Avoidance Decoding for Diverse Multi-Branch Story Generation

Kyeongman Park
Seoul National University
zzangmane@snu.ac.kr

Nakyeong Yang
Seoul National University
yny0506@snu.ac.kr

Kyomin Jung
Seoul National University
kjung@snu.ac.kr

Abstract

Large Language Models (LLMs) often generate repetitive and monotonous outputs, especially in tasks like story generation, due to limited creative diversity when given the same input prompt. To address this challenge, we propose a novel decoding strategy, ***Avoidance Decoding***, that modifies token logits by penalizing similarity to previously generated outputs, thereby encouraging more diverse multi-branch stories. This penalty adaptively balances two similarity measures: (1) Concept-level Similarity Penalty, which is prioritized in early stages to diversify initial story concepts, and (2) Narrative-level Similarity Penalty, which is increasingly emphasized later to ensure natural yet diverse plot development. Notably, our method achieves up to 2.6 times higher output diversity and reduces repetition by an average of 30% compared to strong baselines, while effectively mitigating text degeneration. Furthermore, we reveal that our method activates a broader range of neurons, demonstrating that it leverages the model's intrinsic creativity.

1 Introduction

Human writers can craft entirely different texts from the same ideas. However, Large Language Models (LLMs) still struggle to reach human-level creativity in writing. Previous studies have found that even the state-of-the-art models such as GPT-4o (OpenAI, 2024) generate repetitive and monotonous patterns (Zhang et al., 2025; Wu et al., 2025a; Wenger and Kenett, 2025; Lagzian et al., 2025). This tendency limits LLMs' performance on tasks that require conceptual diversity and broad exploration, such as story generation (Park et al., 2024a; Materzok, 2025; Huang et al., 2024) and complex reasoning (Wang et al., 2024; Sun et al., 2025; Kirk et al., 2023; Wu et al., 2025b). Especially in story generation, the task poses a unique challenge: it must engage readers through creative ideas and narratives, making it crucial to ensure diversity when generating stories with LLMs.

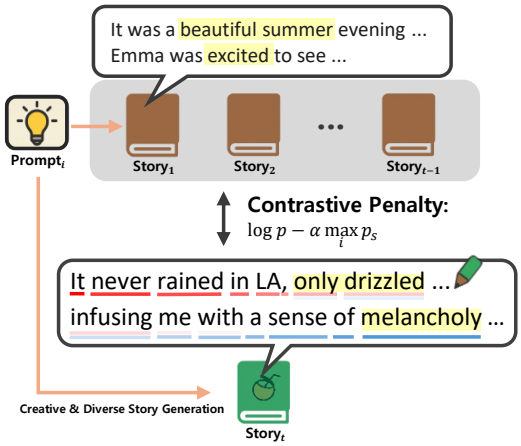

Figure 1: ***Avoidance Decoding*** for discouraging similarity to previously generated stories. Highlighted text demonstrates the most contrast in the story induced by the Similarity-based Contrastive Penalty. The red underlines mark the front regions where the Conceptual Similarity Penalty is primarily applied, and the blue underlines mark the backward regions where the Narrative Similarity Penalty is primarily applied.

Existing studies have attempted to increase the diversity of generated texts through decoding-time methods (Welleck et al., 2019; Holtzman et al., 2019; Nguyen et al., 2024; Vijayakumar et al., 2016). However, they have failed to achieve sufficient diversity since they only induce superficial token-level variations, without enriching conceptual, contextual, or narrative-level diversity. In addition, they have exhibited text degeneration due to an unresolved trade-off between diversity and fluency (Su et al., 2022; Arias et al., 2024).

To address these limitations, we propose a novel decoding strategy, ***Avoidance Decoding***. Our method introduces a new Similarity-based Contrastive Penalty that modifies the model's logits at each decoding step by penalizing similarity between the current output and multiple previously generated stories, which serve as negative samples. As a result, our method can substantially increase

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 7501–7517
November 4-9, 2025

the diversity of multi-branch stories, when given a single input prompt. Specifically, our Similarity-based Contrastive Penalty is a hybrid formulation of two distinct penalties: the Conceptual-level Similarity Penalty (**CSP**) and the Narrative-level Similarity Penalty (**NSP**). First, CSP is computed via similarity between the hidden states of candidate tokens for the next stepand negative samples to diversify initial concept representations. Second, NSP is computed via similarity between the sentence embeddings of the generated output and negative samples to ensure holistic narrative diversity. We assign higher weight to CSP in early stages to ensure diverse story planning, and gradually shift emphasis toward NSP as the current output length increases to ensure natural yet diverse plot progression, and apply the weighted sum as a final penalty.

As a result, without any additional training or stochastic sampling, our method achieves up to 2.6 times higher diversity than the best baseline according to LLM-based diversity evaluation. We also reduce repetition by at least 30% on average compared to the baseline in automated metrics such as BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and sentence similarity (Reimers and Gurevych, 2019), while maintaining robustness to text degeneration. Furthermore, we reveal that our method activates a broader range of neurons during iterative multi-branch story generation. This indicates that it leverages the model's intrinsic creative capacity, rather than merely introducing superficial token-level variations.

2 Related Works

2.1 LLM Decoding Strategies

Various decoding strategies exist for large language models (LLMs), some focusing on promoting diversity (Welleck et al., 2019; Holtzman et al., 2019; Fan et al., 2018; Vijayakumar et al., 2016; Nguyen et al., 2024; Zhu et al., 2023), increasing reliability (Hokamp and Liu, 2017; Wang et al., 2022; Chuang et al., 2023; Guo et al., 2025; Kim et al., 2025), and achieving more fluent and high-quality generation (Meister et al., 2023; Su et al., 2022; Li et al., 2022; Arias et al., 2024). In our case, we utilize contrastive-style decoding strategy to enhance diversity during generation (Su et al., 2022; Welleck et al., 2019; Arias et al., 2024; Li et al., 2022); unlike prior work that reduces token-level similarity to previously generated tokens, we increase context-level diversity among multi-branch samples by penalizing similarity to each previously generated output.

2.2 Diverse Story Generation

There exist various methods to boost diversity and creativity in story generation (Park et al., 2024a; Patel et al., 2024; Fan et al., 2018; Bae and Kim, 2024; Materzok, 2025; Vijayakumar et al., 2016). Some approaches generate diverse story branches in a tree structure from a single prompt, enhancing creativity and engagement (Materzok, 2025; Wen et al., 2023; Nottingham et al., 2024; Jaschek et al., 2019; Alabdulkarim et al., 2021). Others incorporate interactive story generation with human-in-the-loop branching at key decision points (Huang et al., 2024; Ghaffari and Hokamp, 2025). To the best of our knowledge, there is no method to enhance diversity during decoding time for multi-branch story generation, which is the main focus of our work.

