A Universal Avoidance Method for Diverse Multi-branch Generation

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

Minha Jhang
Seoul National University
jminha@snu.ac.kr

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

Abstract

Modern generative models still lack human-level creativity, particularly in multi-branch diversity. Prior approaches to address this problem often incur heavy computation or strong dependency on model architecture. Therefore, we introduce **UAG**(**U**niversal **A**voidance **G**eneration), a model-agnostic and computationally efficient generation strategy that penalizes similarity among previously generated outputs. Thus, UAG can enhance multi-branch diversity across both diffusion and transformer models, with minimal additional computation. In experiments, our method achieves up to 1.9 times higher diversity, runs 4.4 times faster, and requires only 1/64 of the FLOPs compared to state-of-the-art methods. The full code is here.

1 Introduction

Recently, generative models across various domains such as text and image generation (Yang et al., 2025; Dubey et al., 2024; Nie et al., 2025; OpenAI, 2025a; Fortin and Odoom, 2025; OpenAI, 2023; Stability AI, 2024) have demonstrated remarkable performance. Despite this progress, these models still fall short in terms of human-level creativity, particularly in tasks such as story writing (Park et al., 2024a; Wen et al., 2023; Nottingham et al., 2024; Jaschek et al., 2019) and illustration (Lu et al., 2024; Li et al., 2024; Zameshina et al., 2023). These tasks require substantial *multi-branch diversity*, i.e., generating diverse outputs from the same prompts (Park et al., 2025).

Thus, many approaches have been proposed to address the lack of *multi-branch diversity* in these tasks (Park et al., 2025; Nottingham et al., 2024; Wen et al., 2023; Lu et al., 2024; Li et al., 2024). However, these approaches often suffer from substantial computational overhead due to token-wise or pixel-wise operations (Park et al., 2025; Lu et al., 2024), and limited applicability across different models because of their dependence on specific architectures (Park et al., 2025; Corso et al., 2023; Welleck et al., 2019; Li et al., 2024; Sadat et al., 2023). To overcome these limitations, we propose **UAG**(**U**niversal **A**voidance **G**eneration), a novel framework for enhancing *multi-branch diversity* that minimizes additional computation costs while remaining agnostic to model architecture.

UAG enhances *multi-branch diversity* by reducing similarity to previously generated outputs, by penalizing the gradient of a similarity loss on the final outputs. The similarity loss comprises two components:(1) a local similarity loss at the token or pixel level, and(2) a global similarity loss at the hidden-state level. Through loss scheduling, UAG can promote a natural progression of diversity: in the early stages, local similarity encourages concept-level diversity, while in later stages, global similarity fosters semantic-level diversity. Because UAG requires only a few gradient computations per sample, rather than token-wise (Park et al., 2025; Welleck et al., 2019; Garces Arias et al., 2024; Su and Collier, 2022; Ding et al., 2025; Giulianelli et al., 2023) or pixel-wise operations (Ho et al., 2020), it is both highly efficient and fast. Moreover, due to its simplicity, UAG is model-agnostic and thus applicable to both diffusions and transformers, which represent the current state of the art in generative modeling.

Surprisingly, our method achieves on average 1.43 times higher scores in conventional diversity metrics(e.g., BLEU, CLIPscore) and 1.19 times higher scores in LLM-based evaluations compared to baselines. Moreover, our method simultaneously attains 4.4 times faster decoding speed and requires 64 times fewer FLOPs than previous state-of-the-art methods (Park et al., 2025; Lu et al., 2024).

2 Universal Generation Process

Most modern generative models, including autoregressive and diffusion-based models, can be uni-

Findings of the Association for Computational Linguistics: ACL 2026, pages 15857–15870

formly described as sequential processes indexed by steps $t = 1, 2, \ldots, T$. At each step t, the model updates an internal hidden(or latent) state h_t and produces an output representation y_t:

$$(y_t, h_{t+1}) \sim P_\theta(\cdot \mid h_t, p), \tag{1}$$

where p is the conditioning input(e.g., a text prompt or image condition). We define this process as the Universal Generation Process, and this makes it possible to design universal diverse multi-branch generation strategy that operates consistently across different model families.

Autoregressive language models. The autoregressive language models include famous LLM such as GPT-5 (OpenAI, 2025a), LLaMa-3B (Dubey et al., 2024), and Qwen-7B (Yang et al., 2025). For these models, h_t denotes the decoder's last hidden state after processing tokens up to step t, while the model output y_t is the logit vector obtained by applying the output weight matrix to h_t. The next state h_{t+1} is updated once a token is predicted and appended.

Diffusion language models. For denoising diffusion language models (Zhu et al., 2025; Nie et al., 2025), the process starts from random noise and iteratively refines it into a coherent text representation, where the steps t typically count down from a maximum value T to 1. Here, the state h_t is the noisy latent representation of the text at reverse-time step t. The model output y_t is the prediction of the token logits. The next state in the generative sequence, h_{t-1}, is then calculated by a scheduler, which uses the current state h_t and the prediction y_t to produce a slightly less noisy representation.

Diffusion image models. Similarly, latent diffusion models for image generation (Google DeepMind, 2024; Stability AI, 2024) also follow a reverse, denoising process within the VAE's compressed latent space. The state h_t is the noisy latent vector at reverse-time step t. The model's output y_t is the predicted noise present in h_t. The scheduler uses this predicted noise y_t to remove it from the current state h_t, thereby computing the next, cleaner state h_{t-1}. This process is repeated until the final denoised latent h_0 is obtained, which is then decoded by the VAE into the final image.

3 Universal Avoidance Generation

To achieve *diverse multi-branch generation* within the Universal Generation Process, we propose **UAG**(**U**niversal **A**voidance **G**eneration), which applies gradient-based penalties to the output space and adjusts the results step by step. Our method requires only simple differentiation with respect to outputs and hidden states, rather than repetitive token-wise or pixel-wise computation. Thus, it is not only applicable to any generative model within the Universal Generation Process, but also significantly more efficient than previous state-of-the-art methods (Park et al., 2025; Lu et al., 2024).

3.1 Step-wise Loss Scheduling

We denote by $\phi(y_t)$ the output representation of the model. Let $\mathcal{B}_t^{\text{out}}$ and $\mathcal{B}_t^{\text{hid}}$ be the reference banks, which are previously generated outputs or hidden states from cached past runs.

