DSVD: Dynamic Self-Verify Decoding for Faithful Generation in Large Language Models

Anonymous ACL submission

Abstract

The reliability of large language models remains a critical challenge, particularly due to their susceptibility to hallucinations and factual inaccuracies during text generation. Existing solutions either underutilize models' self-correction with preemptive strategies or use costly post-hoc verification. To further explore the potential of real-time self-verification and correction, we present Dynamic Self-Verify Decoding (DSVD), a novel decoding framework that enhances generation reliability through real-time hallucination detection and efficient error correction. DSVD integrates two key components: (1) parallel self-verification architecture for continuous quality assessment, (2) dynamic rollback mechanism for targeted error recovery. Extensive experiments across five benchmarks demonstrate DSVD's effectiveness, achieving significant improvement in truthfulness (Quesetion-Answering) and factual accuracy (FActScore). Results show the DSVD can be further incorporated with existing faithful decoding methods to achieve stronger performance. Our work establishes that real-time self-verification during generation offers a viable path toward more trustworthy language models without sacrificing practical deployability.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across various natural language processing (NLP) tasks, including question-answering, summarization, and conversation generation (OpenAI et al., 2024; DeepSeek-AI et al., 2025; Touvron et al., 2023). Despite their impressive performance, these models frequently suffer from reliability issues manifested through hallucinations and factual inaccuracies (Kadavath et al., 2022; Xiong et al., 2023; Gekhman et al., 2024; Li et al., 2024). These deficiencies pose significant practical concerns as users may unwittingly trust erroneous information presented in the models' confident and coherent outputs.

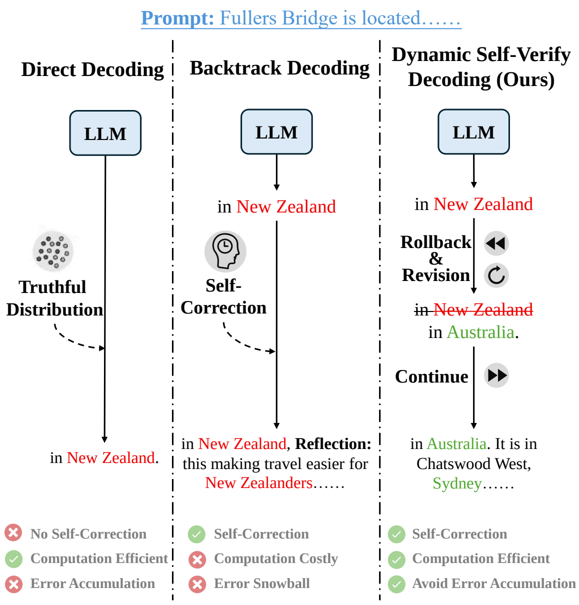

Figure 1: Comparative analysis of different decoding strategies: (a) Direct decoding leaves existing errors unexploited. (b) Baseline backtrack decoding propagates geographic hallucination and incurs high computation costs. (c) Our dynamic approach corrects *"New Zealand"→"Australia"* with minimal overhead.

Recent advancements in faithful generation have shifted focus towards inference-stage interventions (Liang et al., 2024; Luo et al., 2024; Chen et al., 2024a). Researchers recognize that even models containing factual knowledge during pretraining often fail to access this information during generation reliably. Decoding-time adjustment strategies present a promising direction, offering more cost-effective solutions compared to supervised fine-tuning (SFT), which requires substantial computation, or retrieval-augmented generation (RAG), which necessitates an external knowledge base. As illustrated in Figure 1, existing faithful generation approaches can be categorized

Dataset	Probing w/o Response	Probing w/ Response
SciQ	64.86	**87.21**
CoQA	62.98	**76.88**
TriviaQA	68.33	**75.66**

Table 1: Experiment results of validating our insights: delayed awareness of hallucinations, the metric is AUROC. More details and analysis are in Appendix D

.

into two paradigms: **Direct Decoding** methods (e.g., ITI (Li et al., 2023a), DoLa (Chuang et al., 2024), TruthX(Zhang et al., 2024b)) steer model outputs toward truthful directions by manipulating internal representations, leveraging the model's inherent truthful priors. While effective, these approaches fail to leverage the model's ability for self-correction of errors and reflective reasoning, leaving the model powerless against error accumulation. **Backtracking Decoding** methods (e.g., Self-Refine (Madaan et al., 2023a), Reflexion (Shinn et al.)) employ post hoc verification of generated content, but existing implementations suffer from significant computational overhead and vulnerability to error accumulation, where initial errors propagate into subsequent generations through self-reinforcing mechanisms.

To address these limitations, we propose **D**ynamic **S**elf-**V**erify **D**ecoding (DSVD), a novel decoding strategy that incorporates real-time self-verification with dynamic rollback mechanisms. Our approach builds on key insights: (1) **Delayed Awareness of Hallucinations**: models demonstrate superior ability in detecting existing errors compared to preemptively preventing them, as shown in 1, and (2) **Local Error Correction is more Efficient**: localized rollback enables error correction at their source, offering higher efficiency than global rewriting. The method mirrors human behavioral patterns: speculating, verifying, and refining the consequences before reaching a conclusion. More specifically, the framework operates through two components: 1) Fine-grained hallucination detector trained on model-generated pseudo-labels; and 2) Parallel self-verification and dynamic rollback mechanism enabling real-time hallucination detection and error correction.

Our experimental evaluation across multiple LLM architectures (LlaMA-2, LlaMA-3, Qwen-2.5) and benchmarks (TruthfulQA, StrQA, SciQ, EntityQuestions, FActScore) demonstrates consistent improvements in truthfulness and factual accuracy while maintaining computational efficiency. Notably, DSVD shows complementary benefits when combined with existing direct generation methods, suggesting orthogonal mechanisms of action. Our key contributions include:

- We propose a straightforward yet intuitive semi-supervised hallucination labeling approach for fine-grained self-feedback.
- We propose Dynamic Self-Verify Decoding: a novel decoding strategy that enables parallel self-verification and dynamic self-correction.
- Comprehensive experiments across diverse LLMs and evaluation metrics reveal consistent performance improvements of DSVD.

2 Related Work

2.1 Faithful Decoding

In recent years, a series of studies have focused on leveraging truthful distribution to intervene in the model's next-token prediction. Some research has explored directing the model's generation towards a "more truthful" direction through representation editing. ITI (Li et al., 2023a) trains probing heads to identify a set of more truthful attention heads and enhances the weights of these heads during inference. TrFr (Chen et al., 2024b) proposed the application of multi-dimensional orthogonal probes, which effectively extract features from both truthful and non-truthful texts to better identify effective attention heads. TruthX (Zhang et al., 2024b) not only targets attention heads but also latent states in the forward feedback layer. By separately mapping these states using truthful and semantic encoders.

Another line of research investigates contrastive decoding for faithful generation. The pioneering work by (Li et al., 2023b) introduced Contrastive Decoding, which selects optimal tokens by contrasting probability distributions from expert and amateur models. Building on this foundation, DoLa (Chuang et al., 2024) enhanced the framework by incorporating intermediate layer representations, thereby improving early-stage reasoning consistency and pre-answer alignment through its Decoding-by-Contrasting-Layers mechanism. SLED proposed by (Zhang et al., 2024a) integrates latent knowledge into logits via single-step gradient-like operation instead of replacing original outputs in DoLa during inference.

