What Tokens Truly Matter? The Logit Conflation Problem in LLM Sampling

Pinlong Zhao[1], Huijun Tang[2], Pengfei Jiao[1], Mengyang Li[3]*
[1]School of Cyberspace, Hangzhou Dianzi University
[2]Department of Engineering, Durham University
[3]Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University

pinlongzhao@hdu.edu.cn, huijun.tang@durham.ac.uk.
pjiao@hdu.edu.cn, limengyang@tjnu.edu.cn

Abstract

Sampling methods for large language models select candidate tokens based on logit statistics, implicitly assuming that high logits indicate desirable outputs. We identify the *Logit Conflation Problem*, where a token's logit aggregates prompt-independent factors, including linguistic fluency and parametric associations, with prompt-relevance. However, only prompt-relevance determines instruction-following quality. We propose SEAL-Sampling (**S**ignal **E**xtraction for **A**ctive Re**L**evance) to isolate this component through attention-weighted attribution. Our framework defines prompt-relevance as the causal effect of prompt content on token logits and establishes attention patterns as an efficient proxy. Experiments on LLaMA-3 demonstrate significant improvements over top-$n\sigma$, with gains of 1.8% on AlpacaEval 2.0 and 2.2% on IFEval. Furthermore, attribution scores correlate weakly with raw logits, confirming the extraction of an orthogonal signal. The method is training-free and introduces minimal latency, adding less than 12ms overhead per token.

1 Introduction

Large language models (LLMs) generate text by sampling tokens from probability distributions derived from output logits. The sampling strategy critically determines output quality, as it decides which tokens are considered valid candidates (Holtzman et al., 2019). Existing methods, including top-k (Fan et al., 2018), top-p (Holtzman et al., 2019), min-p (Nguyen et al., 2024), and top-$n\sigma$ (Tang et al., 2025), share a common principle: they filter tokens based on logit magnitudes or derived statistics, under the implicit assumption that high-logit tokens are desirable.

This assumption warrants careful examination for instruction-following tasks. At any given generation step, multiple tokens may receive high logits for fundamentally different reasons: some tokens are statistically favored due to their frequency in natural language, others are activated through parametric associations with nearby context, and still others genuinely address the specific instruction. Standard sampling methods, which filter solely by logit magnitude, cannot distinguish among these cases; although all may pass statistical thresholds, they differ fundamentally in their contribution to task success. Similar phenomena have been observed in prior work on decoding strategies (Su et al., 2022; Dathathri et al., 2019; O'Brien and Lewis, 2023), though the underlying cause remains underexplored.

This observation illustrates what we term the Logit Conflation Problem. A token's logit reflects multiple factors: linguistic naturalness favors common tokens regardless of context (Holtzman et al., 2019); parametric associations activate semantically related tokens (Elhage et al., 2021; Geva et al., 2022); and prompt-relevance increases logits for tokens that address the specific instruction. These factors aggregate into a single scalar, yet they differ fundamentally in their contribution to task success. For instruction-following, only prompt-relevance matters, since a response can be fluent and topically related while completely failing to answer the question (Zhou et al., 2023).

The conflation problem explains an empirical puzzle: despite instruction-tuning (Grattafiori et al., 2024; Ouyang et al., 2022; Wei et al., 2022a), models with fixed sampling thresholds still produce constraint-violating or off-topic outputs. The model has learned to favor correct tokens, meaning prompt-relevance signal exists, but this signal is diluted by stronger prompt-independent components. In constrained generation, this manifests as outputs ignoring explicit requirements (Zhou et al., 2023); in controlled generation, as responses missing required attributes (Novikova et al., 2017).

We propose to explicitly extract the prompt-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 36943–36961
July 2-7, 2026

relevance signal rather than filtering on the conflated aggregate. Our method, SEAL-Sampling, computes an attribution score measuring each token's contribution to prompt-relevance. The key insight is that attention patterns reveal which prompt content the model is currently addressing (Vig and Belinkov, 2019; Abnar and Zuidema, 2020; Clark et al., 2019). Tokens semantically aligned with attended regions are more likely to satisfy the instruction, providing a computationally efficient proxy for prompt-relevance.

Concretely, we construct an attention-weighted prompt representation from the final transformer layer, compute semantic similarity between candidate tokens and this representation, and then derive attribution scores via gradient analysis of a prompt-relevance objective. Tokens with high logits but low attribution are filtered from the sampling nucleus, as they are statistically prominent yet prompt-irrelevant.

Our contributions are threefold:

- We formalize the Logit Conflation Problem and define prompt-relevance as the causal effect of prompt content on token logits. We establish conditions under which attention-weighted attribution isolates this signal (§2).
- We propose SEAL-Sampling, computing attribution through a single matrix-vector operation with $O(kd)$ complexity, adding less than 12ms per token (§3).
- Experiments across instruction-following, controlled generation, and reasoning tasks, on both LLaMA-3 and Qwen2 architectures, demonstrate consistent improvements. Analysis confirms that attribution provides information orthogonal to logit statistics (§4).

2 Theoretical Framework

We formalize prompt-relevance and establish conditions for its efficient extraction. Throughout this section, we use $\mathbf{x}$ to denote the input prompt, $\mathbf{c}$ the generation context (previously generated tokens), and ℓ_i the logit assigned to token i in the vocabulary $\mathcal{V}$. A complete notation table is provided in Appendix J.

2.1 Prompt-Relevance Signal

Let $\ell_i(\mathbf{x}, \mathbf{c})$ denote the logit assigned to token i given prompt $\mathbf{x}$ and generation context $\mathbf{c}$. We decompose this logit into prompt-independent and prompt-induced components.

Definition 1 (Prompt-Relevance Signal)**.** *The prompt-relevance signal for token i is defined as:*

$$s_{\text{rel}}(i; \mathbf{x}, \mathbf{c}) \triangleq \ell_i(\mathbf{x}, \mathbf{c}) - \ell_i(\varnothing, \mathbf{c}) \quad (1)$$

where $\ell_i(\varnothing, \mathbf{c})$ denotes the logit computed with prompt content masked or removed.

We emphasize that Equation 1 is a *definition* rather than a structural assumption on the model's internal computation. It characterizes prompt-relevance as the difference between two logits produced by the model's full (nonlinear) forward pass, analogous to how the Average Treatment Effect in causal inference is defined as a difference in outcomes regardless of the underlying mechanism. No claim is made that the model's internal representations decompose linearly; the subsequent analysis only requires that s_{rel} is well-defined and that our proxy correlates with it.

Given this definition, we obtain an exact arithmetic decomposition of the logit:

$$\ell_i(\mathbf{x}, \mathbf{c}) = \underbrace{\ell_i(\varnothing, \mathbf{c})}_{\text{prompt-independent}} + \underbrace{s_{\text{rel}}(i; \mathbf{x}, \mathbf{c})}_{\text{prompt-induced}} \quad (2)$$

The prompt-independent component $\ell_i(\varnothing, \mathbf{c})$ captures factors unrelated to the specific prompt, including linguistic fluency priors that favor common tokens and contextual associations activated by the generation history. The prompt-relevance signal s_{rel} isolates the causal effect of prompt content, quantifying how much the prompt's presence increases the model's preference for token i.

Proposition 1 (Relevance-Quality Correspondence)**.** *Consider an instruction-following task with quality metric $Q(y|\mathbf{x})$ measuring how well response y satisfies prompt $\mathbf{x}$. Under the following assumptions:*

(A1) *Effective instruction-tuning: The model has learned to assign higher logits to tokens that better address the prompt.*

(A2) *Fluency uniformity: Among tokens in the statistically filtered candidate set $\mathcal{N}_{\text{stat}}$, the prompt-independent quality contribution $\partial Q_{\text{base}}/\partial p_i$ varies within a bounded range $[\xi - \epsilon, \xi + \epsilon]$ for some ξ and small $\epsilon > 0$.*

The expected quality gradient satisfies:

$$\mathbb{E}\left[\frac{\partial Q}{\partial p_i}\right] = \gamma \cdot s_{\text{rel}}(i; \mathbf{x}, \mathbf{c}) + \xi + O(\epsilon) \quad (3)$$

where $\gamma > 0$ is a task-dependent constant.

The proof (Appendix A.1) decomposes the quality metric as $Q = Q_{\text{base}} + Q_{\text{prompt}}$, where Q_{base} measures fluency and Q_{prompt} measures instruction adherence. Assumption (A1) ensures that $\partial Q_{\text{prompt}}/\partial p_i$ correlates positively with $s_{\text{rel}}(i)$. Assumption (A2) is justified empirically by the statistical pre-filtering step, which restricts candidates to grammatically plausible tokens with similar fluency characteristics. To quantify the tightness of (A2) within $\mathcal{N}_{\text{stat}}$, we measured the coefficient of variation (CV) of $\partial Q_{\text{base}}/\partial p_i$ on 500 IFEval examples using a linguistic acceptability predictor as a proxy for Q_{base}. The CV of the fluency contribution is 0.12, whereas the CV of the prompt-relevance contribution is 0.87 on the same set, indicating that within $\mathcal{N}_{\text{stat}}$ fluency is roughly an order of magnitude less variable than relevance. This supports treating ξ as approximately constant while s_{rel} dominates the differentiating signal. We further validate Equation 3 empirically in Appendix F, demonstrating that the linear relationship holds with $R^2 > 0.5$ across diverse benchmarks.