The step-t local loss $\mathcal{L}_{\text{local}}^{(t)}$ is defined as $\max_{b \in \mathcal{B}_t^{\text{out}}} \text{sim}\big(\phi(y_t),\, b\big)$, and global loss $\mathcal{L}_{\text{global}}^{(t)}$ is defined as $\max_{b \in \mathcal{B}_t^{\text{hid}}} \text{sim}\big(h_t,\, b\big)$, where $\text{sim}(\cdot, \cdot)$ is a similarity measure that depends on the model type(see Appendix D).

To balance the contributions of the two losses, we apply a logistic schedule, following prior work (Park et al., 2025) :

$$\begin{aligned} s_t &= \frac{1}{1 + e^{\delta\,(t - L_0)}}, \\ w_{\text{local}}(t) &= \alpha \cdot s_t, \quad w_{\text{global}}(t) = \beta \cdot (1 - s_t) \end{aligned} \tag{2}$$

where α, β are maximum weights, L_0 is the transition center, and δ controls sharpness. Thus, we emphasize *local* similarity(e.g., token-level or pixel-level) in the early stages, while in the later stages we emphasize *global* similarity(e.g., hidden-state level). This scheduling is intuitive, since in the earlier stages the model has not yet formed meaningful global representations and should therefore focus on local contextual diversity, whereas in the later stages it becomes more important to ensure global semantic diversity(e.g., story narratives or scene layouts).

Finally, we define the UAG loss at step t as a combined loss as follows:

$$\mathcal{L}_{\text{UAG}}^{(t)} = w_{\text{local}}(t)\, \mathcal{L}_{\text{local}}^{(t)} + w_{\text{global}}(t)\, \mathcal{L}_{\text{global}}^{(t)}. \tag{3}$$

3.2 Gradient-based Penalty Adaptation

To penalize similarity to previous outputs, we adjust the output representation y_t by subtracting the

gradient of the UAG loss:

$$\begin{aligned}\hat{y}_t &= y_t - \nabla_{y_t} L_{\text{UAG}}^{(t)} \\ &= y_t - \Big(w_{\text{local}}(t)\, \nabla_{y_t} \mathcal{L}_{\text{local}}^{(t)} + w_{\text{global}}(t)\, \nabla_{y_t} \mathcal{L}_{\text{global}}^{(t)} \Big)\end{aligned} \tag{4}$$

then we use $\hat{y}_t$ to sample token probabilities or as input for the scheduler of the diffusion model. This penalty introduces a repulsive force that pushes the current generation away from previously generated outputs. For the underlying mathematical motivations of this adjustment, see Appendix B.

The local penalty $\nabla_{y_t} \mathcal{L}_{\text{local}}^{(t)}$ and global penalty $\nabla_{y_t} \mathcal{L}_{\text{global}}^{(t)}$ defined as

$$\begin{aligned}\nabla_{y_t} \mathcal{L}_{\text{local}}^{(t)} &= \left(\frac{\partial \phi(y_t)}{\partial y_t} \right)^{\top} \nabla_{\phi} \mathcal{L}_{\text{local}}^{(t)}, \\ \nabla_{y_t} \mathcal{L}_{\text{global}}^{(t)} &= \left(\frac{\partial h_t}{\partial y_t} \right)^{\top} \nabla_{h_t} \mathcal{L}_{\text{global}}^{(t)}.\end{aligned} \tag{5}$$

However, in LMs, $\frac{\partial h_t}{\partial y_t}$ does not carry gradient flow, because y_t is obtained as $y_t = W h_t + b$, where W and b denote the output projection matrix and the bias term, respectively, so h_t does not depend on y_t. Therefore, we use a heuristic alternative that first differentiates with respect to h_t:

$$g_{\text{global}} = \nabla_{h_t} \mathcal{L}_{\text{global}}^{(t)}, \tag{6}$$

and then project it into the y_t space using the Jacobian

$$\widehat{g}_{\text{hid}} = J_t\, g_{\text{global}}, \qquad J_t = \frac{\partial y_t}{\partial h_t} = W. \tag{7}$$

We then use $\widehat{g}_{\text{hid}}$ as a surrogate for $\nabla_{y_t} \mathcal{L}_{\text{global}}^{(t)}$, which empirically works well. Note that in diffusion models, both the latent state h_t and the predicted noise y_t reside in the same dimensional space and are tightly coupled by the scheduler's update rule, so we make a simplifying approximation for diffusion models by treating the required Jacobian as an identity matrix($J_t \approx I$), which is also effective. Additionally, **we greatly reduce** gradient computation by using analytic formulations rather than automatic differentiation(see Appendix C).

Finally, to ensure stable magnitude, all penalty gradients $g = \nabla_{y_t} \mathcal{L}_{\cdot}^{(t)}$ are normalized using

$$\text{Norm}(g) = \frac{g - \mu(g)}{\sqrt{\text{Var}(g) + \varepsilon}}, \tag{8}$$

where $\mu(g)$ is the mean of g along the last dimension, $\text{Var}(g)$ the variance along the same dimension, and ε a small constant for numerical stability.

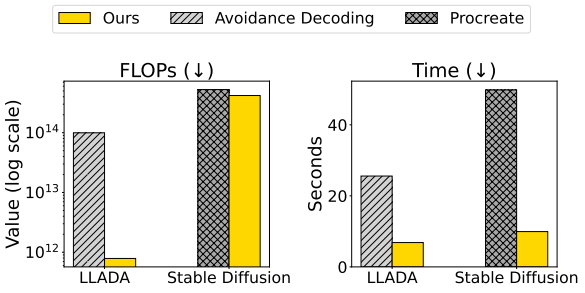

Figure 1: **FLOPs and Time comparison.** Lower values indicate better performance, and FLOPs are log-scaled due to their large differences.

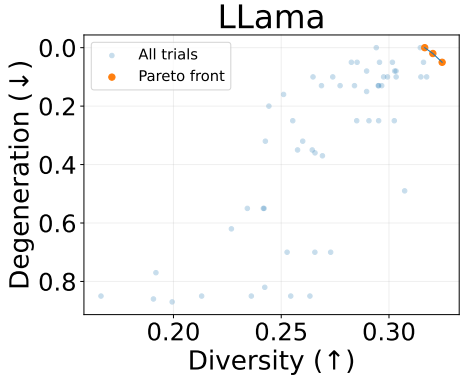

Figure 2: **Extensive hyperparameter sweeping tests** for LLaMA-3B. We select the best hyperparameter set from the Pareto front points.

4 Experiments and Results

All details on experimental setups such as implementation, datasets, baselines and metrics are in Appendix D.