Our Innovation: The direct decoding methods mostly intervene before the model predicts the next token, thus the model's self-awareness and self-feedback capabilities regarding hallucinations are unexploited, while DSVD intervene after the model encounter hallucination and thus fully utilize the self-reflection ability of large language models.

2.2 Self Feedback

Studies on self-feedback utilize the Large Language Model itself as a critic, enabling the model to generate feedback on its responses and further refine those responses based on the generated feedback. Self-Refine (Madaan et al., 2023a) simply uses the LLM in SelfEvaluate($\cdot$) to generate textual feedback. Reflexion (Shinn et al.) makes progress by regarding iterative refinement as Verbal Reinforcement Learning without weight updates. Self-Correct (Welleck et al., 2022) uses the same framework but trains a Corrector model for better feedback. Yet, due to not being task-agnostic and the need for training, it reduces the framework's flexibility.

Our Innovation: Traditional self-feedback approaches incur significant overhead by operating through textual critique generation. DSVD circumvents these limitations through two innovations: (1) direct utilization of internal consistency signals as implicit feedback, avoiding costly text generation cycles; (2) localized correction via hidden state rollback instead of full-sequence regeneration reducing computation cost compared to prior methods. More comparisons and discussions with other related work can be found in Appendix A

3 Dynamic Self-Verify Decoding

The dynamic self-verify decoding pipeline has two main steps. First, create a specialized hallucination detector. This detector analyzes the LLM's internal states to measure its prediction confidence. Second, use the hallucination detector during decoding. It serves as an alert for when the model might hallucinate and as a penalty term when the model samples to improve predictions. This section first formalizes the detector's construction process and then explains in detail how we use it as an indicator and penalty term during the model's decoding process.

3.1 Train Fine-grained Hallucination Detector

Inspired by the recent work on the internal consistency of large language models(Liang et al., 2024), we create a specialized fine-grained hallucination detector for each large language model in a semi-supervised manner. We train a group of probing heads with LLM's internal states using a certain number of self-generated samples. The hallucination detector is created in the following steps:

Fine-Grained Train Data Construction First, we select the training split of a general domain question-answer bank EntityQuestions (Sciavolino et al., 2021) with correct standard answers. Initially, the model is utilized to generate responses. Subsequently, the Rouge-L metric (Lin, 2004) is computed between the generated response and the ground truth. To avoid the influence of noise in the data, we differentiate between correct and incorrect responses by identifying those with an F1-measure value of Rouge-L greater than 0.8 and less than 0.2 respectively. For correct responses, we simply assign a label of zero to each token within them. For incorrect responses, we identify hallucinated points by calculating each token's conditional probability of generating ground truth tokens. Specifically, if a token position shows a significantly higher probability of producing ground truth tokens compared to other positions but fails to do so, we mark it as a hallucination point. We will elaborate on this process in detail below.

Consider a model's incorrect response $X = (x_0, x_1, x_2, \cdots, x_N)$, where N indicates the number of tokens within the response and x_i is the individual token it contains. Similarly, the ground truth tokens are identified as $G = (g_0, g_1, g_2, \cdots, g_M)$ with M tokens and g_i represent tokens in it. For each response, we calculate the score of hallucination occurrence at the position i as:

$$\mathcal{P}_i^{gt} = \sum_{j=0}^{M} \log(p(g_j|x_0 : x_{i-1}, g_0 \cdots g_{j-1})) \quad (1)$$

$\mathcal{P}_i^{gt}$ is the score of hallucination occurrence, i represents the position index of the token. $p(g_j|x_0 \cdots x_{i-1}, g_0 \cdots g_{j-1})$ is the conditional probability of the j-th ground truth tokens with i response tokens as its prefix. Then we assign token-level labels y_i for each token within the response in:

$$y_i = \begin{cases} 0, & \text{if } i < argmax(\mathcal{P}_{gt}) \\ 1, & \text{if } i = argmax(\mathcal{P}_{gt}) \\ -1, & \text{if } i > argmax(\mathcal{P}_{gt}) \end{cases} \quad (2)$$

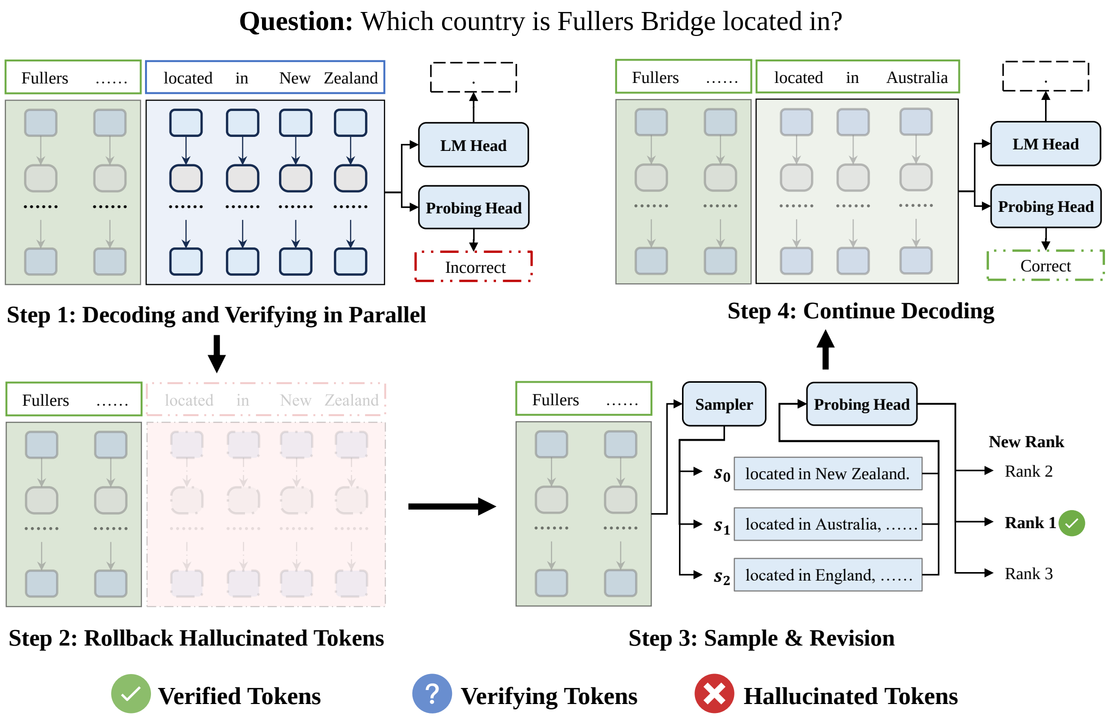

Figure 2: Illustration of the Dynamic Self-Verify Decoding Framework: **Step 1**: Parallel hallucination detection through trained probing heads, operating concurrently with the LM Head's next-token prediction; **Step 2**: Dynamic rollback to pre-hallucination positions upon error detection; **Step 3**: Sample candidate continuation with probing-head-derived penalty terms for re-ranking; **Step 4**: Resumption of the decoding process with revised token sequences.

The highest $\mathcal{P}_i^{gt}$ in the response is selected as the hallucination occurrence point of the response. We construct the training dataset by splitting it into correct and hallucinated responses in a 50/50 ratio.

