

TTGen: Incorporating Test-time Scaling to Diffusion Models

Yuming Qiao, Yuechen Wang, Xudong Zhang, Dan Meng*
OPPO Research Institute, Shanghai, China

qym.2021@tsinghua.org.cn, wyc9725@mail.ustc.edu.cn,
shzhangxd@aliyun.com, mengdan90@163.com

Abstract

Test-time scaling has demonstrated significant potential in enhancing the performance of Large Language Models. Beyond the recent surge in Reinforcement Learning based approaches that bolster models' self-reasoning capabilities, techniques such as Monte Carlo Tree Search (MCTS) sampling and Best-of-N (BoN) sampling have also made remarkable strides in improving the quality of model outputs, thereby advancing the field of Artificial General Intelligence (AGI). Visual generation, exemplified by diffusion models, represents a critical domain within AGI. However, there has been limited research exploring the integration of Test-time Scaling with diffusion models. Motivated by this gap and the inherent compatibility between sampling-based Test-time Scaling and diffusion-based generation, we introduce TTGen, a novel framework that integrates sampling-based test-time scaling methods with diffusion models. TTGen operates through a three-step process: (1) Sampling clean latent within each step, where the clean latent for the current step is sampled based on the predicted noise. (2) Refining step-wise prompts, where the current step's clean latent and the initial query are fed into a LVLM (Large Vision-Language Model), prompting the model to output several revised prompts based on the misalignment between current clean latent and initial query. Revised prompts are then used to guide the denoising direction. (3) BoN selection, where at each sampling step, the best diffusion trajectory is progressively selected based on the CLIP score between the revised latent and the initial prompt, thereby enhancing the quality of image generation. Finally, we conduct a series of experiments to validate the effectiveness of TTGen. It is worth noting that the generation results of TTGen demonstrate a 7.1% improvement in CLIP score and a 13.8% enhancement in FID compared to the direct sampling.

1. Introduction

Recent studies are gradually uncovering a new insight that the gains from scaling law are subject to diminishing returns, suggesting that attempts to further enhance model performance by increasing model size have reached a plateau[9, 11, 15]. Conversely, the research community has pivoted towards a strategy of trading time inference for further performance improvements, namely test-time scaling, which has achieved remarkable success[14, 17].

The emergence of novel reasoning LLMs such as OpenAI-o1[16], Deepseek R1[2], and QwQ[13] has effectively demonstrated the efficacy of test-time scaling approaches. Deepseek R1[2] introduced GRPO, leveraging reinforcement learning to boost the model's self-reasoning capabilities, thereby establishing the think-response paradigm as a new standard for LLMs. Beyond these post-training test-time scaling methodologies, search and sampling strategies based on existing models have also proven effective in enhancing the quality of model responses. OpenAI proposed a step-by-step strategy[7], utilizing human-annotated data to train PRM/ORM for supervising the model's Chain-of-Thought (CoT) process, thus improving output quality. Similarly, V-STaR[6] employs correct and incorrect reasoning trajectories for DPO training of a verifier, which then serves as an ORM during inference to select the optimal reasoning process. ReST-MCTS[19] and MCTS-IPL[18] integrate Monte Carlo Tree Search strategies into the LLM sampling process, treating the model's CoT process as distinct nodes and employing majority voting as an ORM to iteratively update the reasoning tree's node Q-values, ultimately selecting the highest quality reasoning process based on node Q-values to enhance model output quality.