3 Problem Definition

Given a fixed story prompt representing a core concept or initial idea of a story, ***Multi-Branch Story Generation*** focuses on generating multiple coherent and fluent story continuations in parallel, which are mutually diverse in their narrative trajectories. Suppose P_θ is a language model. Given a fixed story prompt p, we denote $\{x^1, ..., x^n\}$ as the set of stories that are generated in parallel by P_θ conditioned on p. Each x^i is generated by predicting the next token x_t^i from the distribution $P_\theta(x_t^i \mid p, x_{1:t-1}^i)$, where $x_{1:t-1}^i$ denotes the previously generated tokens. We define the pairwise similarity function $s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ where $\mathcal{X}$ denotes the space of all possible generated stories (e.g., token sequences), and $s(x_i, x_j)$ quantifies the semantic or structural similarity between two outputs x_i and x_j, such as BLEU or Sentence-Similarity. The objective of this task is to generate a set of semantically divergent stories $\{x^1, ..., x^n\}$ from a single prompt p by minimizing $s(x_i, x_j)$ for all $i \neq j$, thereby ensuring diversity in content and narrative structure while maintaining robustness against degeneration.

4 Methodology

We introduce a novel decoding strategy that steers the language model to generate outputs that do not resemble any of the negative samples while resisting degeneration. In this work, we treat pre-

viously generated outputs for the same input as negative samples, thereby enhancing diversity of multi-branch stories while mitigating degeneration.

4.1 Motivation

As a high-level justification for our method, we present a probabilistic motivation for contrastively modifying token logits using the maximum similarity penalty to negative samples. Suppose we have $P_\theta(x_t|x_{1:t-1}) = \prod_{j=1}^{t} P_\theta(x_j|x_{1:t-1})$ that is a language model, and $P_s(\mathcal{N}|x_{1:t})$ that represents the *match probability* to quantify how closely the generated text (with the next candidate token appended) resembles the negative sample texts, $\mathcal{N} = \{n_1, ..., n_N\}$. We then define $P\big(x_t \mid x_{1:t-1}, \neg\mathcal{N}\big)$ as the probability of generating token x at step t given previous context $x_{1:t-1}$ while explicitly avoiding any negative samples in $\mathcal{N}$, and $P_s\big(\neg\mathcal{N} \mid x_{1:t}\big)$ as the complementary *match-avoidance probability*, the probability that the generated text $x_{1:t}$ does not match any of the negative samples $\mathcal{N}$. Ideally, this avoidance probability should take into account dependencies between negative samples: $P_s\big(\neg\mathcal{N} \mid x_{1:t}\big) = \prod_{i=1}^{N}\big(1 - P_s(n_i \mid x_{1:t}, \neg n_{1:i-1})\big)$. This formulation naturally reflects our main goal: generating outputs that are dissimilar to any previously generated stories (i.e., negative samples). However, in practice, we approximate each conditional match probability as $P_s(n_i \mid x_{1:t}, \neg n_{1:i-1}) \approx P_s(n_i \mid x_{1:t})$, to reduce computational cost. Substituting this into the product and applying Bayes' rule yields:

$$\begin{aligned} P\big(x_t \mid x_{1:t-1}, \neg\mathcal{N}\big) &= \frac{P_\theta\big(x_t \mid x_{1:t-1}\big) P_s\big(\neg\mathcal{N} \mid x_{1:t-1}, x_t\big)}{P_s\big(\neg\mathcal{N} \mid x_{1:t-1}\big)} \\ &\propto P_\theta\big(x_t \mid x_{1:t-1}\big) P_s\big(\neg\mathcal{N} \mid x_{1:t}\big) \\ &= P_\theta\big(x_t \mid x_{1:t-1}\big) \prod_{i=1}^{N}\big(1 - P_s(n_i|x_{1:t})\big). \end{aligned} \tag{1}$$

As $P_s\big(\neg\mathcal{N} \mid x_{1:t-1}\big)$ is constant when optimizing for x_t. We then have an approximated formula:

$$\log P\big(x_t \mid x_{1:t-1}, \neg\mathcal{N}\big) \approx \log P_\theta(x_t \mid x_{1:t-1}) \; - \; \sum_{i=1}^{N} \alpha_i \, P_s(n_i|x_{1:t}). \tag{2}$$

using a scaled first-order approximation $\log(1 - p) \approx -\alpha_i \, p$ for each sample. Therefore, we apply a penalty to the original token logits ℓ_t (corresponding to the probability $P(x_t \mid p, x_{<t})$) resulting in the adjusted logits ℓ_t^* as follows:

$$\ell_t^* = \ell_t \; - \; \sum_{i=1}^{N} \alpha_i \, P_s(n_i|x_{1:t}). \tag{3}$$

The sum-based penalty in Eq. (3) considers all negative evidence, akin to an L_1-style regularization. To prevent over-penalization from negative-sample accumulation, we redefine the adjusted logits as ℓ_t', akin to an L_∞-style regularization:

$$\ell_t^* = \ell_t \; - \; \max_{i=1}^{N} \alpha_i \, P_s(n_i|x_{1:t}). \tag{4}$$

Greedy decoding then proceeds using these modified logits as:

$$x_t^* = \operatorname*{argmax}_{v \in \mathcal{V}} \ell_t^*(v). \tag{5}$$

4.2 Similarity-Based Contrastive Penalties

The *match probability*, $P_s(\mathcal{N}|x_{1:t})$, corresponds to the degeneration penalty introduced in prior work, *Contrastive Search* (Su et al., 2022). The prior work computes the degeneration penalty as the maximum cosine similarity between the hidden state of a candidate token x_t and those of the preceding tokens $x_{1:t-1}$. We extend this probability by aiming to degrade the similarity between the target output and multiple negative samples. Furthermore, we also design novel penalty terms that account for both concept- and narrative-level similarity to those samples.

4.2.1 Concept-level Similarity Penalty

We first propose the **CSP** (**C**oncept-level **S**imilarity **P**enalty) to diversify the concept representations of stories. Specifically, we calculate the CSP as maximum cosine similarity between last hidden representations of candidate tokens and those of all individual tokens of negative samples. Formally, we compute:

$$s_j^{\text{CSP}} = \max_{h' \in H^-} \cos(h_j, h') \tag{6}$$

where h_j is the last hidden state of the j-th candidate token, and H^- denotes the set of hidden states for all tokens in the negative samples.

This mainly encourages diversity in the low-level conceptual space, which significantly influences the diversity of the story planning.

4.2.2 Narrative-level Similarity Penalty

However, applying the CSP may degrade the overall coherence and naturalness of the story, as it directly disrupts the representations, particularly during the later stages of long-form generation. Therefore, we additionally propose the **NSP** (**N**arrative-level **S**imilarity **P**enalty) to encourage narrative distinction from negative samples. The NSP is computed by the cosine similarity between the embedding of the current output sentence (after appending the candidate token) and each negative sample, using Sentence-Bert. Formally, we compute :

$$s_j^{\text{NSP}} = \cos\big(E(y_{1:t} \oplus w_j),\, E(x^-)\big), \qquad (7)$$

where $E(\cdot)$ denotes the Sentence-BERT embedding function, $y_{1:t} \oplus w_j$ is the current output sentence after appending w_j, and x^- is negative sample sentence.

This penalty can significantly enhance plot diversity by reducing semantic similarity at a higher-level context without compromising the story's coherence and naturalness.

4.2.3 Concept-to-Narrative Hybrid Penalty

However, in the early stages of generation, when the output has not yet formed a meaningful length of sentence, the NSP is often too small to be effective.