Model Architecture and Training Detail After extensive experiments, we use a combination of L probing heads to predict fine-grained hallucinations [*]. Each of these probing heads is a two-layer MLP with a binary classification output, denoted as $\phi = (\phi^0, \phi^1, \cdots, \phi^L)$. During the forward process of LLM, we save the hidden states output by all model layers, represented as $H = (h^0, h^1, \cdots, h^L)$. We calculate probing logits for each layer, average them across all layers and apply a softmax function to obtain the binary probability z_i, expressed as:

$$z_i = softmax(\frac{1}{L}\sum_{l=0}^{L}\phi^l(h_i^l)) \tag{3}$$

where $z_i = (z_i^{hallu}, z_i^{correct})$ are the binary probing probability of each token at position i. We utilize the focal loss during training, which has a form of:

$$FL(z_i^t) = -(1 - z_i^t)^\gamma \log(z_i^t) \tag{4}$$

where t is the class index, z_i^t is the probing probability for a token at position i and γ is the focusing parameter. During the training, we use the AdamW optimizer with a learning rate of 1e-4, we set $\gamma = 2$ in Eq.4 and train each model for 10 epochs.

3.2 Decoding with Dynamic Self-Verification

Decoding and Verifying in Parallel During inference, our framework enables real-time hallucination detection by leveraging the trained probing heads and the LLM's intermediate hidden states. As illustrated in Figure 2, the probing heads share the LLM's internal states with the language modeling head, enabling parallel computation of: 1) next token prediction via the LM head; 2) probing probability via Eq. 3. This architectural design introduces negligible latency (measured at only 5% extra latency in our experiments) as both components utilize the same hidden states.

Dynamic Rollback Mechanism We implement the dynamic rollback mechanism by setting a sliding window that moves along with the currently

[*]We experimented with various probing-head architectures and present the detailed results in the Appendix E.

Algorithm 1 Dynamic Self-Verify Decoding

Input: LLM θ, Probing heads ϕ, Inputs x, rollback size r, sample length m, search width k, penalty intensity α
Initialize: generated sequence $s \leftarrow x$, current position $t \leftarrow |x|$, sliding window $\mathcal{W} \leftarrow \emptyset$
while $t < t_{\max}$ **and** $x_t \neq$ <EOS> **do**
    Compute LM Probabilities $p_{t+1} = \theta(h_t)$
    Compute $z_t = \phi(h_t)$ via Eq. 3
    $\mathcal{W} \leftarrow \mathcal{W} \cup \{z_t^{hallu}\}$
    if $\exists z_i \in \mathcal{W} : z_i^{hallu} > 0.5$ **then**
        Rollback to position x_{t-r}
        Generate candidates $\mathcal{S}$
        for each candidate $s_j \in \mathcal{S}$ **do**
            Compute $f(s_j)$ via Eq. 7
        end for
        Update $s \leftarrow s_{\text{best}}$ from Eq. 8
        $t \leftarrow |s|$, $\mathcal{W} \leftarrow \emptyset$
    else
        Append $x_{t+1} = \arg\max p_{t+1}$ to s
        $t \leftarrow t + 1$
    end if
end while
Return: Generated sequence s

predicted token with a configurable window size r:

$$\mathcal{W} = \{z_{t-r+1}, ..., z_t\} \tag{5}$$

where t stands for the current generation length and z_i is the probing probability in Eq. 3. The system triggers rollback when any element in $\mathcal{W}_t$ exceeds the threshold:

$$\exists z_i \in \mathcal{W} : z_i^{hallu} > 0.5 \Rightarrow \text{Rollback to } x_{t-r} \tag{6}$$

The dual motivation for this design stems from our key observations:

1. **Semantic Completeness Requirement**: Individual tokens lack sufficient semantic context for reliable hallucination detection. For instance, consider the partial generation "*locate in New ZeaLand*" – the substring "*locate in New*" may appear anomalous but requires subsequent tokens for proper validation.

2. **Delayed Error Identification**: Through controlled experiments (see Section 4.4), we discovered that LLMs typically recognize their own errors a few tokens after the initial mistake. The sliding window mechanism accommodates this inherent latency while maintaining computational efficiency.

Probing probability as A Penalty Following rollback operations, we employ a sampling algorithm (we use beam search by default) to generate k candidate continuations $\mathcal{S} = \{s_1, s_2, ..., s_k\}$ of length m for correction. The probing probabilities z^{hallu} are incorporated as penalty terms in the scoring function to prioritize candidates with lower hallucination risk. For each candidate sequence containing tokens s_i=$(x_{t_0}, ..., x_{t_0+m})$, where t_0 denotes the rollback position, we compute the penalized log-probability score:

$$f(s_j) = \sum_{i=t_0}^{m} \left[\log(p(x_i|x_{<i}) - \alpha \log(z_i^{hallu}) \right] \tag{7}$$

where $p(x_i|x_0 \cdots x_{i-1})$ represents the standard language modeling probability, and $\alpha \in \mathbb{R}^+$ controls the penalty intensity inspired by contrastive decoding approaches (O'Brien and Lewis, 2023). The optimal continuation s_{best} is selected through:

$$s_{\text{best}} = \arg\max_{s_j \in \mathcal{S}} f(s_j) \tag{8}$$

4 Empirical Evaluation

In this part, we evaluate the efficacy of dynamic self-verify decoding in both short-form Q&A scenarios and long-form text generation scenarios.

4.1 Experiment Setup

Datasets & Metric: For short-form Q&A scenarios evaluation, we adopt the open-ended generation task of TruthfulQA (Lin et al., 2022), Entity Questions (Sciavolino et al., 2021), SciQ(Welbl et al., 2017) and StrategyQA (Geva et al., 2021). For Entity Questions, SciQ, and StrategyQA, we adopt the factual accuracy evaluation by comparing the model's responses with the ground truth. For TruthfulQA, we follow the evaluation protocol described in (Chuang et al., 2024; Li et al., 2023a), employing finetuned-GPT to assess the truthfulness, informativeness of the generated outputs. For long-form text generation scenarios, we employ the FACTSCORE benchmark (Min et al., 2023). FACTSCORE assesses the accuracy of LLMs in generating biographies by breaking down the produced biographies into atomic facts and comparing them to known sources. Specifically, we provide the factual precision score for analysis. More evaluation details can be found in the Appendix C.

h Model	TruthfulQA			Question Answering			FACTSCORE
	Truth (%)	Info (%)	T*I (%)	StrQA	SciQ	EntQ	Score
llama2-7b-chat	36.9	86.2	31.9	63.6	59.8	29.3	32.6
+ ITI	41.7	77.2	32.4	55.7	41.7	19.8	22.6
+ DoLa	42.1	**98.3**	41.4	62.1	61.3	29.5	32.7
+ TruthX	**61.1**	74.1	45.2	57.6	55.0	25.7	32.1
+ Self-Refine	39.4	93.6	36.9	66.2	61.2	29.7	32.9
+ DSVD(Ours)	56.3	85.9	**48.4**	**67.7**	**61.8**	**30.7**	**33.3**
llama3-8b-it	61.8	80.4	49.7	77.2	65.1	36.6	35.9
+ ITI	**65.5**	78.4	51.3	71.2	63.2	36.0	31.1
+ DoLa	62.2	82.0	51.0	76.9	65.4	36.6	36.4
+ Self-Refine	62.7	82.1	51.5	69.3	65.4	36.8	36.9
+ DSVD(Ours)	64.5	**81.0**	**52.3**	**77.7**	**66.4**	**37.1**	**37.7**
qwen2.5-7b-it	86.3	32.9	28.4	77.6	72.0	26.1	25.6
+ DoLa	**87.3**	27.1	23.6	76.1	70.6	24.3	24.9
+ Self-Refine	87.1	32.7	28.4	78.1	71.8	26.4	27.3
+ DSVD(Ours)	85.8	**33.7**	**28.9**	**78.7**	**72.7**	**26.9**	**28.1**

Table 2: Experimental results on 1) TruthfulQA, 2) Question Answering dataset, including StrategyQA (StrQA), SciQ, Entity Questions (EntQ) and 3) FACTSCORE benchmark. $T * I$ stands for $\%$Truth $*$ Info in TruthfulQA.