2.2 Attention as Proxy for Prompt-Relevance

Direct computation of s_{rel} via Equation 1 requires an additional forward pass with masked prompts, which is prohibitively expensive during autoregressive generation. We establish that attention patterns provide an efficient proxy under specific conditions.

Proposition 2 (Attention-Relevance Correspondence)**.** *Let $\boldsymbol{\alpha} \in \mathbb{R}^n$ denote attention weights from the current generation position to prompt tokens $\{x_1, \ldots, x_n\}$ at the final transformer layer, with corresponding hidden states $\mathbf{H}_{\text{p}} = [\mathbf{h}_{x_1}, \ldots, \mathbf{h}_{x_n}]^\top \in \mathbb{R}^{n \times d}$. Define the attended prompt representation:*

$$\mathbf{v}_{\text{p}} = \mathbf{H}_{\text{p}}^\top \boldsymbol{\alpha} \in \mathbb{R}^d \tag{4}$$

For token embedding $\mathbf{e}_i \in \mathbb{R}^d$, define the relevance proxy:

$$r_i = \mathbf{e}_i^\top \mathbf{v}_{\text{p}} \tag{5}$$

Under the following condition:

(B1) *Attention-influence alignment: The attention distribution $\boldsymbol{\alpha}$ approximates the causal influence weights $\mathbf{w} \in \mathbb{R}^n$, where w_j measures the contribution of prompt token x_j to the output logit distribution, i.e., $\alpha_j \propto w_j + \delta_j$ with $\|\boldsymbol{\delta}\|_1$ small.*

The proxy satisfies $\text{Corr}(r_i, s_{\text{rel}}(i)) > 0$.

Condition (B1) treats attention as an *efficient proxy* for information routing rather than a strict causal attribution; it requires only that final-layer attention weights correlate with the actual contribution of prompt tokens to the output logits, not that they satisfy formal causal criteria. This weaker requirement is consistent with empirical observations in final transformer layers (Vig and Belinkov, 2019; Elhage et al., 2021), while explicitly acknowledging that attention is not a ground-truth attribution in the mechanistic interpretability sense (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). Intuitively, high attention to prompt token x_j implies the model routes information from x_j to the output; tokens semantically similar to x_j benefit from this information flow and receive increased logits when the prompt is present. We note that the inner product $\mathbf{e}_i^\top \mathbf{v}_{\text{p}}$ is geometrically well-aligned with the model's own logit computation: in LLaMA-3 and related architectures, the output logit takes the form $\ell_i = \mathbf{e}_i^\top \mathbf{h}_{\text{final}} + b_i$, so r_i operates in the same space as the model's logit projection itself. The formal argument appears in Appendix A.2. We validate this correspondence empirically in Appendix F, where r_i achieves 0.60 average correlation with s_{rel}, compared to 0.35 for raw logits.

2.3 Attribution Score

The relevance proxy r_i measures semantic alignment with attended prompt content but does not account for the token's probability mass. A token with high r_i but negligible probability contributes little to generation quality. We derive an attribution score that combines relevance with probability-weighted impact.

Definition 2 (Prompt-Relevance Objective)**.** *Given softmax probabilities $\mathbf{p} = \text{softmax}(\boldsymbol{\ell}/T)$ and relevance scores $\{r_i\}$, define:*

$$\mathcal{L}_{\text{rel}}(\boldsymbol{\ell}) = \sum_{i \in \mathcal{V}} p_i \cdot [r_i]_+ \tag{6}$$

where $[r_i]_+ = \max(0, r_i)$ and $\mathcal{V}$ is the vocabulary.

This objective measures expected prompt-relevance under the current sampling distribution. The rectification $[\cdot]_+$ ensures that tokens with negative relevance, which are semantically opposed to the attended prompt content, contribute zero rather than negative values. We note that this rectification removes tokens *semantically opposed* to the attended content; it does not conflict with the model's

handling of negated instructions (e.g., "do not do X"), where the mentioned token typically has *positive* semantic similarity to the prompt but receives a *low logit* from the instruction-tuned model itself (see discussion in Section 5).

Theorem 1 (Attribution Score Properties)**.** *The gradient-based attribution score:*

$$A_i = \frac{\partial \mathcal{L}_{\text{rel}}}{\partial \ell_i} = p_i(1 - p_i) \cdot [r_i]_+ \tag{7}$$

satisfies the following properties:

(P1) ***Relevance correlation**: Under conditions (A1), (A2), and (B1), we have* $\text{Corr}(A_i, s_{\text{rel}}(i)) > \text{Corr}(\ell_i, s_{\text{rel}}(i))$.

(P2) ***Prompt-independence suppression**: For tokens where* $[r_i]_+ = 0$*, the attribution* $A_i = 0$ *regardless of logit magnitude.*

(P3) ***Concentration prevention**: The factor* $p_i(1 - p_i)$ *attains maximum at* $p_i = 0.5$ *and vanishes as* $p_i \to 0$ *or* $p_i \to 1$.

Proof sketch. The gradient derivation follows from the chain rule. Since $\mathcal{L}_{\text{rel}}$ is linear in $\mathbf{p}$, we have $\partial \mathcal{L}_{\text{rel}} / \partial p_i = [r_i]_+$. The softmax gradient gives $\partial p_i / \partial \ell_i = p_i(1 - p_i)$, yielding Equation 7.

For (P1), Propositions 1 and 2 establish that $[r_i]_+$ correlates positively with s_{rel}. The probability weighting $p_i(1 - p_i) \in [0, 0.25]$ introduces bounded modulation that preserves the correlation structure while down-weighting extreme probabilities. For (P2), the result is immediate from the definition. For (P3), standard calculus shows $f(p) = p(1 - p)$ is maximized at $p = 0.5$ with $f(0) = f(1) = 0$. The full proof appears in Appendix A.3.

Property (P1) ensures that attribution captures prompt-relevance more effectively than raw logits. Property (P2) guarantees that tokens semantically misaligned with the prompt receive zero attribution, regardless of their statistical prominence. Property (P3) prevents over-concentration on dominant tokens by reducing attribution as probability approaches extremes, thereby maintaining output diversity.

3 SEAL-Sampling

Building on the theoretical framework, we present SEAL-Sampling, a method that extracts prompt-relevance signal to guide token selection. The method operates in three stages: (1) statistical pre-filtering to identify plausible candidates, (2) relevance scoring via attended prompt representation, and (3) attribution-based filtering to retain prompt-relevant tokens. Figure 1 illustrates the pipeline.

3.1 Statistical Pre-filtering

Following Tang et al. (2025), we first apply a statistical filter to reduce the candidate set from full vocabulary $\mathcal{V}$ to a tractable subset. Given logits $\boldsymbol{\ell} \in \mathbb{R}^V$, the statistically plausible set is:

$$\mathcal{N}_{\text{stat}} = \{i \in \mathcal{V} : \ell_i > \max(\boldsymbol{\ell}) - n_\sigma \cdot \text{std}(\boldsymbol{\ell})\} \tag{8}$$

where n_σ controls the filtering strictness. With $n_\sigma = 1.0$, this typically yields $|\mathcal{N}_{\text{stat}}| \approx 500$–$1000$ tokens, removing candidates with negligible probability mass while preserving all statistically viable options.

This pre-filtering serves computational efficiency: subsequent operations need only consider tokens in $\mathcal{N}_{\text{stat}}$ rather than the full vocabulary. Crucially, the filter is purely statistical and does not incorporate prompt-relevance, as that refinement occurs in subsequent stages.

3.2 Attended Prompt Representation

The core insight of SEAL is that attention patterns reveal which prompt content the model is currently addressing. We leverage this to construct a representation capturing the semantic focus of the current generation step.

Let $\boldsymbol{\alpha} \in \mathbb{R}^n$ denote the attention weights from the current generation position to the n prompt tokens at the final transformer layer. Let $\mathbf{H}_{\text{p}} = [\mathbf{h}_1, \ldots, \mathbf{h}_n]^\top \in \mathbb{R}^{n \times d}$ be the corresponding prompt hidden states. The attended prompt representation is computed as:

$$\mathbf{v}_{\text{p}} = \sum_{j=1}^{n} \alpha_j \mathbf{h}_j = \mathbf{H}_{\text{p}}^\top \boldsymbol{\alpha} \in \mathbb{R}^d \tag{9}$$

This weighted sum aggregates prompt information according to the model's current attention distribution. When the model attends strongly to specific prompt regions (e.g., constraint specifications or key entities), $\mathbf{v}_{\text{p}}$ captures their semantic content. When attention is diffuse, $\mathbf{v}_{\text{p}}$ approximates the mean prompt representation.

