LUCiD: Mitigating Hallucinations in Large Language Models with Learned Unilayer Contrastive Decoding

Anonymous authors
Paper under double-blind review

Abstract

Large language models remain prone to hallucination, generating fluent but factually wrong or unsupported content. This is often traced to models falling back on pretraining priors instead of grounding each token in the available context, a failure that is costly in reasoning and knowledge-intensive tasks. Contrastive decoding methods such as DoLa improve reasoning and factuality by contrasting a model's final-layer distribution against a premature earlier-layer distribution. They are training-free, so the premature signal is taken directly from an untrained early exit and the contrast strength is fixed across tokens, which is brittle and often lowers accuracy on tasks where earlier layers are already well calibrated. We introduce **LUCiD** (**L**earned **U**nilayer **C**ontrast**i**ve **D**ecoding), a learned contrastive decoding method. **LUCiD** attaches one lightweight probe to a single middle layer and reuses the model's own frozen final RMSNorm and LM head to extract a premature distribution. This distribution is contrasted against the final layer, scaled by a per-token gate predicted from the last hidden state, under an adaptive plausibility constraint. Across five models from 0.6B to 12B and ten benchmarks, under both greedy decoding and sampling, **LUCiD** outperforms vanilla decoding, DoLa and SLED, in most scenarios.

1 Introduction

Large language models (LLMs) remain prone to hallucination: they produce fluent, confident text that is factually wrong or unsupported by the input (Ji et al., 2023; Huang et al., 2025). Because such outputs are surface-level plausible yet false, they are harder to catch than overt failures, and in high-stakes domains such as medicine and law they carry real consequences (Pal et al., 2023; Magesh et al., 2025). A leading explanation is that these errors arise less from the prompt and more from the model reverting to the statistical associations and priors it absorbed during pretraining, rather than remaining grounded in the knowledge available at inference (McKenna et al.; Longpre et al., 2021; Huang et al., 2025).

A range of methods target this failure. Retrieval-augmented generation grounds outputs in external documents (Lewis et al., 2020). Alignment tuning methods move the model's parameters toward truthful behavior (Ouyang et al., 2022). Post-hoc approaches verify or revise generations after decoding (Dhuliawala et al., 2024; Madaan et al., 2023), while activation-level interventions steer internal representations toward truthful directions (Li et al., 2023a).

Contrastive decoding (CD) is an inference-time alternative that amplifies the disagreement between two next-token distributions to suppress unfaithful predictions. One line contrasts the model against a separate, weaker model (Li et al., 2023b). A second contrasts it against a deliberately degraded version of the input (Shi et al., 2024b; Zhang et al., 2025b; Leng et al., 2024; Wang et al., 2024a; Chen et al., 2024; Woo et al., 2024). A third contrasts against the model's own early-exit predictions (Gera et al., 2023; Chuang et al., 2024; Zhang et al., 2024; Zhou et al., 2025; Zhang et al., 2025a). Other methods draw the contrast from elsewhere in the model: SCMoE from the unchosen experts of a mixture-of-experts model (Shi et al., 2024a), and PruneCD from a pruned copy of the model (Yu et al., 2025).

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

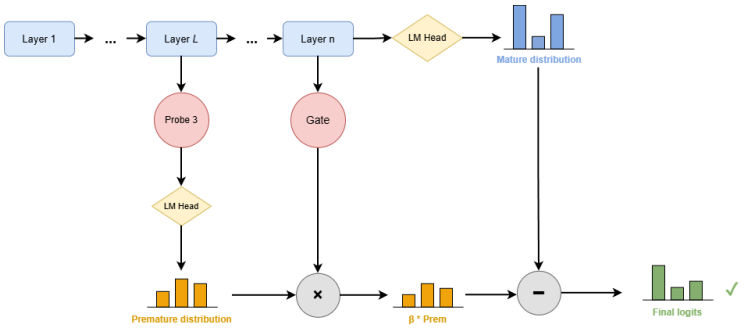


Figure 1: Overview of **LUCiD**. The frozen backbone produces hidden states at every layer. The hidden state at a single middle layer is passed through a lightweight residual probe and then the model's own frozen final RMSNorm and LM head, yielding a premature next-token distribution. This distribution is scaled by a learned gate $\beta \in (0, 1)$ predicted by an MLP on the final hidden state, subtracted from the final-layer distribution, and filtered by an adaptive plausibility constraint (APC) to produce the output distribution. Only the probe and the gate are trained. The backbone, RMSNorm, and LM head stay frozen.

Across the layer-based line, the premature distribution is read directly off the frozen backbone: an early exit, a fusion of early exits, or a pruned copy. Nothing about the distribution itself is trained, and the strength of the contrast is the same for every token. Both choices are brittle, because subtracting an early layer that is already well calibrated lowers accuracy, and a fixed contrast weight applies the same correction to every token regardless of whether it needs one. Learned variants of these methods only decide which layer to read from, or when to apply the contrast at all, leaving the premature distribution untouched. Thus, we introduce **LUCiD** (**L**earned **U**nilayer **C**ontrast**i**ve **D**ecoding), which learns the contrastive distribution itself. **LUCiD** attaches a single lightweight probe to one middle transformer layer and reuses the model's own frozen final RMSNorm and LM head to extract a premature distribution. This distribution is contrasted against the final layer, scaled by a learned gate that sets the contrast strength for each token, under an adaptive plausibility constraint. Only the probe and the gate are trained, while the LLM backbone stays frozen.

Our contributions are as follows:

- We make the layer-contrast distribution trainable: **LUCiD** learns a premature head over a frozen backbone, whereas prior layer-based CD reads the premature distribution directly off the backbone and learns at most which layer to read or when to contrast.
- We introduce a gated contrast that modulates the contrast strength for each token through a learned gate, under an adaptive plausibility constraint. The probe and gate are the only trained parameters, leaving the base model, RMSNorm, and LM head frozen.
- We evaluate across five models (0.6B-12B) under greedy decoding and sampling against Vanilla, DoLa, and SLED, showing that **LUCiD** leads on grounded faithfulness.

2 Related Work

Contrast against a separate model. One line contrasts the model against a distinct, weaker model, letting the disagreement cancel out what that model favors. The original formulation contrasts with a smaller amateur model (Li et al., 2023b).

Contrast against a degraded input. A second line keeps the same model but feeds it a deliberately weakened version of its own input, so the contrast removes input- or prior-driven errors. This has been done by stripping the input context (Shi et al., 2024b), running a pass in which hallucinations are deliberately induced (Zhang et al., 2025b), and perturbing visual or instruction inputs in vision-language models (Leng et al., 2024; Wang et al., 2024a; Chen et al., 2024; Woo et al., 2024).