Therefore, we integrate Concept- and Narrative-level Similarity Penalty into a hybrid formulation. Specifically, we rely primarily on the CSP in the early stage, until the decoding step reaches the inflection point T_0, ensuring diverse story planning. Then we progressively increase the weight of the NSP to ensure the coherence and naturalness of the generated stories. We define the mixing ratio γ as follows:

$$\gamma = \delta + (1 - \delta) \cdot \text{sigmoid}\big(t - T_0\big) \qquad (8)$$

where δ denotes the minimum weight assigned to CSP. Then the Concept-to-Narrative Hybrid Penalty is computed as:

$$s_j^{\text{hybrid}} = \gamma\, s_j^{\text{CSP}} \;+\; (1 - \gamma)\, s_j^{\text{NSP}} \qquad (9)$$

The term s_j^{hybrid} corresponds to the match probability $P_s\big(n_i \mid x_{1:t}\big)$ in Equation 4, after appending the candidate token w_j to the generated text $x_{1:t-1}$.

4.3 Overall Decoding Procedure

As shown in Algorithm 1, we first compute the number of candidate tokens k and the adaptive penalty weight α_{ACS}, following the process in the prior study (Arias et al., 2024). Next, we apply Equations 8 and 9 to compute the penalty for each candidate token with respect to each negative sample, scaling it by a constant hyperparameter β, and take the maximum value across them. Finally, we compute the final score F_j for each candidate by combining the logit probability $p_t(w_j)$ with the penalty term s_j^{final}, weighted by α_{ACS}, and select the token with the highest score via greedy decoding.

Algorithm 1 Avoidance Decoding

Input : p_t, $y_{1:t}$, $\{x_i^-\}_{i=1}^N$, $\{H_i^-\}_{i=1}^N$
Compute k, α_{ACS} (Arias et al., 2024)
Select top-k tokens $\{w_j\}_{j=1}^k$ from p_t
$\gamma = \delta + (1 - \delta) \cdot \text{sigmoid}(t - T_0)$
for $j = 1$ to k **do**
  $h_j \leftarrow$ last hidden state of model$(w_j \mid y_{1:t})$
  for $i = 1$ to N **do**
    $s_j^{\text{CSP}}[i] \leftarrow (\max_{h' \in H_i^-} \cos(h_j, h'))$
    $s_j^{\text{NSP}}[i] \leftarrow \cos(E(y_{1:t} \oplus w_j), E(x_i^-))$
    $s_j^{\text{hybrid}}[i] \leftarrow \gamma \cdot s_j^{\text{CSP}}[i] + (1-\gamma) \cdot s_j^{\text{NSP}}[i]$
  end for
  $s_j^{\text{final}} \leftarrow \max_i \beta \cdot s_j^{\text{hybrid}}[i]$
  $F_j \leftarrow (1 - \alpha_{ACS}) \cdot p_t(w_j) - \alpha_{ACS} \cdot s_j^{\text{final}}$
end for
$w^* \leftarrow \arg\max_j F_j$
return w^*

where H_i^- is the set of last hidden states of all individual tokens in i-th negative sample, p_t is the model's logits, $y_{1:t}$ is generated tokens so far, and x_i^- is i-th negative sample text.

5 Experiments

5.1 Experimental Setup

Implementation Details We run every decoding process on two NVIDIA RTX A5000 GPUs. To accelerate iterative token generation, we reuse the cached KV values of the Attention modules from the previous decoding step for the next one. We collect 20 versatile story prompts from ReedsyPrompts (Park et al., 2024b) and WritingPrompts (Fan et al., 2018) that yielded at least 20 different human stories, and use these prompts as

method	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLMScore(↑)	Degen(↓)
Naive	2.21	16.19	20.95	48.51	20.75	0.02
Top-k	1.92	15.53	21.09	50.47	22.60	0.01
Top-p	1.25	**10.80**	14.68	47.22	23.25	0.09
Typical	1.47	14.13	18.76	50.54	21.75	0.02
Mirostat	9.64	27.94	31.02	55.16	19.75	0.01
Min-p	12.30	28.96	33.26	53.97	19.25	0.00
CS	52.14	65.70	67.35	66.18	17.60	0.00
ACS	50.71	63.80	66.58	63.12	20.50	0.00
DBS	11.99	25.04	31.20	58.81	25.25	0.03
GPT-4o	3.57	18.48	24.61	51.89	22.25	0.04
Ours$_{CSP}$	0.61	9.31	9.95	25.25	69.75	0.15
Ours$_{NSP}$	39.87	48.71	51.23	71.44	25.25	0.00
Ours	**1.04**	12.57	**14.36**	**27.56**	**65.40**	0.02

Table 1: ReedsyPrompts, Mistral 7B

method	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLMScore(↑)	Degen(↓)
Naive	3.32	14.69	20.12	56.22	30.75	0.03
Top-k	2.36	16.85	22.58	57.30	21.85	0.01
Top-p	7.22	23.83	29.50	59.34	19.50	0.00
Typical	1.92	15.18	19.93	58.84	21.50	0.02
Mirostat	15.64	34.05	37.27	65.25	19.60	0.02
Min-p	15.29	33.55	36.70	64.13	26.85	0.00
CS	72.02	80.78	81.62	79.88	15.25	0.00
ACS	72.49	81.89	83.07	79.09	17.00	0.01
DBS	13.40	27.14	33.12	66.79	21.60	0.04
GPT-4o	3.57	18.48	24.61	51.89	24.15	0.00
Ours$_{CSP}$	0.80	10.83	11.62	29.51	51.50	0.12
Ours$_{NSP}$	35.12	45.66	48.75	73.31	23.50	0.00
Ours	**1.45**	**13.96**	**15.72**	**35.42**	**44.90**	0.05

Table 2: Writing Prompts, Mistral 7B

our initial inputs. We set the constant scalar hyperparameter β to 2.0 to meaningfully influence the token logits, and δ to 0.5 to maintain sufficient low-level diversity throughout the generation process, based on empirical tuning (See Appendix J).

Baselines We compare our method against several strong baselines:

- **Naive, Top-k, Top-p, Typical, Mirostat, and Min-P sampling (Holtzman et al., 2019; Meister et al., 2023; Basu et al., 2020; Nguyen et al., 2024):** These are various stochastic decoding strategies that can enhance diversity. Note that for Naive sampling, we apply no special techniques other than adjusting the temperature.
- **Contrastive Search (CS) and Adaptive Contrastive Search (ACS):** We include the original Contrastive Search and its advanced version, Adaptive Contrastive Search, as baselines. These methods serve as the primary motivation for our proposed methodology.
- **Diverse Beam Search (DBS) (Vijayakumar et al., 2016):** For Diverse Beam Search, we set the number of beam groups equal to the number of candidate sentences to maximize diversity.
- **GPT-4o:** We include OpenAI's powerful language model, GPT-4o, as a strong baseline.
- **Ours$_{CSP}$ and Ours$_{NSP}$:** To demonstrate the effectiveness of our Hybrid Penalty, we include two ablated versions as baselines. Ours$_{CSP}$ utilizes only CSP as the penalty, while Ours$_{NSP}$ utilizes only NSP as the penalty. Note that Ours$_{CSP}$ is highly inspired by the formulation of prior work (*CS*), yet integrates our own adjustments.

