SimLens for Early Exit in Large Language Models: Eliciting Accurate Latent Predictions with One More Token

Anonymous ACL submission

Abstract

Intermediate-layer predictions in large language models (LLMs) are informative but hard to decode accurately, especially at early layers. Existing lens-style methods typically rely on direct linear readout, which is simple but often drifts away from the model's eventual prediction. We propose **SimLens**, a simple training-free decoder for single-token decision tasks that keeps only the start token and a candidate answer token (<s> and <a>) and performs one lightweight continuation through the remaining upper layers. This surprisingly small modification recovers much more accurate latent predictions than direct linear decoding. We further introduce **Linear SimLens**, a lightweight linear approximation for entropy-based confidence estimation, and combine the two in **SimExit**, a hybrid early-exit mechanism. On ARC, BoolQ, and HeadQA with LLaMA-7B and Vicuna-7B, SimLens improves Iso-Compute accuracy in all six settings, with an average gain of +0.43 even when fair compute includes the extra two-token post-forward overhead. SimExit yields an average 1.15× speedup at the best-accuracy operating points and 1.40× when allowing up to a 1 percentage-point accuracy drop. Ablations show that <s> and <a> play distinct roles as global condition and semantic anchor, respectively.

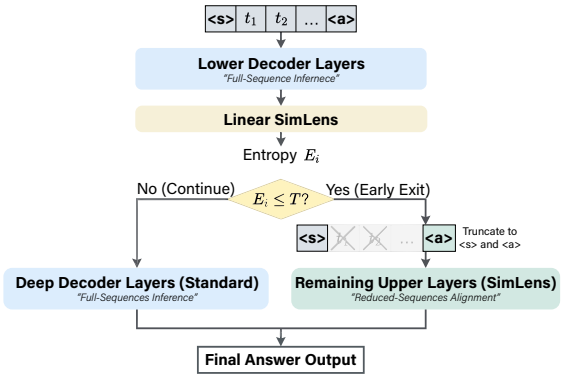

Figure 1: Overview of SimExit. At each monitored layer, Linear SimLens provides a low-cost entropy confidence score for exit decisions. Once the confidence threshold is met, full-sequence forwarding is truncated and SimLens continues with only two tokens (<s> and <a>) to decode the answer.

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).

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 model's final prediction. On ARC, for example, a linear probe recovers answer information much earlier than Logit Lens, 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.

In this paper, we take a different route. Rather than designing another stronger linear lens, we ask whether one more tiny forward step can recover a much better intermediate-layer prediction. We propose **SimLens**, a simple and training-free decoder for single-token decision tasks. Instead of directly reading out a single intermediate token, SimLens keeps only two tokens—the start token <s> and a candidate answer token <a>—and forwards this minimal state through the remaining upper layers. The two tokens play complementary roles: <s> preserves global conditioning, while <a> serves as a semantic anchor for the candidate being scored.

Surprisingly, this minimal continuation is already enough to outperform linear baselines at early layers and often move much closer to the final model prediction. Our experiments focus on single-token decision tasks such as multiple-choice QA and classification-style decoding on ARC, BoolQ, and HeadQA (Clark et al., 2018, 2019; 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 requiring a separate full-model decode to first discover the answer token.

More accurate intermediate decoding naturally enables early exit. We therefore introduce **Linear SimLens**, a lightweight linear approximation used only for low-cost entropy monitoring, and combine it with SimLens in **SimExit**, a hybrid early-exit framework. Across LLaMA-7B and Vicuna-7B, SimLens improves Iso-Compute accuracy in all six model-dataset settings by an average of +0.43 under fair accounting that includes the extra two-token post-forward overhead. SimExit yields an average 1.15× speedup at the best-accuracy operating points and 1.40× when allowing up to a 1 percentage-point accuracy drop.

Contributions. We make four contributions: (1) we introduce SimLens, a simple training-free decoder that uses a two-token continuation to elicit accurate latent predictions from intermediate layers; (2) we show that keeping <s> and <a> is sufficient to markedly improve early-layer readout, and we clarify their distinct roles as global condition and semantic anchor; (3) we introduce Linear SimLens and SimExit, which combine low-cost entropy monitoring with accurate reduced-sequence decoding for adaptive early exit; and (4) we provide quantitative fairness analysis, cross-task transfer results, and ablations that frame space alignment and extra non-linearity as plausible analysis perspectives rather than a single hard mechanism claim.

2 Related Work

Hidden Representation Decoding A growing body of work shows that intermediate layers encode 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), while Tuned Lens (Belrose et al., 2023) and Future Lens (Pal et al., 2023) add learned linear projections; PatchScope adopts a training-free variant (Ghandeharioun et al., 2024). Patching studies replace or ablate subsets of activations to test functional necessity, revealing that many tasks survive with most sequential context removed when intervention occurs 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 such as Tuned Lens and Future Lens, but the core question is different. We focus on single-token decision tasks, where the model scores a small candidate set rather than continuing unrestricted long-form generation. In this regime, SimLens does not learn a better linear projector; instead, it reuses the model's own upper layers on only two tokens (<s> and a candidate answer token). We therefore position SimLens as a simple reduced-sequence alternative to direct linear readout, rather than as another lens variant with a larger learned decoder.

Early Exit Scaling LLMs inflates inference cost, motivating efficiency techniques such as quantization (Zhang et al., 2024b; Hu et al., 2024; Hasan, 2024), pruning (Li et al., 2024; Fang et al., 2024), distillation (Wang et al., 2024; Ko et al., 2024; Di Palo et al., 2024), and early exit (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).

Among these methods, early exit accelerates inference by terminating computation once a confidence threshold is met, either through threshold optimization (Zhang et al., 2024a; Fan et al., 2024; Jazbec et al., 2024; Mofakhami et al., 2024) or