4.1 Flops and Time Comparison

As shown in Figure 1, the previous state-of-the-art textual generation method, *Avoidance Decoding*, incurs approximately 126 times more FLOPs and requires 3.73 times longer runtime than our method. In image generation, the prior state-of-the-art method, *Procreate*, requires about 1.25 times more FLOPs and 5.01 times longer runtime.Therefore, we conclude that our method achieves substantially better computational efficiency by relying on simple gradient computation rather than token-wise penalty operations or external model calls.

4.2 Automatic Evaluation

As shown in Table 1, in the LLaMA-3B test, it achieves strong diversity even when compared with the strong baseline, *Avoidance Decoding*, but exhibits relatively higher degeneration due to the

method	RougeL(↓)	LLM-D(↑)	Degen(↓)
Llama-3B			
Naive	0.257	0.293	0.005
Top-$k^{\dagger}$	*0.006*	*0.930*	*0.980*
Top-p	0.198	0.476	0.193
AD	0.1265	**0.82**	0.25
Ours$_{global}$	0.312	0.442	0.388
Ours$_{local}$	0.212	0.391	**0.000**
Ours	**0.093**	0.680	0.565
Llada-8B			
Naive	0.180	0.316	0.564
Ours$_{global}$	0.493	0.212	0.157
Ours$_{local}$	0.082	0.278	**0.100**
Ours	**0.067**	**0.434**	0.323
method	**CLIP(↓)**	**LLM-D(↑)**	**LLM-Q(↑)**
Stable Diffusion			
Naive	0.830	0.800	**0.720**
PC	0.218	0.76	0.684
Ours$_{global}$	0.830	0.840	0.625
Ours$_{local}$	**0.143**	0.840	0.355
Ours	0.477	**0.860**	0.610

Table 1: **Llama-3B**, **Llada-8B** for ReedsyPrompts and **Stable Diffusion** for COCO datasets. AD denotes Avoidance Decoding, and PC denotes Procreate. Best results are in **bold**, second-best are underlined. Top-k for Llama-3B is excluded from best/second-best ranking due to extremely high degeneration(0.98). See Appendix H for the full results.

method	Div(↑)	Degen(↑)	Crt(↑)	Coh(↑)
LlaMa-3B				
Naive	1.9	1.0	1.2	1.6
Top-k	1.0	1.0	1.0	1.0
Top-p	2.1	1.0	1.3	1.7
Ours$_{local}$	**4.0**	**4.1**	**3.1**	**3.7**
Ours$_{global}$	2.8	1.6	2.6	2.8
Ours	3.7	1.9	2.8	2.6
LlaDa-8B				
Naive	2.3	1.1	1.7	1.8
Ours$_{local}$	3.1	**3.6**	2.6	**3.8**
Ours$_{global}$	1.1	3.2	2.0	3.3
Ours	**3.6**	3.2	**3.3**	3.5
Stable Diffusion				
Naive	4.0	**3.6**	**3.8**	4.0
Ours$_{local}$	3.5	1.4	2.4	3.5
Ours$_{global}$	**4.25**	3.0	3.4	4.0
Ours	4.0	2.8	3.4	**4.25**

Table 2: **Human evaluation results** across three settings with ReedsyPrompts and COCO. Div = Diversity, Degen = Degeneration, Crt = Creativity, Coh = Coherence. Higher is better for all four metrics. Best results are in **bold**, second-best are underlined.

inherent trade-off; nevertheless, under our strict evaluation criteria, the outputs remain acceptably readable (see Appendix F). For the LlaDA-8B diffusion language model, it maintains balanced performance with superior diversity compared to high-temperature baselines. In diffusion-based image generation, our method achieves the highest LLM-Diversity and competitive CLIP scores, with an acceptable trade-off between diversity and quality, even when compared with the strong baseline, *Procreate*. Furthermore, we conduct large-scale model experiments which shows enhanced diversity and degeneration scores (see Appendix G). We therefore conclude that our method enhances multi-branch diversity successfully than either ablated versions or other state-of-the-art methods.

4.3 Human Evaluation

See Appendix A for details including annotator information, rubrics and agreements. As shown in Table 2, our method consistently ranked among the top across all models: second-best overall for LLaMA, best in diversity and creativity for LLada, and best in coherence with strong diversity and creativity for Stable Diffusion. These results demonstrate that the combined penalty in UAG effectively enhances creativity, diversity, and coherence across different models, while maintaining acceptable degeneration compared to naive or ablated versions.

4.4 Searching for the Best Hyperparameters

To identify the best trade-off between diversity and degeneration, we perform extensive hyperparameter sweeping tests **for all baselines**. As a result, we find the optimal hyperparameters that achieve the lowest degeneration while yielding the highest diversity, as shown in Figure 2. Since diversity and degeneration are positively correlated during sweeping, this confirms that our hyperparameter search is a right approach to determine the optimal balance. See Appendix I for additional details and other sweeping experiment results.

5 Conclusion

We introduce UAG, an efficient framework that enhances multi-branch diversity across autoregressive and diffusion-based generative models using gradient-based penalties. UAG achieves top-ranked strong diversity with acceptable degeneration, while significantly reducing FLOPs and runtime compared to prior state-of-the-art methods.

Limitations

Our framework is not applicable to generative models which lie outside the scope of the Universal Generation Process, such as GANs. Developing methods that can be universally applied across all classes of generative models still remains as a challenge for future research.

Ethical Considerations

This work primarily focuses on improving the diversity of generative models and therefore does not directly raise ethical concerns. However, we acknowledge that excessive pursuit of diversity could potentially violate ethical boundaries. Future research should consider safety and ethical safeguards to ensure that enhanced diversity does not compromise responsible use.

Acknowledgement

We thank anonymous reviewers for their constructive and insightful comments. K. Jung is with ASRI, Seoul National University, Korea. This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2022-II220184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics), Institute of Information & Communications Technology Planning & Evaluation(IITP)-ITRC(Information Technology Research Center) grant funded by the Korea government(MSIT)(IITP-2025-RS-2024-00437633, 30%)], and the National Research Foundation of Korea(NRF) grant funded by the Korea goverment(MSIT) (RS-2025-02263628).

References

Stability AI. 2024. Stable diffusion 3.5 large: Multimodal diffusion transformer for text-to-image generation. Model description released via Stability AI and Hugging Face model card.

Gabriele Corso, Yilun Xu, Valentin De Bortoli, Regina Barzilay, and Tommi Jaakkola. 2023. Particle guidance: non-iid diverse sampling with diffusion models. *arXiv preprint arXiv:2310.13102*.