Contrast against early-exit predictions. A third line contrasts the model against its own early-exit predictions. Autocontrastive decoding uses lower-layer predictions to flag tokens worth avoiding (Gera et al., 2023). DoLa selects a premature layer by Jensen-Shannon divergence and contrasts it

Algorithm 1 LUCiD decoding step

function LUCID($x_{<t}$)
  $h_t^{(1)}, \dots, h_t^{(L)} \leftarrow$ FORWARD($x_{<t}$) ▷ frozen backbone
  $q_t \leftarrow \mathrm{softmax}\left(\phi\left(h_t^{(L)}\right)\right)$ ▷ mature distribution
  $\tilde{p}_t \leftarrow \mathrm{softmax}\left(\phi\left(g\left(h_t^{(\ell^*)}\right)\right)\right)$ ▷ probed middle layer ℓ^*
  $\beta_t \leftarrow \sigma\left(W_2\,\mathrm{GELU}\left(W_1\,h_t^{(L)}\right)\right)$ ▷ gate reads the final layer
  $\mathcal{V}_{\text{head}} \leftarrow \{x \in \mathcal{V} : q_t(x) \geq \alpha \max_w q_t(w)\}$ ▷ APC
  for all $x \in \mathcal{V}$ **do**
    if $x \in \mathcal{V}_{\text{head}}$ **then**
      $s_t(x) \leftarrow \log q_t(x) - \beta_t \log \tilde{p}_t(x)$
    else
      $s_t(x) \leftarrow -\infty$
    end if
  end for
  return s_t
end function

against the final layer (Chuang et al., 2024). SLED fuses information across many layers rather than one (Zhang et al., 2024). ALW trains a predictor to select the contrast layer (Zhou et al., 2025). ActLCD decides when to contrast through a sequence-level reinforcement-learning policy (Zhang et al., 2025a).

Other contrastive signals. Finally, some methods draw the contrast from elsewhere in the model. SCMoE uses the unchosen experts of a mixture-of-experts model (Shi et al., 2024a), while PruneCD contrasts against a pruned copy of the same model instead of an earlier layer (Yu et al., 2025).

3 METHOD

3.1 CONTRASTIVE DECODING

Contrastive decoding was introduced to improve open-ended generation by contrasting a strong *expert* model against a weaker *amateur* model (Li et al., 2023b). Given a prefix $x_{<t}$, the next-token score is the difference of their log-probabilities,

$$\log p_{\text{exp}}(x \mid x_{<t}) - \log p_{\text{ama}}(x \mid x_{<t}), \tag{1}$$

so that generic, degenerate continuations favored by the amateur are suppressed and content the expert adds is amplified. The next token is drawn from the renormalized contrasted distribution, restricted to a plausible subset of the vocabulary, as described in Section 3.3.

3.2 SELF-CONTRASTIVE DECODING

Running a separate amateur model is costly. Self-contrastive decoding removes it, drawing both distributions from a single model by contrasting its deeper layers against its shallower ones (Gera et al., 2023). Let a transformer with L layers produce a hidden state $h_t^{(\ell)} \in \mathbb{R}^d$ at each layer ℓ, read out through the model's shared, frozen head

$$\phi(h) = W_o\,\mathrm{RMSNorm}(h). \tag{2}$$

Applying ϕ to the final layer gives the *mature* distribution $q_t = \mathrm{softmax}(\phi(h_t^{(L)}))$. Alternatively, applying ϕ to an earlier layer ℓ gives a *premature* distribution $p_t^{(\ell)} = \mathrm{softmax}(\phi(h_t^{(\ell)}))$. DoLa instantiates this by dynamically selecting the premature layer M from a candidate set C as the one most divergent from the mature layer, then contrasting the two (Chuang et al., 2024).

3.3 Adaptive Plausibility Constraint

Subtracting log-probabilities can misbehave at the tails. A token can score high after the contrast for the wrong reason: not because the mature layer supports it, but because its premature probability was near zero, where the log difference is unstable (Chuang et al., 2024). Symmetrically, when the mature layer is already near-certain, the contrast barely moves and can drown out the answer it should keep. The adaptive plausibility constraint of Li et al. (Li et al., 2023b) addresses both by evaluating the contrast only over tokens the mature model already ranks near the top,

$$\mathcal{V}_{\text{head}} = \Big\{ x \in \mathcal{V} : q_t(x) \geq \alpha \max_{w \in \mathcal{V}} q_t(w) \Big\}, \tag{3}$$

where $\alpha \in (0, 1)$ sets the threshold. The contrast score is evaluated on $\mathcal{V}_{\text{head}}$ and set to $-\infty$ elsewhere.

3.4 **LUCiD**: Learning the Contrastive Signal

LUCiD keeps the self-contrastive setup but replaces the fixed premature signal with a learned one. It attaches a single lightweight probe to one middle layer $\ell^\star$, reads the probe out through the same frozen head, and contrasts the resulting distribution against the mature layer under a learned gate. Only the probe and the gate are trained. The backbone, RMSNorm, and head stay frozen. The two learned components are shown in Figure 2.

Premature probe. The probed layer is fixed at $\ell^\star = \lfloor L/2 \rfloor$ and equipped with a residual bottleneck probe (Figure 2a)

$$g(h) = h + W^{\uparrow} \,\text{GELU}\Big(W^{\downarrow} h\Big). \tag{4}$$

The up-projection $W^{\uparrow}$ is zero-initialized, so at the start of training g is the identity and the probe reproduces the layer's early-exit logits. Training then moves it away from the identity. The probe yields a learned premature distribution

$$\tilde{p}_t(\cdot) = \text{softmax}\Big(\phi\Big(g\Big(h_t^{(\ell^\star)}\Big)\Big)\Big). \tag{5}$$

Contrast gate. A gate sets how strongly this premature signal is subtracted. A two-layer MLP on the final hidden state (Figure 2b) outputs a scalar

$$\beta_t = \sigma\Big(W_2 \,\text{GELU}\Big(W_1 \, h_t^{(L)}\Big)\Big) \in (0, 1), \tag{6}$$

whose output layer W_2 is zero-initialized, so $\beta_t = 0.5$ at the start of training. The final score contrasts the mature distribution against the gated premature distribution, under the constraint of Eq. equation 3:

$$s_t(x) = \begin{cases} \log q_t(x) - \beta_t \log \tilde{p}_t(x), & x \in \mathcal{V}_{\text{head}}, \\ -\infty, & \text{otherwise}, \end{cases} \tag{7}$$

and the next token is drawn from $\text{softmax}(s_t)$. At initialization the probe is the identity and $\beta_t = 0.5$, so **LUCiD** begins from a half-strength early-exit contrast at layer $\ell^\star$ and learns the probe and gate from there.

