ConceptMix++: Leveling the Playing Field in Text-to-Image Benchmarking via Iterative Prompt Optimization

Haosheng Gan, Berk Tinaz, Mohammad Shahab Sepehri, Zalan Fabian, Mahdi Soltanolkotabi
University of Southern California
Los Angeles, CA 90089
{woodygan, tinaz, sepehri, zfabian, soltanol}@usc.edu

Abstract

Current text-to-image (T2I) benchmarks evaluate models on rigid prompts, potentially underestimating true generative capabilities due to prompt sensitivity and creating biases that favor certain models while disadvantaging others. We introduce ConceptMix++, a framework that disentangles prompt phrasing from visual generation capabilities by applying iterative prompt optimization. Building on ConceptMix, our approach incorporates a multimodal optimization pipeline that leverages vision-language model feedback to refine prompts. Through extensive experiments across multiple diffusion models, we show that optimized prompts significantly improve compositional generation performance, revealing previously hidden model capabilities and enabling fairer comparisons across T2I models. Our analysis reveals that certain visual concepts – such as spatial relationships and shapes – benefit more from optimization than others, suggesting that existing benchmarks systematically underestimate model performance in these categories. Additionally, we find strong cross-model transferability of optimized prompts, indicating shared preferences for effective prompt phrasing across models.

1. Introduction

Text-to-image (T2I) generation aims to synthesize images based on user-specified textual descriptions. Diffusion models (DM) [4, 17] have established state-of-the-art in image [3, 5, 11, 14, 16], audio [9], and video generation [6]. In the specific domain of T2I generation, DM-based approaches have become the dominant paradigm [14, 15]. To enable systematic evaluation and comparison of DM-based T2I models, several benchmarks have been proposed [2, 7, 19]. However, evaluating and comparing these models remains challenging due to their prompt sensitivity [18]. For example, as illustrated in Figure 1, two prompts expressing the same scene ("four cows", "glass texture", "tiny rose") yield drastically different results depending on how they are phrased. One prompt produces a successful generation (bottom), while the other fails entirely (top).

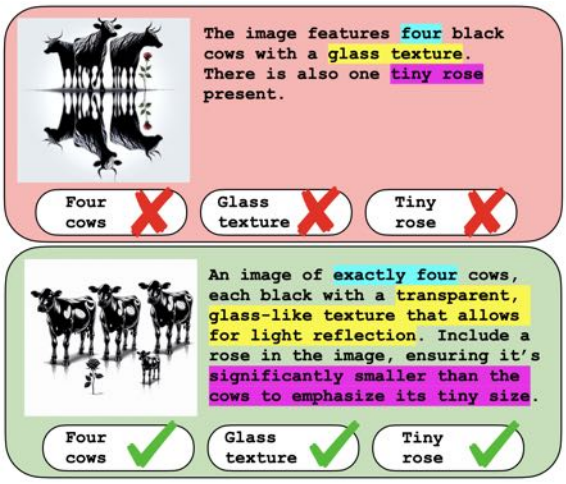


Figure 1. Text-to-image models are sensitive to the specific phrasing of the input prompt, thus using a rigid prompt format may underestimate generation capabilities.

To address this limitation, we propose ConceptMix++, a framework that disentangles prompt understanding from visual generation capabilities via prompt optimization. Building on ConceptMix [19], our approach incorporates a multimodal optimization module that leverages feedback from vision-language models to iteratively refine prompts. Through extensive experiments with multiple state-of-the-art diffusion models, we demonstrate that:

- Optimized prompts substantially improve compositional generation performance across architectures.
- Visual concept categories (e.g., spatial relations, shapes) benefit unevenly from optimization, highlighting category-specific bottlenecks.
- Optimized prompts exhibit cross-model transferability, suggesting shared preferences in effective prompting.

ConceptMix++ offers a more nuanced lens for evaluating T2I models, surfacing capabilities that conventional benchmarks may overlook and exposing fundamental limitations in current evaluation methodologies.

2. Method

Overview of our framework– ConceptMix++ is a framework designed to disentangle prompt phrasing from visual synthesis abilities of T2I models. Our key insight is that by adapting and optimizing prompts for each model individually, we can more accurately assess their true visual generation potential.