The aforementioned sampling-based test-time scaling methods rely on a prerequisite: the base model must inherently possess substantial output diversity, as higher output diversity can better unleash the potential of sampling-based test-time scaling. It is well-known that diffusion models in the visual generation domain exhibit strong output diversity[10], where even minor variations in the ini-

*Corresponding author.

tial noise can lead to significantly different sampling results. Therefore, diffusion models exhibit a high degree of compatibility with sampling-based Test-time Scaling methods. However, there has been scarce research exploring the integration of Test-time Scaling within diffusion models. Inspired by these characteristics, we propose TTGen, an innovative approach that integrates sampling-based test-time scaling strategies with the sampling process of diffusion models. Although the initial noise has a considerable impact on the sampling results, once the initial noise is fixed, the sampling process becomes deterministic. To incorporate test-time scaling laws into the sampling process, rather than manipulating the initial noise, we focus on refining the predicted noise during the diffusion model’s sampling process. TTGen consists of three key components: 1. Sampling clean latent at each sampling step based on the predicted noise. 2. Utilizing a prompting mechanism with LVLM(Large Vision-Language Model) to analyze the discrepancies between the clean latent and the initial prompt, thereby rewriting several prompts to manipulate the predicted noise and generate multiple candidate intermediate latents. 3. Employing the CLIP score[3] between the step-wise clean latent and the initial prompt as an ORM to select the highest-quality latent from the intermediate latents as the input for the next sampling step. Through this refine-supervise paradigm, TTGen effectively enhances the robustness of the diffusion model’s sampling process and improves the quality of the final generated images.

2. Methodology

As illustrated in Fig.1, the TTGen pipeline takes a init prompt q as input and initiates the test-time diffusion process. At each sampling step, leveraging the capabilities of LVLM, the pipeline identifies misalignment between the current step’s clean latent and the initial prompt q , progressively refining q into $\{q_t^i\}_{i=1:N}^{t=1:T}$, where t represents the current sampling step, T denotes the total number of sampling steps. i and N are the order and total number of re-written prompts. Subsequently, at each step, a CLIP score-based ORM is employed to filter the diffusion trajectories, ultimately yielding an enhanced generation result.

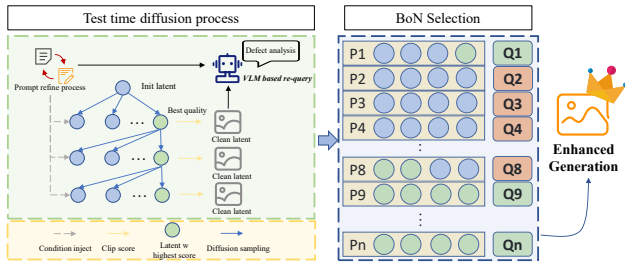


Figure 1. Overview of our proposed TTGen, a plug and play framework incorporates test-time scaling to diffusion models.

2.1. Test-time diffusion process

The diffusion model takes the initial noise and the initial prompt q as input for sampling. At each sampling step t , the model outputs the corresponding noise ε_t . Based on ε_t and Eq.1, one can sample x_{t-1} from x_t .

$$x_t = \alpha_t x_{t-1} + \beta_t \varepsilon_t \quad (1)$$

Further expansion of this formula yields Eq.2, from which we can also obtain the approximate clean latent x_0^t at timestep t .

$$x_t = (\alpha_t \cdots \alpha_1) x_0^t + \sqrt{1 - (\alpha_t \cdots \alpha_1)^2} \bar{\varepsilon}_t \quad (2)$$

The predicted noise is given by Eq.3, and it can be observed that altering the condition, i.e., the prompt embedding ϕ_{cond} , can change the noise prediction results ε_t , thereby influencing the denoising direction of the diffusion process.

$$\varepsilon_t = \varepsilon_{\theta}(x_t, t, \phi_{none}) + \omega_g(\varepsilon_{\theta}(x_t, t, \phi_{cond}) - \varepsilon_{\theta}(x_t, t, \phi_{none})) \quad (3)$$

Therefore, at each step, we use x_0^t and the initial query q as inputs to prompt the Vision-Language Model (VLM), allowing it to rewrite q based on the alignment between q and x_0^t , resulting in a set of revised prompts $\{r q_i^j\}_{i=1:T}^{j=1:M}$, where M is the number of revised prompts in one step. These revised prompts tend to emphasize aspects of q that are not well-represented by x_0^t . Finally, at the current sampling step t , the candidate intermediate latents x_t are sampled using $\{r q_t^j\}_{j=1:M}$ and x_{t+1} as inputs.