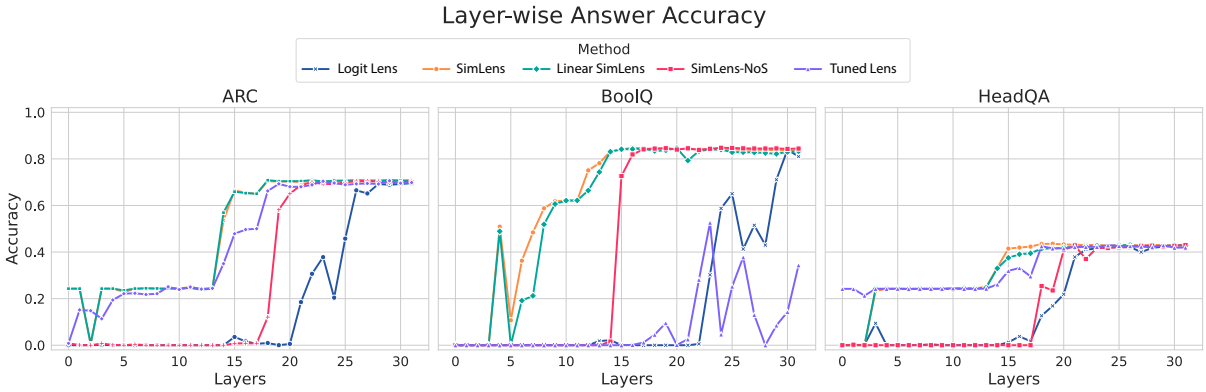

Figure 2: Layer-wise answer accuracy on ARC (left), BoolQ (middle), and HeadQA (right). We compare Logit Lens, Tuned Lens, SimLens, Linear SimLens, and SimLens-NoS (SimLens without <s>). Dashed lines show naive full-model accuracy. SimLens variants decode useful answers earlier than linear baselines,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

architectural designs (Pal et al., 2023; Din et al., 2024; Valade, 2024; Kavehzadeh et al., 2024; Fan et al., 2025; Zarch et al., 2025) that support intermediate predictions. System work also studies the inference framework and KV-cache management needed to make early-exit LLMs practical in deployment (Miao et al., 2024).

We propose a hybrid early-exit mechanism. It uses linear mappings as fast confidence estimators and uses a more accurate reduced-sequence propagation mechanism to generate accurate answers.

3 Preliminaries

We consider a decoder-only language model $\mathcal{M}$ with L layers, indexed by $l \in \{1, \ldots, L\}$. For an input sequence $S = \{x_1, \ldots, x_n\}$, let $h_i^l \in \mathbb{R}^d$ denote the hidden state at token position i and layer l, with

$$h_i^0 = E(x_i), \tag{1}$$

where $E(\cdot)$ is the token embedding function.

We write the l-th layer transformation as T_l, so

$$h_i^l = T_l(h_i^{l-1}), \quad l = 1, \ldots, L. \tag{2}$$

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

$$z_i^L = W h_i^L \tag{3}$$

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

$$p_i = \mathrm{softmax}(z_i^L) \tag{4}$$

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

4 SimLens

Here, we present SimLens, a simple training-free decoder that reuses the model's native forward dynamics.

In this paper, we instantiate SimLens for single-token decision tasks. The candidate answer token comes from the task format itself (e.g., an answer-choice token or label token), so SimLens scores candidates directly at layer l and does not require running the full model once to first obtain the answer. For an intermediate layer l, we form a minimal two-token state with the start token <s> and the candidate answer token <a>. Let T_k denote the k-th transformer block. We define the layer-transition operator from layer l to layer L as

$$T_l^L \triangleq T_L \circ T_{L-1} \circ \cdots \circ T_{l+1}. \tag{5}$$

Then SimLens computes

$$[\texttt{<s>}^L, \texttt{<a>}^L] = T_l^L([\texttt{<s>}^l, \texttt{<a>}^l]) \tag{6}$$

where $[\cdot, \cdot]$ denotes an ordered two-token state (not an interval), and T_l^L is the forward map from layer l to layer L restricted to this state.

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

Model	Dataset	SimLens Best Layer	SimLens Acc.	Naive Acc. (same layer)	Δ Acc.
LLaMA-7B	ARC	24	0.706	0.204	+0.501
LLaMA-7B	BoolQ	16	0.849	0.000	+0.849
LLaMA-7B	HeadQA	19	0.436	0.169	+0.267
Vicuna-7B	ARC	26	0.754	0.729	+0.025
Vicuna-7B	BoolQ	21	0.846	0.101	+0.746
Vicuna-7B	HeadQA	20	0.512	0.306	+0.206
Mean	-	21.0	0.684	0.251	+0.432

Table 1: Selected layer-wise accuracy summary at SimLens' best layer. For each model-dataset pair, we report the best SimLens layer, SimLens accuracy at that layer, naive accuracy at the same layer, and their difference.

then convert <a>^{L} to vocabulary probabilities via Eq. (3) and Eq. (4), score each candidate answer, and select the highest-scoring candidate.

To evaluate SimLens, we compare layer-wise accuracy against other methods (Figure 2). On ARC, SimLens rises above the baseline at layer 14 and reaches the naive LLM's accuracy by layer 18. In comparison, Logit Lens starts to rise above the baseline only at layer 21 and remains below SimLens until layer 28.

The start token <s> is crucial for SimLens. When it is removed, performance on ARC is significantly degraded, reaching the naive LLM's accuracy only after layer 20. The baseline in this case is also lower than SimLens and only comparable to Logit Lens, suggesting that the start token, widely used as a positional anchor in transformer models, contributes crucial task information that SimLens can exploit. This is consistent with recent evidence that the first token can act as a privileged routing point or attention sink in LLM computation (Barbero et al., 2025; Gu et al., 2024).

The same trend also appears on BoolQ and HeadQA (Figure 2), suggesting that SimLens generalizes across our single-token decision benchmarks.

Figure 2 also reports results for an ARC-trained Tuned Lens. Tuned Lens is competitive on the in-domain ARC setting, reaching 70.60% at layer 29, but remains slightly below Linear SimLens (70.91%). Its cross-dataset transfer is less stable: on BoolQ, the best Tuned Lens accuracy is 52.63%, far below SimLens (84.86%), Linear SimLens (84.59%), and even Logit Lens (83.94%); on HeadQA, Tuned Lens reaches 42.92%, again below SimLens (43.58%). This pattern is consistent with the CE-based comparison in Figure 3, where SimLens remains strongest across most layers.

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 in early exit, we further approximate SimLens with a lightweight linear mapping, termed Linear SimLens.

Specifically, for an early layer l ($0 < l \leq L$), our objective is to obtain a linear mapping that transforms representational space of layer l to that of layer L:

$$\hat{h}_i^L = \mathcal{F}(h_i^l) \tag{7}$$

where h_i^l is the hidden state at layer l, $\mathcal{F}$ is the linear mapping, and $\hat{h}_i^L$ is the predicted final-layer representation.

Distinct from methods such as Tuned Lens (Belrose et al., 2023), the target of training h_i^L is not the output of the original transformer model. Instead, it is obtained from SimLens, as described in Eq. (6). $\mathcal{F}$ is trained by minimizing the cross-entropy loss between the two logits z_i^L and $\hat{z}_i^L$ computed from h_i^L and $\hat{h}_i^L$ with Eq. (3):

$$\mathcal{L} = CE(z_i^L, \hat{z}_i^L) \tag{8}$$

where CE is the cross-entropy loss, defined as:

$$CE(p) = -\sum_i p_i log(p_i)$$

This approximation is a distillation from SimLens. By distilling to SimLens targets, Linear SimLens focuses on representational space shift while avoiding full-sequence propagation during training and inference-time monitoring.