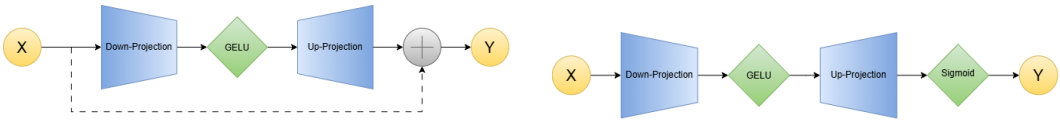
Training. The backbone, RMSNorm, and head are frozen. The only trainable parameters are the probe g and the gate MLP. We minimize the negative log-likelihood of the gold token under the contrasted distribution,

$$\mathcal{L} = -\log \big[\text{softmax}(s_t)\big]_{x_t^\star}, \tag{8}$$

computed on the answer tokens of each training example, so the probe and gate learn a contrast that raises the probability of the correct continuation.

4 Experimental Setup

Models. We apply **LUCiD** to five models spanning 0.6B to 12B parameters: Mistral-Nemo-12B (Jiang et al., 2023), LLaMA3-8B and LLaMA3.2-3B (Grattafiori et al., 2024), and Qwen3-4B and



(a) Premature residual probe. (b) Contrast gate.

Figure 2: Components of **LUCiD**. (a) A single middle layer gets a residual bottleneck probe (down-projection to rank r, GELU, up-projection back to hidden size, plus a skip connection). The up-projection is zero-initialized, so the probe starts as the identity and its premature distribution begins as the layer's early-exit logits. (b) The gate is a two-layer MLP on the final hidden state that outputs a scalar $\beta \in (0, 1)$, setting how strongly the premature distribution is subtracted from the final-layer distribution. Its output layer is zero-initialized, so β starts at 0.5.

Qwen3-0.6B (Yang et al., 2025). For every model the backbone, final RMSNorm, and LM head are frozen. Only the probe and the gate are trained.

Training data. The probe and gate are trained on a small structured mixture: MetaMathQA (Yu et al., 2023) for mathematics, and the ARC-Challenge train split (Clark et al., 2018) and SciQ (Welbl et al., 2017) for science. Supervision is on the answer only, the final numeric or symbolic answer for math and the option letter for multiple choice, never the intermediate reasoning.

Training details. The probe is attached to layer $\ell^\star = \lfloor L/2 \rfloor$ and is a residual bottleneck of rank $r = 1024$. The gate is a two-layer MLP with a hidden width of 256. These are the only trainable parameters. We optimize the negative log-likelihood of the gold answer tokens under the contrasted distribution with AdamW in bfloat16, and set the adaptive plausibility constraint threshold to $\alpha = 0.1$.

Baselines. We compare against standard decoding, DoLa (Chuang et al., 2024), and SLED (Zhang et al., 2024), under both greedy decoding and sampling.

Decoding regimes. All methods are evaluated under two regimes: greedy decoding, and sampling with temperature 0.7 (Chuang et al., 2024) and top-p 0.9 (Holtzman et al., 2019).

Benchmarks and metrics. Our headline benchmark is SQuAD (Rajpurkar et al., 2016), scored by token-level F1 against the reference spans. SQuAD requires answers to be grounded in a provided passage, which makes it a direct test of factuality. It is also out-of-distribution with respect to our training mix (MetaMathQA, ARC-Challenge train, SciQ), covering a domain and task format not seen during training.

General capability benchmarks. Beyond SQuAD, we evaluate on nine further benchmarks. For mathematics we use GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021), scored by exact match on the final answer. The remaining tasks are multiple choice, scored by accuracy: ARC-Challenge (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), MMLU-Pro (Wang et al., 2024b), GPQA-Diamond (Rein et al., 2023), ANLI-R1 (Nie et al., 2020), CommonsenseQA (Talmor et al., 2019), and HellaSwag (Zellers et al., 2019). We report each benchmark's relation to the training mix. GSM8K and ARC-Challenge are in-distribution (ID): GSM8K shares MetaMathQA's domain and ARC-Challenge is evaluated on the held-out test split of a training source. MATH-500 and OpenBookQA are near-distribution (ND), matching the math and science domains seen in training but drawn from unseen datasets. The remaining five benchmarks are out-of-distribution (OOD), covering domains and tasks not seen during training. The purpose of this suite is to check that the faithfulness gains reported in the main body do not come at the cost of general capability. We therefore read these tables as a retention check rather than as a second headline result.

5 Results

5.1 Factuality Results

Table 1 reports F1 score under sampling and greedy decoding for all five models. **LUCiD** is the strongest method on four of the five models under both decoding regimes, reaching 0.797 on Mistral-12B and 0.872 on Qwen-4B under greedy decoding, and 0.786 and 0.872 respectively

Table 1: SQuAD F1 for Vanilla, DoLa, SLED, and **LUCiD**, under sampling (temperature 0.7, top-p 0.9) and greedy decoding. SQuAD measures factuality, since answers must be grounded in the passage, and is out-of-distribution with respect to the training mix (MetaMathQA, ARC-Challenge train, SciQ).

	Sampling				**Greedy**			
Model	Vanilla	DoLa	SLED	**LUCiD**	Vanilla	DoLa	SLED	**LUCiD**
Mistral-12B	0.780	0.765	0.759	**0.786**	0.793	0.767	0.787	**0.797**
LLaMA-8B	0.696	0.778	0.677	**0.808**	0.711	0.749	0.713	**0.815**
LLaMA-3B	**0.791**	0.776	0.766	0.771	**0.821**	0.781	0.813	0.786
Qwen-4B	0.870	0.869	0.868	**0.872**	0.868	0.869	0.869	**0.872**
Qwen-0.6B	0.075	0.067	0.053	**0.076**	**0.078**	0.067	0.056	**0.078**

Table 2: Sampling accuracy (temperature 0.7, top-p 0.9) for Vanilla, DoLa, SLED, and **LUCiD** on the extended benchmark suite. Superscripts mark each benchmark's relation to the training mix (MetaMathQA, ARC-Challenge train, SciQ): $^{\text{ID}}$ in-distribution, $^{\text{ND}}$ near-distribution, $^{\text{OOD}}$ out-of-distribution.