2.2. BoN selection

During the Best-of-N (BoN) selection process, we employ the CLIP score as the Objective Reward Model (ORM) to supervise the diffusion process. For each intermediate latent x_t^j sampled from $\{r q_t^j\}_{j=1:M}$ and x_{t+1} , we compute the CLIP score between its corresponding clean latent $x_0^{t,j}$ and the initial prompt q . The intermediate latent with the highest CLIP score is then selected as the input for the next step. We visualize several examples of this process, as illustrated in Fig.2.

With the same initial prompt and noise, the first row illustrates the optimal diffusion sampling process refined by TTGen. The middle row depicts the sampling process directly using the initial prompt without any refining procedure. The last row, labeled as the worst sampling, represents the inverse process of TTGen’s Best-of-N (BoN) sampling, where, instead of selecting the trajectory with the highest CLIP score, the trajectory with the lowest CLIP score is chosen to serve as the lower bound for the diffusion model’s generation. From these comparisons, it is evident that TTGen significantly enhances the alignment between

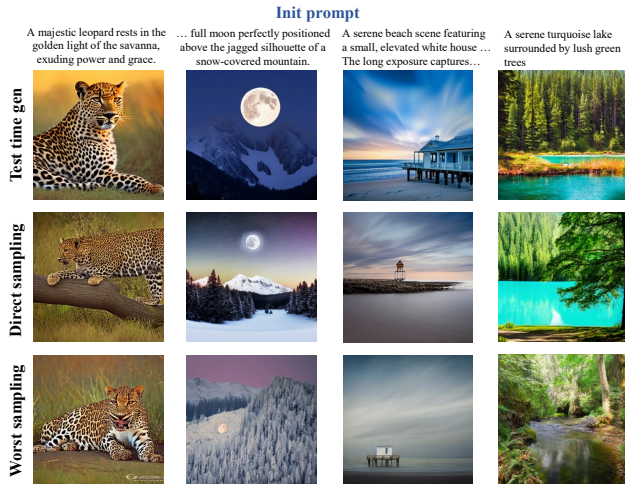


Figure 2. Generation results visualization of TTGen, Direct sampling and worst sampling.

the generated results and the initial prompt. For instance, TTGen distinctly better captures the “golden light of the savanna” mentioned in the prompt, a feature that is notably absent in both the direct sampling and worst sampling outcomes. In the context of the second prompt, TTGen more accurately renders the characteristic “above the jagged silhouette of a snow-covered mountain.” Similarly, for the third and fourth prompts, TTGen improves the realism of the entities “beach” and “turquoise lake” respectively.

The step-wise CLIP score trends of the aforementioned process are illustrated in Fig.3. It can be observed that TTGen consistently maintains a significant advantage in alignment between the sampling results and the initial prompt throughout the sampling process. This further demonstrates the superiority and practical utility of TTGen.

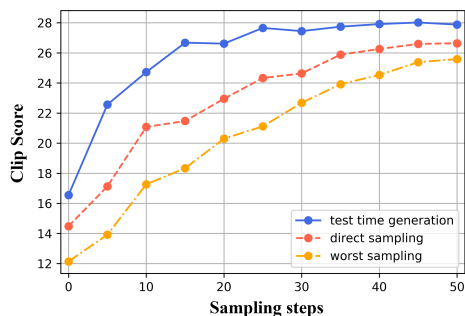


Figure 3. The variation in CLIP scores for TTGen, Direct Sampling, and Worst Sampling as a function of sampling steps.