For all baselines except the ablated versions (i.e., Ours$_{CSP}$ and Ours$_{NSP}$), we feed all previously generated outputs from the same story prompt back into the next input and instruct the model to "create a story that does not resemble any of the already generated outputs." See Appendix G for more detailed information. In contrast, all Ours variants receive the exact same instruction at each iteration step within a story prompt, while internally storing all generated outputs in a *Negative Examples*

method	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLMScore(↑)	Degen(↓)
Top-k	1.12	**11.85**	17.94	48.49	34.25	0.04
Top-p	6.22	17.96	24.34	52.50	29.35	0.00
Min-p	20.05	30.30	36.20	60.27	26.85	0.00
ACS	59.78	64.91	67.68	77.95	19.00	0.00
GPT-4o	3.57	18.48	24.61	51.89	22.25	0.04
Ours$_{CSP}$	0.85	11.61	14.06	31.73	60.10	0.23
Ours$_{NSP}$	31.16	40.02	43.65	68.16	27.20	0.00
Ours	**1.09**	12.40	**15.63**	**32.66**	**54.25**	0.09

Table 3: ReedsyPrompts, Llama 3B

method	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLMScore(↑)	Degen(↓)
Top-k	1.12	12.18	18.53	51.74	32.00	0.05
Top-p	4.66	16.98	23.67	55.70	27.00	0.08
Min-p	42.59	48.20	53.44	77.84	17.90	0.00
ACS	68.96	72.09	73.45	86.12	18.50	0.01
GPT-4o	3.57	19.71	26.11	58.97	24.15	0.00
Ours$_{CSP}$	0.82	11.54	14.03	32.96	57.25	0.13
Ours$_{NSP}$	28.60	38.13	42.54	69.82	23.25	0.00
Ours	**1.05**	**11.68**	**15.10**	**34.03**	**50.60**	0.09

Table 4: Writing Prompts, Llama 3B

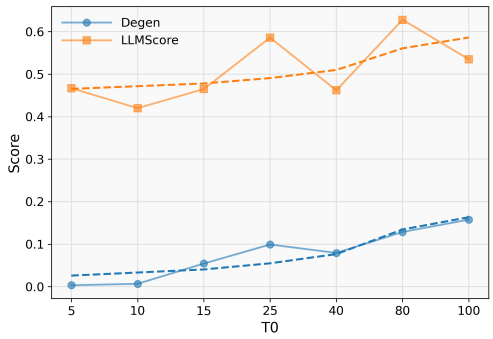

Figure 2: Average Degeneration and LLMScore versus T_0.

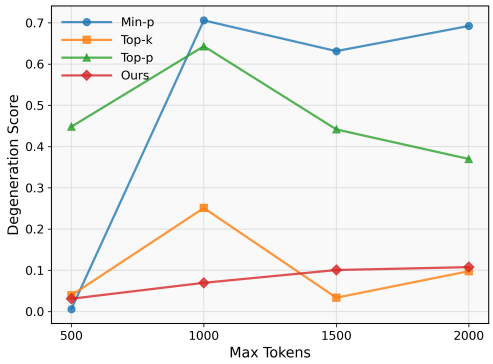

Figure 3: Average Degeneration scores versus max tokens.

memory and computing the maximum hidden-state similarity and sentence-similarity from them.

5.2 Main Experiment Results

5.2.1 Automatic Evaluation Metrics

We assess the performance of baseline methods using standard automatic evaluation metrics. Specifically, we first compute the average similarity among all stories generated for each prompt. The final score is obtained by averaging these values across all prompts.

- **BLEU, ROUGE-L, METEOR:** We utilize these n-gram or unigram matching metrics as primary diversity indicators.
- **Sent-sim:** To quantify semantic similarity, we first convert generated outputs into embedding vectors using Sentence-BERT, then compute the pairwise cosine similarity between these vectors.
- **LLM-Score:** We use OpenAI's recently revealed powerful reasoning model, GPT-o4-mini (OpenAI, 2025), to directly assess the diversity of multiple generated outputs for the same story prompt. See Appendix F for more detailed rubric.
- **Degen:** We also use GPT-o4-mini to evaluate the level of degeneration in each generated output of the scale 0-1. The rubric for this evaluation consists of "Syntactic Integrity (0.25)", "Semantic Trajectory (0.25)", "Lexical Sanity (0.25)", and "Noise Symptoms (0.25)". See Appendix E for the more detailed rubric.

Prompt: Write a story from the different perspectives of two people meeting for a blind date.	
Negative Sample	**Generation Result**
It was a beautiful summer evening, and the sun was setting over the bustling city. Emma was excited to see if their chemistry was real, but she was also a little nervous. (...) Meanwhile, Ryan was getting ready for the date, feeling a mix of excitement and nerves. He had been looking forward to this all week, but he was also a little...	On this particular evening, the drizzle seemed to seep into the very marrow of my bones, infusing me with a sense of melancholy (...) I stood outside the nondescript bistro, clutching my glass of Pinot Grigio (...) "So, how's your day been?" "What do you like to do in your free time?" Ugh, who came up with these boring conversation starters? ...

Table 5: Two multi-branch stories generated from the same prompt. The left story is a prior output, and the right story is a new generation. We highlighted the most contrastive parts, including mood, setting, and emotions.

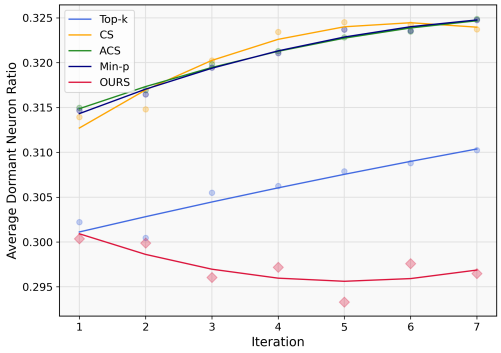

Figure 4: Average dormant neuron ratios per iteration.

5.2.2 Quantitative Analysis of Diversity and Degeneration

We apply our method and all baselines across three temperature values (low=0.7, moderate=1.0, high=1.3), four widely used LLMs (Mistral 7B (Jiang et al., 2023), Llama 3B (Touvron et al., 2024), Llama 8B, Qwen 7B (Bai et al., 2023)), and two story datasets that contain explicit story prompts (ReedsyPrompts, WritingPrompts). For each prompt, we generate 15 multi-branch stories of 200 tokens each. We then evaluate diversity and degeneration metrics on these outputs. For reporting, we select the temperature setting that yields the greatest number of cases with an average degeneration score ≤ 0.1 among the three temperature, since we empirically observe a sharp performance drop once the degeneration score exceeds 0.1. If a model exceeds the average degeneration score 0.1 across all temperature settings, then we report the result of the lowest temperature setting. Due to space constraints, we only present the full results for Mistral-7B and the top-8 results for LLaMA-3B in this section. More comprehensive results can be found in Appendix D.

As shown in Tables 1, 2, 3, and 4, our model achieves the best performance on most metrics. Although Ours$_{CSP}$ exhibits higher diversity, it is not considered the best-performing model, as its degeneration rates exceed the acceptable 0.1 threshold across all temperature settings. Ours$_{NSP}$, on the other hand, produces no degenerated samples but suffers from low overall diversity. These results indicate that our Hybrid Penalty approach achieves an effective balance: It significantly enhances diversity compared to existing baselines while also demonstrating greater robustness to degeneration than the version relying solely on CSP.