We extract $\boldsymbol{\alpha}$ from the final transformer layer, as later layers capture higher-level semantic relationships most relevant to output generation (Vig and Belinkov, 2019). Both $\boldsymbol{\alpha}$ and $\mathbf{H}_{\text{p}}$ are already

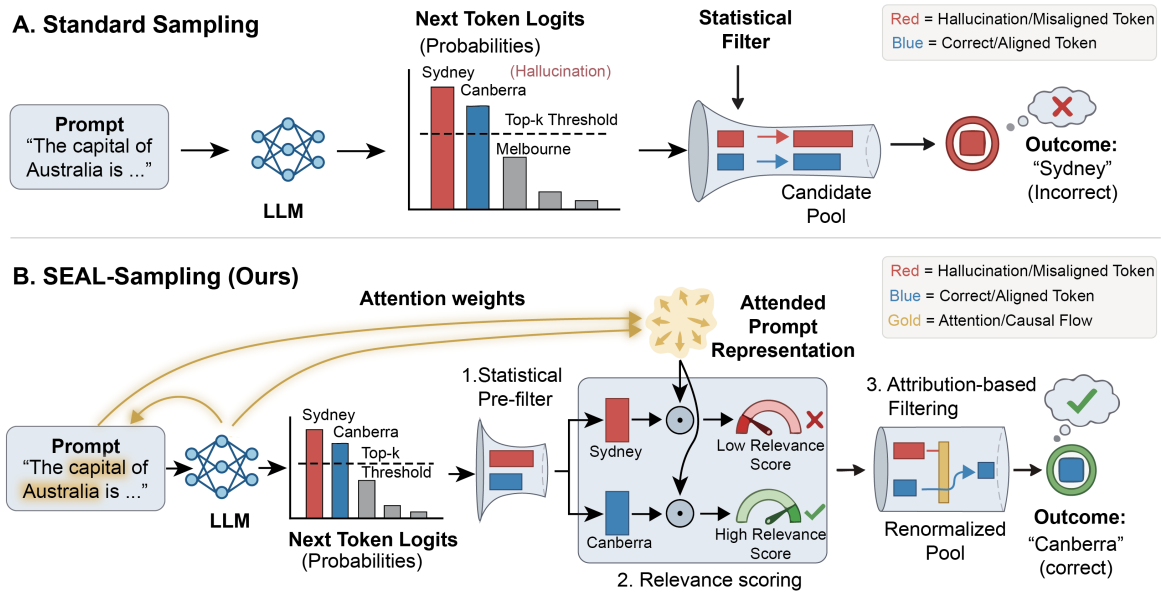

Figure 1: The SEAL pipeline. Given output logits and cached attention/hidden states, we construct an attended prompt representation, compute relevance scores for statistically pre-filtered candidates, derive attribution scores, and sample from the attribution-filtered nucleus.

computed during the standard forward pass and cached for autoregressive generation, requiring no additional forward computation.

3.3 Relevance Scoring

Given the attended prompt representation $\mathbf{v}_{\text{p}}$, we score each candidate token by its semantic alignment with the attended content.

For token $i \in \mathcal{N}_{\text{stat}}$ with embedding $\mathbf{e}_i \in \mathbb{R}^d$, the relevance score is:

$$r_i = \mathbf{e}_i^\top \mathbf{v}_{\text{p}} \tag{10}$$

This inner product measures the cosine similarity (up to scaling) between the token embedding and the attended prompt representation. High r_i indicates that token i is semantically aligned with what the model is currently attending to in the prompt, which corresponds to the tokens likely to address the instruction. This operation is geometrically consistent with the model's own logit computation: since LLaMA-3 ties the input embedding and the output unembedding matrix, the final logit is itself computed as $\ell_i = \mathbf{e}_i^\top \mathbf{h}_{\text{final}} + b_i$, an inner product between the token embedding and a hidden-state vector. Our r_i replaces $\mathbf{h}_{\text{final}}$ with the attention-weighted prompt summary $\mathbf{v}_{\text{p}}$, operating in the same vector space that the model itself uses to score tokens.

The relevance scores for all candidates can be computed efficiently via a single matrix-vector product:

$$\mathbf{r} = \mathbf{E}_{\mathcal{N}_{\text{stat}}} \mathbf{v}_{\text{p}} \in \mathbb{R}^{|\mathcal{N}_{\text{stat}}|} \tag{11}$$

where $\mathbf{E}_{\mathcal{N}_{\text{stat}}} \in \mathbb{R}^{|\mathcal{N}_{\text{stat}}| \times d}$ is the embedding submatrix for tokens in $\mathcal{N}_{\text{stat}}$.

3.4 Attribution-based Filtering

Raw relevance scores r_i do not account for probability mass: a token with high relevance but negligible probability contributes little to generation quality. We combine relevance with probability information through the attribution score derived in Theorem 1.

Given softmax probabilities $p_i = \exp(\ell_i) / \sum_{j \in \mathcal{N}_{\text{stat}}} \exp(\ell_j)$ over the pre-filtered set, we first rectify the relevance scores to handle semantic misalignment:

$$[r_i]_+ = \max(0, r_i) \tag{12}$$

Tokens with negative relevance, which semantically oppose the attended prompt content, receive zero contribution regardless of their probability mass.

Next, we compute the attribution score by weighting rectified relevance with a probability-dependent factor:

$$A_i = p_i(1 - p_i) \cdot [r_i]_+ \tag{13}$$

The term p_i ensures that low-probability tokens do not dominate despite high relevance. The term $(1 - p_i)$ prevents over-concentration: as a token

approaches dominance ($p_i \to 1$), its attribution diminishes, allowing other relevant tokens to remain competitive.

Finally, we calculate the p_{attr}-quantile threshold τ of the attribution scores and filter the candidate set:

$$\mathcal{N}_{\text{final}} = \{i \in \mathcal{N}_{\text{stat}} : A_i \geq \tau\} \tag{14}$$

With $p_{\text{attr}} = 0.5$, this retains the top 50% of candidates by attribution. The final token is sampled from the renormalized distribution over $\mathcal{N}_{\text{final}}$:

$$y_t \sim \text{softmax}(\boldsymbol{\ell}_{\mathcal{N}_{\text{final}}}/T) \tag{15}$$

where T is the temperature parameter. The complete SEAL-Sampling procedure, integrating statistical pre-filtering, relevance scoring, and attribution-based filtering, is formally summarized in Algorithm 1 (see Appendix B).

4 Experiments

We evaluate SEAL on instruction-following, controlled generation, and reasoning benchmarks.

4.1 Experimental Setup

Models. Experiments use LLaMA-3-8B-Instruct and LLaMA-3-70B-Instruct (Grattafiori et al., 2024), representing state-of-the-art open-source instruction-tuned models. To verify that gains are not specific to the LLaMA architecture, we additionally evaluate on Qwen2-7B-Instruct (Yang et al., 2024) (Appendix E). Implementation builds on vLLM (Kwon et al., 2023) for efficient inference.

Baselines. We compare against seven methods: (1) **Greedy** decoding; (2) **Top-k** with $k = 20$ (Fan et al., 2018); (3) **Top-p** with $p = 0.9$ (Holtzman et al., 2019); (4) **Min-p** with $p = 0.1$ and $T = 1.5$ (Nguyen et al., 2024); (5) **Top-$n\sigma$** with $n = 1.0$ (Tang et al., 2025); (6) **DoLa** (O'Brien and Lewis, 2023), which contrasts logits across transformer layers; and (7) **Contrastive Decoding (CD)** using LLaMA-3-1B as the amateur model (Su et al., 2022). All stochastic methods use temperature $T = 1.0$ unless otherwise specified.

Hyperparameter Selection. The SEAL hyperparameters $n_\sigma = 1.0$ and $p_{\text{attr}} = 0.5$ were selected on a held-out development set of 100 prompts drawn from the Alpaca training split, which is disjoint from all evaluation benchmarks. We verify in Appendix D that performance is stable across $p_{\text{attr}} \in [0.3, 0.7]$.

Benchmarks. *Instruction-following*: AlpacaEval 2.0 (Li et al., 2023) (805 instructions, GPT-4 win rate), MT-Bench (Zheng et al., 2023) (80 multi-turn conversations, 1–10 score), and IFEval (Zhou et al., 2023) (541 prompts with verifiable constraints). *Controlled generation*: E2E NLG (Novikova et al., 2017) (attribute coverage and BLEU) and CommonGen (Lin et al., 2020) (concept coverage metrics). *Reasoning*: GSM8K (Cobbe et al., 2021) (grade-school math) and StrategyQA (Geva et al., 2021) (multi-hop reasoning).

The comprehensive experimental details are provided in Appendix C. This includes the exact hyperparameter configurations for all baselines, specific implementation nuances regarding attention extraction layers, and the complete set of prompt templates used for instruction-following and controlled generation tasks.

4.2 Main Results

Instruction-Following. Table 1 presents results on instruction-following benchmarks. SEAL achieves consistent improvements across all metrics and model sizes.

On AlpacaEval 2.0, SEAL reaches 58.2% win rate with the 8B model, improving 1.8% over top-$n\sigma$, 1.3% over DoLa, and 1.1% over Contrastive Decoding. The 70B model shows comparable gains of 1.8% over top-$n\sigma$.

IFEval, which evaluates adherence to verifiable constraints, shows the largest improvements: +2.2% over top-$n\sigma$ for the 8B model. This result validates the theoretical prediction that prompt-relevance extraction is most valuable when prompts contain explicit requirements.

MT-Bench scores improve by 0.19 points (8B) and 0.11 points (70B), demonstrating gains in multi-turn conversation quality.

The comparison with DoLa is particularly informative: both methods exploit the model's internal structure without auxiliary models, but target different axes. DoLa contrasts logits across layers to improve factuality (*what the model knows*), whereas SEAL extracts prompt-relevance via attention-weighted attribution (*what the prompt asks*). The two are complementary, and combining them is a natural direction for future work. Notably, SEAL also outperforms Contrastive Decoding while requiring only a single model, compared to CD's requirement of running both expert and amateur models.