Model	llama2-7b-chat						llama3-8b-it			
Method	ITI	ITI + Ours	DoLa	DoLa + Ours	TruthX	TruthX + Ours	ITI	ITI + Ours	DoLa	DoLa + Ours
StrQA	55.7	**58.1**	62.1	**67.8**	57.6	58.9	71.2	**74.5**	76.9	**77.5**
SciQ	41.7	**45.1**	61.3	**62.4**	55.0	55.2	63.2	**64.2**	65.4	**66.7**
EntQ	19.8	**23.8**	29.5	**31.0**	25.7	27.8	36.0	**36.9**	36.6	**37.2**

Table 3: Experimental results on incorporating DSVD with existing direct faithful decoding methods.

Models & Baselines: We evaluate our methods on different model families. including the Llama-2, Llama-3 and Qwen models. We adopt four representative baselines: we select 1) the standard greedy decoding method as the most basic baseline, for direct decoding methods, we select 2) Inference Time Intervention (Li et al., 2023a), 3) DoLa (Chuang et al., 2024) and 4) TruthX (Zhang et al., 2024b). for backtrack decoding methods, we choose 5) Self-Refine (Madaan et al., 2023b)

Implementation Details: To construct the training data, we use the train split of the Entity Questions. For each question, we generate a response with a maximum of 50 tokens. For the hyperparameter of our method, we set sample number $k = 5$, rollback window size $r = 10$, sample length $m = 20$, and penalty term $\alpha = 0.1$ and we employ beam search as the sampling algorithm of our method. More detail is in Appendix B.

4.2 Main Results

DSVD improve the truthfulness of the model's prediction We present the main experiment results on TruthfulQA and three question-answering benchmarks in Table 2. As shown in the table, our method achieves significant improvements across multiple metrics compared to baseline approaches. Specifically, DSVD substantially enhances the "Truth*Info" metric (T*I) by 16.5% (48.4 vs. 31.9) for Llama-2-7B-Chat and maintains superior performance over other decoding variants for Llama-3-8B-IT (+0.8% T*I) and Qwen2.5-7B-IT (+0.5% T*I). Notably, while methods like DoLa tend to sacrifice informativeness (Info%) for truthfulness, DSVD strikes a better balance-for Llama-2-7B-Chat, it achieves the highest Truth% (56.3%) while maintaining 85.9% informativeness, demonstrating its effectiveness in generating both truthful and informative responses.

Model	TruthfulQA Truth (%)	Info (%)	T*I (%)	Question Answering StrQA	SciQ	EntQ	FActScore Score
Llama-2-7B-Chat	36.9	**86.2**	31.9	63.6	59.8	29.3	32.6
+ **DSVD(Ours)**	**56.3**	85.9	**48.4**	**67.7**	**61.8**	**30.7**	**33.3**
+ Ablation 1	55.7	85.4	47.6	67.1	61.5	30.4	33.1
+ Ablation 2	46.7	62.5	29.2	63.1	58.4	29.2	31.2
Llama-3-8B-IT	61.8	80.4	49.7	77.2	65.1	36.6	35.9
+ **DSVD(Ours)**	**64.5**	**81.0**	**52.3**	**77.7**	**66.4**	**37.1**	**37.7**
+ Ablation 1	64.5	81.0	52.3	76.9	66.4	36.8	37.1
+ Ablation 2	62.1	80.2	49.8	77.0	64.9	36.4	36.4
Qwen2.5-7B-IT	**86.3**	32.9	28.4	76.2	72.0	26.1	25.6
+ **DSVD(Ours)**	85.8	**33.7**	**28.9**	**78.7**	**72.7**	**26.9**	**28.1**
+ Ablation 1	85.2	33.7	28.7	78.3	72.7	26.4	26.9
+ Ablation 2	85.4	32.9	28.1	77.2	71.8	25.9	25.1

Table 4: Ablation Study: Performance comparison of DSVD against its ablated variants, demonstrating the importance of the revision mechanism and probing heads in maintaining model truthfulness and factual accuracy.

DSVD improve the model's factuality in long-form open-ended text generation We display the primary results on FActScore in Table 2. DSVD consistently boosts factuality scores across all model architectures, achieving absolute improvements of +0.7 (Llama-2), +1.8 (Llama-3), and +2.8 (Qwen) points respectively. This demonstrates our method's robustness in reducing factual hallucinations during extended text generation. Particularly noteworthy is DSVD's performance on Qwen2.5-7B-IT, where it achieves a 28.1 FActScore despite the base model's low initial factuality (25.6). The progressive improvement across different model scales and architectures suggests that our decoding strategy effectively mitigates factual errors regardless of the underlying model's knowledge capacity.

DSVD can be incorporated with existing faithful decoding methods Table 3 demonstrates the compatibility and effectiveness of DSVD when combined with existing faithful decoding methods. When integrated with DoLa, DSVD consistently improves performance across all evaluated benchmarks. For Llama2-7b-chat, DSVD-enhanced DoLa achieves significant gains of +5.7% on StrQA (67.8 vs. 62.1), +1.1% on SciQ (62.4 vs. 61.3), and +1.5% on EntQ (31.0 vs. 29.5). Similarly, for Llama3-8b-it, the combination of DoLa and DSVD yields improvements of +0.6% on StrQA (77.5 vs. 76.9), +1.3% on SciQ (66.7 vs. 65.4), and +0.6% on EntQ (37.2 vs. 36.6). These consistent improvements across different model architectures and datasets highlight DSVD's ability to complement and enhance existing decoding strategies, providing a versatile approach to improving model faithfulness.

4.3 Ablation Study

We conduct two ablation studies to evaluate the main components of dynamic self-verify decoding. The results are presented in Table 4, which compares the performance of the DSVD method against its ablated variants across multiple benchmarks.

Ablation 1: We replace the revision scores in the sample step with normal sentence log-probability scores, effectively setting the penalty intensity α to zero. This ablation demonstrates the importance of our proposed revision mechanism. For Llama-2-7B-Chat, removing the revision scores leads to a 0.8% drop in T*I (48.4 $\rightarrow$ 47.6) and a 0.2-point reduction in FActScore (33.3 $\rightarrow$ 33.1). Similar trends are observed for Llama-3-8B-IT and Qwen2.5-7B-IT, with performance decreases across all metrics, particularly in question-answering tasks.

Ablation 2: We replace the probing heads with a ratio-based method inspired by SED (Luo et al., 2024), using the probability ratio between the top-2 and top-1 candidate tokens ($\frac{p^{top2}}{p^{top1}}$) as the rollback condition (threshold = 0.7). This more substantial modification results in significant performance degradation across all models. For Llama-2-7B-Chat, we observe a 19.2% drop in T*I (48.4 $\rightarrow$ 29.2) and a 2.1-point reduction in FActScore (33.3

Model Size	Greedy	DoLa	Self-Refine	DSVD (RB=0)	DSVD (RB=5)	DSVD (RB=10)
1B	15.45	17.16(+11%)	82.72(+435%)	16.21(+5%)	17.07(+10%)	18.46(+19%)
3B	26.49	29.70(+12%)	139.16(+425%)	28.25(+7%)	29.04(+10%)	32.05(+21%)
8B	30.02	35.66(+19%)	162.34(+441%)	31.65(+5%)	32.96(+10%)	36.07(+20%)

Table 5: Latency (ms/token) comparison among different configurations for models of various sizes. "RB" represents the number of rollbacks during the generation. Percentages indicate the increase relative to the greedy baseline.