4.2 Performance

We evaluate SimLens, Linear SimLens, Logit Lens, and Tuned Lens on ARC, BoolQ, and HeadQA

Model	Dataset	SimLens Fair Cost	Naive Cost (Iso-Compute)	Δ Acc. at Iso-Compute	SimLens Cost (Iso-Accuracy)
LLaMA-7B	ARC	0.788	0.781	+0.501	0.606
LLaMA-7B	BoolQ	0.537	0.531	+0.849	0.506
LLaMA-7B	HeadQA	0.633	0.625	+0.267	0.602
Vicuna-7B	ARC	0.849	0.844	+0.025	0.849
Vicuna-7B	BoolQ	0.691	0.688	+0.746	0.506
Vicuna-7B	HeadQA	0.663	0.656	+0.206	0.602
Mean	-	0.693	0.688	+0.432	0.612

Table 2: Iso-Compute / Iso-Accuracy fairness alignment. Fair cost explicitly includes SimLens' extra two-token post-forward overhead. SimLens wins all 6 pairs in Iso-Compute accuracy and reaches the matched target accuracy with substantially lower fair cost.

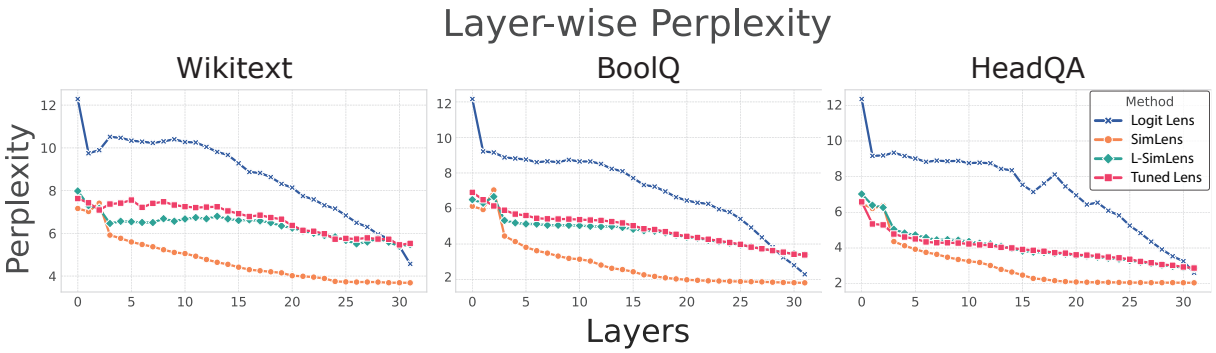

Figure 3: Layer-wise cross-entropy/perplexity on WikiText-103 (left), BoolQ (middle), and HeadQA (right). Linear SimLens and Tuned Lens mappings are trained once on ARC and directly evaluated on target datasets. Lower is better. SimLens is training-free and remains strongest across most layers.

(Clark et al., 2018, 2019; Vilares and Gómez-Rodríguez, 2019). Figure 2 shows that both SimLens and Linear SimLens decode useful answers at earlier layers and reach the naive model's accuracy with shallower depth.

Table 1 quantifies the layer-wise advantage: across 6 model-dataset pairs, mean SimLens accuracy is 0.68 versus 0.25 for naive decoding at matched layers, giving an average improvement of +0.43.

To test generalization, we train Linear SimLens once on ARC and directly apply it to BoolQ, HeadQA, and WikiText-103 (Merity et al., 2016) without retraining. Figure 3 shows that SimLens (training-free) consistently gives the lowest CE, while Linear SimLens remains competitive with much lower monitoring overhead. This supports our motivation that improving linear readout alone is not enough to replace upper-layer non-linearity, especially under cross-dataset transfer.

To make the fairness 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 target accuracy. In both cases, fair cost includes SimLens' extra two-token post-forward overhead.

Table 2 addresses fairness directly: even after adding the extra two-token post-forward cost, SimLens still improves Iso-Compute accuracy in all 6 settings (average +0.43) and reaches the matched target accuracy with mean fair cost 0.612. The corresponding bar-chart visualization is provided in Appendix B.1.

5 SimExit: Accelerating LLMs with SimLens

While Linear SimLens provides fast decoding with a simple linear transformation, SimLens remains more accurate, especially under task transfer. We therefore combine the two to build SimExit, a hybrid early-exit mechanism that takes advantage of both methods.

5.1 Entropy-Based Confidence Metrics with Linear SimLens

First, we need a decision mechanism for early exit. By adopting a suitable confidence metric, we can determine whether the model can already produce a reliable output at an intermediate layer, thereby skipping subsequent layers and improving inference speed. Because confidence must be evaluated repeatedly before exit, we adopt Linear SimLens for efficiency.

One of the most commonly used metrics for making early-exit decisions is the probability gap between the top two predicted tokens (Fan et al., 2024; Din et al., 2024; Jazbec et al., 2024; Mofakhami et al., 2024). A larger gap indicates higher confidence in the model's prediction. However, this metric is not ideal: when multiple tokens are similarly plausible, the model may be overly conservative, which reduces the efficiency of potential early exits (Schuster et al., 2022). To address this limitation, we employ an entropy-based metric and compute an entropy score for each candidate early-exit layer:

$$H^{(l)} = -\sum_{v=1}^{V} p_v^{(l)} \log p_v^{(l)}, \tag{9}$$

where $p^{(l)} \in \mathbb{R}^V$ is the Linear SimLens soft-max distribution at layer l. Lower $H^{(l)}$ means the probability mass is more concentrated, i.e., higher confidence.

5.2 Adaptive Early-Exit Mechanism

Based on the entropy-based confidence metric, we establish an adaptive early-exit mechanism. When the entropy falls below a predefined threshold, the inference process is truncated and SimLens completes decoding with the reduced sequence. Similar thresholding mechanisms have been studied in biological decision-making (Gold and Shadlen, 2002; Gold and Shadlen, 2007; Kira et al., 2015). Adjusting the threshold allows us to balance accuracy and speed for different application scenarios.

As we have demonstrated the performance advantage of SimLens, we use it to produce answers once the inference is stopped. In Algorithm 1, $\text{SimLens}_i(\cdot)$ denotes the i-th transformer block applied to the reduced two-token state after truncation, i.e., the stepwise implementation of the restricted map in Eq. (6). The complete SimExit algorithm is shown in Algorithm 1.