Method	ARC$^{\text{ID}}$	OBQA$^{\text{ND}}$	GPQA$^{\text{OOD}}$	MMLU-P$^{\text{OOD}}$	GSM8K$^{\text{ID}}$	MATH$^{\text{ND}}$	ANLI$^{\text{OOD}}$	CSQA$^{\text{OOD}}$	HSwag$^{\text{OOD}}$
Mistral-12B									
Vanilla	**0.807**	**0.783**	0.342	0.320	0.845	0.319	0.402	0.700	0.564
DoLa	0.740	0.709	**0.358**	0.292	0.826	0.356	0.400	0.665	0.701
SLED	0.790	0.774	0.352	0.301	0.711	0.125	0.407	0.678	**0.730**
LUCiD	0.787	0.756	0.135	**0.321**	**0.862**	**0.384**	**0.412**	**0.703**	0.702
LLaMA-8B									
Vanilla	0.793	**0.766**	0.383	0.339	**0.817**	**0.259**	**0.394**	**0.751**	0.509
DoLa	0.788	0.741	**0.513**	0.338	0.800	0.232	0.386	0.739	0.604
SLED	0.783	0.733	0.332	0.330	0.656	0.127	0.376	0.740	**0.682**
LUCiD	**0.798**	0.760	0.358	**0.341**	0.810	0.246	0.389	0.748	0.604
LLaMA-3B									
Vanilla	**0.719**	0.715	0.342	0.253	0.794	**0.402**	0.345	0.692	0.479
DoLa	0.701	0.685	**0.663**	0.229	0.741	0.317	**0.369**	0.685	0.584
SLED	0.670	0.683	0.301	0.245	0.448	0.111	0.357	0.669	**0.623**
LUCiD	0.710	**0.725**	0.337	**0.262**	**0.807**	0.398	0.345	**0.716**	0.582
Qwen-4B									
Vanilla	0.889	0.832	0.264	0.437	**0.935**	**0.729**	**0.635**	0.764	0.485
DoLa	0.890	**0.834**	**0.715**	**0.439**	**0.935**	0.727	0.614	0.759	0.584
SLED	**0.891**	0.826	0.264	**0.439**	0.932	0.699	0.632	0.764	**0.618**
LUCiD	0.889	0.822	0.275	0.431	**0.935**	0.723	0.631	**0.766**	0.586
Qwen-0.6B									
Vanilla	0.339	0.448	0.093	0.178	0.708	**0.487**	0.333	**0.443**	0.350
DoLa	0.318	0.410	**0.285**	0.159	0.668	0.366	**0.339**	0.433	0.396
SLED	**0.349**	**0.459**	0.114	0.176	0.645	0.353	0.332	**0.443**	**0.414**
LUCiD	0.325	0.406	0.223	**0.180**	**0.726**	0.485	0.333	0.447	0.397

under sampling. The largest gain is on LLaMA-8B, where it raises F1 from 0.696 to 0.808 under sampling and from 0.711 to 0.815 under greedy decoding, well beyond either contrastive baseline. LLaMA-3B is the exception: standard decoding remains strongest there, and **LUCiD** places behind it under both regimes.

5.2 General Capability Retention

Capability retention. Tables 2 and 3 show that the factuality gains do not come at a broad capability cost. On the knowledge and commonsense benchmarks **LUCiD** stays within about one point of standard decoding on most model and benchmark pairs, and it takes the top cell on CommonsenseQA for four of five models under sampling. The clearest positive movement is on mathematics, where **LUCiD** leads or ties GSM8K on four of five models under both decoding regimes, raising GSM8K on Mistral-12B from 0.845 to 0.862 under sampling and MATH-500 on Qwen-0.6B from 0.446 to 0.479 under greedy decoding. ANLI-R1 is near-flat for every method, and HellaSwag separates the contrastive methods from standard decoding in both directions depending on the considered model.

Table 3: Greedy decoding accuracy for Vanilla, DoLa, SLED, and **LUCiD** on the extended benchmark suite. Superscripts mark each benchmark's relation to the training mix (MetaMathQA, ARC-Challenge train, SciQ): $^{\text{ID}}$ in-distribution, $^{\text{ND}}$ near-distribution, $^{\text{OOD}}$ out-of-distribution.

Method	ARC$^{\text{ID}}$	OBQA$^{\text{ND}}$	GPQA$^{\text{OOD}}$	MMLU-P$^{\text{OOD}}$	GSM8K$^{\text{ID}}$	MATH$^{\text{ND}}$	ANLI$^{\text{OOD}}$	CSQA$^{\text{OOD}}$	HSwag$^{\text{OOD}}$
Mistral-12B									
Vanilla	0.822	0.804	0.389	0.339	0.845	**0.406**	0.421	**0.716**	**0.804**
DoLa	0.743	0.703	0.238	0.297	0.844	0.309	**0.425**	0.659	0.702
SLED	**0.823**	**0.806**	**0.399**	**0.342**	0.760	0.194	0.421	0.714	0.732
LUCiD	0.792	0.729	0.140	0.341	**0.848**	0.400	0.425	0.715	0.703
LLaMA-8B									
Vanilla	0.800	**0.762**	0.425	0.356	0.818	0.277	0.377	0.762	**0.741**
DoLa	0.799	0.741	**0.513**	0.349	0.802	0.265	**0.399**	0.742	0.604
SLED	**0.804**	0.760	0.409	**0.359**	0.702	0.131	0.377	0.762	0.685
LUCiD	0.802	0.762	0.425	0.358	**0.812**	**0.289**	0.377	**0.765**	0.605
LLaMA-3B									
Vanilla	0.742	**0.762**	0.409	0.299	0.811	0.406	0.335	0.728	**0.701**
DoLa	0.719	0.705	**0.767**	0.227	0.737	0.327	**0.359**	0.685	0.584
SLED	0.740	0.756	0.415	**0.302**	0.578	0.162	0.335	0.727	0.624
LUCiD	**0.741**	0.758	0.446	0.300	**0.814**	**0.414**	0.335	**0.727**	0.584
Qwen-4B									
Vanilla	0.889	0.824	0.244	0.438	**0.941**	**0.731**	0.632	**0.766**	**0.673**
DoLa	**0.891**	**0.834**	**0.427**	0.437	**0.941**	0.713	0.614	0.758	0.584
SLED	0.889	0.824	0.238	**0.439**	0.940	0.709	**0.635**	**0.766**	0.619
LUCiD	0.888	0.830	0.233	0.431	0.932	0.717	0.634	0.765	0.586
Qwen-0.6B									
Vanilla	0.347	**0.467**	0.067	**0.186**	0.705	0.446	**0.333**	0.460	**0.448**
DoLa	0.290	0.436	**0.187**	0.158	0.661	0.349	0.332	0.430	0.397
SLED	**0.353**	**0.467**	0.073	0.185	0.680	0.414	**0.333**	**0.463**	0.414
LUCiD	0.294	0.444	0.067	0.181	**0.716**	**0.479**	**0.333**	0.442	0.399

Table 4: Probe-depth ablation on SQuAD, under sampling (temperature 0.7, top-p 0.9) and greedy decoding. Each ℓ_i column trains its own probe and gate at that depth. Probed layers are 5/10/15/20/25/30/35 for Mistral-12B, 4/8/12/16/20/24/28 for LLaMA-8B, 4/7/11/14/18/21/25 for LLaMA-3B and Qwen-0.6B, and 5/9/14/18/23/27/32 for Qwen-4B. In every model $\ell_4 = \lfloor L/2 \rfloor$, the depth used by **LUCiD** throughout the paper.

