate that SimExit generalizes well across different tasks.

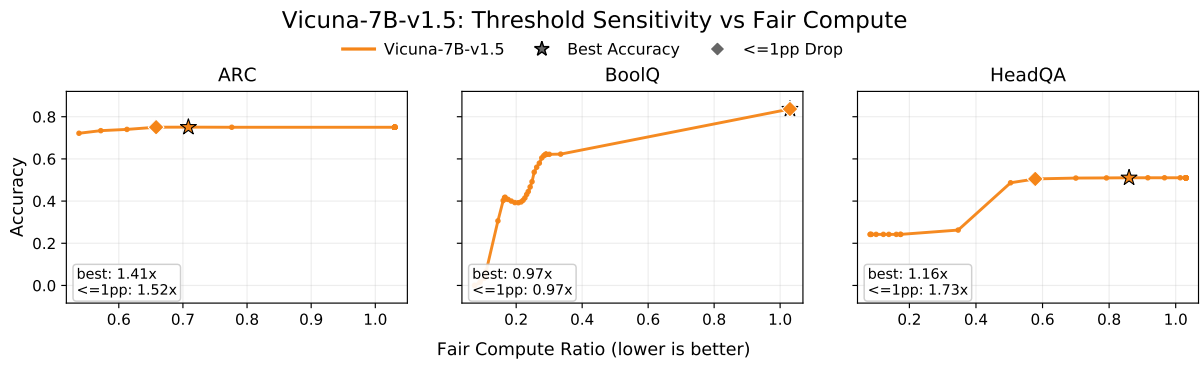

Figure 12: Accuracy vs. fair compute under threshold sweeping for Vicuna-7B. Fair compute includes the +2-token post-forward overhead after truncation. Star markers indicate best-accuracy operating points; diamond markers indicate operating points within 1pp accuracy drop.

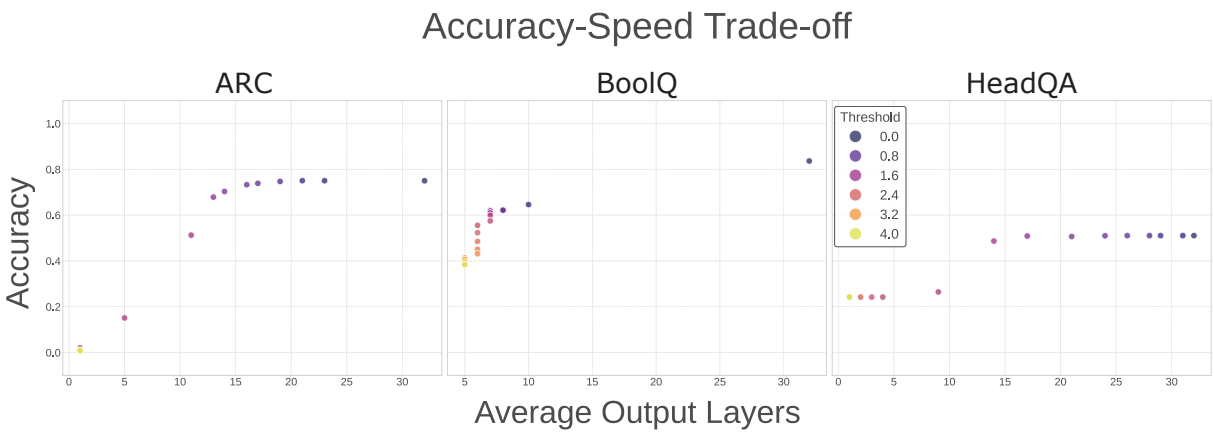

Figure 13: SimExit speed-accuracy curves on Vicuna-7B across ARC, BoolQ, and HeadQA under different entropy thresholds.

Model	Dataset	Best Th	Best Acc (%)	Best Fair Cost	Best Speedup	Th @≤1pp Drop	Acc @≤1pp Drop (%)	Fair Cost @≤1pp Drop	Speedup @≤1pp Drop
LLaMA-7B	ARC	3.80	70.4	0.865	1.16×	3.80	70.4	0.865	1.16×
LLaMA-7B	BoolQ	7.40	83.2	0.951	1.05×	7.20	83.1	0.744	1.34×
LLaMA-7B	HeadQA	1.20	43.0	0.875	1.14×	0.80	42.4	0.597	1.67×
Vicuna-7B	ARC	0.80	75.1	0.709	1.41×	0.60	75.0	0.658	1.52×
Vicuna-7B	BoolQ	7.80	83.6	1.030	0.97×	7.80	83.6	1.030	0.97×
Vicuna-7B	HeadQA	2.80	51.1	0.860	1.16×	2.20	50.5	0.578	1.73×
Mean	-	3.97	67.7	0.882	1.149×	3.73	67.5	0.745	1.399×

Table 7: Threshold sensitivity of SimExit under fair-compute accounting, with explicit fair-cost fields at both operating points (best accuracy and within 1pp accuracy drop).