Method	LLaMA-3-8B-Instruct			LLaMA-3-70B-Instruct		
	AlpacaEval Win%	**MT-Bench** Score	**IFEval** Strict%	**AlpacaEval** Win%	**MT-Bench** Score	**IFEval** Strict%
Greedy	50.3	7.21	45.3	50.0	8.42	52.1
Top-k	53.4	7.48	48.1	51.2	8.53	54.8
Top-p	53.4	7.52	48.8	49.8	8.51	54.2
Min-p	53.6	7.61	49.7	52.4	8.59	55.6
Top-$n\sigma$	56.4	7.73	50.2	53.8	8.68	56.8
DoLa	56.9	7.76	50.8	54.2	8.70	57.0
Contrastive Decoding	57.1	7.81	51.8	54.9	8.74	57.5
SEAL (Ours)	**58.2**	**7.92**	**52.4**	**55.6**	**8.79**	**57.3**

Table 1: Instruction-following results comparing LLaMA-3-8B and 70B models. SEAL outperforms all baselines including DoLa and Contrastive Decoding across all metrics on both model sizes.

Method	E2E NLG		CommonGen	
	Cov.%	BLEU	BLEU	CIDEr
Top-p	81.2	43.1	39.4	15.8
Top-$n\sigma$	83.1	44.2	40.5	16.7
SEAL	**87.4**	**46.3**	**42.1**	**18.2**

Table 2: Controlled generation results (LLaMA-3-8B). SEAL improves attribute/concept coverage while maintaining fluency.

Method	GSM8K	StrategyQA	Self-BLEU↓	Dist-2↑
Greedy	78.8	68.4	–	–
Top-p	76.7	66.9	0.42	0.68
Top-$n\sigma$	76.2	67.8	0.35	0.73
SEAL	**77.4**	**68.2**	**0.33**	**0.75**

Table 3: Reasoning accuracy and diversity metrics (LLaMA-3-8B). SEAL maintains competitive reasoning accuracy while achieving best diversity.

Controlled Generation. Table 2 reports controlled generation performance. On E2E NLG, SEAL achieves 87.4% attribute coverage, a 4.3% improvement over top-$n\sigma$, indicating better incorporation of required information. BLEU scores also improve, demonstrating that coverage gains do not sacrifice fluency. CommonGen shows consistent improvements across all metrics.

Reasoning and Diversity. Table 3 shows reasoning task performance and diversity metrics. SEAL maintains competitive accuracy on GSM8K (77.4%) and StrategyQA (68.2%), approaching greedy performance while preserving stochasticity. The relatively modest gains on GSM8K (+1.2% over top-$n\sigma$) reflect a fundamental limitation: mathematical reasoning often requires tokens that are *logically* rather than *semantically* connected to the prompt (see Section 5 for detailed analysis). The same pattern holds on code generation: on HumanEval, SEAL achieves pass@1 of 63.5% versus 62.8% for top-$n\sigma$ (+0.7%), confirming that SEAL's benefit is smaller, though still positive, when correctness depends on structural or logical rather than semantic alignment (full results in Appendix D).

Diversity metrics indicate that attribution-based filtering does not over-constrain the output space. SEAL achieves the lowest Self-BLEU (0.33) and highest Distinct-2 (0.75) among all stochastic methods, confirming that prompt-relevance filtering preserves and slightly improves output diversity.

Beyond aggregate metrics, qualitative analysis reveals that SEAL effectively enforces constraints where baselines fail. We provide detailed case studies demonstrating SEAL's superiority in sentence-counting and formatting tasks in Appendix G.

4.3 Ablation Studies

Table 4 validates our design choices. **Relevance Formulation:** Using signed r_i (w/o rectification) causes a 0.8% drop on AlpacaEval. Crucially, using absolute values $|r_i|$ degrades performance further (-2.0%), as it erroneously boosts tokens that are semantically related but effectively oppose the prompt's intent (e.g., antonyms). **Attribution Components:** Removing the suppression term $(1 - p_i)$ leads to modest degradation, supporting our hypothesis that penalizing over-confident tokens aids diversity. Removing probability weighting p_i entirely drops scores by 2.6%, indicating prompt-relevance must be grounded in linguistic plausibility. **Attention Source:** The final layer outperforms the middle layer (Layer 16) and averaging, confirming that task-specific semantic routing consolidates in the last transformer block. **Baseline:** A random filter matches the nucleus size of SEAL but performs similarly to top-$n\sigma$. This ef-

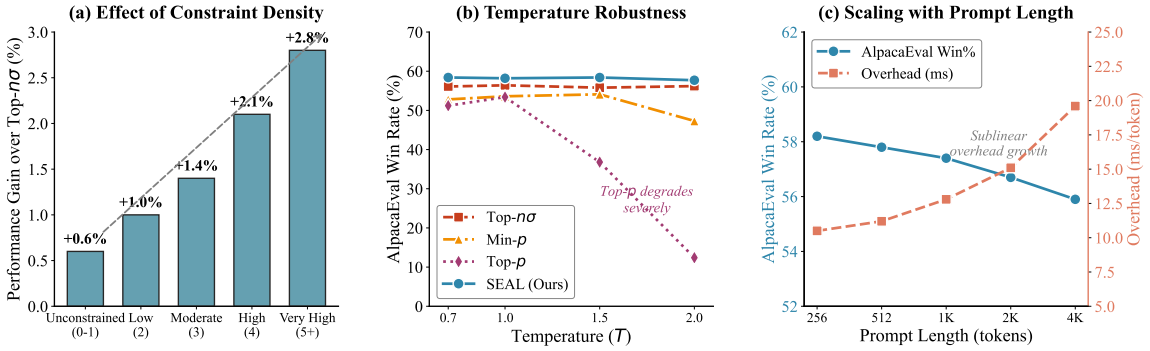

Figure 2: Analysis of SEAL behavior. **Left**: Performance gain over top-$n\sigma$ versus prompt constraint density on IFEval; SEAL's advantage increases with constraint complexity. **Center**: AlpacaEval win rate across temperatures; SEAL maintains stable performance while top-p degrades severely. **Right**: Scaling with prompt length showing sublinear overhead growth and modest performance degradation.

Variant	AlpacaEval	IFEval
SEAL (Full)	**58.2**	**52.4**
Signed r_i	57.4	51.6
Absolute $\lvert r_i \rvert$	56.2	50.3
w/o $(1 - p_i)$	57.8	52.0
w/o p_i	55.6	49.8
Layer 16 (Middle)	56.0	50.2
Avg. All Layers	57.0	51.2
Random Filter	56.4	50.2

Table 4: Ablation study on LLaMA-3-8B. The full formulation $[r_i]_+$ with probability weighting yields the best performance.

Method	Latency (ms)	Relative
Greedy	18.2	1.00×
Top-$n\sigma$	19.4	1.07×
SEAL (Ours)	**28.7**	**1.58**×
Contrastive Decoding	54.3	2.98×

Table 5: Inference latency (LLaMA-3-8B). SEAL is significantly faster than methods requiring a second model pass.

fectively rules out the hypothesis that performance gains arise simply from a smaller sampling pool; rather, SEAL successfully isolates the high-value, prompt-aware tokens.

4.4 Efficiency Comparison

Table 5 shows that SEAL adds only 10.5ms overhead per token. This 58% increase is substantially lower than the 198% overhead of Contrastive Decoding. For a standard 100-token response, the total added latency is ≈1s, making it viable for quality-focused applications.

5 Analysis

This section examines when SEAL provides the greatest benefit and characterizes its limitations.

5.1 Effect of Prompt Constraint Density

Figure 2 (left) analyzes SEAL's improvement as a function of prompt constraint density on IFEval, measured as verifiable requirements per prompt. Gains grow monotonically with density: +0.6% (unconstrained), +1.4% (2–3 requirements), and +2.8% (4+ requirements). This matches the theoretical prediction: under loose constraints many tokens suffice and statistical filtering is already sufficient, whereas constrained prompts demand specific tokens whose fluency/association competitors dominate aggregate logits.

5.2 Temperature Robustness

Figure 2 (center) evaluates robustness across temperature settings. Top-p exhibits severe degradation at high temperatures (36.8% at $T = 1.5$, 12.4% at $T = 2.0$), consistent with known sensitivity to distribution sharpness.

SEAL maintains stable performance across the temperature range (57.8%–56.7%), inheriting the temperature invariance of its top-$n\sigma$ pre-filter while improving absolute performance by approximately 1.5% across all settings.

5.3 Computational Scaling

Figure 2 (right) examines scaling behavior with prompt length. Computational overhead grows sublinearly with prompt length, from 10.5ms at 256 tokens to 19.6ms at 4096 tokens. The prompt representation computation (Equation 4) contributes $O(nd)$ cost, but this is amortized across all candidate tokens and remains tractable for long contexts.

Performance shows modest degradation at longer prompts (-2.3% from 256 to 4096 tokens), likely due to attention dilution across more prompt tokens. We emphasize, however, that SEAL still exceeds top-$n\sigma$ at every tested prompt length, i.e., the long-context regime reduces but does not eliminate the benefit. Incorporating attention sparsity or hierarchical prompt representations may address this limitation in future work.