	Sampling							**Greedy**						
Model	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ_7	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ_7
Mistral-12B	0.787	0.787	0.786	0.786	**0.789**	0.785	0.784	0.794	0.796	0.795	**0.797**	0.795	0.795	0.796
LLaMA-8B	0.749	0.755	0.763	**0.808**	0.793	0.776	0.765	0.764	0.765	0.775	**0.815**	0.808	0.785	0.777
LLaMA-3B	0.774	0.782	0.766	0.771	0.776	**0.794**	0.775	0.788	0.799	0.781	0.786	0.791	**0.815**	0.789
Qwen-4B	0.871	0.872	0.871	0.872	0.872	0.873	**0.874**	0.871	0.872	0.872	0.872	0.873	**0.874**	0.873
Qwen-0.6B	0.075	0.075	**0.076**	**0.076**	0.075	0.075	**0.076**	0.078	0.078	**0.079**	0.078	0.078	0.077	0.078

6 Ablation Studies

6.1 Layer Selection

LUCiD places its probe at a single fixed depth, $\ell^\star = \lfloor L/2 \rfloor$. We ablate that choice by sweeping seven depths per model, training one residual probe at each. Every run trains its own probe and its own contrast gate β_t from scratch, over the same frozen RMSNorm and LM head. Table 4 reports greedy and sampling results.

On Mistral-12B, Qwen-4B, and Qwen-0.6B the spread across all seven depths is at most a few thousandths, so any mid-network choice performs alike. LLaMA-8B is the exception in the other direction: the middle layer is clearly the best one, and moving one step either way costs one to four points. The one model where the fixed middle depth is not the best available choice is LLaMA-3B, where ℓ_6 reaches 0.794 under sampling and 0.815 under greedy decoding against 0.771 and 0.786 at ℓ_4. This accounts for the LLaMA-3B row in Table 1, where **LUCiD** trails standard decoding.

Table 5: SQuAD F1. Vanilla and **LUCiD** are absolute; Head-only, Gate-only and Probe-only are reported as differences against Vanilla. Head-only: frozen RMSNorm + LM head at $\lfloor L/2 \rfloor$, no probe, no gate ($\beta = 1$). Gate-only: frozen RMSNorm + LM head at $\lfloor L/2 \rfloor$ with a trained gate but no probe. Probe-only: trained probe at $\lfloor L/2 \rfloor$ with no gate ($\beta = 1$). **LUCiD**: trained probe and gate at the same layer.

	Sampling					Greedy				
Model	Vanilla	Head	Gate	Probe	**LUCiD**	Vanilla	Head	Gate	Probe	**LUCiD**
Mistral-12B	0.780	−0.029	+0.005	−0.002	0.786	0.793	−0.041	+0.003	−0.014	0.797
LLaMA-8B	0.696	+0.083	+0.085	+0.119	0.808	0.711	+0.069	+0.080	+0.106	0.815
LLaMA-3B	0.791	−0.001	−0.023	−0.016	0.771	0.821	−0.026	−0.040	−0.045	0.786
Qwen-4B	0.870	+0.006	+0.002	+0.005	0.872	0.868	+0.008	+0.003	+0.006	0.872
Qwen-0.6B	0.075	−0.011	+0.001	−0.033	0.076	0.078	−0.014	+0.000	−0.035	0.078

Table 6: Multi-layer variants against **LUCiD** on SQuAD F1, under sampling (temperature 0.7, top-p 0.9) and greedy decoding. Both variants attach a probe to every layer except the last. Mean-pool averages the premature log-probabilities uniformly; weighted-pool mixes them with per-token weights from a small per-layer MLP, softmaxed across layers. **LUCiD** probes the single layer $\lfloor L/2 \rfloor$.

	Sampling				**Greedy**			
Model	Vanilla	Mean-pool	Weighted-pool	**LUCiD**	Vanilla	Mean-pool	Weighted-pool	**LUCiD**
Mistral-12B	0.780	**0.787**	0.781	0.786	0.793	0.795	0.794	**0.797**
LLaMA-8B	0.696	0.752	0.778	**0.808**	0.711	0.776	0.789	**0.815**
LLaMA-3B	**0.791**	0.785	0.752	0.771	**0.821**	0.808	0.764	0.786
Qwen-4B	0.870	**0.874**	0.873	0.872	0.868	**0.874**	**0.874**	0.872
Qwen-0.6B	0.075	0.075	0.075	**0.076**	**0.078**	**0.078**	0.077	**0.078**

6.2 Component Ablation

LUCiD adds two trained components on top of a middle-layer early-exit contrast: the residual probe that forms the premature distribution and the per-token gate that sets the contrast strength. We remove them one at a time at the same depth $\ell^\star = \lfloor L/2 \rfloor$. *Head-only* drops both: the premature distribution is the frozen final RMSNorm and LM head applied to $h_t^{(\ell^\star)}$, contrasted at a fixed $\beta = 1$. *Gate-only* keeps the frozen head and restores the trained gate, isolating what the probe contributes. *Probe-only* does the reverse: the trained probe with a fixed $\beta = 1$, isolating what the gate contributes. Table 5 reports all three against vanilla decoding and full **LUCiD**.

The probe and the gate contribute in different ways: one supplies the gain, the other provides the floor. The probe is what produces the gains: on LLaMA-8B it adds 11.9 points under sampling and 10.6 under greedy, well past what either Head-only or Gate-only reaches, and on Qwen-4B it is the largest of the three deltas even though all of them are small. The gate is what keeps the contrast from hurting. An ungated frozen early exit costs Mistral-12B 2.9 points under sampling and 4.1 under greedy, and probe-only leaves Qwen-0.6B 3.3 to 3.5 points below vanilla, its worst result in the table, while adding the gate alone brings both models back to or above vanilla. Neither component is sufficient on its own. On Qwen-4B all variants sit within a few thousandths of each other, so the learned components neither add to nor spoil an already strong early-exit signal. LLaMA-3B stays the exception noted in Table 1: vanilla is strongest and every contrastive variant sits below it, with Gate-only being the weakest.

6.3 Pooling Across Layers

We train two multi-layer variants that attach a probe to every layer except the last. Mean-pool averages the resulting premature log-probabilities into a single signal. Weighted-pool mixes them instead with per-token weights, one small MLP per layer emitting a scalar logit from that layer's hidden state, softmaxed across layers. Both are contrasted against the mature distribution under the

same gate and adaptive plausibility constraint, and training data, token budgets, and all other settings match Section 4. Table 6 compares them against **LUCiD**.

Neither pooling scheme beats a single middle-placed probe. On Mistral-12B, Qwen-4B, and Qwen-0.6B all three sit within a few thousandths of each other in both decoding regimes. On LLaMA-8B both are clearly worse, 0.752 and 0.778 against 0.808 under sampling, because spreading the contrast across depths dilutes a premature signal that one depth carries almost entirely. Learning the mixture recovers part of that loss but not all of it. The two variants also disagree on LLaMA-3B, the one model where mean-pool leads: weighted-pool drops to 0.752 sampled and 0.764 greedy, below every other variant, so the added flexibility is not reliably an improvement even where pooling helps. Since both train one probe per layer rather than one in total, we adopt the single-layer formulation.