3. Experiment

3.1. Implementations

All experiments were implemented based on Stable Diffusion v1.5[10]. For the diffusion model, the DDIM

scheduler[12] was selected, with inference steps set to 50 and a guidance scale of 7.5[5]. The Vision-Language Model (VLM) used to rewrite prompts during the diffusion model sampling process was InternVL2.5-26B[1], a recently advanced VLM that can be co-loaded with SD1.5 on a single A100 GPU. Additionally, we compared TTGen, Direct Sampling, and Worst Sampling on a subset of the MS COCO text-to-image benchmark[8] using FID[4] and CLIP score[3] metrics. The FID metric effectively measures the diversity and quality of the generated images, while the CLIP score evaluates the alignment between the generated results and the textual prompts.

3.2. Evaluations

The evaluation results are presented in Table 1, demonstrating that TTGen exhibits marginal advantages in prompt alignment, generation quality, and diversity.

Method	Mscoco	
	clip score \uparrow	FID \downarrow
TTGen	27.1	20.9
Direct sampling	25.3	23.8
Worst sampling	23.1	24.6

Table 1. Quantitative experiments on MScoco T2I benchmark subset.

A more detailed illustration of the TTGen process is shown in Fig.4.

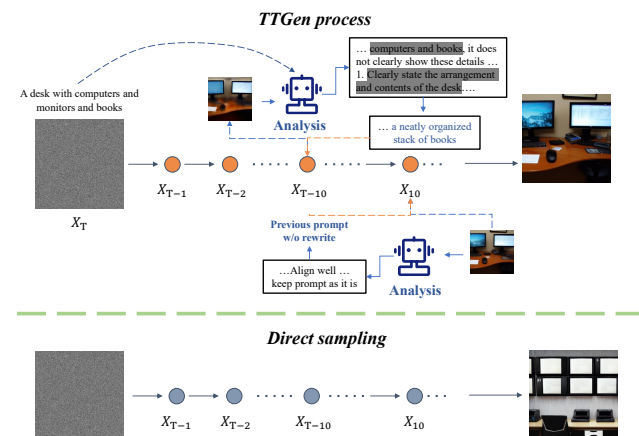


Figure 4. An illustration of the distinctions between the sampling processes of TTGen and Direct Sampling.

It can be observed that the direct sampling results using the initial prompt, “A desk with computers and monitors and books” are suboptimal, lacking adequate representation of entities such as computers and books. In contrast, within the TTGen process, the LVLM adjusts the prompt to address the insufficient representation of these entities, thereby enhancing their presence in the generated results.

For instance, at the 10th sampling step x_{T-10} , although the sampled clean latent is relatively blurry, the LVLM effectively identifies the absence of book-related information in the current clean latent and refines the prompt to emphasize the depiction of books in the image, such as "...a neatly organized stack of books." This adjustment influences the sampling process, strengthening the representation of books in the final generated image. By the 40th sampling step (x_{10}), the model determines that the current clean latent sufficiently represents the entities described in the initial prompt, and thus refrains from further modifying the prompt in this sampling iteration. The generated results clearly demonstrate that TTGen surpasses direct sampling in terms of generation quality and prompt adherence, proving the effectiveness of modifying the denoising direction through prompt adjustments during the sampling process.

3.3. Ablation study

Inspired by the observations in Fig.4, it can be noted that if the early sampling stages yield favorable results, the frequency of prompt modifications in the later stages of sampling tends to decrease. As the sampling process progresses, the overall direction of denoising becomes relatively stable, and adjusting the prompt to alter the denoising direction may have limited impact on the final generated outcome. In this section, we investigate the effect of the injection step (the point at which the prompt is adjusted during the sampling process) on the final generated results. As illustrated in Fig.5, it is evident that as the injection step decreases, the clip score also gradually declines. Notably, in the overlapping regions of the injection steps, such as when the injection step is less than 5, the corresponding sampling segments align almost entirely with those of other injection steps. However, once the sampling process surpasses the designated injection step, the clip score is significantly affected and decreases markedly.