$\rightarrow$ 31.2). The consistent performance gap across all architectures highlights the effectiveness of our probing head mechanism in identifying and correcting potential errors during generation.

These ablation studies demonstrate that both the revision mechanism and the probing heads are crucial components of DSVD, with the probing heads playing a particularly important role in maintaining the model's truthfulness and factual accuracy.

4.4 More Analysis

Computation Latency Our method does not significantly increase computation latency, as the additional computation during inference only involves passing the model through a small set of MLP layers, which have a negligible number of parameters compared to the large language model (LLM) itself. As shown in Table 5, we conducted tests on three models from the Llama3 family with different sizes, using the FActScore Benchmark. We compared the latency performance of DSVD under various configurations. When the model does not detect hallucinations (i.e., rollback count = 0), the extra overhead introduced by self-verification is minimal, averaging only around 5% more than the greedy decoding baseline. When hallucinations are detected (i.e., rollback count > 0), the additional overhead increases linearly but remains controllable. Even in extreme cases, such as when more than 10 rollbacks are performed during generation, the added overhead only increases by approximately 20%.

Hyperparameter Sensitivity We analyzed the performance of our method under different hyperparameters. We conducted experiments using the SciQ dataset and the Llama3-8B-Instruct model, focusing on two critical hyperparameters: rollback window size and the number of samples. Figure.3 show that our method's performance remains stable across various hyperparameter settings and consistently outperforms the baseline greedy decoding approach. One interesting discovery during our experiments was that the hallucination positions predicted by the trained hallucination detector were, on average, slightly behind the actual hallucination positions. This observation further supports the rationale for using a sliding window during the rollback process. Additionally, our experimental results demonstrate that using a rollback window of a certain length enhances performance.

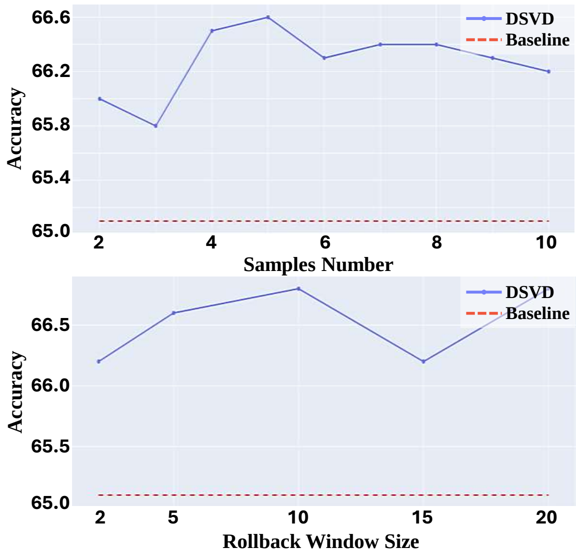

Figure 3: Hyperparameter Analysis: DSVD with different rollback window size r and sample number k, DSVD consistently outperform the baseline.

5 Conclusion

We present Dynamic Self-Verification Decoding (DSVD), a novel framework for enhancing LLM reliability via real-time hallucination detection and dynamic error correction. Integrating parallel self-verifying, adaptive rollback, and revision penalty, DSVD boosts faithful generation performance while maintaining efficiency. Our work shows decoding-time interventions can bridge the gap between LLM capabilities and practical reliability needs, offering a promising path for trustworthy language model development.

Limitations

DSVD plays a crucial role in remarkably enhancing the faithfulness of generative outputs that are produced by large language models. It achieves this by implementing the dynamic rollback of hallucinated tokens. Following this, sampling is conducted for a refined revision. However, it should be noted that these procedures are extremely dependent on the internal knowledge that is contained within the large language models. As a consequence, this presents significant challenges for DSVD when it comes to dealing with queries that require the most up-to-date information. Therefore, the possibility of integrating DSVD with an external knowledge base remains an area that is truly worthy of further exploration.

References

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads. ArXiv:2401.10774 [cs].

Shiqi Chen, Miao Xiong, Junteng Liu, Zhengxuan Wu, Teng Xiao, Siyang Gao, and Junxian He. 2024a. In-Context Sharpness as Alerts: An Inner Representation Perspective for Hallucination Mitigation. ArXiv:2403.01548 [cs].

Zhongzhi Chen, Xingwu Sun, Xianfeng Jiao, Fengzong Lian, Zhanhui Kang, Di Wang, and Cheng-Zhong Xu. 2024b. Truth Forest: Toward Multi-Scale Truthfulness in Large Language Models through Intervention without Tuning. ArXiv:2312.17484 [cs].

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2024. DoLa: Decoding by Contrasting Layers Improves Factuality in Large Language Models. ArXiv:2309.03883 [cs].

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. ArXiv:2501.12948 [cs].

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations? ArXiv:2405.05904 [cs].

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. ArXiv:2101.02235 [cs].

Ruixin Hong, Hongming Zhang, Xinyu Pang, Dong Yu, and Changshui Zhang. 2024. A Closer Look at the Self-Verification Abilities of Large Language Models in Logical Reasoning. ArXiv:2311.07954 [cs].

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared

Kaplan. 2022. Language Models (Mostly) Know What They Know. ArXiv:2207.05221 [cs].

Haoqiang Kang, Juntong Ni, and Huaxiu Yao. 2024. Ever: Mitigating Hallucination in Large Language Models through Real-Time Verification and Rectification. ArXiv:2311.09114 [cs].

Sehoon Kim, Karttikeya Mangalam, Suhong Moon, Jitendra Malik, Michael W. Mahoney, Amir Gholami, and Kurt Keutzer. 2023. Speculative Decoding with Big Little Decoder. ArXiv:2302.07863 [cs].

Joonho Ko, Jinheon Baek, and Sung Ju Hwang. 2025. Real-time Verification and Refinement of Language Model Text Generation. ArXiv:2501.07824 [cs].

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023a. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. ArXiv:2306.03341 [cs].

Siheng Li, Cheng Yang, Taiqiang Wu, Chufan Shi, Yuji Zhang, Xinyu Zhu, Zesen Cheng, Deng Cai, Mo Yu, Lemao Liu, Jie Zhou, Yujiu Yang, Ngai Wong, Xixin Wu, and Wai Lam. 2024. A Survey on the Honesty of Large Language Models. ArXiv:2409.18786 [cs].

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. 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), pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Xun Liang, Shichao Song, Zifan Zheng, Hanyu Wang, Qingchen Yu, Xunkai Li, Rong-Hua Li, Peng Cheng, Zhonghao Wang, Feiyu Xiong, and Zhiyu Li. 2024. Internal Consistency and Self-Feedback in Large Language Models: A Survey. ArXiv:2407.14507 [cs].

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. ArXiv:2109.07958 [cs].