Our framework operates in three-stages:

1. Baseline prompt evaluation: Evaluate model performance using standard benchmark prompts.
2. Prompt optimization: Use our prompt optimization module to iteratively refine prompts tailored to each model.
3. Capability analysis: Assess each model's full visual synthesis ability under optimized prompting.

This methodology allows for a more nuanced analysis of model capabilities, revealing: (1) performance gains (2) persistent limitations that remain even with optimized prompts.

Evaluating baseline performance– For baseline evaluation, we adopt the ConceptMix [19] benchmark, which measures compositional generation capabilities across a broad range of visual concept categories. This benchmark is particularly well-suited for capability analysis because:

- It provides fine-grained evaluations across 8 diverse visual concept categories.
- It supports scalable compositional complexity denoted with complexity level k (where k ranges between 1 and 7)

At each complexity level k, ConceptMix specifies $k+1$ criteria, each corresponding to one of the 8 visual concept categories. The initial prompt p_0 is generated by GPT-4o by providing it with full list of criteria. The diffusion model then synthesizes an image $\mathcal{I}$ from this prompt. To evaluate the generated image, ConceptMix utilizes GPT-4o acting as a verifier $\mathcal{V}$ which answers a yes/no question for each criterion to determine whether it has been satisfied. The image receives a score of 1 only if all criteria are met:

$$s(\mathcal{I}) = \prod_{i=1}^{k+1} \mathbb{1}(\mathcal{V}(\text{“Is criteria } i \text{ satisfied ?”}|\mathcal{I}) = \texttt{"Yes"})$$

where $\mathbb{1}$ is the indicator function that equals 1 if the verifier thinks the criteria is satisfied and 0 otherwise.

Multimodal prompt optimization– To systematically enhance prompts, we introduce a prompt optimization module inspired by TextGrad [20], tailored specifically for the text-to-image (T2I) domain. As illustrated in Figure 2, our module operates end-to-end across the T2I generation and evaluation pipeline, refining prompts based on visual outcomes.

Given a T2I diffusion model $\mathcal{D}$ and a vision-language evaluator $\mathcal{V}$, the goal is to find an optimal prompt p^* that maximizes the evaluation score:

$$p^* = \arg\max_p \mathcal{V}(\mathcal{D}(p))$$

Here, $\mathcal{D}(p)$ denotes the image generated by the diffusion model from prompt p, and $\mathcal{V}(\mathcal{I})$ returns a scalar score assessing how well image $\mathcal{I}$ satisfies the specified visual concepts.

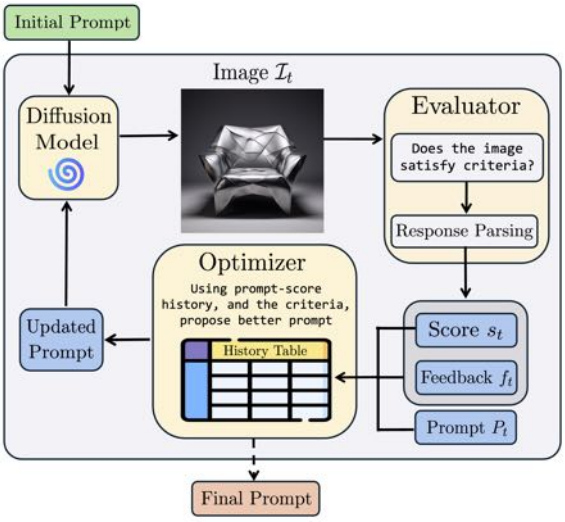


Figure 2. Overview of our prompt optimization framework. Starting with an initial prompt, our iterative process first generates images using a diffusion model. Next, an evaluator scores and provides feedback. Finally, LLM optimizer proposes improved prompts based on past prompt-score pairs.

The optimization process (Figure 2) follows an iterative loop:

1. Generate an image $\mathcal{I}_t = \mathcal{D}(p_t)$ from the current prompt.
2. Evaluate $\mathcal{I}_t$ using the VLM to obtain a score $s_t = \mathcal{V}(\mathcal{I}_t)$ and feedback f_t.
3. Store the prompt, score, and feedback tuple (p_t, s_t, f_t) in a history buffer H, sorted by score.
4. Update the prompt $p_{t+1} = \mathcal{U}_{\text{LLM}}(p_{best}, H)$, where $\mathcal{U}_{\text{LLM}}(\cdot)$ is an LLM-based update function.