7 Conclusion

We introduced **LUCiD**, a layer-contrastive decoding method that replaces the untrained premature distribution used by prior work with a learned one, drawn from a single probe at one middle layer and subtracted under a per-token gate. Across five models from 0.6B to 12B parameters, under both greedy and sampled decoding, **LUCiD** leads on SQuAD for four of five models, reaching 0.872 on Qwen-4B and 0.797 and 0.786 on Mistral-12B under greedy decoding and sampling respectively, with its largest gain on LLaMA-8B, from 0.711 to 0.815 under greedy decoding.

Future work includes extending training beyond answer-token supervision to full generated sequences, testing whether the learned contrast helps throughout an entire generation rather than only at the answer position, and learning where to place the probe rather than fixing it at the middle layer.

AI use statement

In this work, we used generative AI tools to help implement methods, to assist with writing and editing code, and to improve the readability of the paper. We did not use generative AI tools for any other task with required disclosure. In particular, no research idea, hypothesis, method design, or interpretation of results originated from a generative AI tool.

We have reviewed all AI-assisted work. All AI-assisted code was tested and verified for correctness by the authors, and all AI-assisted text edits were checked to confirm they preserved the intended meaning. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Ethics statement

This work does not involve human subjects, personally identifiable information, or the collection of new data. All experiments use publicly available datasets and models, used in accordance with their licenses and intended use. We report our experimental setup and hyperparameters in full so that our results can be independently verified. We declare no conflicts of interest or external sponsorship that could have influenced this work. We do not anticipate direct harmful applications of our method beyond the risks already present in the pretrained models we build on.

Reproducibility statement

We have taken several steps to make our results reproducible. The main text describes our method in full and states the models, datasets, and evaluation protocol used, including the metrics we report and how they are computed. All datasets and pretrained models used in this work are publicly available. We specify the training and inference configurations, including the relevant hyperparameters, so that our setup can be matched. Taken together, these details are intended to be sufficient for an independent reimplementation of our method and for a faithful reproduction of the experiments we report.

REFERENCES

Zhaorun Chen, Zhuokai Zhao, Hongyin Luo, Huaxiu Yao, Bo Li, and Jiawei Zhou. Halc: Object hallucination reduction via adaptive focal-contrast decoding. *arXiv preprint arXiv:2403.00425*, 2024.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*, volume 2024, pp. 54158–54183, 2024.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pp. 3563–3578, 2024.

Ariel Gera, Roni Friedman, Ofir Arviv, Chulaka Gunasekara, Benjamin Sznajder, Noam Slonim, and Eyal Shnarch. The benefits of bad advice: Autocontrastive decoding across model layers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10406–10420, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The llama 3 herd of models, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13872–13882, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023a.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 12286–12312, 2023b.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7052–7063, 2021.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. Hallucination-free? assessing the reliability of leading ai legal research tools. *Journal of empirical legal studies*, 22(2):216–242, 2025.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. Sources of hallucination by large language models on inference tasks (2023). *arXiv preprint arXiv:2305.14552*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4885–4901, 2020.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Med-halt: Medical domain hallucination test for large language models. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pp. 314–334, 2023.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Chufan Shi, Cheng Yang, Xinyu Zhu, Jiahao Wang, Taiqiang Wu, Siheng Li, Deng Cai, Yujiu Yang, and Yu Meng. Unchosen experts can contribute too: Unleashing moe models' power by self-contrast. *Advances in Neural Information Processing Systems*, 37:136897–136921, 2024a.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 783–791, 2024b.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15840–15853, 2024a.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024b.

Johannes Welbl, Nelson F Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, 2017.

Sangmin Woo, Jaehyuk Jang, Donguk Kim, Yubin Choi, and Changick Kim. Ritual: Random image transformations as a universal anti-hallucination lever in large vision language models. *arXiv preprint arXiv:2405.17821*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen3 technical report, 2025.

Byeongho Yu, Changhun Lee, Jun-gyu Jin, and Eunhyeok Park. Prunecd: Contrasting pruned self model to improve decoding factuality. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32450–32461, 2025.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics.

Hongxiang Zhang, Hao Chen, Muhao Chen, and Tianyi Zhang. Active layer-contrastive decoding reduces hallucination in large language model generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3028–3046, 2025a.

Jianyi Zhang, Da-Cheng Juan, Cyrus Rashtchian, Chun-Sung Ferng, Heinrich Jiang, and Yiran Chen. Sled: Self logits evolution decoding for improving factuality in large language models. *Advances in Neural Information Processing Systems*, 37:5188–5209, 2024.

Yue Zhang, Leyang Cui, Shuming Shi, et al. Alleviating hallucinations of large language models through induced hallucinations. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8218–8232, 2025b.

Yuechi Zhou, Chuyue Zhou, Jianxin Zhang, Juntao Li, and Min Zhang. Alw: Adaptive layer-wise contrastive decoding enhancing reasoning ability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8506–8524, 2025.

A Performance Metrics

Table 7: Inference cost across all five models. Within each block, the baseline row gives absolute values and the remaining rows are relative to it.

Method	Params	Prefill FLOPs	TTFT	FLOP/tok	Tok/s
Mistral-Nemo-Instruct-2407					
Baseline	12.25 G	4.75 T	48.4 ms	23.29 G	21.5
DoLa	+0	1.00×	1.05×	1.58×	0.93×
SLED	+0	1.01×	1.09×	3.31×	0.94×
LUCiD	+11.80 M	1.00×	1.05×	1.06×	0.94×
LUCiD (probed, no gate)	+10.49 M	1.00×	1.03×	1.06×	0.95×
LUCiD (no probe, gated)	+1.31 M	1.00×	1.04×	1.06×	0.94×
LUCiD (no probe, no gate)	+0	1.00×	1.04×	1.06×	0.96×
Meta-Llama-3-8B-Instruct					
Baseline	8.03 G	3.04 T	38.8 ms	15.12 G	26.9
DoLa	+0	1.00×	1.04×	1.56×	0.94×
SLED	+0	1.01×	1.07×	3.22×	0.93×
LUCiD	+9.44 M	1.00×	1.04×	1.07×	0.95×
LUCiD (probed, no gate)	+8.39 M	1.00×	1.04×	1.07×	0.95×
LUCiD (no probe, gated)	+1.05 M	1.00×	1.06×	1.07×	0.95×
LUCiD (no probe, no gate)	+0	1.00×	1.06×	1.07×	0.95×
Llama-3.2-3B-Instruct					
Baseline	3.21 G	1.47 T	35.1 ms	6.50 G	29.8
DoLa	+0	1.01×	1.02×	1.85×	0.97×
SLED	+0	1.01×	1.03×	4.40×	0.95×
LUCiD	+7.08 M	1.00×	1.01×	1.12×	0.96×
LUCiD (probed, no gate)	+6.30 M	1.00×	1.05×	1.12×	0.97×
LUCiD (no probe, gated)	+786.95 K	1.00×	1.03×	1.12×	0.98×
LUCiD (no probe, no gate)	+0	1.00×	1.00×	1.12×	0.98×
Qwen3-4B-Instruct-2507					
Baseline	4.02 G	1.71 T	55.6 ms	8.17 G	18.6
DoLa	+0	1.00×	1.01×	1.86×	0.96×
SLED	+0	1.02×	1.04×	4.43×	0.93×
LUCiD	+5.90 M	1.00×	1.02×	1.10×	0.95×
LUCiD (probed, no gate)	+5.25 M	1.00×	1.00×	1.10×	0.96×
LUCiD (no probe, gated)	+655.87 K	1.00×	1.01×	1.10×	0.96×
LUCiD (no probe, no gate)	+0	1.00×	1.03×	1.10×	0.96×
Qwen3-0.6B					
Baseline	596.05 M	259.14 G	43.1 ms	1.24 G	24.1
DoLa	+0	1.01×	1.03×	2.76×	0.96×
SLED	+0	1.03×	1.03×	8.02×	0.95×
LUCiD	+2.36 M	1.00×	1.05×	1.26×	0.96×
LUCiD (probed, no gate)	+2.10 M	1.00×	1.01×	1.26×	0.97×
LUCiD (no probe, gated)	+262.66 K	1.00×	1.01×	1.25×	0.98×
LUCiD (no probe, no gate)	+0	1.00×	1.00×	1.25×	0.97×