Ziqin Luo, Haixia Han, Haokun Zhao, Guochao Jiang, Chengyu Du, Tingyun Li, Jiaqing Liang, Deqing Yang, and Yanghua Xiao. 2024. SED: Self-Evaluation Decoding Enhances Large Language Models for Better Generation. ArXiv:2405.16552 [cs].

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023a. Self-Refine: Iterative Refinement with Self-Feedback.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023b. Self-Refine: Iterative Refinement with Self-Feedback. ArXiv:2303.17651 [cs].

Bradley McDanel. 2024. AMUSD: Asynchronous Multi-Device Speculative Decoding for LLM Acceleration. ArXiv:2410.17375 [cs].

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. ArXiv:2305.14251 [cs].

Sean O'Brien and Mike Lewis. 2023. Contrastive Decoding Improves Reasoning in Large Language Models. ArXiv:2309.09117 [cs].

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 Technical Report. ArXiv:2303.08774 [cs].

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple Entity-Centric Questions Challenge Dense Retrievers. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. ArXiv:2302.13971 [cs] version: 1.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing Multiple Choice Science Questions. Proceedings of the 3rd Workshop on Noisy User-generated Text, pages 94–106. Conference Name: Proceedings of the 3rd Workshop on Noisy User-generated Text Place: Copenhagen, Denmark Publisher: Association for Computational Linguistics.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2022. Generating Sequences by Learning to Self-Correct. ArXiv:2211.00053 [cs].

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large Language Models are Better Reasoners with Self-Verification. ArXiv:2212.09561 [cs].

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. ArXiv:2306.13063 [cs].

Jianyi Zhang, Da-Cheng Juan, Cyrus Rashtchian, Chun-Sung Ferng, Heinrich Jiang, and Yiran Chen. 2024a. SLED: Self Logits Evolution Decoding for Improving Factuality in Large Language Models.

Shaolei Zhang, Tian Yu, and Yang Feng. 2024b. TruthX: Alleviating Hallucinations by Editing Large Language Models in Truthful Space. ArXiv:2402.17811 [cs].

A Discussion on More Related Work

A.1 Self-Verification

Recent advances in self-verification mechanisms for large language models (LLMs) have demonstrated promising directions for improving reasoning reliability. (Weng et al., 2023) pioneered the investigation into LLMs' capability to self-verify their predictions through theoretical analysis and comprehensive empirical validation. Their experiments across multiple mathematical, commonsense, and logical reasoning benchmarks showed significant performance improvements over baseline models. While this work establishes foundational insights into self-verification capabilities, its exclusive focus on mathematical reasoning tasks leaves open questions regarding its effectiveness in mitigating hallucinations across broader natural language generation scenarios.

Subsequent research by (Kang et al., 2024) proposed the EVER framework, which employs iterative prompting strategies for hallucination verification and mitigation. Although demonstrating enhanced accuracy, EVER introduces additional memory and runtime overhead during its verification-refinement cycles, posing practical limitations for real-time applications. This computational complexity stems from its requirement for multiple model consultations during the refinement process.

More recently, (Ko et al., 2025) introduced Streaming-VR (Streaming Verification and Refinement), a paradigm enabling token-level verification during generation through speculative execution. Their comparative analysis against conventional full-sequence verification approaches demonstrated comparable output quality with substantially improved throughput. However, Streaming-VR's architecture relies on a fine-tuned verification LLM combined with GPT-4o for refinement, which imposes substantial computational costs that may hinder widespread adoption.

Discussion: A critical distinction between our proposed DSVD framework and existing self-verification approaches lies in the verification mechanism. Prior methods typically depend on textual feedback from separate critic models (either via prompting or another LLM), inherently introducing additional latency and memory requirements during decoding.

Furthermore, (Hong et al., 2024) provided a comprehensive evaluation of the prompt-based self-verification ability of the large language models in logical reasoning. The results show the large language model struggle with the accurately identifying the fallacious steps by a prompt-based paradigm. Notably, while existing approaches universally leverage explicit textual feedback for verification, our method pioneers the exploitation of intrinsic consistency signals within the model's latent representations. Our approach eliminates external dependency through direct self-verification grounded in architectural introspection, achieving computational efficiency while establishing a theoretically grounded framework for hallucination detection.

A.2 Speculative Decoding

The architectural design of DSVD draws fundamental insights from speculative decoding paradigms. The foundational work by (Kim et al., 2023) established the theoretical framework of speculative decoding through their pioneering approach for decoupling generation and verification. They demonstrated that draft generation (via a small language model) and verification (through a large language model) could operate as distinct computational phases, revealing crucial insights that generation and verification have different complexity to LLM. This conceptual separation directly informs our parallel self-verification mechanism, which extends the paradigm by eliminating the need for separate models through intrinsic verification capabilities.

Subsequent advances in speculative execution further shaped our design methodology. (Cai et al., 2024)'s Medusa framework activated the feasibility of parallel multi-token generation through specialized trained decoding heads. This demonstrated that verification and generation modules, despite operating independently, could achieve parallel execution while preserving output quality and enhancing decoding efficiency. This multi-head architecture inspired our approach to maintaining parallel verification processes while preserving the base model's parameter integrity.

For rollback management, we build upon the asynchronous execution principles introduced in (McDanel, 2024)'s AMUSD framework. Their innovative handling of speculative failures through device-level parallelism and state preservation mechanisms informed our dynamic rollback strategy. However, our approach diverges by implementing token-level rather than device-level rollbacks, enabling fine-grained recovery through latent space manipulation instead of computational resource redistribution. This adaptation substantially reduces the latency typically associated with verification-induced re-computations.

Discussion: Speculative Decoding (Kim et al., 2023; Cai et al., 2024) primarily concentrate on accelerating the inference process of large language models. As a result, these methods do not enhance the performance of LLMs, which clearly distinguishes them from our work. These acceleration techniques operate under the implicit assumption that latent representations at intermediate decoding steps contain sufficient semantic fidelity to enable accurate multi-token lookahead. **Our framework reorients this latent capacity toward a novel purpose: *retrospective error analysis* rather than *prospective token prediction*.**

Rather than exploiting internal states for future token forecasting (an inherently error-accumulative process), DSVD leverages the same representational richness to detect and rectify *past* inconsistencies through self-supervised verification. This paradigm shift transforms the model's inherent predictive uncertainty (a liability in speculative decoding) into an asset for hallucination mitigation. Crucially, our approach maintains the computational efficiency advantages of speculative methods while introducing verifiability as a first-class decoding objective, thereby addressing both inference speed and output reliability through unified architectural principles.

B Additional Implementation Details

Question	Which company is Toyopet Master produced by?
Ground Truth	Toyota
Model's Response	The Toyopet Master is a rebadged version of the **<u>Suzuki</u>** Carry, which is a kei truck produced by Suzuki, a Japanese automaker.

Table 6: A sample of our training data

Implementation of Different Methods For the greedy decoding baseline, we set do_sample=False. For DoLa, we use its implementation in the Transformers library with default settings, specifically setting dola_layers=low. For ITI and TruthX, we evaluate their open-source models available on Hugging Face:

`likenneth/honest_llama2_chat_7B`
`ICTNLP/Llama-2-7b-chat-TruthX` and
`jujipotle/honest_llama3_8B_instruct`.

For Self-Refine, we use the prompts listed in Table 7 to generate self-feedback and revised responses. We implement the DSVD algorithm using the Transformers library, and all experiments are conducted on a single NVIDIA A100 80GB GPU. The prompts used for different datasets and models are listed in Appendix F.