Algorithm 1 introduces an evaluation interval N: we compute the confidence score only when $i \bmod N = 0$. Thus, N controls a direct trade-off between monitoring granularity and inference overhead. A smaller N checks confidence more frequently, which can trigger earlier exits but adds more Linear SimLens evaluations; a larger N reduces monitoring cost but may miss the best exit point and delay truncation. In practice, we select N on a validation set by maximizing speedup under an accuracy-drop constraint δ:

Algorithm 1 SimExit

Require: Pre-trained Linear SimLens, threshold T, evaluation interval N

```
for i from 0 to L do
    if trunc then
        x_i = SimLens_i(x_{i-1})
        Cache_i = updateCache(x_i)
    else
        x_i = Layer_i(x_{i-1})
    end if
    if i mod N = 0 then
        E_i = Entropy(LinearSimLens(x_i))
        if E_i ≤ T then
            trunc = True
        end if
    end if
end for
```

$$N^\star = \arg\max_N \text{Speedup}(N, T) \quad \text{s.t. } \Delta\text{Acc}(N, T) \le \delta, \tag{10}$$

where T is the entropy threshold.

With the original LLM model, a full forward pass over sequence length n has $\mathcal{O}(n^2)$ attention cost. After exit is triggered, SimExit only forwards two tokens, which makes the post-exit cost effectively independent of n.

5.3 Performance

Finally, we evaluate SimExit on ARC, BoolQ, and HeadQA using LLaMA (Figure 4); additional Vicuna results are provided in Appendix C.4. By tuning the entropy threshold, we obtain a clear speed-accuracy trade-off within this single-token decision setting. Because Linear SimLens transfers across tasks, SimExit trained on one task also performs well on other tasks (Figure 5).

5.4 Threshold Sensitivity

We quantify the threshold trade-off using fair compute (including the extra two-token post-forward cost). On average, SimExit achieves 1.15× speedup at the best-accuracy operating points, and 1.40× when allowing up to a 1 percentage-point accuracy drop. This indicates a stable and tunable speed–accuracy frontier in practice. Figure 6 shows the corresponding fair-compute frontier for LLaMA-7B. For deployment, a simple policy is to first choose an allowable accuracy drop on a

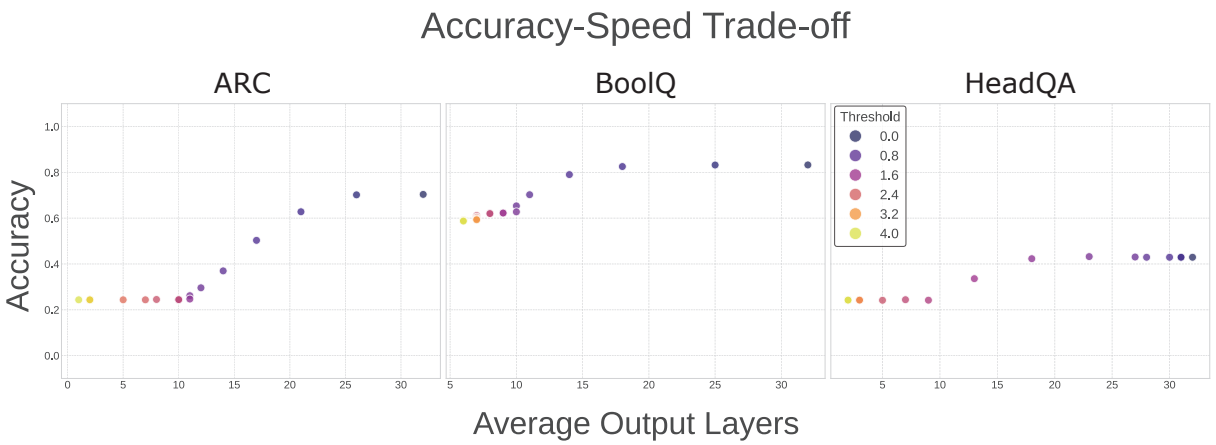

Figure 4: SimExit performance on ARC (left), BoolQ (middle), and HeadQA (right). Colors indicate different entropy thresholds. Lower thresholds delay exits and preserve accuracy; higher thresholds exit earlier and increase speed with a larger accuracy trade-off.

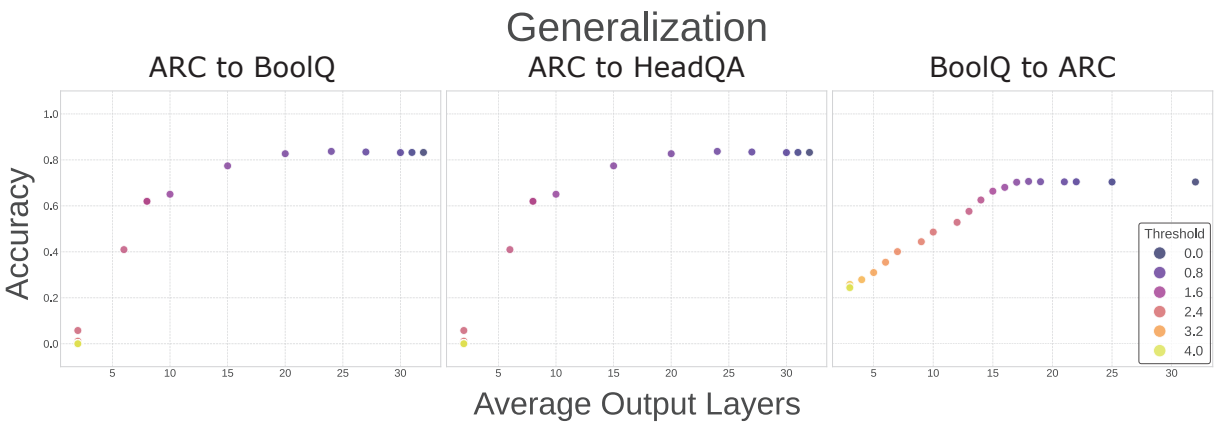

Figure 5: Cross-task transfer for SimExit. Left/middle: evaluate on BoolQ and HeadQA using Linear SimLens trained on ARC. Right: evaluate on ARC using Linear SimLens trained on BoolQ. The method preserves a favorable speed-accuracy trade-off without per-task retraining.

validation set (e.g., at most 1pp), and then select the threshold with the lowest fair cost under that constraint. The exact operating-point tables and the corresponding Vicuna-7B frontier are provided in Appendix C.1.

6 Ablation Studies

6.1 Why <a> Matters: Semantic Anchor

We run a Wrong-A ablation on ARC (layer 32), keeping all settings fixed and only replacing the answer anchor <a>. Results are shown in Table 3.

Replacing the correct anchor with a wrong option causes a 13.24-point drop. Replacing it with random/zero/neutral anchors causes even larger drops (35.48–36.53 points). This confirms that <a> is not a formatting token; it provides essential semantic anchoring for alignment.