In Table 7 we measure inference cost on a single NVIDIA A100 80GB GPU at batch size 1, across five backbones spanning 0.6B to 12B parameters. Baseline measurements are absolute and every other row is expressed relative to it.

Metrics. *Params* is the total parameter count (added weights for non-baseline rows). *Prefill FLOPs* is the compute to encode the prompt, before any token is generated. *TTFT* (Time-To-First-Token) is the latency from receiving the prompt to emitting the first output token, dominated by prefill and the lowest-latency signal a user perceives. *FLOP/tok* is the compute spent per generated token during decoding. *Tok/s* is decode throughput.

Prefill Performance. **LUCiD** leaves prefill untouched. FLOPs rise by at most 0.1% on every backbone, so prompt encoding is effectively unchanged. The added weights are a negligible slice of the backbone, from 0.10% on Mistral-Nemo to 0.40% on Qwen3-0.6B. TTFT moves by 1–5% (mean 1.03×), which is at the level of run-to-run jitter: the zero-parameter ablation row varies over the same range.

Decoding Performance. **LUCiD** costs 1.06–1.26× baseline FLOP/tok, against 1.56–2.76× for DoLa and 3.22–8.02× for SLED. It is the cheapest of the three on every backbone, and the gap widens as the model shrinks, since a fixed contrastive step is amortized over less backbone compute. Decode throughput holds at 0.95× baseline on average (0.94–0.96×), a slowdown of 1.04–1.06×, on par with DoLa and SLED despite the far lower FLOP count. Peak memory rises by at most 1.2%, against 6.1% for SLED on Qwen3-0.6B.

Per-Component Analysis. Removing both the probe and the gate leaves FLOP/tok within 0.01× of full **LUCiD** on every backbone (1.06 vs 1.06, 1.07 vs 1.07, 1.12 vs 1.12, 1.10 vs 1.10, 1.26 vs 1.25), and throughput within a single point. All of the decode overhead therefore comes from the contrastive step itself, which every variant shares, and none of it from the adapters or the gate. The residual throughput spread across ablations is smaller than the variance between repeated runs, so we do not read an ordering into it.

Summary. Full **LUCiD** is the most complete configuration and carries the same near-free profile as its ablations: unchanged prefill, a TTFT bump inside measurement noise, a 1.04–1.06× decode slowdown, and the lowest FLOP/tok of any contrastive decoding method we compare against. The gate and adapters can be kept in every deployment at no practical cost.

B Extended Single-Layer Ablation

Tables 9 and 8 extend the probe-depth ablation to the remaining nine benchmarks, sweeping seven depths on Mistral-12B, LLaMA-8B, LLaMA-3B, Qwen-4B, and Qwen-0.6B. In each block the row at $\ell = \lfloor L/2 \rfloor$ is the configuration **LUCiD** uses throughout the paper.

No depth wins across all nine benchmarks, and the differences between them are mostly small and unsystematic. The middle depth stays close to the best available configuration on nearly every task, so the fixed choice costs little in general capability. Mistral-12B is the exception, where the mid-network depths collapse on GPQA-Diamond to between 0.140 and 0.187 under greedy decoding while ℓ=5 reaches 0.440, with the same pattern under sampling.

Table 8: Sampling single-layer ablation on the extended benchmark suite (temperature 0.7, top-p 0.9).

Method	ARC^{ID}	OBQAND	GPQA^{OOD}	MMLU-P^{OOD}	GSM8K^{ID}	MATHND	ANLI^{OOD}	CSQA^{OOD}	HSwag^{OOD}
Mistral-12B									
ℓ=5	**0.811**	0.774	0.342	0.320	0.846	0.358	0.394	0.697	**0.702**
ℓ=10	0.802	**0.790**	0.306	0.319	0.842	0.364	0.414	0.693	**0.702**
ℓ=15	0.803	0.760	0.187	0.319	0.855	**0.400**	0.399	0.700	0.701
ℓ=20	0.787	0.756	0.135	0.321	**0.862**	0.384	0.412	0.703	**0.702**
ℓ=25	0.759	0.737	0.192	0.312	0.848	0.384	0.402	0.665	**0.702**
ℓ=30	0.785	0.729	0.207	0.320	0.861	0.360	**0.422**	0.688	**0.702**
ℓ=35	0.800	0.772	0.321	**0.325**	0.855	0.372	0.407	0.694	**0.702**
LLaMA-8B									
ℓ=4	0.795	0.752	**0.409**	0.337	0.805	**0.275**	**0.409**	0.750	0.604
ℓ=8	0.793	**0.766**	**0.409**	0.340	0.803	0.259	0.383	0.754	**0.605**
ℓ=12	0.799	0.756	**0.409**	**0.341**	0.813	0.248	0.382	0.748	0.604
ℓ=16	0.798	0.760	0.358	**0.341**	0.810	0.246	0.389	0.748	0.604
ℓ=20	0.799	0.762	0.389	0.337	0.804	0.253	0.398	0.757	0.604
ℓ=24	0.796	0.764	0.358	0.339	**0.817**	0.246	0.371	**0.759**	0.603
ℓ=28	0.795	0.756	0.383	**0.341**	0.802	0.257	0.384	0.755	0.603
LLaMA-3B									
ℓ=4	0.704	**0.741**	0.295	0.257	0.799	0.404	0.344	0.714	0.581
ℓ=7	0.704	0.715	0.342	0.257	0.791	**0.406**	**0.351**	0.699	**0.584**
ℓ=11	0.722	0.731	0.316	0.258	0.797	0.398	0.349	0.708	0.581
ℓ=14	0.710	0.725	0.337	**0.262**	**0.807**	0.398	0.345	**0.716**	0.582
ℓ=18	0.715	0.711	0.306	0.258	0.798	0.341	0.344	0.710	0.582
ℓ=21	0.716	0.735	**0.368**	0.258	0.801	0.396	0.341	0.706	0.582
ℓ=25	0.711	0.731	0.275	0.258	0.788	0.368	0.348	0.698	**0.584**
Qwen-4B									
ℓ=5	**0.891**	0.820	0.269	0.429	0.938	**0.749**	0.636	0.764	0.583
ℓ=9	0.889	0.828	0.259	**0.431**	0.935	0.723	0.629	0.771	0.583
ℓ=14	0.888	**0.830**	0.259	**0.431**	0.939	0.731	0.629	0.764	0.582
ℓ=18	0.889	0.822	**0.275**	**0.431**	0.935	0.723	0.631	0.766	**0.586**
ℓ=23	0.889	0.826	**0.275**	0.428	0.934	0.723	**0.638**	0.765	0.585
ℓ=27	0.888	0.816	0.218	**0.431**	0.939	0.733	0.633	**0.772**	0.584
ℓ=32	0.889	0.828	0.264	0.430	**0.942**	0.717	0.634	0.768	0.585
Qwen-0.6B									
ℓ=4	0.327	**0.440**	0.166	0.177	0.718	0.453	**0.333**	0.432	0.398
ℓ=7	0.328	0.424	0.228	0.178	0.712	0.455	**0.333**	0.435	0.398
ℓ=11	0.327	**0.440**	0.166	0.178	0.717	0.446	**0.333**	0.439	0.398
ℓ=14	0.325	0.406	0.223	0.180	0.726	**0.485**	**0.333**	**0.447**	0.397
ℓ=18	0.330	0.428	**0.254**	0.178	**0.728**	0.446	**0.333**	0.430	0.398
ℓ=21	**0.335**	0.434	0.197	0.175	0.707	0.459	**0.333**	0.437	0.398
ℓ=25	0.310	0.424	0.218	0.174	0.718	0.473	**0.333**	0.427	0.398