Additionally, the significantly low diversity of CS and ACS may stem from modified instruction by accumulating previously generated stories in the input prompt as negative samples. This leads language models to represent different hidden states for even identical tokens across stories. Consequently, even when the current candidate token is identical to one in the negative sample, their cosine similarity becomes small. In contrast, our method uses the exact same instruction across branches, ensuring identical tokens yield identical hidden states and thus enabling accurate per-position penalties.

5.2.3 Human Evaluation

Human Evaluation. We recruit seven human annotators and let them rate each model's outputs on a 5-point Likert scale according to four criteria:

- **Diversity:** The extent to which each story fundamentally differs from the others, beyond simple changes to character or theme names.
- **Degeneration:** The degree to which the text maintains grammatical correctness and lexical coherence without breakdown.

method	**Diversity(↑)**	**Creativity(↑)**	**Degen(↑)**
Naive	2.14 ± 0.8	2.33 ± 0.9	1.53 ± 1.3
Top-k	3.38 ± 1.2	3.38 ± 1.5	3.94 ± 1.3
Top-p	2.71 ± 1.1	3.24 ± 1.5	4.00 ± 1.3
Typical	1.62 ± 0.7	2.81 ± 1.5	4.06 ± 1.0
Mirostat	2.43 ± 1.1	3.14 ± 1.4	**4.18** ± 1.2
Min-p	2.05 ± 0.9	3.10 ± 1.4	3.76 ± 1.3
CS	1.10 ± 0.5	2.95 ± 1.4	3.76 ± 1.2
ACS	1.14 ± 0.5	3.14 ± 1.4	3.71 ± 1.2
DBS	2.95 ± 1.2	3.29 ± 1.4	3.88 ± 1.4
GPT-4o	2.62 ± 1.1	3.33 ± 1.5	4.00 ± 1.3
Ours_{CSP}	3.33 ± 1.4	3.19 ± 1.3	2.24 ± 1.2
Ours_{NSP}	1.43 ± 0.6	2.62 ± 1.5	4.06 ± 1.0
Ours	**3.48** ± 1.3	**3.71** ± 1.3	3.06 ± 1.4

Table 6: Human evaluation scores across decoding methods (mean ± std). Best means in each column are in **bold**.

- **Creativity:** The overall originality and intrigue of the generated stories.

Annotators evaluate only the last five generated outputs (samples 11–15) from each set, which exhibit the most prominent diversity differences. For more detailed information about human evaluation, see Appendix C, I.

As shown in Table 6, our model achieves the highest scores from human annotators for both Diversity and Creativity, proving the effectiveness of the Similarity-based Contrastive Penalty in enhancing narrative richness. Furthermore, our model shows superior performance on Degeneration compared to Ours_{CSP}, demonstrating the enhanced robustness of the concept-to-narrative hybrid penalty against degeneration. These results align with the quantitative analysis in Section 5.2.2, providing additional reliability.

5.2.4 Hyper-parameter Search: T_0

We vary the inflection point hyperparameter T_0 (equation 8) from 5 to 100 to find the best value considering the trade-off between diversity and text degeneration using Llama-3.1-8B. We fix the total generation length at 200 tokens. As shown in Figure 2, larger T_0 values tend to increase LLMScore and Degeneration score, with trendline slopes of 0.0012 and 0.0014, respectively. Considering this trade-off, we select an T_0 value of 25 as the optimal setting, as it yields the highest Diversity among outputs with Degeneration ≤ 0.1.

5.2.5 Impact of Maximum Token

We observe how the degeneration scores of various models change as the maximum token increases using Llama-3.2-3B. As shown in Figure 3, our model demonstrates greater robustness to degeneration than other sampling methods, only exceeding a degeneration score of 0.1 when the maximum token length reaches 2000, while other baselines exhibit a significantly higher degeneration score at shorter lengths. These results indicate that, as sequence length grows, our method provides substantially higher robustness to degeneration than directly feeding accumulated outputs back into instructions.

5.2.6 Analysis of Dormant Neuron Ratios

To examine the range of neuron activation during iterative multi-branch story generation, we perform dormant neuron analysis (Sokar et al., 2023) using the LLaMA-3.1-8B model. Due to space limitations, the full set of results is presented in Appendix A. Specifically, we consider a neuron in a fully connected layer to be *dormant* if its GELU activation falls below 5×10^{-5}. As shown in Figure 4, all baseline methods exhibit progressively reduced neural activation, indicating a decline in latent creativity. However, our method exhibits a decreasing dormant neuron ratio, thus increasing the range of neuron activation as the number of iterations increases. These results suggest that it more effectively enhances the model's inherent creativity, not by simply avoiding repetition of previously generated tokens.

5.2.7 Qualitative Analysis

Table 5 illustrates how our method encourages conceptual and narrative divergence from a negative sample. In the example shown, the previously generated story (left) features a bright and cheerful setting (e.g., *"beautiful summer evening"*, *"bustling city"*) and expresses anticipation and nervousness. The new decoded story (right), by contrast, adopts a somber tone with a rainy setting (e.g., *"drizzle"*, *"nondescript bistro"*) and conveys emotions such as melancholy and boredom. The plot also shifts from a planned date to a coincidental bar encounter. These differences highlight the effectiveness of our method's Similarity-based Contrastive Penalty in promoting both conceptual- and narrative-diversity in multi-branch stories.

6 Conclusion

We introduce Avoidance Decoding, a novel decoding strategy designed to enhance the diversity of multi-branch story generation. Our method contrastively penalizes token logits based on similarity

across different branch stories, using a hybrid of Concept-level and Narrative-level Similarity Penalties. Automatic and human evaluations consistently demonstrate that Avoidance Decoding significantly outperforms existing baselines in terms of diversity, while effectively mitigating the common trade-off between diversity and degeneration. Furthermore, dormant neuron analysis suggests that our method fosters deeper model creativity, as evidenced by broader neuron activation during generation.

7 Limitations

Although Avoidance Decoding demonstrates strong diversity and effective suppression of degeneration without any additional training or stochastic sampling, it has the drawback of increased decoding time. This issue becomes more pronounced as the number of negative samples increases. To mitigate the computational overhead, one potential solution is to store only a fixed-size window of recent outputs in the negative example memory rather than maintaining the entire output history. Furthermore, additional hyperparameter tuning may lead to better trade-offs between degeneration and diversity.

8 Acknowledgments

We thank anonymous reviewers for their constructive and insightful comments. K. Jung is with ASRI, Seoul National University, Korea. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea goverment(MSIT) (RS-2025-02263628). This work was also partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University) & No.RS-2021-II212068, Artificial Intelligence Innovation Hub & No.RS-2022-II220184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics].

References

Amal Alabdulkarim, Winston Li, Lara J Martin, and Mark O Riedl. 2021. Goal-directed story generation: Augmenting generative language models with reinforcement learning. *arXiv preprint arXiv:2112.08593*.

Esteban Garces Arias, Julian Rodemann, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. 2024. Adaptive contrastive search: Uncertainty-guided decoding for open-ended text generation. *arXiv preprint arXiv:2407.18698*.