Yuanhao Ding, Esteban Garces Arias, Meimingwei Li, Julian Rodemann, Matthias Aßenmacher, Danlu Chen, Gaojuan Fan, Christian Heumann, and Chongsheng Zhang. 2025. GUARD: Glocal uncertainty-aware robust decoding for effective and efficient open-ended text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 7202–7226, Suzhou, China. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

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

Alisa Fortin and Seth Odoom. 2025. Announcing imagen 4 fast and the general availability of the imagen 4 family in the gemini api. Google Developers Blog. Accessed: 2025-09-29.

Esteban Garces Arias, Julian Rodemann, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. 2024. Adaptive contrastive search: Uncertainty-guided decoding for open-ended text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15060–15080, Miami, Florida, USA. Association for Computational Linguistics.

Mario Giulianelli, Joris Baan, Wilker Aziz, Raquel Fernández, and Barbara Plank. 2023. What comes next? evaluating uncertainty in neural text generators against human production variability. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14349–14371, Singapore. Association for Computational Linguistics.

Google DeepMind. 2024. Imagen 3. Technical report, Google DeepMind. Accessed: 2025-09-29.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

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.

Shuangqi Li, Hieu Le, Jingyi Xu, and Mathieu Salzmann. 2024. Enhancing compositional text-to-image generation with reliable random seeds. In *The Thirteenth International Conference on Learning Representations*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer.

Jack Lu, Ryan Teehan, and Mengye Ren. 2024. Procreate, don't reproduce! propulsive energy diffusion for creative generation. In *European Conference on Computer Vision*, pages 397–414. Springer.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *arXiv preprint arXiv:2502.09992*.

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

OpenAI. 2023. Dall·e 3 system card. Technical report, OpenAI. Accessed: 2025-09-29.

OpenAI. 2025a. Gpt-5 system card. Technical report, OpenAI. Accessed: 2025-09-29.

OpenAI. 2025b. Introducing gpt-4.1 in the api. OpenAI Blog. Accessed: 2025-09-29.

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.

Kyeongman Park, Nakyeong Yang, and Kyomin Jung. 2025. Avoidance decoding for diverse multi-branch story generation. *arXiv preprint arXiv:2509.02170*.

Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M Weber. 2023. Cads: Unleashing the diversity of diffusion models through condition-annealed sampling. *arXiv preprint arXiv:2310.17347*.

Stability AI. 2024. Stable diffusion 3. Stability AI News. Accessed: 2025-09-29.

Yixuan Su and Nigel Collier. 2022. Contrastive search is what you need for neural text generation. In *arXiv preprint*. Preprint.

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*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Mariia Zameshina, Olivier Teytaud, and Laurent Najman. 2023. Diverse diffusion: Enhancing image diversity in text-to-image generation. *arXiv preprint arXiv:2310.12583*.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and 1 others. 2025. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*.

A Human Evaluation Details

A.1 Annotators Details

We recruited five 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 five annotators was given six samples—each consisting of five outputs from all baselines—and answered four questions for each sample. For the payment of the annotators, the coauthors conducted annotations for 6 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.

A.2 Rubrics

The detailed rubrics for human evaluation are in Table 3,4.

A.3 Agreements

To assess annotator agreement, we computed both Kendall's coefficient of concordance(W) and intraclass correlation coefficients(ICC) across five raters. Kendall's W reached 0.61, suggesting moderate to substantial agreement among annotators. The single-measure ICC(1,1) was 0.49, indicating moderate reliability. Taken together, these results show that annotator ratings exhibit a reasonable level of consistency, though not uniformly high across all items.

B Mathematical Motivation of UAG

We recall the standard smoothness assumption: a differentiable function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is said to

Question 1. Diversity
To what extent are the generated stories fundamentally different from each other, beyond merely changing character names or surface-level details? (1–5)

Question 2. Degeneration
To what extent are the sentences fluent, grammatical, and lexically natural, without exhibiting broken syntax or incoherent word usage? (1 = very poor, 5 = excellent)

Question 3. Creativity
To what extent are the stories novel, engaging, and imaginative, rather than conventional or obvious? (1 = very poor, 5 = excellent)

Question 4. Coherence
To what extent does each story develop in a consistent and logically connected manner, maintaining narrative coherence throughout? (1 = very poor, 5 = excellent)

Figure 3: Human Evaluation Rubric for Measuring Textual Diversity, Degeneration, Creativity, and Coherence.

Diversity: Are the stories fundamentally different from each other?(1 = very poor, 5 = excellent)
Degeneration: Are the sentences natural and fluent, without being broken or degraded?(1 = very poor, 5 = excellent)
Creativity: Are the stories non-trivial, engaging, and imaginative rather than obvious?(1 = very poor, 5 = excellent)
Coherence: Are the stories consistent with the given prompt?(1 = very poor, 5 = excellent)

Figure 4: Human Evaluation Rubric for Measuring Image Diversity, Degeneration, Creativity, and Coherence.

have an L-Lipschitz continuous gradient if for all $x, y \in \mathbb{R}^d$,

$$\|\nabla\mathcal{L}(x) - \nabla\mathcal{L}(y)\| \;\le\; L\|x - y\|.$$

This condition is satisfied by all similarity functions we employ in $\mathcal{L}^{(t)}_{\text{UAG}}$: dot-product similarity(used in autoregressive and diffusion language models), cosine similarity (used in diffusion latent penalties), and CLIP-based similarity (used in diffusion image penalties, where CLIP encoders are smooth neural networks). Hence $\nabla\mathcal{L}^{(t)}_{\text{UAG}}$ is L-Lipschitz for some constant $L > 0$.

Under this assumption, a first-order Taylor expansion yields

$$\mathcal{L}^{(t)}_{\text{UAG}}(\hat{y}_t) \le \mathcal{L}^{(t)}_{\text{UAG}}(y_t) - \eta\left\|\nabla_{y_t}\mathcal{L}^{(t)}_{\text{UAG}}\right\|^2 + O(\eta^2).$$

so for sufficiently small step size η the penalty strictly decreases, thereby reducing similarity to reference bank items. More generally, the smoothness inequality implies that for any update $y_t^+ = y_t - \eta g$ with $g = \nabla_{y_t}\mathcal{L}^{(t)}_{\text{UAG}}$,

$$\mathcal{L}^{(t)}_{\text{UAG}}(y_t^+) \le \mathcal{L}^{(t)}_{\text{UAG}}(y_t) - \eta\Big(1 - \tfrac{L\eta}{2}\Big)\|g\|^2,$$

$$\text{for } \eta < 2/L.$$

Thus, the update guarantees a monotonic decrease in the UAG penalty, ensuring that each step progressively increases dissimilarity from past generations and thereby promotes diversity across branches.