Method	Accuracy	Δ vs. Correct-A (pp)
SimLens (Correct-A)	0.614	0.00
SimLens (Wrong-A)	0.482	-13.24
SimLens (Random-A)	0.249	-36.53
SimLens (Zero-A)	0.252	-36.23
SimLens (Neutral-A)	0.259	-35.48

Table 3: Wrong-A ablation on ARC (layer 32). Replacing <a> with a wrong or non-semantic anchor causes large accuracy drops.

6.2 Why <s> Matters: Global Condition

We further test SimLens-NoS, which removes the start token <s>. For each model-dataset pair, we compare SimLens-NoS at the same layer where SimLens reaches its best accuracy.

As shown in Table 4, removing <s> degrades performance in all 6 settings (average drop: 10.68 points). Together with the Wrong-A results in Table 3, this indicates that robust SimLens decoding

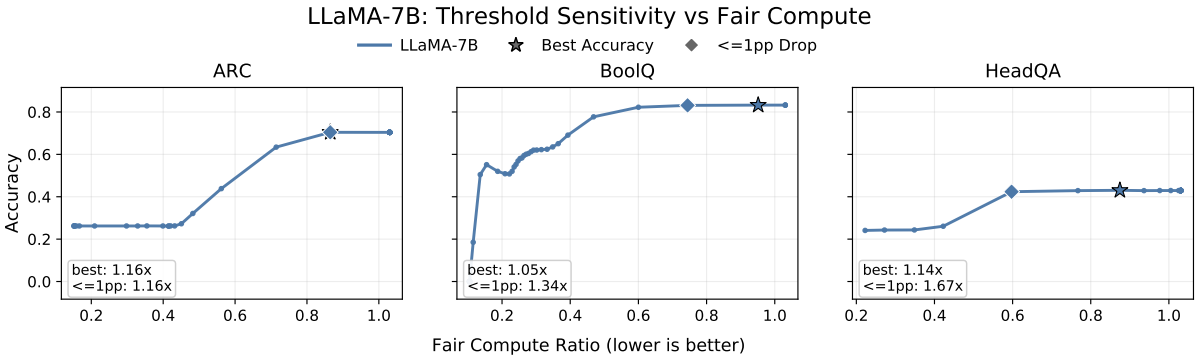

Figure 6: Accuracy vs. fair compute under threshold sweeping for LLaMA-7B. Fair compute includes the +2-token post-forward overhead after truncation. Star markers indicate best-accuracy operating points; diamond markers indicate operating points within 1pp accuracy drop.

Model	Dataset	SimLens (Best Layer)	SimLens-NoS (Same Layer)	Drop (pp)	Best Layer
LLaMA-7B	ARC	0.706	0.692	-1.35	24
LLaMA-7B	BoolQ	0.849	0.819	-2.94	16
LLaMA-7B	HeadQA	0.436	0.235	-20.06	19
Vicuna-7B	ARC	0.754	0.750	-0.42	26
Vicuna-7B	BoolQ	0.846	0.833	-1.31	21
Vicuna-7B	HeadQA	0.512	0.132	-38.00	20
Mean	-	0.684	0.577	-10.68	21.0

Table 4: SimLens-NoS ablation (removing <s>) at SimLens' best layer for each setting.

depends on both semantic anchoring from <a> and global conditioning from <s>. The effect is especially strong on HeadQA (20.06 and 38.00 points), where option semantics are more complex.

7 Discussion

Our empirical results show that a simple two-token continuation can decode latent predictions much more accurately than direct linear readout at early layers. We intentionally do not present a single hard mechanism claim for why this works. Instead, we view the results through two analysis perspectives. One is *space alignment*: forwarding <s> and <a> through the upper layers may better align intermediate states with the final output space. The other is *extra non-linearity*: compared with linear lenses, the additional forward step may recover task-relevant transformations that a linear map cannot express.

The ablations indicate that the two retained tokens play distinct roles—<s> supplies global conditioning and <a> supplies semantic anchoring—but we treat these findings as analysis rather than definitive mechanistic proof. Looking forward, it would be valuable to test whether stronger cross-layer consistency during pretraining, or more general reduced-sequence decoding designs, can preserve the same simple-yet-effective behavior beyond the current single-token setting.

8 Conclusion

In this work, we introduce SimLens, a simple training-free method for eliciting accurate latent predictions from intermediate layers with one more token. We further develop Linear SimLens, a lightweight and transferable approximation used for entropy-based confidence estimation.

Building on these methods, we develop SimExit, a hybrid early-exit algorithm for single-token decision tasks with LLMs. Our results show that, within this task setting, SimLens improves fair-compute decoding quality, transfers across datasets, and enables SimExit to reduce inference cost while maintaining high accuracy. More broadly, the results suggest that a very small amount of additional forward computation can be more effective than purely linear readout for turning intermediate representations into usable predictions.

9 Limitations

Our main experiments target single-token decision tasks (multiple-choice / classification style decod-

ing). In this regime, SimLens only propagates <s> and <a> and achieves consistent gains under fair compute accounting.

As a short autoregressive pilot beyond the single-token setting, we run SimLens on Opus100 (Zhang et al., 2020) using only the first two target tokens; details are reported in Appendix C.3. At threshold 6.5, SimLens cached decoding matches vanilla full forward on prefix-2 token accuracy (0.02) while exiting much earlier on average (13.79 layers). We view this as evidence that SimLens is not inherently restricted to a single output token, but not as validation for general long-form generation.

For multi-token autoregressive generation, skipped layers require KV-cache backfilling to preserve decoding consistency, which can dominate the runtime. Using an upper-bound proxy analysis, we find that as generation length grows from 1 to 5/20 tokens, the backfill cost ratio rises from 0.69 to 1.92/6.16, i.e., 2.79×/8.22× over the ideal no-backfill setting. This suggests that the key bottleneck for multi-token extension is cache backfilling on skipped layers, not the one-step SimLens readout itself. The detailed proxy table and scaling plot are provided in Appendix C.2. We also compare against representative linear lenses, but not every recent variant (e.g., Future Lens); a broader head-to-head comparison would further clarify when simple reduced-sequence propagation is preferable to increasingly strong learned linear decoders.

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, Zach Furman, Logan Smith, Danny Halawi, and Igor Ostrovsky. 2023. Eliciting latent predictions from transformers with the tuned lens. *Preprint*, arXiv:2303.08112.

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, and Bram Wasti. 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.

Siqi Fan, Xin Jiang, Xiang Li, Xuying Meng, and Peng Han. 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.

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, and Dan Zhang. 2024. Fast yet safe: Early-exiting with risk control. *Preprint*, arXiv:2405.20915.

Parsa Kavehzadeh, Mojtaba Valipour, Marzieh Tahaei, Ali Ghodsi, and Boxing Chen. 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, and Dara Bahri. 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, and Jalal Naghiyev. 2025. Layer by layer: Uncovering hidden representations in language models. *Preprint*, arXiv:2502.02013.