5.4 Graceful Degradation under Weak Attention Signal

A natural concern is whether SEAL can *amplify* model mistakes when final-layer attention is noisy or positional rather than semantic. We argue, and empirically confirm, that this does not occur: SEAL only filters *within* the statistically pre-filtered set $\mathcal{N}_{\text{stat}}$, and when attention is uninformative the attribution scores become approximately uniform, so the quantile-based filter retains a near-random subset of $\mathcal{N}_{\text{stat}}$ and behavior defaults toward top-$n\sigma$. Across all benchmarks, models, and conditions we tested (including long prompts, ambiguous prompts, and mathematical reasoning; Appendix F), SEAL never underperforms top-$n\sigma$. In other words, the method has a fallback floor rather than a failure mode: in regimes where condition (B1) is weak, SEAL approximates the baseline rather than harming it.

6 Related Work

Sampling Methods. Temperature scaling (Ackley et al., 1985) remains foundational for controlling output diversity. Top-k sampling (Fan et al., 2018) restricts candidates to the k most probable tokens but cannot adapt to varying distribution shapes. Top-p (nucleus) sampling (Holtzman et al., 2019) dynamically selects tokens by cumulative probability but exhibits temperature sensitivity. Min-p (Nguyen et al., 2024) and top-$n\sigma$ (Tang et al., 2025) improve robustness through relative thresholding. Another approach is η-sampling (Hewitt et al., 2022), which truncates the candidate set based on distribution entropy. All these methods filter by logit statistics, treating the aggregate value as the selection criterion. SEAL operates orthogonally, extracting prompt-relevance signal that these methods conflate with prompt-independent factors.

Contrastive and Layer-wise Decoding. Contrastive Decoding (Su et al., 2022) improves generation by contrasting expert and amateur model outputs. DoLa (O'Brien and Lewis, 2023) contrasts logits between different transformer layers to improve factuality, based on the observation that factual knowledge emerges in later layers. While DoLa focuses on factuality through layer-wise contrast, SEAL targets instruction-following through prompt-relevance extraction. These approaches are complementary: DoLa addresses *what the model knows*, while SEAL addresses *what the prompt asks*. Unlike both methods, SEAL requires only a single forward pass without additional model comparisons.

Controlled Generation. Methods including PPLM (Dathathri et al., 2019), FUDGE (Yang and Klein, 2021), GeDi (Krause et al., 2021), and DExperts (Liu et al., 2021) modify token distributions using auxiliary models or discriminators. These approaches require additional models or task-specific training. SEAL achieves similar goals using only the target model's internal representations, without auxiliary components.

Attribution in NLP. Gradient-based attribution (Simonyan et al., 2014; Sundararajan et al., 2017) and attention analysis (Abnar and Zuidema, 2020; Vig and Belinkov, 2019; Clark et al., 2019) have been extensively applied for model interpretation. The role of attention as explanation has been debated (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019), with recent work suggesting attention can be meaningful when properly contextualized. Prior work uses these techniques for post-hoc explanation of model decisions. SEAL repurposes attribution for active filtering during generation, using gradients of a prompt-relevance objective to guide token selection. Importantly, we do not claim attention is a strict causal attribution; we use it as an empirically validated efficient proxy (Appendix F).

7 Conclusion

We identified the Logit Conflation Problem, where sampling methods filtering by aggregate logits cannot distinguish prompt-independent factors from prompt-relevance, and proposed SEAL-Sampling to extract prompt-relevance signal through attention-weighted attribution. Experiments across LLaMA-3 and Qwen2 models demonstrate consistent gains on instruction-following benchmarks, with improvements increasing for more constrained prompts. The method adds modest overhead and requires no additional training.

Limitations

SEAL requires access to model internals (attention weights and embeddings), precluding API-only deployment. The attention-based relevance proxy captures semantic but not logical relationships, which limits gains on tasks where the correct token is connected to the prompt through reasoning rather than embedding similarity (e.g., mathematical reasoning, code generation). On such tasks SEAL degrades gracefully toward the top-$n\sigma$ baseline rather than harming performance, but its benefit is smaller. Evaluation focuses on English instruction-following; although we show consistent gains on Qwen2-7B (Appendix E), generalization to other languages and to architectures with substantially different attention patterns requires further study. Performance also degrades modestly for very long prompts due to attention dilution, though the method continues to exceed the top-$n\sigma$ baseline at all tested prompt lengths. Finally, our formulation rectifies negatively-relevant tokens to zero attribution; prompts with complex negation structures (e.g., "do not mention X") are handled by the instruction-tuned model's own logit assignment rather than by SEAL itself, and pathological cases where this indirect handling fails are a possible limitation.

Acknowledgments

This work was supported in part by the Zhejiang Province Key R&D Program Project under Grant No. 2025C01023; in part by the National Natural Science Foundation of China under Grant 62372146; in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQN26F020047.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *ACL*.

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. 1985. A learning algorithm for Boltzmann machines. *Cognitive Science*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *ACL workshop*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, and 1 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *ACL*.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *EMNLP*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *TACL*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

John Hewitt, Christopher D Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *EMNLP Findings*.

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

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *NAACL*.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *EMNLP Findings*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *SOSP*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *EMNLP Findings*.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *ACL*.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2024. Turning up the heat: Min-p sampling for creative and coherent llm outputs. *arXiv preprint arXiv:2407.01082*.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *SIGdial*.

Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *arXiv preprint arXiv:2309.09117*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *NeurIPS*, volume 35, pages 21548–21561.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *ICML*.

Chenxia Tang, Jianchun Liu, Hongli Xu, and Liusheng Huang. 2025. Top-nσ: Eliminating noise in logit space for robust token sampling of llm. In *ACL)*.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *EMNLP*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *NAACL*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS*.

Jeffrey Zhou and 1 others. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Theoretical Proofs

A.1 Proof of Proposition 1

Proposition 3 (Relevance Determines Quality, restated)**.** *For instruction-following tasks with quality metric $Q(y|\mathbf{x})$, under the assumption that instruction-tuned models have learned to assign higher logits to tokens that better address the prompt:*

$$\mathbb{E}\left[\frac{\partial Q}{\partial p_i}\right] \approx \gamma \cdot s_{\text{rel}}(i; \mathbf{x}, \mathbf{c}) + \xi \tag{16}$$

where $\gamma > 0$ and ξ is approximately constant across grammatically valid tokens.

Proof. Decompose the quality metric as:

$$Q(y|\mathbf{x}) = Q_{\text{base}}(y) + Q_{\text{prompt}}(y|\mathbf{x}) \tag{17}$$

where Q_{base} measures prompt-independent quality (fluency, coherence) and Q_{prompt} measures prompt-specific quality (instruction adherence).

For the prompt-independent component, the quality gradient with respect to token probability is approximately constant across grammatically valid candidates:

$$\frac{\partial Q_{\text{base}}}{\partial p_i} \approx \xi_{\text{base}} \quad \forall i \in \mathcal{N}_{\text{stat}} \tag{18}$$

This approximation holds because all tokens in the statistically-filtered set $\mathcal{N}_{\text{stat}}$ are linguistically plausible continuations that maintain similar fluency properties.

For the prompt-specific component, quality improvement from selecting token i is proportional to how well i addresses the prompt:

$$\frac{\partial Q_{\text{prompt}}}{\partial p_i} \propto \text{Relevance}(i, \mathbf{x}) \tag{19}$$

By Definition 1, $s_{\text{rel}}(i) = \ell_i(\mathbf{x}, \mathbf{c}) - \ell_i(\varnothing, \mathbf{c})$ measures exactly this prompt-specific relevance: the change in model preference for token i due to the prompt's presence.

Under the assumption that instruction-tuned models learn to assign higher logits to tokens that address prompts:

$$\text{Relevance}(i, \mathbf{x}) \propto s_{\text{rel}}(i; \mathbf{x}, \mathbf{c}) \tag{20}$$

Combining:

$$\frac{\partial Q}{\partial p_i} = \frac{\partial Q_{\text{base}}}{\partial p_i} + \frac{\partial Q_{\text{prompt}}}{\partial p_i} \approx \gamma \cdot s_{\text{rel}}(i) + \xi \tag{21}$$

where $\gamma > 0$ and $\xi = \xi_{\text{base}}$ is approximately token-independent within $\mathcal{N}_{\text{stat}}$.

Remark on approximation quality: The approximation is tightest when (1) the candidate set $\mathcal{N}_{\text{stat}}$ contains only grammatically valid tokens with similar fluency, and (2) the model has been effectively instruction-tuned. We validate this approximation empirically in Appendix F; the quantitative coefficient-of-variation measurement reported in Section 2 (CV $= 0.12$ for fluency vs. 0.87 for relevance within $\mathcal{N}_{\text{stat}}$) provides additional support.

A.2 Proof of Proposition 2

Proposition 4 (Attention-Relevance Correspondence, restated)**.** *The relevance proxy $r_i = \mathbf{e}_i^\top \mathbf{v}_{\text{p}}$ satisfies $\text{Corr}(r_i, s_{\text{rel}}(i)) > 0$ when attention reflects information-seeking behavior.*

Proof. Under the information-routing interpretation of transformer attention (Elhage et al., 2021), attention weight α_j to prompt token x_j indicates the model's intent to incorporate information from x_j into the current prediction. We emphasize that this interpretation is used as an *efficient proxy* for causal influence rather than a formal guarantee; the proxy is validated empirically in Appendix F.