Unlike gradient-based optimization in continuous spaces, our method exploits the LLM's capability to learn from qualitative feedback and historical patterns, generating refined prompts in natural language.

To enable more fine-grained and stable updates, we extend ConceptMix's binary evaluation to a probabilistic one. Specifically, we use the likelihood of the affirmative answer ("Yes") predicted by the vision-language model for each criterion:

$$\mathbb{E}[s(\mathcal{I})] = \prod_{i=1}^{k+1} P(\mathcal{V}(\text{“Is criteria } i \text{ satisfied ?”}|\mathcal{I}) = \texttt{"Yes"})$$

We provide further implementation details and analysis of the optimization loop in Appendix A.

3. Experiments

3.1. Setup

We apply our ConceptMix++ framework to benchmark three state-of-the-art diffusion models: stable-diffusion-3.5-medium [1], playground-v2.5-1024px-aesthetic [10], and DALL·E 3 [12]. In this setup, we use GPT-4o (2024-08-06) [8] as both the evaluator and the optimizer.

For each complexity level $k = 1$ to $k = 7$, we select 300 datapoints from the ConceptMix benchmark. To reduce the effect of randomness, we generate 5 images per prompt and evaluate performance based on two metrics. The *average* score is the average across the 5 generations, while *best-of-5* score is the highest score among them.

3.2. Performance Analysis

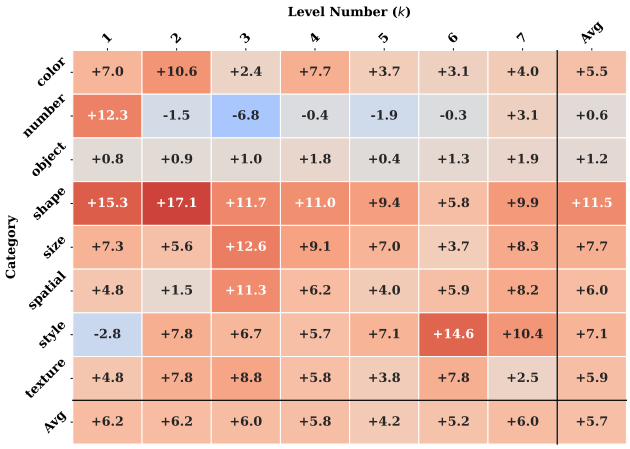
Table 1 presents the performance comparison between original and optimized prompts across all diffusion models and complexity levels. We observe substantial improvements following optimization – up to $\approx 20\%$ absolute gains in both *average* and *best-of-5* scores for mid-range complexity levels ($k = 3$ to $k = 5$) across all models. These gains reflect the significant value of prompt optimization in unlocking true capabilities, particularly for compositions that require understanding and generating multiple visual concepts simultaneously. Interestingly, the performance gap between original and optimized prompts widens with increasing k in the low-to-mid range, suggesting that prompt refinement becomes increasingly beneficial as compositional complexity grows. However, at the highest complexity levels ($k = 6$, $k = 7$), the gains taper off. This diminishing return likely stems from an upper bound imposed by the model's capacity itself: the prompts may become too detailed or domain-shifted compared to the model's training distribution, limiting the effectiveness of optimization.

Overall, these results reveal that fixed-prompt benchmarks substantially underestimate the potential of diffusion models, especially in the moderate complexity regime. Prompt optimization not only elevates baseline performance but also provides a fairer, more accurate view of model capability in real-world, compositional scenarios.