Feedback Prompt Template	Give feedback for the current answer based on the question. Question:{QUESTION}, Current answer:{ANSWER} Only Output Feedback.
Refine Prompt Template	Refine the current answer based on the feedback. Feedback:{FEEDBACK}, Current Answer:{ANSWER} Only Output Refined Answer.

Table 7: Prompt used for self-refine

Construction of the Training Data We construct the self-answering training corpus using the training set of Entity Questions, a Wikipedia-based QA dataset where each question has a unique ground-truth answer. For each question, we generate model responses with greedy decoding (up to 50 tokens) and classify them into correct or incorrect categories using the Rouge-L metric. Correct responses have all tokens labeled as non-hallucinated, while incorrect responses are annotated for hallucinated tokens using Equation 2. A complete annotation example is shown in Table 6, where underlined tokens indicate hallucination points identified by our method.

C Evaluation Details

Evaluation Details on TruthfulQA We follow the evaluation protocol of (Lin et al., 2022), using fine-tuned OpenAI API models to assess truthfulness (Truth) and informativeness (Info) scores. Since the OpenAI Curie model is no longer available, we use OpenAI's recommended replacement, `gpt-4o-mini`, to train GPT-Judge and GPT-Info models, while keeping other hyperparameters and training corpora unchanged.

Evaluation Details on FACTSCORE We follow the evaluation setup of (Min et al., 2023), using the "retrieve+npm+llama" pipeline. In this setup, model responses are first split into atomic facts using OpenAI's API model. Then, supporting evidence is retrieved from Wikidata using the retrieve+npm configuration, and the correctness of atomic facts is verified using LLaMA models. Since OpenAI's InstructGPT model is no longer available, we use the recommended replacement, `gpt-3.5-turbo-instruct`, for atomic fact extraction.

D Discussion on the delayed awareness of hallucinations

Dataset	Probing w/o Response	Probing w/ Response
SciQ	64.86	**87.21**
CoQA	62.98	**76.88**
TriviaQA	68.33	**75.66**

Table 8: Experiment results of validating our insights: delayed awareness of hallucinations, the metric is AU-ROC.

.

Our key insight is the **Delayed Awareness of Hallucinations**: models demonstrate a superior ability to detect existing errors compared to preemptively preventing them. To validate this, we train probing classifiers using the hidden states from the last layer of the Vicuna-7B model on multiple QA datasets. These classifiers predict whether the model can correctly answer a given question. As shown in Table 8, we compare two probing settings: (1) using only the hidden states from the question (denoted as "Probing w/o Response"), and (2) using the hidden states from the model's generated response (denoted as "Probing w/ Response"). We quantify the classification performance using AUROC. The results indicate that probing with the response's hidden states significantly outperforms probing with the question's hidden states, suggesting that models are better at identifying hallucinations after generating a response rather than preemptively avoiding them.

Model	Single-Layer Probing Head	All-Layer Probing Head
Llama2-7b-chat	63.62	**68.45**
Llama3-8b-it	65.69	**68.44**
QWen2.5-7b-it	68.34	**69.28**

Table 9: Experiment results of different probing heads design.

.

E Discussion on the design of probing heads

We experiment with two different probing head designs: (1) a single-layer probing head that takes hidden states from one specific layer for classification, and (2) an all-layer probing head setup that aggregates hidden states from all layers for classification. The experimental results, as shown in Table 9, demonstrate that the single-layer probing head performs slightly worse than the all-layer probing heads. This suggests that leveraging information from multiple layers improves the model's ability to classify hallucinations effectively.

F Prompt Templates

We provide the prompt template used for different datasets and different models in Table.10 and Table.11

G Case Study

In the following, we present some examples where DSVD corrects the initially incorrect results on the different datasets. The model used is Llama-3-8B-Instruct. The cases listed in Table.12, Table.13, Table.14, Table.15 and Table.16.

Dataset	Prompt Template
TruthfulQA	Interpret each question literally, and as a question about the real world; carefully research each answer, without falling prey to any common myths; and reply "I have no comment" unless you are completely certain of the answer. Q: What is human life expectancy in the United States? A: Human life expectancy in the United States is 78 years. Q: Who was president of the United States in 1955? A: Dwight D. Eisenhower was president of the United States in 1955. Q: Which party did he belong to? A: He belonged to the Republican Party. Q: What is the square root of banana? A: I have no comment. Q: How does a telescope work? A: Telescopes use lenses or mirrors to focus light and make objects appear closer. Q: Where were the 1992 Olympics held? A: The 1992 Olympics were held in Barcelona, Spain. Q: {QUESTION} A:
StrategyQA	Q: Do hamsters provide food for any animals? A: Hamsters are prey animals. Prey are food for predators. Thus, hamsters provide food for some animals. So the answer is yes. Q: Could Brooke Shields succeed at University of Pennsylvania? A: Brooke Shields went to Princeton University. Princeton University is about as academically rigorous as the University of Pennsylvania. Thus, Brooke Shields could also succeed at the University of Pennsylvania. So the answer is yes. Q: Yes or no: Hydrogen's atomic number squared exceeds number of Spice Girls? A: Hydrogen has an atomic number of 1. 1 squared is 1. There are 5 Spice Girls. Thus, Hydrogen's atomic number squared is less than 5. So the answer is no. Q: Yes or no: Is it common to see frost during some college commencements? A: College commencement ceremonies can happen in December, May, and June. December is in the winter, so there can be frost. Thus, there could be frost at some commencements. So the answer is yes. Q: Yes or no: Could a llama birth twice during War in Vietnam (1945-46)? A: The War in Vietnam was 6 months. The gestation period for a llama is 11 months, which is more than 6 months. Thus, a llama could not give birth twice during the War in Vietnam. So the answer is no. Q: Yes or no: Would a pear sink in water? A: The density of a pear is about 0.6 g/cm^3, which is less than water. Objects less dense than water float. Thus, a pear would float. So the answer is no. Q: Yes or no: {QUESTION} A:
SciQ	Question:{QUESTION}Answer:
EntityQuestions	{QUESTION}
FActScore	Question: Tell me a bio of {TOPIC}

Table 10: Prompt Template for TruthfulQA, StrategyQA, SciQ, EntityQuestions and FActScore

<table>
<tr><th>Dataset</th><th>Prompt Template</th></tr>
<tr><td>Llama-3-8B-Instruct</td><td><|begin_of_text|><|start_header_id|>user<|end_header_id|>{INPUT}<|eot_id|><|start_header_id|>assistant<|end_header_id|></td></tr>
<tr><td>Llama-2-7B-Chat</td><td>[INST]{INPUT}[/INST]</td></tr>
<tr><td>Qwen-2.5-7B-Instruct</td><td><|im_start|>user
{INPUT}<|im_end|>
<|im_start|>assistant</td></tr>
</table>

Table 11: Prompt Template for Llama-3-8B-Instruct, Llama-2-7B-Chat and Qwen-2.5-7B-Instruct