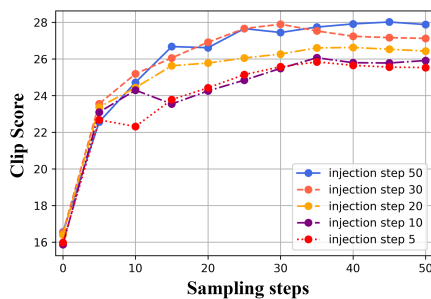


Figure 5. An ablation study on the impact of different injection steps on the generation quality of TTGen.

We visualize two examples to more intuitively demonstrate the impact of the injection step on the generation results of TTGen as shown in Fig.6. In the first row, it is evident that the generation result with an injection step of 5

fails to effectively express the concept of "jagged silhouette of a snow-covered mountain." However, when the injection step is set to 10, the generation result begins to exhibit a preliminary and meaningful representation of the entity. As the injection step gradually increases, the quality of the entity representation in the generated results progressively improves. In the second row, when the injection step is 5, the generation results poorly represent both "beach" and "white house." In contrast, with injection steps of 10, 20, and 30, the representation of these two subjects becomes increasingly detailed and enriched. Notably, the generation result with an injection step of 30 shows strong alignment with the result of full injection, achieving an enhanced generation effect. This observation aligns with the trend of clip score changes depicted in Fig.5. Therefore, in practical applications, it is often unnecessary to adopt a full injection refinement strategy. Instead, using an injection step of 30 or even 20 can ensure the quality of the generated results while significantly improving sampling efficiency.



Figure 6. Visualization of the Impact of Different Injection Steps on the Generation Quality of TTGen. Higher injection steps, such as 20, 30, and 50, significantly enhance the generation quality. Considering both generation quality and sampling time, an injection step of 20 or 30 emerges as an optimal default choice.

4. Conclusion

In this paper, we introduce TTGen, a plug-and-play framework that integrates Test-time Scaling with diffusion models. Inspired by the application of sampling-based Test-time Scaling methods in Large Language Model (LLM), TTGen employs this strategy during the diffusion model sampling process to identify the optimal denoising trajectory. Specifically, TTGen first samples clean latents at each denoising step and then utilizes a Large Vision-Language Model (LVLM) to refine the prompt by addressing the misalignment between the clean latent and the initial prompt, thereby enhancing the expression of specific entities. Furthermore, the sampling process is supervised using the CLIP score, enabling the selection of the highest-quality latents and refined prompts at each step to reinforce the model's generation results. Quantitative experiments and ablation studies further validate the efficacy of TTGen and demonstrate the significant potential of Test-time Scaling in the domain of visual generation.

References

- [1] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 3
- [2] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1
- [3] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP (1)*, 2021. 2, 3
- [4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637, 2017. 3
- [5] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3
- [6] Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-star: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling*. 1
- [7] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023. 1
- [8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision—ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014. 3
- [9] Charles Luo. Has llm reached the scaling ceiling yet? unified insights into llm regularities and constraints. *arXiv preprint arXiv:2412.16443*, 2024. 1
- [10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 3
- [11] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. 1
- [12] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3
- [13] Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, 2024. 1
- [14] Fengwei Teng, Zhaoyang Yu, Quan Shi, Jiayi Zhang, Chenglin Wu, and Yuyu Luo. Atom of thoughts for markov llm test-time scaling. *arXiv preprint arXiv:2502.12018*, 2025. 1
- [15] Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*. 1
- [16] Siwei Wu, Zhongyuan Peng, Xinrun Du, Tuney Zheng, Minghao Liu, Jialong Wu, Jiachen Ma, Yizhi Li, Jian Yang, Wangchunshu Zhou, et al. A comparative study on reasoning patterns of openai’s o1 model. *arXiv preprint arXiv:2410.13639*, 2024. 1
- [17] Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang Hong, Shihan Do, Wenyu Zhan, et al. Enhancing llm reasoning via critique models with test-time and training-time supervision. *arXiv preprint arXiv:2411.16579*, 2024. 1
- [18] Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*, 2024. 1
- [19] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772, 2024. 1