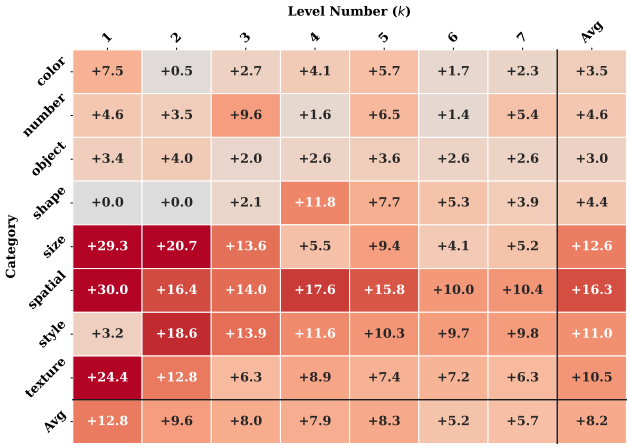
3.3. Category-wise Analysis

To gain deeper insight into the impact of our optimization framework, we conduct a category-wise analysis across eight visual concept categories defined in the ConceptMix benchmark: *color*, *number*, *object*, *shape*, *size*, *spatial*, *style*, and *texture*. This analysis allows us to examine how prompt optimization influences specific dimensions of compositional generation and to identify which aspects of diffusion model capabilities benefit most from our framework.

For each diffusion model and complexity level, we compute category-specific scores by evaluating performance



(a) DALL·E 3



(b) Stable Diffusion 3.5

Figure 3. Heatmaps showing improvement magnitude across visual concept categories and complexity levels. Darker red indicates higher improvements, while blue indicates a decline in performance.

on questions associated with each visual concept category. Formally, for a model M, complexity level k, and category c with its corresponding set of questions Q_c, the score $S_{M,k,c}$ for an image $\mathcal{I}$ is defined as:

$$S_{M,k,c}(\mathcal{I}) = \frac{1}{|Q_c|} \sum_{q \in Q_c} \mathbb{1}(\mathcal{V}(q|\mathcal{I}) = \texttt{"yes"})$$

Figure 3 presents heatmap visualizations of category-wise improvement achieved by our optimization framework for Stable Diffusion 3.5 and DALL·E 3 across all complexity levels. Corresponding heatmap for Playground v2.5 is provided in Appendix C.

For Stable Diffusion 3.5, the most significant improvements are observed in *spatial* and *size* categories, each showing an average gain exceeding 12%. Meanwhile, for DALL·E 3, the most improvement happens for *shape* with an average improvement of 11.5%. These observations

Table 1. Performance comparison between original and optimized prompts across different complexity levels k. For each model, we report both average and best-of-5 scores under default and optimized prompts, highlighting performance gaps revealed through prompt optimization.

Model	Metric	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$
DALL·E 3	average (default)	0.824 ± 0.021	0.621 ± 0.015	0.460 ± 0.025	0.292 ± 0.025	0.207 ± 0.024	0.139 ± 0.028	0.085 ± 0.012
	average (optimized)	0.882 ± 0.009	0.723 ± 0.023	0.605 ± 0.024	0.415 ± 0.028	0.313 ± 0.029	0.218 ± 0.009	0.127 ± 0.019
	best-of-5 (default)	0.945	0.853	0.668	0.537	0.449	0.306	0.191
	best-of-5 (optimized)	0.996	0.943	0.868	0.761	0.648	0.505	0.364
SD 3.5	average (default)	0.736 ± 0.018	0.503 ± 0.024	0.351 ± 0.010	0.226 ± 0.020	0.157 ± 0.009	0.089 ± 0.009	0.067 ± 0.006
	average (optimized)	0.877 ± 0.006	0.662 ± 0.008	0.512 ± 0.020	0.356 ± 0.019	0.277 ± 0.017	0.172 ± 0.016	0.123 ± 0.016
	best-of-5 (default)	0.913	0.762	0.595	0.483	0.360	0.232	0.201
	best-of-5 (optimized)	0.973	0.856	0.776	0.678	0.579	0.423	0.298
PG v2.5	average (default)	0.732 ± 0.017	0.451 ± 0.023	0.241 ± 0.020	0.117 ± 0.010	0.062 ± 0.009	0.017 ± 0.004	0.003 ± 0.002
	average (optimized)	0.831 ± 0.010	0.572 ± 0.006	0.351 ± 0.014	0.199 ± 0.009	0.116 ± 0.013	0.047 ± 0.009	0.019 ± 0.005
	best-of-5 (default)	0.856	0.640	0.440	0.233	0.167	0.047	0.014
	best-of-5 (optimized)	0.936	0.783	0.601	0.373	0.278	0.128	0.057

highlight the fact that *spatial* and *size* categories for Stable Diffusion 3.5 and *shape* category for DALL·E 3 require precise prompt formulations. On the other hand, the *object* category exhibits the least improvement for both models, with an average gain of 3% for Stable Diffusion 3.5 and 1.2% for DALL·E 3 respectively. This is mainly because the original prompts already perform well in this category, leaving limited room for further gains. Additionally, the *number* category shows marginal improvements in both models. This can be attributed to the well-known challenge that generative models have with accurately representing specific quantities ([7, 13, 16]), compounded by the fact that specifying a number offers little flexibility for refinement. As a result, the performance in this category remains low both before and after optimization, suggesting that prompt refinement alone is insufficient to overcome this limitation.