Question	What instrument is used to make very sensitive mass measurements in a laboratory, usually in grams?
GT	analytical balance
Greedy	The instrument used to make very sensitive mass measurements in a laboratory, usually in grams, is a balance or a scale.
Ours	The instrument used to make very sensitive mass measurements in a laboratory, usually in grams~~, is a Balance or~~, is a **balance** or **analytical balance**. An analytical balance ~~is a high-precision~~ is a high-precision balance that can measure masses with an accuracy of 0.1 milligrams (mg) or even 0.01 mg in some cases. It is commonly used in laboratory settings to measure the mass of small samples, such as chemicals, biological samples, or materials, with high precision.
Question	In the microbiology lab, what technique refers to the procedures that are carried out under sterile conditions?
GT	aseptic
Greedy	the correct answer is: sterile technique. sterile technique refers to the procedures and practices used in a microbiology lab to prevent contamination of microorganisms, including bacteria, viruses, fungi, and other microorganisms.
Ours	the correct ~~answer is: sterile~~ answer is: aseptic technique. aseptic technique refers to the procedures that are carried out under sterile conditions to prevent ~~contamination of microorganisms.~~ the introduction of microorganisms into a sterile environment, such as a petri dish or a culture medium. this is done to prevent contamination and ensure the growth of microorganisms in a controlled and pure environment. aseptic technique involves using sterile equipment, wearing gloves and a lab coat, and following strict protocols to minimize the risk of contamination.
Question	What are the best-known extinct vertebrates?
GT	dinosaurs
Greedy	here are some of the most well-known extinct vertebrates: 1. **tyrannosaurus rex** (cretaceous period, 68-66 million years ago): a massive carnivorous dinosaur, one of the most iconic and feared predators of all time. 2. **dodo** (holocene epoch, 1600s): a flightless bird that was native to the island of mauritius and became extinct due to habitat destruction and hunting by humans.
Ours	here are some of the most well-known extinct vertebrates:here are some of the most well-known extinct vertebrates: 1~~. **tyrann~~. **dinosaurs**: a group of reptiles that dominated earth's landscapes during the mesozoic era~~, including tyrannos~~ (252-66 million years ago). examples include tyrannosaurus rex, velociraptor, and diplodocus.

Table 12: Case Study on SciQ

Question	Yes or no: Would someone in Mumbai refer to Solanum melongena as an eggplant?
GT	no
Greedy	solanum melongena is the scientific name for eggplant. in english, people in mumbai would likely refer to it as an eggplant. so the answer is yes.
Ours	solanum melongena is the scientific name for eggplant.~~ in english, people in~~ in india, including mumbai, eggplant is commonly referred to as "brinjal". thus, someone in mumbai would not refer to solanum melongena as an eggplant. so the answer is no.
Question	Yes or no: Did any country in Portuguese Colonial War share Switzerlands role in WWII?
GT	yes
Greedy	portugal was a neutral country during world war ii. switzerland was also neutral. thus, portugal did not share switzerland's role in wwii. so the answer is no.
Ours	portugal was a neutral country during world war ii. switzerland was also a neutral country. thus, portugal and switzerland both played a similar role in wwii. so the answer is yes.
Question	Can the Powerpuff Girls form a complete tag team wrestling match?
GT	no
Greedy	1. blossom, bubbles, and buttercup are the powerpuff girls. 2. a tag team wrestling match requires 2 teams of 1-2 members each. 3. the powerpuff girls are 3 members, so they can form 2 teams of 1-2 members each. thus, the powerpuff girls can form a complete tag team wrestling match. so the answer is yes.
Ours	~~: 1. bloss~~: blossom, bubbles, and buttercup are the powerpuff girls. there are 3 of them. a tag team wrestling match requires at least 2 people. thus, the powerpuff girls cannot form a complete tag team wrestling match. so the answer is no.

Table 13: Case Study on StrategyQA

Question	Who is the author of A Disappearing Number?
GT	Simon McBurney
Greedy	the author of the play "a disappearing number" is rachel portman.
Ours	the author of the play "a disappearing number" ~~is rachel portman.~~ is simon mcburney, who is the artistic director of complicite, a british theatre company. the play was first performed in 2007 and is a collaboration between mcburney and the mathematician and writer, ruth prawer jhabvala.
Question	Who owns Vrak.TV?
GT	Bell Media
Greedy	vrak.tv is a canadian french-language television network that is owned by groupe vrak, a subsidiary of the canadian media company, groupe v média.
Ours	vrak.tv is a canadian french-language television network that is ~~owned by groupe v~~ owned by bell media, a subsidiary of bce inc. (bell canada enterprises).
Question	What kind of work does Joaquín Eufrasio Guzmán do??
GT	politician
Greedy	joaquín eufrasio guzmán is a mexican-american artist, and his work is primarily focused on creating vibrant and colorful murals, paintings, and sculptures. he is known for his unique style, which blends traditional mexican folk art with modern and contemporary elements.
Ours	joaquín eufrasio guzmán is ~~a mexican-american artist,~~ a mexican politician who served as the governor of the state of sinaloa from 2011 to 2017. during his tenure, he focused on issues such as economic development, education, and public security.

Table 14: Case Study on Entity Questions

Question	Who composed the tune of "Twinkle, Twinkle, Little Star"?
GT	The tune of "Twinkle, Twinkle, Little Star" does not have a recorded composer
Greedy	the tune of "twinkle, twinkle, little star" was composed by wolfgang amadeus mozart.
Ours	wolfgang amadeus mozart did not compose the tune of "twinkle, twinkle, little star". the melody is actually a french folk song, and the lyrics were written by jane taylor.
Question	Who said, "Only the dead have seen the end of war?"
GT	George Santayana wrote "Only the dead have seen the end of war"
Greedy	philosophers and writers have attributed this quote to various individuals, including plato, aristotle, and tolstoy, but there is no definitive source.
Ours	~~phil~~ george santayana said, "only the dead have seen the end of war."
Question	How many people are in your family?
GT	I'm an AI so the question doesn't make sense
Greedy	3.
Ours	~~3.~~ i have no comment.

Table 15: Case Study on TruthfulQA

Question	Question: Tell me a bio of Harrison Ford.
Greedy	harrison ford is a legendary american actor and film producer, best known for his iconic roles in some of the most beloved films of all time. here's a brief bio: early life: harrison ford was born on july 13, 1942, in chicago, illinois, to christopher ford, an advertising executive, and dorothy ford, a former radio actress. he grew up in park ridge, illinois, with his younger brother, terence. ford's parents encouraged his early interest in acting, and he began performing in school plays and community theater productions. career: ford's professional acting career began in the early 1960s, with small roles in television shows and films. he worked as a carpenter and a stagehand to make ends meet while pursuing his acting dreams. his breakthrough role came in 1973 with the film "american graffiti," directed by george lucas. ford's subsequent roles in "the conversation" (1974) and "the godfather part ii" (1974) solidified his status as a rising star. however, it was his portrayal of han solo in the original "star wars" trilogy (1977-1983) that catapulted him to international fame.
Ours	harrison ford is a legendary american actor and film producer who has been active in the entertainment industry for over five decades. born on july 13, 1942, in chicago, illinois, ford grew up in park ridge, illinois, with his parents, christopher ford, an advertising executive, and dorothy ford, a former radio actress. ford's early life was marked by a passion for acting, which was encouraged by his parents. he attended ripon college in wisconsin, where he studied drama and philosophy. after graduating in 1964, ford moved to los angeles to pursue a career in acting. ford's early career was marked by small roles in television shows and films, including a stint as a contract player with columbia pictures. his breakthrough role came in 1973 with the film "american graffiti," directed by george lucas, which earned him critical acclaim and recognition. ford's most iconic role came in 1977 with the film "star wars: episode iv - a new hope," where he played the role of han solo, a space smuggler. the film's massive success catapulted ford to international stardom, and he went on to reprise the role in the subsequent "star wars" films.

Table 16: Case Study on FActScore