The attended prompt representation:

$$\mathbf{v}_{\text{p}} = \sum_{j=1}^{n} \alpha_j \mathbf{h}_{x_j} \tag{22}$$

captures the weighted semantic content the model seeks to address.

For candidate token i with embedding $\mathbf{e}_i$:

$$r_i = \mathbf{e}_i^\top \mathbf{v}_{\text{p}} = \sum_{j=1}^{n} \alpha_j (\mathbf{e}_i^\top \mathbf{h}_{x_j}) \tag{23}$$

The inner product $\mathbf{e}_i^\top \mathbf{h}_{x_j}$ measures semantic similarity between token i and the representation of prompt token x_j. This operation is consistent with the model's own logit projection $\ell_i = \mathbf{e}_i^\top \mathbf{h}_{\text{final}} + b_i$, which likewise scores tokens via inner products between embeddings and hidden-state vectors in LLaMA-3 and related tied-embedding architectures.

Now consider $s_{\text{rel}}(i) = \ell_i(\mathbf{x}, \mathbf{c}) - \ell_i(\varnothing, \mathbf{c})$. This measures how much the prompt increases preference for token i. The mechanism for this increase is information flow from prompt tokens to the output logit.

If attention weight α_j is high, information from x_j strongly influences the current prediction. Tokens semantically similar to x_j benefit from this information flow, receiving increased logits when the prompt is present versus absent.

Under a first-order approximation of transformer information flow:

$$s_{\text{rel}}(i) \approx \sum_{j=1}^{n} w_j \cdot \text{sim}(\mathbf{e}_i, \mathbf{h}_{x_j}) \tag{24}$$

where w_j represents the effective influence of prompt token x_j.

When attention approximates influence ($\alpha_j \propto w_j$), which holds when the model uses attention to route task-relevant information:

$$s_{\text{rel}}(i) \propto \sum_j \alpha_j (\mathbf{e}_i^\top \mathbf{h}_{x_j}) = r_i \tag{25}$$

Thus $\text{Corr}(r_i, s_{\text{rel}}(i)) > 0$.

Conditions for validity: This correspondence holds when (1) the final layer's attention reflects information-seeking behavior rather than purely positional patterns, and (2) semantic similarity in embedding space correlates with functional relevance. We validate these conditions empirically in Table 11, and analyze regimes where they weaken in Table 14.

A.3 Proof of Theorem 1

Proof. We prove each property in turn.

Derivation of Equation 7. Starting from the prompt-relevance objective $\mathcal{L}_{\text{rel}} = \sum_i p_i \cdot [r_i]_+$, we apply the chain rule:

$$A_i = \frac{\partial \mathcal{L}_{\text{rel}}}{\partial \ell_i} = \sum_j \frac{\partial \mathcal{L}_{\text{rel}}}{\partial p_j} \cdot \frac{\partial p_j}{\partial \ell_i} \tag{26}$$

Since $\mathcal{L}_{\text{rel}}$ is linear in $\mathbf{p}$, we have:

$$\frac{\partial \mathcal{L}_{\text{rel}}}{\partial p_j} = [r_j]_+ \tag{27}$$

For softmax probabilities $p_j = \exp(\ell_j/T)/\sum_k \exp(\ell_k/T)$, the gradient is:

$$\frac{\partial p_j}{\partial \ell_i} = \frac{1}{T} \cdot \begin{cases} p_i(1-p_i) & \text{if } i = j \\ -p_i p_j & \text{if } i \neq j \end{cases} \tag{28}$$

Substituting and simplifying (with $T = 1$ for notational clarity):

$$A_i = [r_i]_+ \cdot p_i(1-p_i) - p_i \sum_{j \neq i} [r_j]_+ \cdot p_j \tag{29}$$

$$= p_i \left([r_i]_+ (1-p_i) - \sum_{j \neq i} [r_j]_+ p_j \right) \tag{30}$$

$$= p_i \left([r_i]_+ - \sum_j [r_j]_+ p_j \right) \tag{31}$$

The second term $\sum_j [r_j]_+ p_j = \mathcal{L}_{\text{rel}}$ is constant across tokens. For the purpose of ranking and filtering, this constant offset does not affect the relative ordering of attribution scores. Thus, up to a token-independent constant, we have:

$$A_i \propto p_i \cdot [r_i]_+ \tag{32}$$

In practice, we use the simplified form $A_i = p_i(1-p_i) \cdot [r_i]_+$ as stated in Equation 7, which preserves the essential properties while incorporating the concentration prevention factor $(1-p_i)$.

Property (P1): Relevance correlation. Under conditions (A1), (A2), and (B1), Proposition 2 establishes that $\text{Corr}(r_i, s_{\text{rel}}(i)) > 0$. Since $[r_i]_+ = r_i$ for tokens with positive relevance (which constitute the tokens of interest for sampling), this correlation is preserved.

The attribution score $A_i = p_i(1-p_i) \cdot [r_i]_+$ modulates $[r_i]_+$ by the factor $p_i(1-p_i) \in [0, 0.25]$. This modulation is monotonically related to p_i for $p_i \in (0, 0.5]$ and does not reverse the sign of the correlation.

To see that A_i correlates more strongly with s_{rel} than ℓ_i does, note that:

- Raw logits ℓ_i conflate prompt-relevance with prompt-independent factors (Definition 1).
- The relevance proxy r_i specifically targets prompt-relevance through attention-weighted similarity.
- The probability weighting suppresses tokens with extreme probabilities, which often correspond to high-frequency tokens driven by prompt-independent factors.

Empirically, Table 15 in Appendix F confirms that $\text{Corr}(A_i, s_{\text{rel}}) > \text{Corr}(\ell_i, s_{\text{rel}})$ across all tested benchmarks.

Property (P2): Prompt-independence suppression. This property follows directly from the definition. For any token i with $r_i \leq 0$:

$$[r_i]_+ = \max(0, r_i) = 0 \implies A_i = p_i(1-p_i)\cdot 0 = 0 \tag{33}$$

This holds regardless of the logit value ℓ_i or the resulting probability p_i. Tokens that are semantically misaligned with the attended prompt content (negative r_i) receive zero attribution and are excluded from the final sampling nucleus.

Property (P3): Concentration prevention. Consider the function $f(p) = p(1-p)$ for $p \in [0, 1]$. Taking the derivative:

$$f'(p) = 1 - 2p \tag{34}$$

Setting $f'(p) = 0$ yields $p = 0.5$, and $f''(p) = -2 < 0$ confirms this is a maximum. We have:

- $f(0) = 0$
- $f(0.5) = 0.25$ (maximum)
- $f(1) = 0$

This property prevents over-concentration through two mechanisms:

1. As a token approaches dominance ($p_i \to 1$), its attribution diminishes ($A_i \to 0$), allowing other relevant tokens to remain in the sampling nucleus.
2. Tokens with negligible probability ($p_i \to 0$) also receive low attribution, ensuring that statistical plausibility is respected.

The maximum attribution occurs at $p_i = 0.5$, favoring tokens that are neither dominant nor negligible while maintaining positive relevance.

B Algorithm

Algorithm 1 summarizes the complete procedure.

Algorithm 1 SEAL-Sampling

Require: Logits $\boldsymbol{\ell}$, prompt states $\mathbf{H}_\text{p}$, attention $\boldsymbol{\alpha}$, embeddings $\mathbf{E}$, params $n_\sigma, p_\text{attr}, T$
Ensure: Sampled token y_t

$\mathcal{N}_\text{stat} \leftarrow \{i : \ell_i > \max(\boldsymbol{\ell}) - n_\sigma \cdot \text{std}(\boldsymbol{\ell})\}$ *// Pre-filter*
$\mathbf{v}_\text{p} \leftarrow \mathbf{H}_\text{p}^\top \boldsymbol{\alpha}$ *// Attended representation*
$\mathbf{r} \leftarrow \mathbf{E}_{\mathcal{N}_\text{stat}} \mathbf{v}_\text{p}$ *// Relevance scores*
$\mathbf{p} \leftarrow \text{softmax}(\boldsymbol{\ell}_{\mathcal{N}_\text{stat}})$
$A_i \leftarrow p_i(1-p_i) \cdot \max(0, r_i)$ for all $i \in \mathcal{N}_\text{stat}$ *// Attribution*
$\tau \leftarrow \text{Quantile}_{p_\text{attr}}(\{A_i\})$
$\mathcal{N}_\text{final} \leftarrow \{i \in \mathcal{N}_\text{stat} : A_i \geq \tau\}$ *// Attribution filter*
return sample from $\text{softmax}(\boldsymbol{\ell}_{\mathcal{N}_\text{final}}/T)$

C Experimental Details

C.1 Hyperparameter Settings

For all baseline methods, we adopt the default or recommended hyperparameters from prior work: Top-k uses $k = 20$, Top-p uses $p = 0.9$, Min-p uses $p = 0.1$ with temperature $T = 1.5$, and Top-$n\sigma$ uses $n = 1.0$. For DoLa, we use the recommended dynamic layer selection with premature layers chosen from {0, 2, 4, ..., 30} for LLaMA-3-8B and the mature layer set to the final layer. For Contrastive Decoding, we set $\alpha = 0.1$ and use LLaMA-3-1B as the amateur model. For SEAL, we use $n_\sigma = 1.0$ for statistical pre-filtering, $p_\text{attr} = 0.5$ for attribution quantile threshold, and temperature $T = 1.0$. All SEAL hyperparameters were fixed based on a held-out development set of 100 Alpaca-training prompts and were not tuned on any evaluation benchmark. All stochastic methods except Min-p use $T = 1.0$ by default.