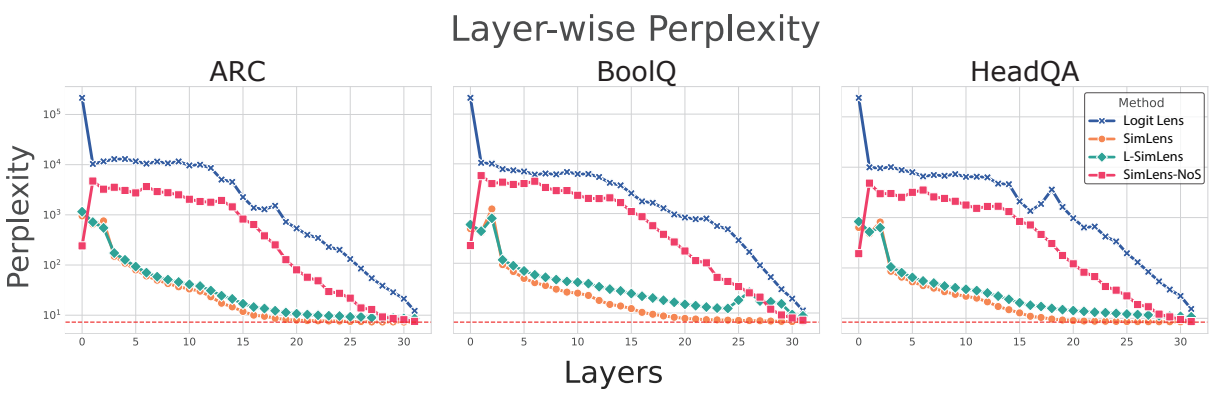

Figure 14: Layer-wise perplexity with vanilla LLaMA-7B across decoding methods. Lower perplexity indicates better readout quality.