Table 9: Greedy single-layer ablation on the extended benchmark suite.

Method	ARC^{ID}	OBQAND	GPQA^{OOD}	MMLU-P^{OOD}	GSM8K^{ID}	MATHND	ANLI^{OOD}	CSQA^{OOD}	HSwag^{OOD}
Mistral-12B									
ℓ=5	**0.824**	**0.804**	**0.440**	0.338	0.848	0.396	0.422	**0.716**	**0.704**
ℓ=10	0.810	0.802	0.295	0.339	0.851	0.388	0.415	0.715	0.703
ℓ=15	0.806	0.794	0.155	0.339	0.855	**0.408**	0.429	0.711	0.703
ℓ=20	0.792	0.792	0.140	0.341	0.848	0.400	0.425	0.715	0.703
ℓ=25	0.761	0.729	0.187	0.328	0.854	0.390	**0.476**	0.693	0.703
ℓ=30	0.787	0.790	0.150	0.338	0.855	0.372	0.461	0.699	**0.704**
ℓ=35	0.821	0.794	0.249	**0.344**	0.856	0.378	0.432	0.714	0.703
LLaMA-8B									
ℓ=4	0.799	0.762	0.409	**0.358**	0.812	0.275	**0.377**	0.763	**0.605**
ℓ=8	**0.802**	**0.764**	**0.440**	0.357	0.820	0.269	0.376	0.763	**0.605**
ℓ=12	0.801	0.760	0.425	0.357	0.818	0.275	0.376	0.764	**0.605**
ℓ=16	**0.802**	0.762	0.425	**0.358**	0.812	**0.289**	**0.377**	**0.765**	**0.605**
ℓ=20	**0.802**	0.762	0.435	**0.358**	0.809	0.265	**0.377**	**0.765**	**0.605**
ℓ=24	**0.802**	**0.764**	0.430	0.357	0.817	0.265	**0.377**	**0.765**	**0.605**
ℓ=28	0.800	0.760	0.430	0.356	0.813	0.265	**0.377**	0.764	**0.605**
LLaMA-3B									
ℓ=4	0.741	0.762	0.409	0.302	**0.817**	0.380	**0.335**	**0.733**	0.582
ℓ=7	0.742	0.760	0.420	0.304	0.808	0.392	**0.335**	0.731	0.582
ℓ=11	0.743	**0.764**	0.389	**0.305**	0.803	0.390	**0.335**	0.730	0.581
ℓ=14	0.741	0.758	**0.446**	0.300	0.814	**0.414**	**0.335**	0.727	**0.584**
ℓ=18	0.741	0.760	0.409	0.301	0.813	0.404	**0.335**	0.729	**0.584**
ℓ=21	0.743	0.758	**0.446**	0.299	0.804	0.370	**0.335**	0.732	**0.584**
ℓ=25	0.744	0.760	**0.446**	0.298	0.804	0.392	**0.335**	0.730	**0.584**
Qwen-4B									
ℓ=5	**0.889**	0.828	**0.233**	**0.431**	0.935	0.717	**0.635**	**0.766**	0.585
ℓ=9	0.888	0.828	**0.233**	**0.431**	0.933	0.717	0.634	**0.766**	0.585
ℓ=14	0.888	0.828	**0.233**	**0.431**	0.933	0.707	**0.635**	0.765	0.586
ℓ=18	0.888	**0.830**	**0.233**	**0.431**	0.932	0.717	0.634	0.765	0.586
ℓ=23	0.888	**0.830**	**0.233**	**0.431**	0.933	0.711	0.634	0.765	0.586
ℓ=27	**0.889**	0.824	0.228	0.430	**0.936**	0.723	**0.635**	**0.766**	**0.587**
ℓ=32	0.888	0.826	0.228	**0.431**	0.935	**0.725**	**0.635**	0.764	**0.587**
Qwen-0.6B									
ℓ=4	**0.294**	**0.446**	**0.073**	**0.181**	0.712	0.469	**0.333**	**0.442**	0.399
ℓ=7	**0.294**	0.444	**0.073**	0.180	**0.716**	0.475	**0.333**	**0.442**	0.399
ℓ=11	**0.294**	**0.446**	0.067	**0.181**	**0.716**	0.479	**0.333**	**0.442**	0.399
ℓ=14	**0.294**	0.444	0.067	0.180	0.712	**0.503**	**0.333**	**0.442**	0.399
ℓ=18	**0.294**	**0.446**	0.067	**0.181**	0.712	0.483	**0.333**	**0.442**	**0.400**
ℓ=21	**0.294**	**0.446**	0.067	**0.181**	0.709	0.485	**0.333**	**0.442**	0.398
ℓ=25	**0.294**	**0.446**	0.067	**0.181**	0.704	0.457	**0.333**	**0.442**	0.398