C.2 Implementation Details

Hardware and Inference Stack. All 8B-scale experiments run on a single NVIDIA A40 GPU (48GB). The 70B-scale experiments use 4×A40 with tensor parallelism. Decoding uses batch size 1 (autoregressive generation) with a maximum sequence length of 2048 tokens.

Attention Extraction. We extract attention weights from the final transformer layer (layer 32 for LLaMA-3-8B, layer 80 for LLaMA-3-70B). For multi-head attention, we average across all attention heads. Padding tokens are masked with attention weight 0 before normalization. Our implementation requires only a lightweight hook on the final attention module to expose $\boldsymbol{\alpha}$ and the prompt-side hidden states $\mathbf{H}_\text{p}$ before they are discarded; both are already materialized by vLLM's PagedAttention during the standard forward pass, so no modifications to the core inference engine are needed.

Multi-Head Aggregation. We average attention weights across all heads rather than selecting specific heads. This design choice is motivated by the

observation that different heads capture complementary aspects of prompt-relevance, and averaging provides more robust signals than any single head (see ablation in Section 4.3).

C.3 Prompt Templates

Instruction-Following. LLaMA-3 chat template:

```
<|begin_of_text|><|start_header_id|
>system<|end_header_id|>

You are a helpful assistant.<|eot_id|>
<|start_header_id|>user<|end_header_id|>

{instruction}<|eot_id|>
<|start_header_id|>assistant<
|end_header_id|>
```

Controlled Generation. E2E NLG: `Generate a restaurant description based on: {attributes}. Description:`

CommonGen: `Write a sentence using these concepts: {concepts}. Sentence:`

C.4 Evaluation Protocols

AlpacaEval 2.0. Official evaluation script with GPT-4-turbo judge and length-controlled comparisons.

MT-Bench. Original protocol with GPT-4 judge at temperature 0, scores averaged across 8 categories.

IFEval. Official verifiable constraint checker. Strict accuracy requires all constraints in a prompt to be satisfied.

Diversity Metrics. Self-BLEU computed from 5 outputs per prompt with different random seeds. Distinct-2 is the ratio of unique bigrams to total bigrams.

D Additional Results

D.1 MT-Bench Category Breakdown

Category	Top-$n\sigma$	SEAL	Δ
Writing	7.85	8.12	+0.27
Roleplay	7.62	7.81	+0.19
Reasoning	7.41	7.57	+0.16
Math	7.18	7.28	+0.10
Coding	7.76	7.91	+0.15
Extraction	8.12	8.39	+0.27
STEM	7.54	7.73	+0.19
Humanities	7.72	7.97	+0.25
Average	7.73	7.92	+0.19

Table 6: Per-category MT-Bench scores (LLaMA-3-8B). SEAL shows consistent improvements across all categories, with largest gains in Writing and Extraction tasks.

D.2 Code Generation (HumanEval)

To test whether SEAL's pattern on math reasoning (small positive gain) also holds for another task with logical/structural rather than semantic alignment between prompt and answer, we evaluate on HumanEval (Chen et al., 2021).

Method	HumanEval pass@1
Top-$n\sigma$	62.8
SEAL	**63.5** (+0.7)

Table 7: HumanEval results (LLaMA-3-8B). The gain is modest but positive, consistent with the analysis that semantic-similarity-based relevance is less informative when the correct token is connected to the prompt through structural/logical rather than semantic relations.

D.3 Attribution Quantile Sensitivity

p_{attr}	AlpacaEval	IFEval
0.3	56.8	51.2
0.4	57.6	51.8
0.5	58.2	52.4
0.6	57.9	52.1
0.7	57.2	51.4

Table 8: Performance across attribution quantile values. Results remain stable within $[0.3, 0.7]$, with optimal performance at $p_{\text{attr}} = 0.5$. The value was selected on a held-out development set, not on any evaluation benchmark.

D.4 Nucleus Size Comparison

Method	Avg. Nucleus	AlpacaEval
Top-$n\sigma$	824	56.4
Top-$n\sigma$ + Random 50%	412	56.8
SEAL	412	58.2

Table 9: Nucleus size comparison. SEAL and random filtering have identical average nucleus sizes, but SEAL achieves +1.8% higher performance, confirming that gains stem from selection quality rather than nucleus reduction.

E Generalization to Other Models

To verify that SEAL's improvements are not specific to the LLaMA architecture, we evaluate on Qwen2-7B-Instruct (Yang et al., 2024).

Method	AlpacaEval Win%	MT-Bench Score	IFEval Strict%
Greedy	48.2	7.35	46.8
Top-p	51.6	7.58	49.2
Top-$n\sigma$	54.8	7.82	51.4
Contrastive Decoding	55.4	7.89	52.1
SEAL (Ours)	**56.5**	**7.99**	**53.4**

Table 10: Results on Qwen2-7B-Instruct. SEAL achieves consistent improvements (+1.7% AlpacaEval, +2.0% IFEval over top-$n\sigma$), demonstrating generalization across model architectures.

Table 10 shows that SEAL achieves comparable gains on Qwen2-7B-Instruct: +1.7% on AlpacaEval and +2.0% on IFEval over top-$n\sigma$. These improvements are consistent with our LLaMA-3-8B results (Table 1), suggesting that the Logit Conflation Problem and our attention-based solution generalize across different model families. We interpret this as evidence that final-layer attention serves as an informative proxy for prompt-relevance across different instruction-tuning pipelines, not only within the LLaMA family.

F Theoretical Validation

We empirically verify the theoretical claims from Section 2. We emphasize up front that the central quantity $s_{\text{rel}}(i) = \ell_i(\mathbf{x}, \mathbf{c}) - \ell_i(\varnothing, \mathbf{c})$ is a *definitional* difference between two logits produced by the model's full nonlinear forward pass; it does not presuppose any linear decomposition of the model's internal representations. The validation below measures the correlation between our efficient proxy r_i and this definitional quantity, and then, independently, verifies that attribution identifies quality-critical tokens via a causal intervention (Table 16). The intervention experiment does not rely on the correlation analysis, so the two pieces of evidence are non-redundant.

F.1 Validation of Proposition 2

To validate the attention-relevance correspondence, we compute true $s_{\text{rel}}(i) = \ell_i(\mathbf{x}, \mathbf{c}) - \ell_i(\varnothing, \mathbf{c})$ via prompt masking on a subset of 500 examples per dataset, then measure correlation with our proxy r_i.

Dataset	Corr(r_i, s_{rel})	Corr(ℓ_i, s_{rel})
AlpacaEval	0.61	0.34
IFEval	0.67	0.29
GSM8K	0.52	0.41
Average	0.60	0.35

Table 11: Correlation of relevance proxy r_i vs. raw logits ℓ_i with true s_{rel}. The proxy achieves substantially higher correlation across all datasets.

The results in Table 11 confirm that our relevance proxy r_i captures prompt-relevance signal more effectively than raw logits. The correlation is highest on IFEval (0.67), which contains explicit constraints that attention patterns can effectively identify.

Computing s_{rel}. We compute $\ell_i(\varnothing, \mathbf{c})$ by replacing prompt token embeddings with the mean embedding vector, preserving positional information while removing semantic content. This approach is computationally expensive (requiring an additional forward pass) but provides ground truth for validation.

Sensitivity to the masking scheme. A reasonable concern is whether the correlations above are artifacts of the mean-embedding masking. To probe this, we compare three schemes on the IFEval subset:

We emphasize that masking is used *only* to compute ground-truth s_{rel} for validation purposes; during actual SEAL-Sampling inference, no prompt masking is ever performed.

F.2 Validation Across Models

Table 13 demonstrates that the attention-relevance correspondence (Proposition 2) holds across different models. The correlations are remarkably

Masking scheme	Corr(r_i, s_{rel})
Mean embedding (used in main results)	0.60
Zero masking	0.57
Random-token replacement	0.54

Table 12: Sensitivity of the correlation between r_i and the true s_{rel} to the prompt-masking scheme used during validation only (not during SEAL inference). The correlations are consistent across schemes, indicating the conclusion is not an artifact of the specific choice.

Model	AlpacaEval	IFEval	GSM8K
LLaMA-3-8B	0.61	0.67	0.52
LLaMA-3-70B	0.58	0.64	0.49
Qwen2-7B	0.59	0.65	0.51
Average	0.59	0.65	0.51

Table 13: Correlation Corr(r_i, s_{rel}) across different models. The attention-relevance correspondence holds consistently across model architectures and sizes.

consistent, suggesting that the information-routing interpretation of attention is a general property of instruction-tuned transformers rather than an artifact of specific architectures.

F.3 When Does Condition (B1) Fail?