C Analytic Gradients for Penalties

We avoid `torch.autograd.grad` over the model graph as much as possible. Only CLIP loss traverses the external differentiable path(VAE→CLIP), thus uses `torch.autograd`.

Repulsion loss(logit-based). Let $\mathcal{B}_t^{\text{out}} = \{q_1, \ldots, q_N\}$ with each q_r a reference distribution. Then as $\phi(y_t) = p_t = \text{softmax}(y_t)$,

$$\nabla_{y_t}\mathcal{L}^{(t)}_{\text{rep}} = \frac{1}{N}\sum_{r=1}^{N}\big(p_t \odot q_r - (p_t^\top q_r)\, p_t\big).$$

Hidden-state loss. If $b^* = \arg\max_{b \in \mathcal{B}_t^{\text{hid}}} \langle h_t, b \rangle$, then

$$\nabla_{y_t} \mathcal{L}_{\text{hid}}^{(t)} = W b^*.$$

Latent cosine loss(Diffusion). Flatten the latent $z_t \in \mathbb{R}^M$, and let $\{y_j\}$ be past latents. Define $\cos(z_t, y) = \frac{\langle z_t, y \rangle}{\|z_t\| \|y\|}$ and $y^* = \arg\max_j \cos(z_t, y_j)$. Then

$$\nabla_{z_t} \cos(z_t, y^*) = \frac{y^*}{\|z_t\| \|y^*\|} - \frac{\cos(z_t, y^*)}{\|z_t\|^2} z_t.$$

Noise(CLIP) loss. For the CLIP-guided objective that depends on VAE decoding and CLIP embeddings, we rely on automatic differentiation `torch.autograd` to obtain $\nabla_{z_t} \mathcal{L}_{\text{noise}}^{(t)}$.

D Experimental Setup

D.1 Datasets

We use the ReedsyPrompts (Park et al., 2024b) and WritingPrompts (Fan et al., 2018) datasets for story generation, and COCO (Lin et al., 2014) for image generation. For each dataset, we take the first 20 prompts or annotations and generate 15 multi-branch stories or images from the same prompt or annotation, while avoiding previously generated outputs by our method.

D.2 Implementation Details

All training and evaluation are performed on a single NVIDIA A100 GPU(40 GB memory). For all LLM-based qualitative evaluation of *diversity* and *quality*, we employ OpenAI's `GPT-4.1-2025-04-14` (OpenAI, 2025b), following the rubrics in Appendix J. To identify optimal hyperparameters, we conduct ∼300 runs with a dense sweep over the parameter space, analyzing both diversity and degeneration metrics(see Appendix I for details). The resulting best configurations are:

- **LLada-8B**: $\alpha = 1.766$, $\beta = 1.077$, $L_0 = 38$, $\delta = 0.8024$
- **LLaMA-3B**: $\alpha = 0.3395$, $\beta = 1.3339$, $L_0 = 5$, $\delta = 0.5479$
- **Stable Diffusion v1.5**: $\alpha = 0.0579$, $\beta = 0.0208$, $L_0 = 51$, $\delta = 1.8268$

Additional details on the settings of the ablation versions are provided in Appendix I. Unless otherwise noted, text output length is fixed to 200 tokens. For Stable Diffusion v1.5 we use 50 diffusion steps, and for LLada-8B we use 200 steps. For $\text{sim}(\cdot, \cdot)$, both autoregressive and diffusion LMs use dot-product for L_{local} and L_{global}, whereas diffusion image models use cosine similarity for L_{local} and CLIP-based similarity on VAE-decoded images for L_{global}. Additionally, all Language models project hidden states gradient to logit space via W.

D.3 Baselines

Our baselines include ablated versions of our method, $\text{Ours}_{\text{local}}$ and $\text{Ours}_{\text{global}}$, as well as standard sampling methods such as top-k, top-p, and naive sampling with high temperature. $\text{Ours}_{\text{local}}$ applies only the model local similarity penalty, while $\text{Ours}_{\text{global}}$ applies only the global similarity penalty. We also compare the runtime and FLOPs of our method with *Avoidance Decoding*, the current state-of-the-art approach for diverse multi-branch story generation.

D.4 Metrics across Models

We conduct experiments with our method on three types of generative models.

First, we evaluate our method on an autoregressive language model, the LLaMA-3B. Second, we evaluate our method on a diffusion-based language model, the LLada-3B. For both models, to measure the diversity, we use n-gram repetition-based metrics such as BLEU, ROUGE-L, and METEOR, as well as cosine similarity between sentence embeddings obtained from Sentence-BERT. In addition, we employ OpenAI's GPT-4.1 to assess the overall diversity(LLM-D) and the degeneration(Degen) of the samples. Third, we evaluate a diffusion-based image model by applying UAG to Stable Diffusion v1.5. For diversity evaluation, we calculate the CLIP-Score between generated images to assess pairwise similarity, and further employ OpenAI's GPT-4.1 to evaluate overall diversity(LLM-D) and quality(LLM-Q). Finally, we conduct a human evaluation to assess creativity, diversity, degeneration, and coherence across all types of generation models. Additionally, section 4.1 shows significantly lower *runtime* and *computational cost* of our method than previous state-of-the-art methods, such as Avoidance Decoding (Park et al., 2025) and Procreate (Lu et al., 2024). Other additional experimental results are provided in Appendix H, which consistently align with the results here.

The Most Degenerated Story
"It started as an accident. She woke up one morning and found her car parked in front of her house, 100 yards from where it was supposedly parked last night! She brushed it off as a fluke, but the next morning, her car was still parked 100 yards away from where it was supposed to be parked! She freaked out! She swore she swore she swore she hadn't moved the parking spot! The next morning! Her car was still parked 100 yards away from her parking spot! She freaked out! She swore she swore she swore she hadn't moved the parking spot! The next morning! Her car was still parked 100 yards away from her parking spot. She freaked out! She swore she swore she swore she hadn't moved the parking spot! And there she goes! A character who keeps ending up in the same place!"

Table 3: The most degenerated sample in the LLaMA-3B experiments.