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, and Hao Zhou. 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*.

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*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, and Ke Chen. 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. Distributed rule vectors is a key mechanism in large language models' in-context learning. *arXiv preprint arXiv:2406.16007*.

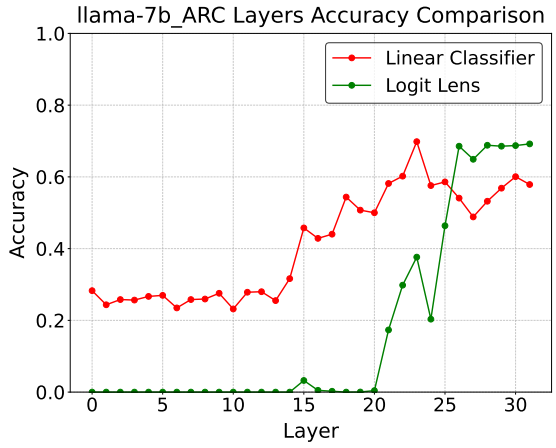

Figure 7: Motivation from representation-output mismatch on ARC. A linear probe recovers answer information much earlier than Logit Lens, indicating that useful task information emerges before final-layer alignment.

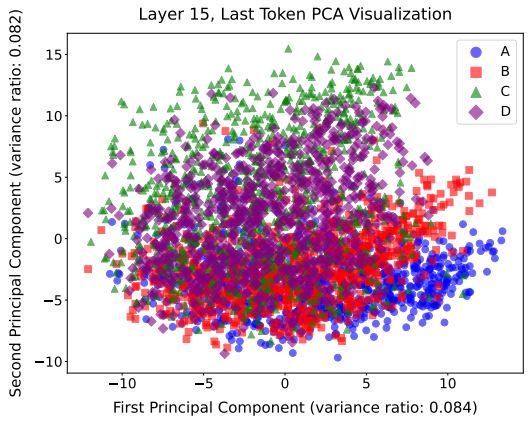

Figure 8: PCA snapshot of layer-15 last-token representations on ARC. Answer choices already form partially separable clusters, consistent with the early linear-probe advantage discussed in the main text.

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 Additional Fairness Visualization

Figure 9 provides a visual summary of the same fairness comparison reported numerically in Table 2.

This appendix presents SimLens and Linear SimLens results on the Vicuna-7B model. All hyper-

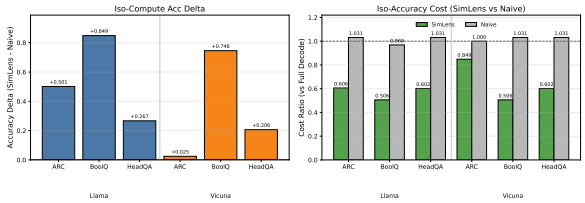

Figure 9: Iso-Compute and Iso-Accuracy alignment. Left: accuracy gain of SimLens over Naive at matched fair compute. Right: fair-cost comparison of SimLens vs. Naive at matched target accuracy.

parameters 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.

When using LLaMA and Vicuna models, we first perform LoRA-based fine-tuning (Hu et al., 2022) on the downstream datasets to ensure good performance. We also evaluate their performance without fine-tuning. Although the overall perplexity is higher without fine-tuning, SimLens still outperforms the other readout baselines across most layers in Figure 11.

C Supplementary Experiments for SimExit

C.1 Detailed Threshold Tables

Tables 5 and 6 report the exact operating points underlying the threshold-sensitivity summary in the main text. Figure 12 provides the corresponding fair-compute frontier for Vicuna-7B.

C.2 Multi-Token KV-Cache Backfill Proxy

Table 7 and Figure 13 provide the detailed upper-bound proxy analysis for multi-token KV-cache backfilling discussed in the limitation section.

C.3 Short-Prefix Autoregressive Pilot

We also report a short autoregressive pilot on Opus100 (Zhang et al., 2020) using only the first two target tokens. We evaluate the first 200 test examples with a LLaMA-7B checkpoint used in the SimLens experiments. This is a short-prefix pilot rather than a full translation benchmark: the complete target length in this subset still has mean 63.1 tokens.

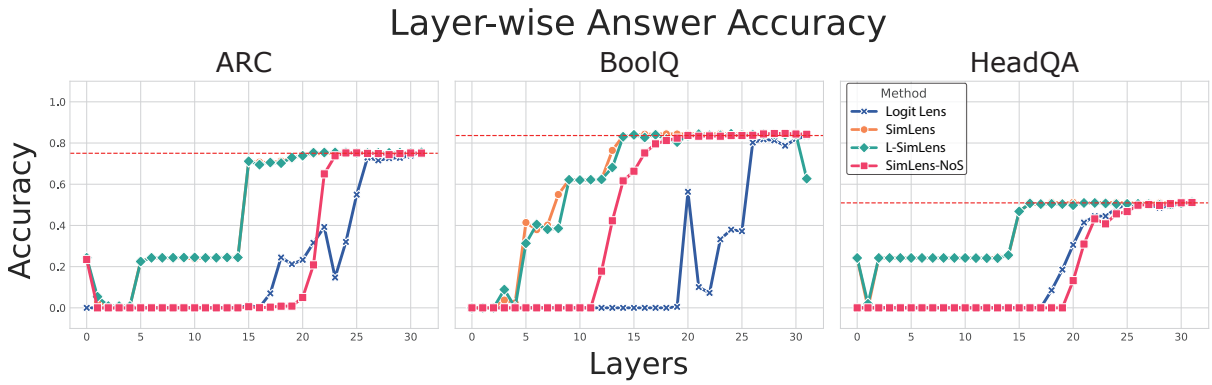

Figure 10: Vicuna-7B layer-wise decoding performance on ARC, BoolQ, and HeadQA for Logit Lens, SimLens, Linear SimLens, and SimLens-NoS.

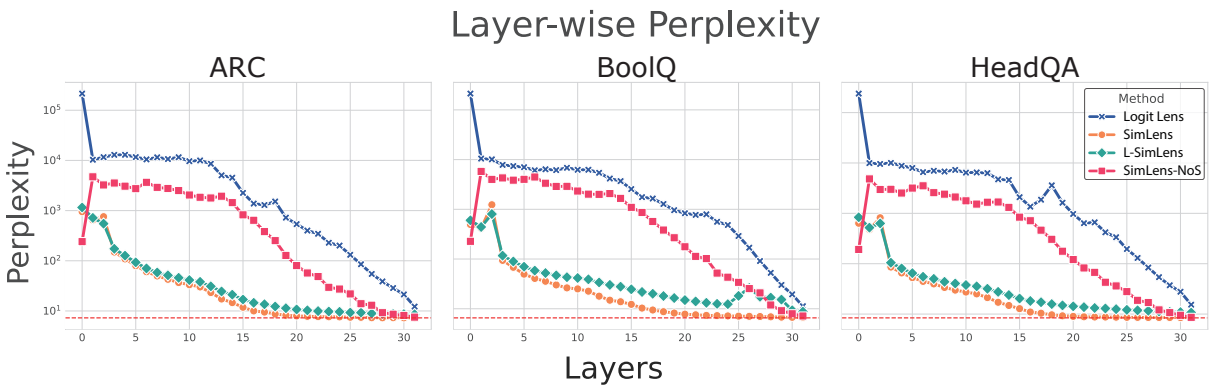

Figure 11: Layer-wise perplexity with vanilla LLaMA-7B across decoding methods. Lower perplexity indicates better readout quality.