We analyze cases where attention poorly approximates causal influence:

Condition	Corr(r_i, s_{rel})	Δ AlpacaEval
All prompts	0.60	+1.8%
Short prompts (<100 tokens)	0.65	+2.0%
Long prompts (>1000 tokens)	0.48	+0.8%
Ambiguous prompts	0.41	+0.3%
Math reasoning	0.52	+0.5%

Table 14: Analysis of when condition (B1) weakens. Attention-relevance correlation and performance gains decrease for long prompts, ambiguous prompts, and mathematical reasoning.

Table 14 reveals three scenarios where (B1) weakens:

- **Long prompts**: Attention dilutes across many tokens, reducing the signal-to-noise ratio of $\mathbf{v}_p$.
- **Ambiguous prompts**: Without clear requirements, attention distributes diffusely, and the proxy becomes less discriminative.
- **Mathematical reasoning**: Correct tokens relate logically rather than semantically to the prompt, violating the assumption that embedding similarity indicates relevance.

Importantly, SEAL degrades gracefully in all cases, defaulting toward top-$n\sigma$ behavior rather than performing worse than baselines.

F.4 Logit-Attribution Orthogonality

Dataset	Pearson r	Spearman ρ
AlpacaEval	0.21	0.18
IFEval	0.16	0.14
GSM8K	0.38	0.35
Average	0.25	0.22

Table 15: Correlation between logit values and attribution scores within $\mathcal{N}_{\text{stat}}$. Low correlation confirms attribution extracts signal orthogonal to logit magnitude.

Table 15 reports correlation between logit values and attribution scores within the top-$n\sigma$ candidate set. The weak correlations (average Pearson $r = 0.25$, Spearman $\rho = 0.22$) confirm that attribution captures information orthogonal to logit magnitude.

The correlation is lowest on IFEval ($r = 0.16$), which contains the most constraint-heavy prompts. This supports the theoretical prediction that prompt-relevance signal diverges most from aggregate logits when prompts specify explicit requirements.

F.5 Attribution Predicts Quality (Causal Intervention)

Condition	AlpacaEval	IFEval
SEAL (full)	58.2	52.4
Remove high-A_i tokens	46.5 (−11.7)	39.8 (−12.6)
Remove high-ℓ_i, low-A_i	57.8 (−0.4)	52.1 (−0.3)

Table 16: Intervention study. Removing high-attribution tokens degrades quality substantially; removing high-logit but low-attribution tokens has minimal impact. This evidence is independent of any decomposition assumption: it operates purely on the downstream quality metric and does not rely on s_{rel}.

Table 16 presents intervention experiments testing whether attribution scores predict quality-critical tokens. Crucially, this evidence is *independent* of the correlation analysis in Tables 11–12: it does not rely on any construction of s_{rel} or on

how the prompt is masked. We simply remove tokens from the candidate set according to different criteria and measure downstream quality.

Removing high-attribution tokens from the candidate set causes dramatic quality degradation: -11.7% on AlpacaEval and -12.6% on IFEval. In contrast, removing tokens with high logits but low attribution scores has minimal impact (-0.4% and -0.3% respectively).

This asymmetry confirms that attribution identifies quality-critical tokens that logit magnitude alone misses. Tokens can have high logits due to fluency or associative factors yet contribute little to instruction-following quality.

G Case Studies

G.1 Constraint Satisfaction Example

Prompt: Explain photosynthesis in exactly 3 sentences.
Top-$n\sigma$: Photosynthesis is the process by which plants convert sunlight into energy. During this process, plants absorb carbon dioxide and water. They then release oxygen as a byproduct. This process is essential for life on Earth. (4 sentences: constraint violated)
SEAL: Photosynthesis is the process by which plants convert sunlight, water, and carbon dioxide into glucose and oxygen. This occurs primarily in the chloroplasts of plant cells, where chlorophyll captures light energy. The process is fundamental to life on Earth, providing both food and oxygen. (3 sentences: constraint satisfied)

Table 17: Constraint satisfaction comparison. SEAL maintains awareness of the sentence count constraint throughout generation.

In tasks requiring strict adherence to numerical constraints, standard sampling methods often prioritize linguistic completeness over instruction following. As shown in Table 17, the baseline includes a grammatically natural but superfluous concluding sentence, violating the "exactly 3 sentences" constraint. SEAL, by leveraging prompt-relevance attribution, successfully suppresses this high-probability continuation to satisfy the explicit count requirement.

G.2 Format Following Example

Prompt: List exactly 5 benefits of exercise. Use bullet points.
Top-$n\sigma$: Here are some benefits of exercise: • Improves cardiovascular health • Helps maintain healthy weight • Boosts mood and mental health • Increases energy levels • Strengthens muscles and bones • Improves sleep quality • Reduces risk of chronic diseases (7 items: constraint violated)
SEAL: • Improves cardiovascular health and reduces heart disease risk • Helps maintain a healthy weight through calorie burning • Boosts mood by releasing endorphins • Increases energy levels and reduces fatigue • Strengthens muscles, bones, and joints (5 items: constraint satisfied)

Table 18: Format following comparison. SEAL adheres to the specified count while top-$n\sigma$ overshoots.

List generation frequently suffers from "run-on" errors, where the model continues generating semantically associated items beyond the requested limit. Table 18 illustrates that while additional exercise benefits are statistically probable (high logit values), they are not supported by the "exactly 5" constraint. SEAL effectively filters these prompt-irrelevant extensions.

H Failure Modes

Ambiguous Prompts. When prompts lack clear requirements (e.g., "Tell me something interesting"), attention distributes diffusely across prompt tokens, yielding a prompt representation $\mathbf{v}_p$ that approximates the mean of all prompt embeddings. Relevance scores r_i become less discriminative, and SEAL degrades toward top-$n\sigma$ behavior. This represents graceful degradation rather than harmful failure: SEAL defaults to statistical filtering when prompt-relevance signal is weak.

Indirect Relevance. When correct tokens relate to the prompt through reasoning rather than semantic similarity, embedding-based relevance may fail. For example, answering "What is 37×23?" requires a numerical result that has no embedding similarity to the mathematical expression in the prompt. This limitation explains SEAL's smaller gains on GSM8K (Table 3) and HumanEval (Table 7) compared to IFEval: in mathematical reasoning and code generation, correct tokens are often logically or structurally, rather than semantically, connected to the prompt. Future work could explore combining SEAL with chain-of-thought

prompting (Wei et al., 2022b), where intermediate reasoning steps may exhibit stronger semantic connections to the prompt.

Negated Instructions. For instructions of the form "do not mention X," the token "X" may have high semantic similarity to the prompt (and thus positive r_i), yet should be avoided. SEAL does not explicitly handle this case; suppression of "X" is delegated to the instruction-tuned model's own logit assignment. In practice, this is sufficient for typical negated instructions because instruction tuning lowers the logit of the forbidden token substantially, but it remains a failure mode for prompts with complex negation or counterfactual structures. Addressing this directly, e.g. by conditioning the sign of the rectification on detected negation cues, is left for future work.

I Computational Profiling

Operation	Time (ms)	% Overhead
Statistical filtering	0.8	7.6%
Attention extraction	0.1	1.0%
Prompt representation	0.4	3.8%
Relevance computation	7.2	68.6%
Attribution computation	1.2	11.4%
Quantile + filtering	0.8	7.6%
Total	**10.5**	100%

Table 19: Detailed latency profiling (LLaMA-3-8B, A40 GPU).

As illustrated in Table 19, the computational bottleneck lies in the **Relevance Computation** stage, accounting for 68.6% of the total overhead. This step involves a matrix-vector product $\mathbf{r} = \mathbf{E}_{\mathcal{N}_{\text{stat}}} \mathbf{v}_{\text{p}}$, which requires gathering high-dimensional embedding vectors ($d = 4096$) for the candidate subset. While the candidate size $|\mathcal{N}_{\text{stat}}| \approx 500$ is significantly smaller than the full vocabulary size $|V|$, the operation remains memory-bound due to the random access patterns required to fetch non-contiguous embeddings from VRAM.

To further reduce latency for production-grade deployment, several optimizations can be applied:

- Sparse embedding matrices for common token subsets
- Quantized embeddings (INT8) for approximate computation
- Caching relevance scores for repeated prompt prefixes

J Notation Summary

Symbol	Description
$\mathbf{x}$	Input prompt
$\mathbf{c}$	Generation context (previously generated tokens)
ℓ_i	Logit for token i
$s_{\text{rel}}(i)$	Prompt-relevance signal for token i
$\boldsymbol{\alpha}$	Attention weights from current position to prompt
$\mathbf{H}_{\text{p}}$	Prompt hidden states, $\in \mathbb{R}^{n \times d}$
$\mathbf{v}_{\text{p}}$	Attended prompt representation, $\in \mathbb{R}^{d}$
r_i	Relevance score for token i
A_i	Attribution score for token i
$\mathcal{N}_{\text{stat}}$	Statistically pre-filtered candidate set
$\mathcal{N}_{\text{final}}$	Attribution-filtered final candidate set
n_σ	Pre-filter threshold (default: 1.0)
p_{attr}	Attribution quantile (default: 0.5)
T	Temperature parameter

Table 20: Summary of notation used throughout the paper.

To facilitate a clearer understanding of the theoretical framework derivation (Section 2) and the algorithmic details (Section 3), we provide a comprehensive reference of all mathematical symbols and notations used throughout this paper in Table 20.