E Scheduling Methods Ablation Studies

We additionally experimented with two ablation version of scheduling method, the constant and linear scheduling methods. The constant scheduling method keeps the weights of the two penalties fixed throughout generation, whereas the linear scheduling method adjusts the weights linearly while preserving a constant sum between them. As table 4, 5, 6, logistic scheduling achieves the highest diversity.

F Qualitative Analysis of Degeneration

The relatively high degeneration score of our method is expected because of our highly strict degeneration evaluation rubric (in both automatic and human evaluation). For reference, table 3 is the most severely degenerated output at LLaMA-3B's score of 0.565, which still exhibit storytelling ability.

G Large Scale Model Adaptation

We additionally performed large-scale experiments on LLaMA-70B (Dubey et al., 2024), stable-diffusion-3.5-large (8B) (AI, 2024), under the same settings as in the main experiments of the paper. These include full hyperparameter sweeps; the best results are shown in table 7, 8. The results show that as model size increases, degeneration is better suppressed, proving wide applicability.

H Full Evaluation results

The full evaluation results using the same metrics on different datasets(ReedsyPrompts and WritingPrompts) are reported in Tables 9, 10, 11, and 12. Additionally, we conduct the same human evaluation of the section 4.3 on WritingPrompts, as reported in Table 13.

I Extensive Experiments for Best Hyper-parameters

I.1 Autoregressive Language Model Test

We conduct a total of 33, 33, 33, and 200 experiments to determine the best hyperparameter settings for Ours$_{local}$, Ours$_{global}$, Naive, and Ours, respectively. The ablation and Naive versions involve only a single variable—α for Ours$_{local}$, β for Ours$_{global}$, and temperature for Naive—so we run far fewer experiments for them. For top-k and top-p tests, we set $k = 20$ and $p = 0.9$. In the naive image generation tests, we use a different random seed for each sample. For all textual experiments, we exclusively use the ReedsyPrompts dataset. Results with severe degeneration(Degen > 0.9) are omitted from the figures.

As shown in figures 5–15, there exist points where the trade-off between diversity and degeneration is optimal, i.e., the upper-rightmost points in the plots. We select the variables corresponding to these points as the best hyperparameters. Furthermore, figures 2, 16, 17 present extensive experiments for Ours, where four variables are tuned simultaneously. Again, we choose the variable set that achieves the best trade-off between diversity and degeneration.

J LLM Evaluation Rubrics

The detailed rubrics for LLM evaluation are in Table 18,19,20,21.

K 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

Method	BLEU ↓	Rouge-L ↓	METEOR ↓	SBERT ↓	LLM-D ↑	Degen ↓
logistic (ours)	**0.016**	**0.093**	**0.1364**	**0.430**	**0.6794**	0.5645
const	0.0776	0.2018	0.2844	0.6572	0.44	**0.0**
linear	0.1059	0.2246	0.2973	0.6644	0.32	**0.0**

Table 4: Llama-3B Logitstic Scheduling Ablation Study.

Method	BLEU ↓	Rouge-L ↓	METEOR ↓	SBERT ↓	LLM-D ↑	Degen ↓
logistic (ours)	0.0679	**0.1811**	**0.2270**	**0.5242**	**0.434**	0.3235
const	**0.0533**	0.1869	0.2512	0.5839	0.29	0.035
linear	0.0927	0.2277	0.2704	0.6445	0.33	**0.0**

Table 5: Llada-8B Logitstic Scheduling Ablation Study.

Method	CLIP ↓	LLM-D ↑	LLM-Q ↑
logistic (ours)	**0.477**	**0.860**	0.610
const	0.8759	0.750	0.6625
linear	0.9089	0.750	**0.7375**

Table 6: Stable Diffusion Logitstic Scheduling Ablation Study.

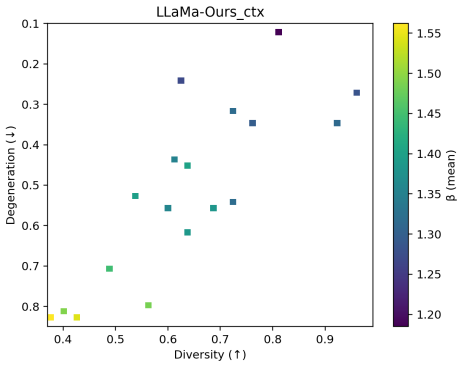

Figure 5: Hyperparameter sweep results for Ours$_{global}$ on the LlaMA-3B.

core methodology, experiments, and analysis, were conducted independently by the authors.

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

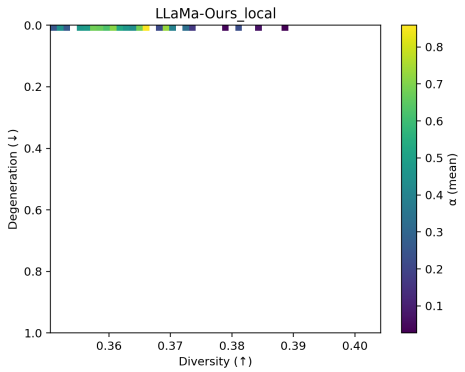

Figure 6: Hyperparameter sweep results for Ours$_{local}$ on the LlaMA-3B.

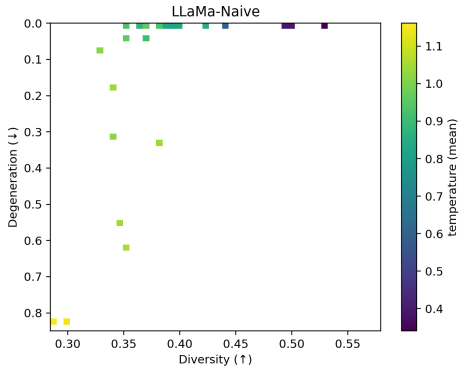

Figure 7: Hyperparameter sweep results for $Naive$ on the LlaMA-3B.

Method	BLEU ↓	ROUGE-L ↓	METEOR ↓	SBERT ↓	Degen ↓	LLM-D ↑
ours	**0.0788**	0.2010	0.2801	0.6267	0.0	0.52
Naive (high temp)	0.1069	0.2281	0.3005	0.6768	**0.0**	0.34

Table 7: LLaMA-70B experiment results.

Method	CLIP ↓	LLM-D ↑	LLM-Q ↑
ours	**0.9023**	**0.700**	0.7313
Naive (random seed)	0.9090	0.450	**0.7438**

Table 8: Stable Diffusion Large (8B) Results.