Minwook Bae and Hyounghun Kim. 2024. Collective critics for creative story generation. *arXiv preprint arXiv:2410.02428*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R Varshney. 2020. Mirostat: A neural text decoding algorithm that directly controls perplexity. *arXiv preprint arXiv:2007.14966*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Parsa Ghaffari and Chris Hokamp. 2025. Narrative studio: Visual narrative exploration using llms and monte carlo tree search. *arXiv preprint arXiv:2504.02426*.

YiQiu Guo, Yuchen Yang, Zhe Chen, Pingjie Wang, Yusheng Liao, Ya Zhang, Yanfeng Wang, and Yu Wang. 2025. Dsvd: Dynamic self-verify decoding for faithful generation in large language models. *arXiv preprint arXiv:2503.03149*.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138*.

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

Runsheng Huang, Lara J Martin, Chris Callison-Burch, et al. 2024. What-if: Exploring branching narratives by meta-prompting large language models. *arXiv preprint arXiv:2412.10582*.

Corinna Jaschek, Tom Beckmann, Jaime A Garcia, and William L Raffe. 2019. Mysterious murder-mcts-driven murder mystery generation. In *2019 IEEE Conference on Games (CoG)*, pages 1–8. IEEE.

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

Minbeom Kim, Kang-il Lee, Seongho Joo, Hwaran Lee, Thibaut Thonet, and Kyomin Jung. 2025. Drift: Decoding-time personalized alignments with implicit user preferences. *arXiv preprint arXiv:2502.14289*.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2023. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*.

Arash Lagzian, Srinivas Anumasa, and Dianbo Liu. 2025. Multi-novelty: Improve the diversity and novelty of contents generated by large language models via inference-time multi-views brainstorming. *arXiv preprint arXiv:2502.12700*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Tobias Materzok. 2025. Cos (m+ o) s: Curiosity and rl-enhanced mcts for exploring story space via language models. *arXiv preprint arXiv:2501.17104*.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121.

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*.

Kolby Nottingham, Ruo-Ping Dong, Ben Kasper, and Wesley N Kerr. 2024. Improving branching language via self-reflection.

OpenAI. 2024. Gpt-4o system card. https://arxiv.org/abs/2410.21276.

OpenAI. 2025. gpt-o4-mini [large language model]. Accessed: 2025-05-17.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Kyeongman Park, Minbeom Kim, and Kyomin Jung. 2024a. A character-centric creative story generation via imagination. *arXiv preprint arXiv:2409.16667*.

Kyeongman Park, Nakyeong Yang, and Kyomin Jung. 2024b. Longstory: Coherent, complete and length controlled long story generation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 184–196. Springer.

Zeeshan Patel, Karim El-Refai, Jonathan Pei, and Tianle Li. 2024. Swag: Storytelling with action guidance. *arXiv preprint arXiv:2402.03483*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. 2023. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, pages 32145–32168. PMLR.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561.

Haoran Sun, Yekun Chai, Shuohuan Wang, Yu Sun, Hua Wu, and Haifeng Wang. 2025. Curiosity-driven reinforcement learning from human feedback. *arXiv preprint arXiv:2501.11463*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, et al. 2024. Llama 3: Open foundation and instruction models. https://ai.meta.com/blog/meta-llama-3/.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Danqing Wang, Jianxin Ma, Fei Fang, and Lei Li. 2024. Typedthinker: Diversify large language model reasoning with typed thinking.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Zhihua Wen, Zhiliang Tian, Wei Wu, Yuxin Yang, Yanqi Shi, Zhen Huang, and Dongsheng Li. 2023. Grove: a retrieval-augmented complex story generation framework with a forest of evidence. *arXiv preprint arXiv:2310.05388*.

Emily Wenger and Yoed Kenett. 2025. We're different, we're the same: Creative homogeneity across llms. *arXiv preprint arXiv:2501.19361*.

Zongqian Wu, Tianyu Li, Baoduo Xu, Jiaying Yang, Mengmeng Zhan, Xiaofeng Zhu, and Lei Feng. 2025a. Is depth all you need? an exploration of iterative reasoning in llms.

Zongqian Wu, Tianyu Li, Baoduo Xu, Jiaying Yang, Mengmeng Zhan, Xiaofeng Zhu, and Lei Feng. 2025b. Is depth all you need? an exploration of iterative reasoning in llms. *arXiv preprint arXiv:2502.10858*.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. 2025. Noveltybench: Evaluating language models for humanlike diversity.

Wenhong Zhu, Hongkun Hao, and Rui Wang. 2023. Penalty decoding: Well suppress the self-reinforcement effect in open-ended text generation. *arXiv preprint arXiv:2310.14971*.

A Full Results of Dormant Neuron Analysis

As shown in Fig. 5, among all baselines only our model exhibits a decrease in dormant neuron ratio over iterations. Specifically, the slopes of the linear trend lines are:

$$\begin{aligned}
\text{Top-k} &= +0.001540,\\
\text{CS} &= +0.001875,\\
\text{ACS} &= +0.001634,\\
\text{Typical} &= +0.002215,\\
\text{Mirostat} &= +0.001754,\\
\text{CD} &= +0.000615,\\
\text{Min-p} &= +0.001741,\\
\text{Top-p} &= +0.002260,\\
\text{Naive} &= +0.002439,\\
\text{OURS} &= -0.000675.
\end{aligned}$$

B Degeneration Trend versus Iteration number

As shown in Figure 6, the average degeneration score increases as the number of iterations grows in large-branch settings. This trend suggests that the model's linguistic structure gradually breaks down as it exhausts its intrinsic creative capacity due to the accumulation of previously generated outputs used as negative samples. Nevertheless, our model consistently yields lower degeneration scores than the ablated version using only CSP, demonstrating the robustness of the Hybrid Penalty even under high-branch configurations.

C Human Evaluation Details

We recruited graduate and undergraduate students fluent in English. The recruited annotators were provided with a detailed description of task definitions, instructions, and samples of each model. Also, all applicants were informed that their annotations would be used for academic purposes and would be published in paper material through the recruitment announcement and instructions.

Each of the seven annotators was given three samples—each consisting of five outputs from 13 baselines—and answered three questions for each sample. For the payment of the annotators, the coauthors conducted annotations for 5 hours first to estimate the average number of annotations that could be completed in the same time. Based on this estimation, a rate of 0.5 dollars per example was established to ensure that the annotators would be paid at least the minimum wage.

To assess annotator agreement, we computed average-measure ICC(2,k) values using each annotator's model-level mean scores: 0.93 for Diversity, 0.91 for Creativity, and 0.62 for Degen, indicating that all metrics achieved acceptable to excellent reliability.

D Comprehensive Experimental Results

Table 9, 10, 11, 12, 13, and 14 present the full experimental results on the ReedsyPrompts and WritingPrompts datasets using LLaMA 3B, LLaMA 8B, and Qwen 7B models. Our model achieves the best performance in most cases; however, on LLaMA 8B, its performance is highly competitive with that of Ours_{CSP}, with only a marginal difference.