Model	Dataset	Best Th	Best Acc	Best Speedup	Th @≤1pp Drop	Acc @≤1pp Drop	Speedup @≤1pp Drop
LLaMA-7B	ARC	3.80	0.704	1.16×	3.80	0.704	1.16×
LLaMA-7B	BoolQ	7.40	0.832	1.05×	7.20	0.831	1.34×
LLaMA-7B	HeadQA	1.20	0.430	1.14×	0.80	0.424	1.67×
Vicuna-7B	ARC	0.80	0.751	1.41×	0.60	0.750	1.52×
Vicuna-7B	BoolQ	7.80	0.836	0.97×	7.80	0.836	0.97×
Vicuna-7B	HeadQA	2.80	0.511	1.16×	2.20	0.505	1.73×
Mean	-	3.97	0.677	1.149×	3.73	0.675	1.399×

Table 5: Threshold sensitivity of SimExit under fair compute accounting.

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	0.704	0.865	1.16×	3.80	0.704	0.865	1.16×
LLaMA-7B	BoolQ	7.40	0.832	0.951	1.05×	7.20	0.831	0.744	1.34×
LLaMA-7B	HeadQA	1.20	0.430	0.875	1.14×	0.80	0.424	0.597	1.67×
Vicuna-7B	ARC	0.80	0.751	0.709	1.41×	0.60	0.750	0.658	1.52×
Vicuna-7B	BoolQ	7.80	0.836	1.030	0.97×	7.80	0.836	1.030	0.97×
Vicuna-7B	HeadQA	2.80	0.511	0.860	1.16×	2.20	0.505	0.578	1.73×
Mean	-	3.97	0.677	0.882	1.149×	3.73	0.675	0.745	1.399×

Table 6: Extended threshold sensitivity table with explicit fair-cost fields at both operating points (best accuracy and within 1pp drop).

Gen Tokens	Ideal/Full	Backfill/Full	Backfill Over Ideal
1	0.693	0.693	1.00×
5	0.716	1.915	2.79×
20	0.787	6.159	8.22×

Table 7: Proxy upper-bound analysis of multi-token KV-cache backfilling cost.

Method	Threshold	Token Acc	Exact Match	Avg Exit Layer
Vanilla full forward	-	0.023	0.000	-
SimLens cached decoding	5.0	0.013	0.000	10.340
SimLens cached decoding	5.5	0.013	0.000	10.890
SimLens cached decoding	6.0	0.018	0.000	13.000
SimLens cached decoding	6.5	0.023	0.000	13.790
SimLens cached decoding	7.0	0.023	0.000	14.275
SimLens cached decoding	7.5	0.023	0.000	14.930
SimLens cached decoding	8.0	0.023	0.000	15.785

Table 8: Short-prefix autoregressive pilot on Opus100 using only the first two target tokens (200 test examples). At threshold 6.5, SimLens cached decoding matches vanilla full forward on prefix-2 token accuracy while exiting much earlier on average.

At the best operating point ($\mathrm{th} = 6.5$), SimLens cached decoding reaches the same prefix-2 token accuracy as vanilla full forward (0.02) while exiting at 13.79 layers on average. This suggests that SimLens can extend to very short autoregressive outputs. At the same time, we do not interpret this result as evidence for general long-form generation, since longer outputs remain constrained by skipped-layer KV-cache backfilling.

C.4 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 14 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.

C.5 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 15 and 16 demonstrate that SimExit generalizes well across different tasks.

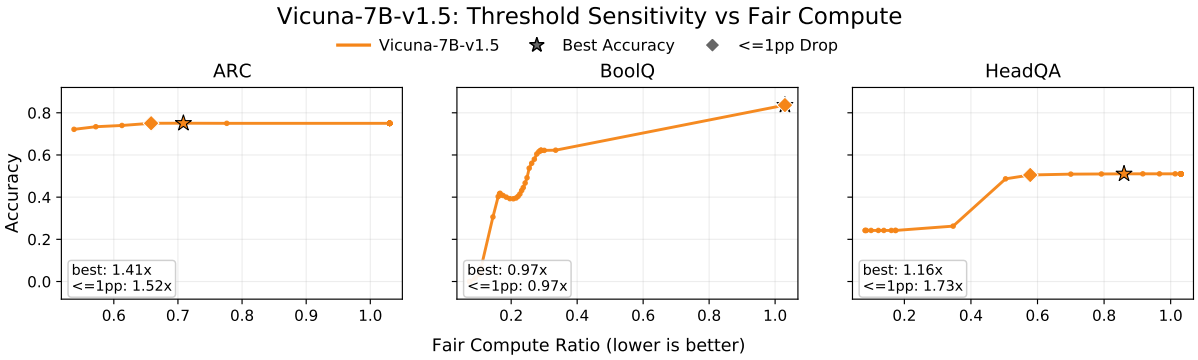

Figure 12: Accuracy vs. fair compute under threshold sweeping for Vicuna-7B. Fair compute includes the +2-token post-forward overhead after truncation. Star markers indicate best-accuracy operating points; diamond markers indicate operating points within 1pp accuracy drop.

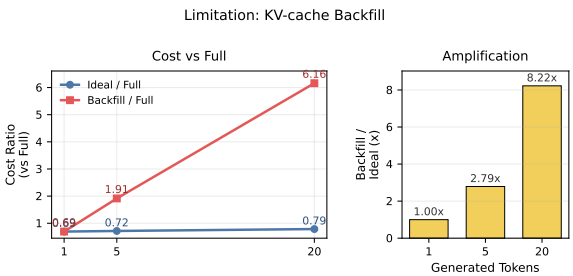

Figure 13: Multi-token limitation under KV-cache backfill assumptions. Backfill cost increases much faster than ideal no-backfill extension as generation length grows.