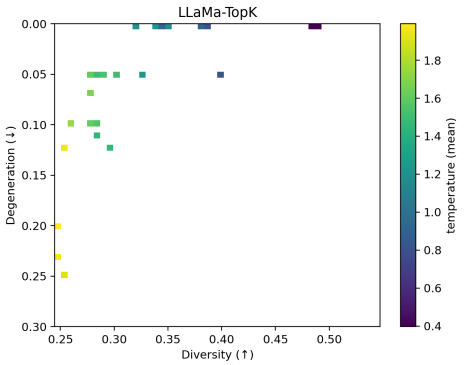

Figure 8: Hyperparameter sweep results for Top-k on the LlaMA-3B.

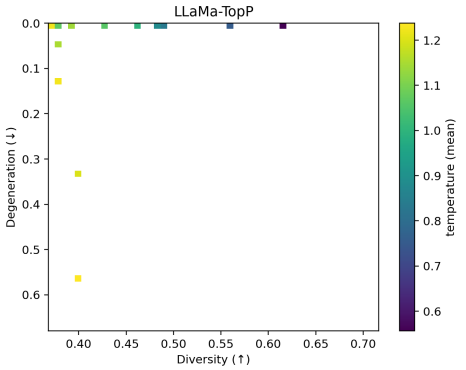

Figure 9: Hyperparameter sweep results for Top-p on the LlaMA-3B.

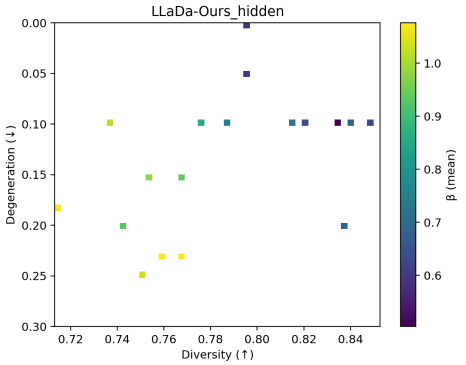

Figure 10: Hyperparameter sweep results for Ours$_{global}$ on the LlaDA-8B.

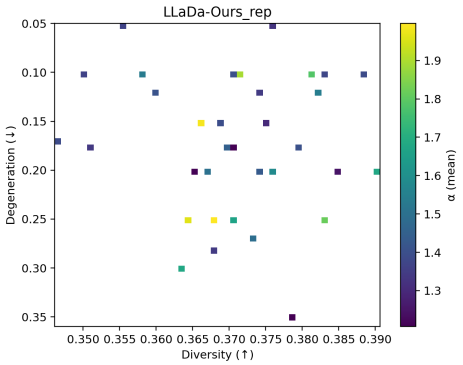

Figure 11: Hyperparameter sweep results for Ours$_{local}$ on the LlaDA-8B.

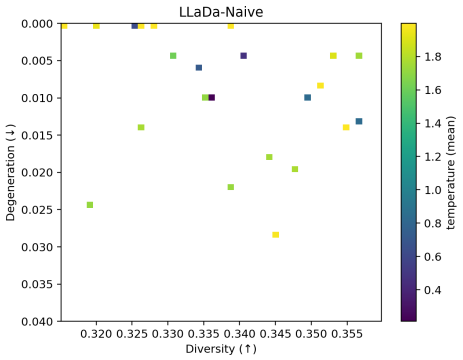

Figure 12: Hyperparameter sweep results for $Naive$ on the LlaDA-8B.

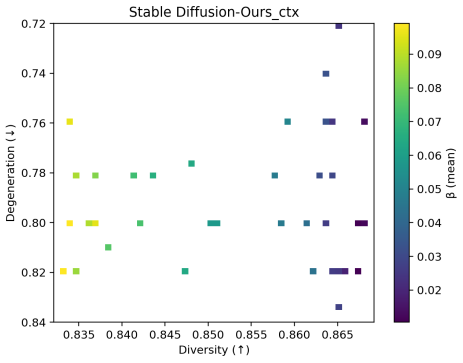

Figure 13: Hyperparameter sweep results for Ours$_{global}$ on the Stable Diffusion.

method	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLM-D(↑)	Degen(↓)
Naive	0.0840	0.2410	0.2570	0.4700	0.2930	0.0050
Top-$k^{\dagger}$	*0.0036*	*0.0059*	*0.0180*	*0.2457*	*0.9300*	*0.9800*
Top-p	0.0573	0.1978	0.2026	**0.4041**	0.4755	0.1930
Avoidance Decoding	**0.0112**	0.1265	0.1483	0.4136	**0.82**	0.25
Ours$_{global}$	0.2540	0.3119	0.3190	0.6570	0.4415	0.3875
Ours$_{local}$	0.1014	0.2120	0.2943	0.6100	0.3905	**0.0000**
Ours	0.0165	**0.0930**	**0.1360**	0.4300	0.6795	0.5645

Table 9: **Llama-3B** with ReedsyPrompts multi-branch story generation results. Lower is better for BLEU, RougeL, METEOR, Sent-Sim, Time, and Degen; higher is better for LLM-D. Best results are highlighted in **bold** and second-best results are underlined. Top-k achieved the best scores on several metrics but is excluded from best/second-best ranking due to extremely high degeneration(0.98).

method	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLM-D(↑)	Degen(↓)
Naive	0.180	0.344	0.356	**0.517**	0.316	0.564
Ours$_{global}$	0.493	0.552	0.565	0.778	0.212	0.157
Ours$_{local}$	0.082	0.213	0.269	0.625	0.278	**0.100**
Ours	**0.067**	**0.181**	**0.227**	0.524	**0.434**	0.323

Table 10: **Llada-8B** with ReedsyPrompts multi-branch story generation results. Lower is better for BLEU, RougeL, METEOR, Cos, and Degen; higher is better for LLM-D. Best results are highlighted in **bold** and second-best results are underlined.

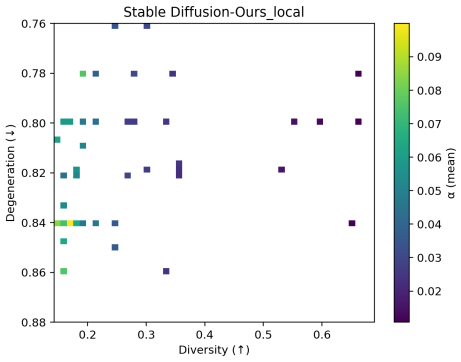

Figure 14: Hyperparameter sweep results for Ours$_{local}$ on the Stable Diffusion.