E Degeneration Evaluation Details

Figure 7 defines the rubric used to compute the **Degeneration** score, which measures the degree of degeneration in GPT-4o outputs. The evaluation incorporates four equally weighted dimensions: syntactic integrity, semantic coherence, lexical sanity,

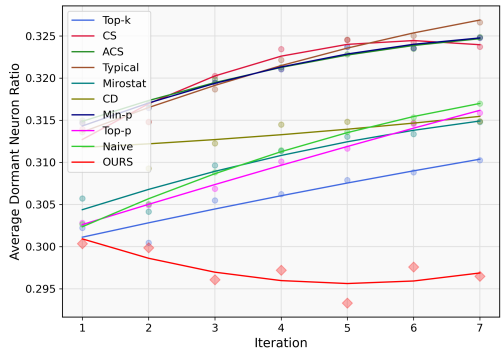

Figure 5: All dormant neuron ratios per iteration.

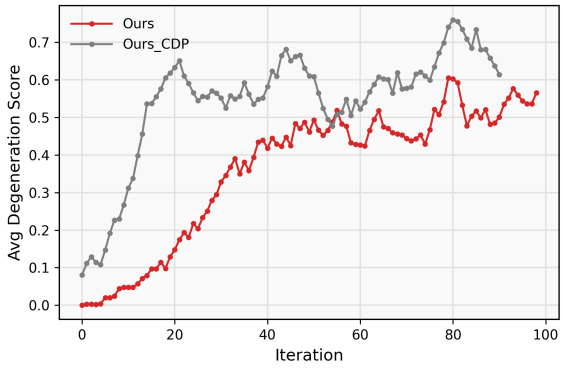

Figure 6: Smoothed average degeneration scores (window=10) versus iteration number.

and noise symptoms. We specifically design the rubric to avoid false positives caused by poetic, metafictional, or stylistic sentences.

F LLMScore Evaluation Details

Figure 8 defines the prompt and rubric used to compute **LLMScore**, which evaluates the diversity of GPT-4o outputs. It emphasizes four key dimensions—*perspective*, *style*, *plot structure*, and *language variation*. We encourage generous recognition of creative or surface-level differences, even in the presence of shared semantic themes, ensuring fair reward for imaginative or structurally distinct outputs beyond token-level repetition.

G Detailed Instruction for Baseline Generation

As shown in Figure 9, we configure all baselines—except the Ours variants—to receive all prior generations along with an instruction that encourages dissimilar outputs in subsequent branches during repetitive multi-branch story generation.

Metric	Spearman r	Pearson r
BLEU (↓)	-0.73	-0.68
ROUGE-L (↓)	-0.69	-0.67
METEOR (↓)	-0.70	-0.68
Cosine Sim (↓)	-0.72	-0.72
LLMScore (↑)	0.51	0.46

Table 7: Correlation between metrics and human scores.

β	δ	Degen Score	LLMScore
2.0	0.5	0.090	54.25
1.0	0.5	0.006	39.50
3.0	0.5	0.175	67.95
2.0	0.2	0.028	47.25
2.0	0.8	0.120	61.04

Table 8: Degen Score and LLMScore across (β, δ) configurations.

H Use of ChatGPT and Compliance with OpenAI's Terms

We utilized **OpenAI's ChatGPT** for limited assistance in refining the writing and formatting of this paper. All substantive contributions, including the core methodology, experiments, and analysis, were conducted independently by the authors.

Our usage complies with OpenAI's Terms of Use and Usage Policies.

I Correlation Between Automatic Metrics and Human Ratings

As shown in the Table 7, we conducted a correlation analysis between human ratings and automatic metrics. Since human evaluators assigned higher scores to outputs with greater diversity, in contrast to similarity-based metrics such as BLEU, ROUGE-L, METEOR, and Cosine Similarity, but in line with LLMScore, we conclude that each metric exhibits a strong correlation with human judgments.

Additionally, we did measure inter-annotator agreement among our human evaluators, which showed strong consistency, thereby supporting further reliability of our human evaluations.

J Empirical Experiments for Tuning Hyperparameters

As shown in Table 8, we conducted several empirical trials to determine hyperparameter values. Although some configurations yield higher LLMScore, they also result in degeneration scores above 0.1, which we consider to indicate serious performance degradation.

Table 9: ReedsyPrompts, Llama 3B

version	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLMScore(↑)	Degen
Naive	8.40	21.22	27.60	52.53	30.50	0.00
Top-k	1.12	**11.85**	17.94	48.49	34.25	0.04
Top-p	6.22	17.96	24.34	52.50	29.35	0.00
Typical	32.71	41.47	47.06	67.73	26.25	0.00
Mirostat	9.88	22.48	28.90	55.71	24.90	0.04
Min-p	20.05	30.30	36.20	60.27	26.85	0.00
CS	66.75	70.64	73.33	80.60	21.10	0.00
ACS	59.78	64.91	67.68	77.95	19.00	0.00
DBS	10.39	21.30	30.20	61.83	26.25	0.01
CD	29.68	39.62	43.90	62.87	27.50	0.01
GPT-4o	3.57	18.48	24.61	51.89	22.25	0.04
Ours$_{CSP}$	0.85	11.61	14.06	31.73	60.10	0.23
Ours$_{NSP}$	31.16	40.02	43.65	68.16	27.20	0.00
Ours	**1.09**	12.40	**15.63**	**32.66**	**54.25**	0.09

Table 10: Writing Prompts, Llama 3B

version	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLMScore(↑)	Degen
Naive	10.45	22.32	29.59	60.01	29.50	0.00
Top-k	1.12	12.18	18.53	51.74	32.00	0.05
Top-p	4.66	16.98	23.67	55.70	27.00	0.08
Typical	41.65	47.65	53.21	79.54	19.00	0.01
Mirostat	29.35	37.53	43.50	71.27	21.00	0.00
Min-p	42.59	48.20	53.44	77.84	17.90	0.00
CS	61.05	64.72	68.27	82.43	18.75	0.00
ACS	68.96	72.09	73.45	86.12	18.50	0.01
DBS	9.31	20.50	29.60	63.60	25.50	0.01
CD	29.08	37.91	43.91	67.57	21.75	0.02
GPT-4o	3.57	19.71	26.11	58.97	24.15	0.00
Ours$_{CSP}$	0.82	11.54	14.03	32.96	57.25	0.13
Ours$_{NSP}$	28.60	38.13	42.54	69.82	23.25	0.00
Ours	**1.05**	**11.68**	**15.10**	**34.03**	**50.60**	0.09

Table 11: ReedsyPrompts, Llama 8B

version	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLMScore(↑)	Degen
Naive	8.96	21.71	28.73	53.61	22.95	0.00
Top-k	1.45	12.23	19.15	52.20	27.75	0.02
Top-p	4.49	16.90	24.05	51.97	28.75	0.00
Typical	29.83	39.26	45.72	63.11	21.10	0.00
Mirostat	8.88	21.89	28.45	54.65	28.25	0.00
Min-p	14.04	26.07	32.90	57.77	20.75	0.04
CS	49.13	56.86	60.38	67.30	24.85	0.00
ACS	50.81	58.29	61.63	68.98	20.25	0.01
DBS	9.66	21.15	29.67	62.08	29.90	0.01
CD	17.29	28.94	35.05	53.30	27.00	0.00
GPT-4o	3.57	18.48	24.61	51.89	22.25	0.04
Ours$_{CSP}$	**1.00**	**11.93**	**15.71**	**35.05**	58.30	0.08
Ours$_{NSP}$	32.14	41.04	45.59	70.56	24.15	0.00
Ours	1.19	12.53	16.94	35.73	**58.60**	0.10