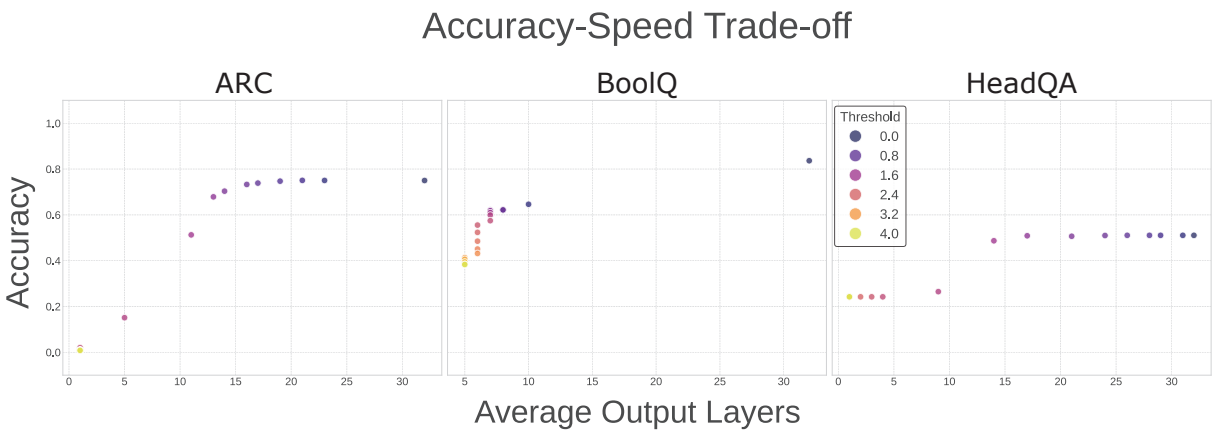

Figure 14: SimExit speed-accuracy curves on Vicuna-7B across ARC, BoolQ, and HeadQA under different entropy thresholds.

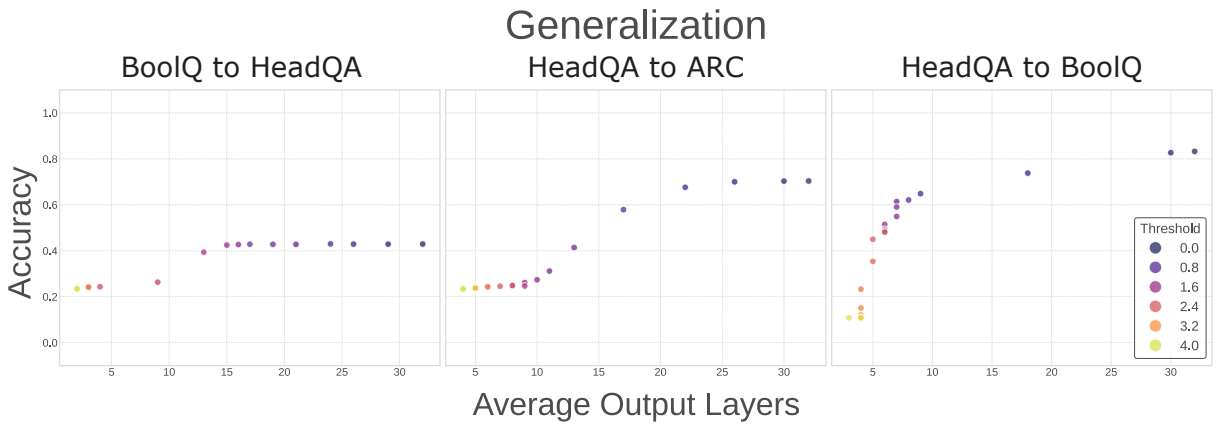

Figure 15: Cross-task generalization of SimExit with Linear SimLens on LLaMA-7B.

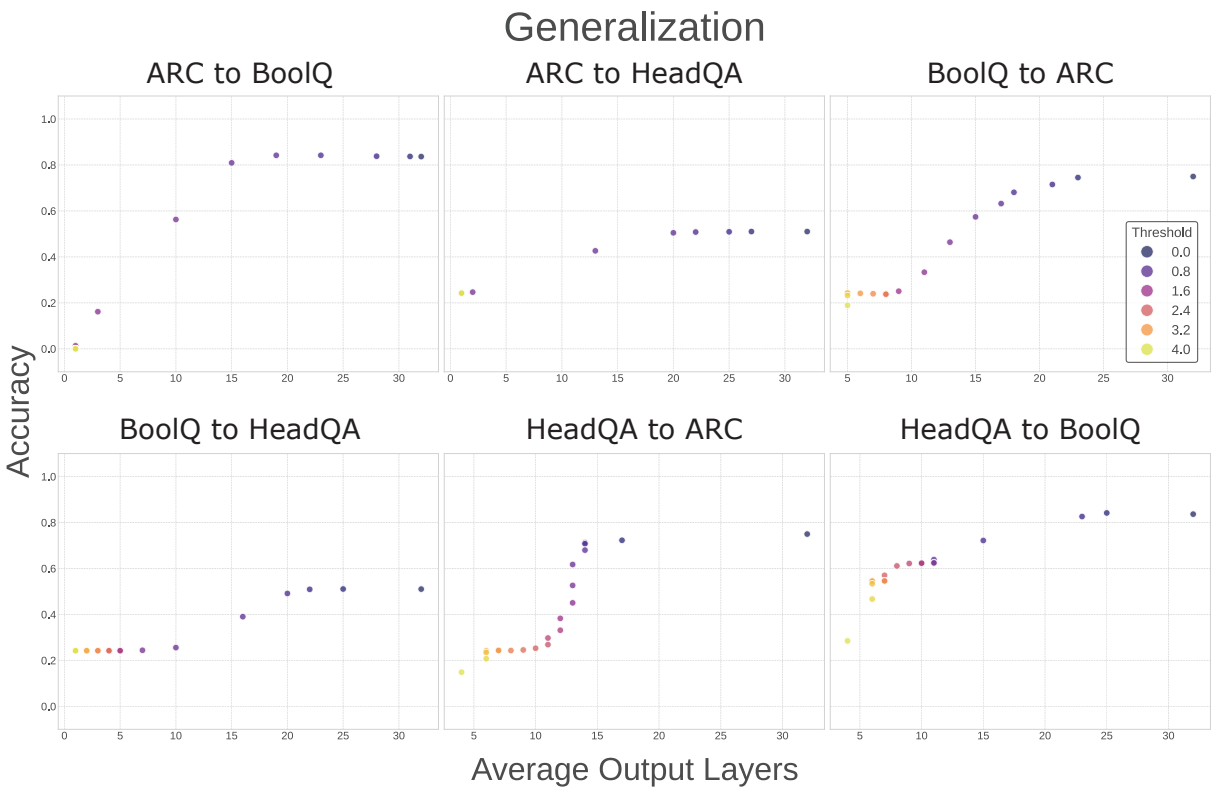

Figure 16: Cross-task generalization of SimExit with Linear SimLens on Vicuna-7B.