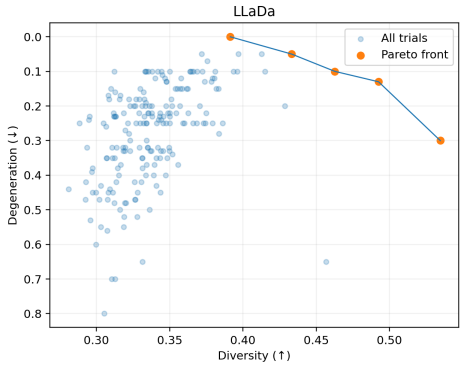

Figure 16: Hyperparameter sweep results for Ours on the LlaDa-8B.

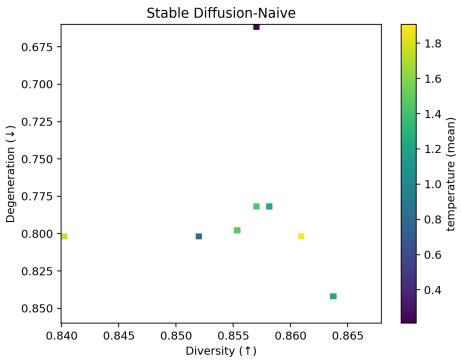

Figure 15: Hyperparameter sweep results for Naive on the Stable Diffusion.

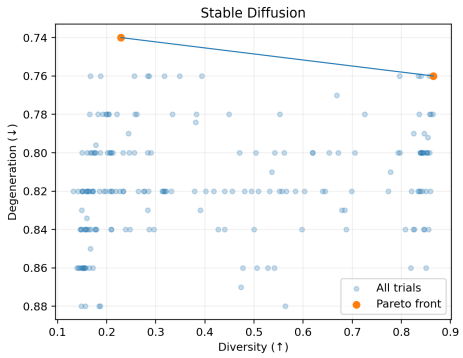

Figure 17: Hyperparameter sweep results for Ours on the Stable Diffusion.

method	BLEU(↓)	RougeL(↓)	METEOR(↓)	Sent-Sim(↓)	LLM-D(↑)	Degen(↓)
Naive	0.0646	0.2402	0.2410	0.4705	0.3790	0.0175
Top-$k^{\dagger}$	*0.0036*	*0.0095*	*0.0173*	*0.2781*	*0.9300*	*0.9800*
Top-p	0.0388	0.1878	0.1850	0.3851	0.5190	0.2395
Avoidance Decoding	**0.0130**	0.1262	0.1632	**0.3708**	**0.7965**	0.27
Ours$_{global}$	0.2192	0.2898	0.2957	0.6118	0.4510	0.4690
Ours$_{local}$	0.1109	0.2244	0.3065	0.6575	0.3770	**0.0075**
Ours	0.0170	**0.1033**	**0.1488**	0.4699	0.6805	0.4875

Table 11: **Llama-3B** with WritingPrompts multi-branch story generation results. Lower is better for BLEU, RougeL, METEOR, Sent-Sim, Time, and Degen; higher is better for LLM-D. Best results are highlighted in **bold** and second-best results are underlined. Top-k achieved the best scores on several metrics but is excluded from best/second-best ranking due to extremely high degeneration(0.98).

method	BLEU(↓)	RougeL(↓)	METEOR(↓)	Cos(↓)	LLM-D(↑)	Degen(↓)
High temperature	0.281	0.479	0.444	**0.453**	0.178	0.398
Ours$_{global}$	0.586	0.638	0.631	0.829	0.193	0.258
Ours$_{local}$	0.072	0.195	0.233	0.593	0.353	0.320
Ours	**0.069**	**0.186**	**0.205**	0.530	**0.435**	0.506

Table 12: **LLADA-8B** with WritingPrompts multi-branch story generation results. Lower is better for BLEU, RougeL, METEOR, Cos, and Degen; higher is better for LLM-D. Best results are highlighted in **bold** and second-best results are underlined.

method	Div(↑)	Degen(↑)	Crt(↑)	Coh(↑)
LLaMA				
Ours$_{global}$	3.0	2.0	3.0	2.8
Ours$_{local}$	**3.7**	**3.7**	**3.6**	**4.0**
Naive	2.3	1.4	1.4	1.7
Top-k	1.0	1.0	1.0	1.0
Top-p	2.1	1.5	1.6	1.8
Ours	**3.7**	2.1	3.3	3.3
LLaDA				
Ours$_{global}$	1.0	2.3	1.7	2.6
Ours$_{local}$	**3.4**	**3.2**	**3.3**	3.3
Naive	1.6	1.0	1.2	1.0
Ours	**3.4**	2.2	**3.3**	**3.5**

Table 13: **Human evaluation results with WritingPrompts.** Div = Diversity, Degen = Degeneration, Crt = Creativity, Coh = Coherence. Higher is better for all metrics. Best results are highlighted in **bold**, and second-best in underlined.

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.

Return only a valid JSON object and nothing else.

Figure 18: Prompt and Evaluation Rubric for Measuring Textual Diversity.

You are a strict judge of IMAGE DIVERSITY.
You will be given multiple images that were generated from the SAME text prompt. Judge how different these images are from each other in content, composition, style, and color palette. Return ONLY a JSON object: {score: <float 0.0 1.0>, reason: <short>}. 0.0 = nearly identical; 1.0 = maximally diverse. Do NOT evaluate prompt-image alignment; only cross-image diversity.

Figure 19: Prompt and Evaluation Rubric for Measuring Image Diversity.

You are a strict judge of text degeneration.
Degeneration includes garbled tokens, random symbols, language mixing, nonsense, broken Unicode, repetitive babble, or clear loss of coherence. Don't judge the repetitiveness across answers, assess the individual quality and average them. Rate on 0.0 1.0: 0.0 clean/coherent, 1.0 severely degenerated. Consider ALL provided answers jointly and set the score to reflect the average observed degeneration. Return pure JSON: {score: <float>, reason: <short>}

Figure 20: Prompt and Evaluation Rubric for Measuring Textual Degeneration.

You are a strict judge of IMAGE GENERATION QUALITY.
You will receive multiple images that were generated from the SAME text prompt. Rate EACH image individually on a 0.0–1.0 scale for: coherence, absence of artifacts, composition, lighting, and overall aesthetics. Do NOT compare images to each other; judge absolute quality per image. Return ONLY JSON of the form: { per_image: [{idx: <int>, score: <float 0.0 1.0>, reason: <short>}, ...], score_mean: <float> } Keep reasons short.

Figure 21: Prompt and Evaluation Rubric for Measuring Image Degeneration.