Table 12: Writing Prompts, Llama 8B

version	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLMScore(↑)	Degen
Naive	11.88	23.82	30.55	63.89	25.35	0.00
Top-k	1.69	12.68	19.36	55.48	27.00	0.02
Top-p	21.76	32.18	38.47	68.25	22.50	0.00
Typical	32.39	46.06	51.93	75.99	21.50	0.00
Mirostat	14.64	25.71	33.28	65.24	23.75	0.00
Min-p	17.80	28.65	35.61	67.99	21.25	0.07
CS	64.48	68.59	71.30	78.95	17.75	0.00
ACS	65.76	68.93	72.48	82.56	20.25	0.00
DBS	10.21	21.60	30.35	65.40	23.80	0.03
CD	31.45	42.18	47.56	70.16	22.00	0.01
GPT-4o	3.57	18.48	24.61	51.89	22.25	0.00
Ours$_{CSP}$	**0.98**	**12.09**	**15.76**	39.77	**49.15**	0.07
Ours$_{NSP}$	35.58	43.69	48.02	75.59	20.65	0.00
Ours	1.25	12.15	16.50	**38.05**	49.00	0.07

Table 13: ReedsyPrompts, Qwen 7B

version	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLMScore(↑)	Degen
Naive	2.89	14.69	20.12	49.66	30.75	0.00
Top-k	2.00	13.75	19.88	51.91	27.75	0.00
Top-p	6.77	17.96	24.34	52.50	29.35	0.00
Typical	3.64	41.47	47.06	52.52	25.75	0.03
Mirostat	19.21	32.01	36.74	63.94	22.85	0.01
Min-p	14.54	28.40	33.26	61.97	25.90	0.01
CS	76.32	81.64	82.14	84.39	14.25	0.00
ACS	70.98	76.83	77.86	81.86	16.00	0.00
DBS	7.87	20.29	28.37	58.22	30.50	0.03
GPT-4o	3.57	18.48	24.61	51.89	22.25	0.04
Ours$_{CSP}$	0.70	10.07	10.67	30.37	63.75	0.19
Ours$_{NSP}$	16.75	28.67	33.60	57.50	24.15	0.00
Ours	**1.59**	**12.40**	**15.63**	**35.13**	**54.25**	0.09

Table 14: Writing Prompts, Qwen 7B

version	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLMScore(↑)	Degen
Naive	6.08	18.34	23.78	56.14	24.75	0.01
Top-k	3.05	15.27	21.44	55.98	23.25	0.00
Top-p	18.44	30.91	35.70	65.52	24.25	0.00
Typical	5.51	17.12	23.21	55.74	26.00	0.04
Mirostat	26.32	38.59	43.11	71.10	20.50	0.01
Min-p	29.05	41.20	45.00	73.36	23.25	0.01
CS	83.32	87.27	87.75	87.66	14.25	0.02
ACS	77.46	82.98	83.03	84.68	17.25	0.00
DBS	9.35	21.83	29.97	63.61	25.00	0.01
CD	31.45	42.18	47.56	70.16	22.00	0.02
GPT-4o	4.67	19.71	26.11	58.97	24.15	0.00
Ours$_{CSP}$	1.30	12.20	13.37	36.18	48.00	0.13
Ours$_{NSP}$	20.61	32.27	36.92	66.77	24.60	0.00
Ours	**1.63**	**12.08**	**14.93**	**37.40**	**49.25**	0.06

You are a rigorous text degeneration evaluator.
You will be given ONE passage of text. Your task is to assess whether it exhibits signs of LLM degeneration.
Return ONLY valid JSON with:

- `"degeneration_score"`: float in [0.0, 1.0]
- `"label"`: `"OK"` or `"DEGENERATED"` (threshold: 0.30)
- `"issues"`: a list of up to 4 bullet-point style strings, explaining key problems if any

Use the following rubric. Each of the 4 dimensions contributes equally (0.25 weight each) to the final score.

– **1. Syntactic Integrity (grammar & sentence structure)**
• 0.0 = flawless grammar, logical punctuation, complete sentences
• 0.2–0.4 = minor run-ons or awkward phrasing
• 0.6–0.8 = frequent structural errors or confusing syntax
• 1.0 = broken or chaotic sentence structure
Do NOT penalize complex or poetic sentence structures if grammatically valid.

– **2. Semantic Trajectory (logical & thematic coherence)**
• 0.0 = clear, logical development of ideas
• 0.2–0.4 = some loose transitions or unexpected inserts
• 0.6–0.8 = idea flow breaks down across sentences
• 1.0 = no coherent message or progression
Do NOT penalize metafictional commentary or stylistic digressions if meaning is preserved.

– **3. Lexical Sanity (word choices & in-context appropriateness)**
• 0.0 = all words match the tone and meaning
• 0.2–0.4 = some unusual word choices but interpretable
• 0.6–0.8 = strange phrasing, tone mismatches, rare word combinations
• 1.0 = nonsensical or surreal word combinations (e.g. "fractional nut satisfaction")
Do NOT penalize poetic, archaic, or stylized language if used intentionally.

– **4. Noise Symptoms (repetition, rambling, word salad)**
• 0.0 = no unusual patterns
• 0.2–0.4 = light stylistic repetition or verbosity
• 0.6–0.8 = distracting repetition, filler, or randomness
• 1.0 = clear signs of uncontrolled generation: token loops, hallucinations, nonsense
*Do NOT, Never penalize if the **final sentence** of the passage is truncated — this is because of the length limits.*

Important:

- Use professional judgment to apply these scores.
- Focus on detecting *true* degeneration (e.g., broken logic, nonsense, hallucinated text).
- Never return anything except the JSON object. No extra explanation or comments.

Figure 7: Prompt and Rubric for Rigorous Degeneration Evaluation

You are a text diversity evaluator.
Below are 15 numbered text samples. Your task is to assess how diverse they are in terms of **perspective, style, plot structure, and language variation.**
Your output must be a JSON object with:

- `"diversity_score"`: a float between 0.0 and 1.0 (where 0 = all samples are nearly identical, and 1 = samples are maximally diverse)
- `"justification"`: a one-sentence explanation of your reasoning

Scoring guidance:

- 0.0: All samples are structurally and semantically almost identical.
- 0.1–0.3: Slight variation in phrasing or detail, but mostly follow the same template.
- 0.4–0.6: Notable variation in perspective, tone, setting, or content development.
- 0.7–0.9: Substantial differences in narrative framing, imaginative detail, or genre shifts.
- 1.0: Samples are maximally different in form, function, and voice.

Be generous when minor shifts in character, setting, or literary device occur. Do not penalize shared themes if surface features differ meaningfully.
Return only a valid JSON object and nothing else.

Figure 8: Prompt and Evaluation Rubric for Measuring Textual Diversity

You are a helpful and creative assistant that always responds in English and avoids undesired responses.
Please write a story from the following prompt.
Do **NOT** generate responses that resemble the following examples:
`{all_previous_outputs}`

Figure 9: Prompt for Conditional Story Generation with Explicit Negative Constraint