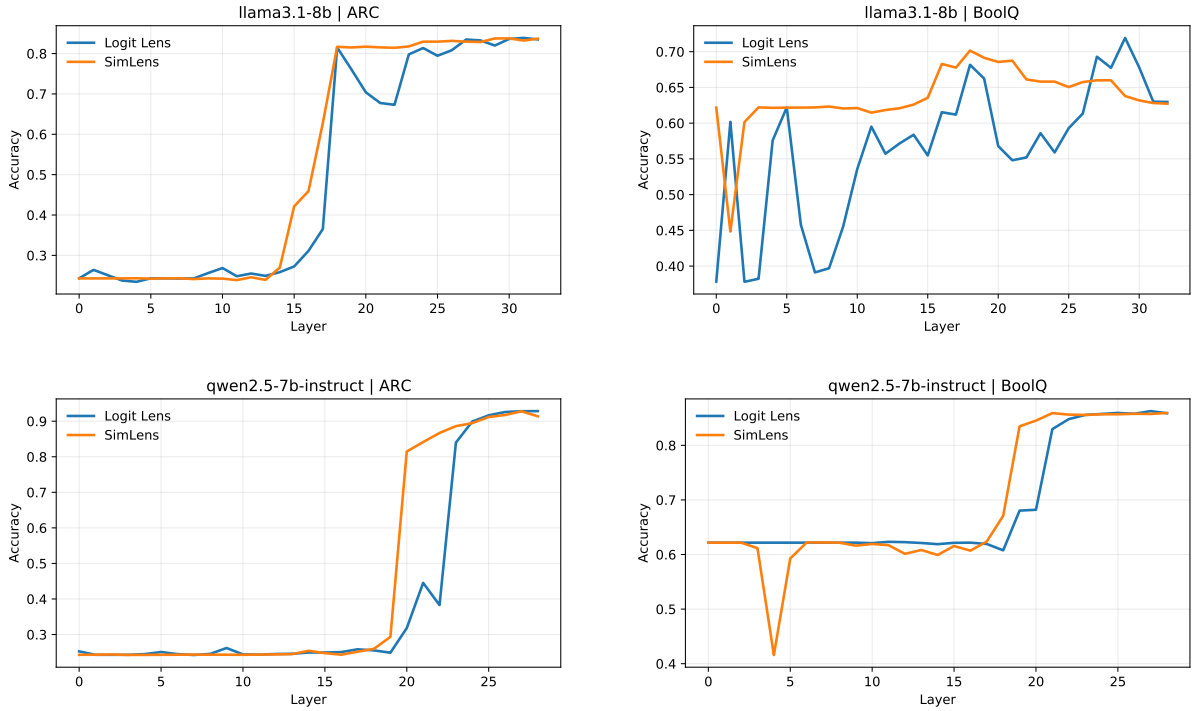

Figure 15: Layer-wise readout accuracy on Llama-3.1-8B and Qwen2.5-7B-Instruct under the candidate-scoring protocol. Top: Llama-3.1-8B on ARC (left) and BoolQ (right). Bottom: Qwen2.5-7B-Instruct on ARC (left) and BoolQ (right). The summary numbers underlying Table 2 are derived from these curves; the qualitative pattern (small best-layer gap against Logit Lens, larger advantage at earlier intermediate layers) is consistent across all four settings.

Figure 16: Cross-task generalization of SimExit with Linear SimLens on LLaMA-7B.

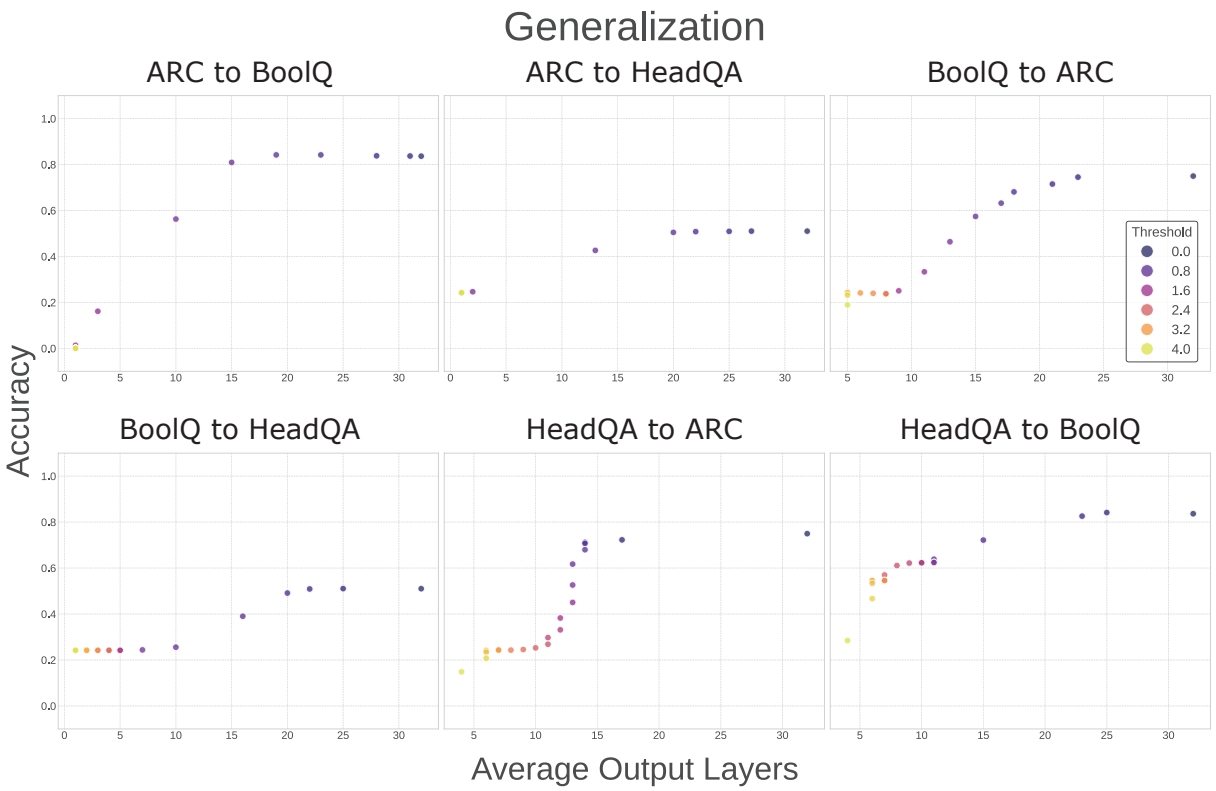

Figure 17: Cross-task generalization of SimExit with Linear SimLens on Vicuna-7B.