3.4. Cross-Model Prompt Transferability

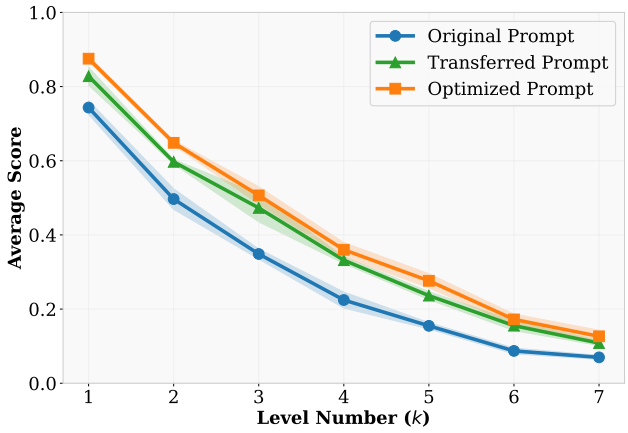


Figure 4. Average score of Stable Diffusion 3.5 using original prompts, prompts optimized for Stable Diffusion 3.5 itself, and prompts optimized for DALL·E 3.

A key question about our framework is whether the prompts optimized for one model can effectively generalize to another. To investigate this, we evaluate the performance of Stable Diffusion 3.5 when using prompts that were optimized for DALL·E 3. The results, shown in Figure 4, compare the average scores of Stable Diffusion 3.5 using original prompts, prompts optimized for Stable Diffusion 3.5 itself, and prompts optimized for DALL·E 3. Appendix D provides additional experiments addressing this question.

We observe that Stable Diffusion 3.5 performs significantly better with DALL·E 3-optimized prompts compared to original prompts, and its performance closely approaches that achieved with prompts optimized for itself. This indicates a high degree of cross-model transferability, suggesting that the optimized prompts capture phrasing patterns that are effective among all models. These findings imply that different diffusion models may share underlying prompt preferences, making cross-model prompt reuse a viable strategy for enhancing generation quality with reduced optimization overhead.

4. Conclusion

In this work, we introduce ConceptMix++, a novel framework for fair benchmarking of text-to-image diffusion models through iterative prompt optimization. Our experiments show that ConceptMix++ consistently enhances model performance, revealing their true compositional generation capabilities. We further analyze the impact of our framework across different visual concept categories, identifying which aspects benefit most from prompt refinement. Additionally, we demonstrate that optimized prompts exhibit strong cross-model transferability, suggesting shared prompt preferences among models. These findings indicate that fixed-prompt benchmarks substantially underestimate the capabilities of text-to-image models and highlight the critical role of prompt formatting in fully realizing their generative potential.

5. Acknowledgements

We would like to thank Microsoft for an Accelerating Foundation Models Research grant that provided the OpenAI credits enabling this work.

References

[1] Stability AI. Stable diffusion 3.5, 2024. 3

[2] Eslam Mohamed Bakr, Pengzhan Sun, Xiaoqian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20041–20053, 2023. 1

[3] Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. *arXiv preprint arXiv:2105.05233*, 2021. 1

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *arXiv preprint arXiv:2006.11239*, 2020. 1

[5] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(47):1–33, 2022. 1

[6] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022. 1

[7] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023. 1, 4

[8] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 3

[9] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. 1

[10] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024. 3

[11] Alex Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. *arXiv preprint arXiv:2102.09672*, 2021. 1

[12] OpenAI. Dall·e 3, 2024. 3

[13] Vitali Petsiuk, Alexander E Siemenn, Saisamrit Surbehera, Zad Chin, Keith Tyser, Gregory Hunter, Arvind Raghavan, Yann Hicke, Bryan A Plummer, Ori Kerret, et al. Human evaluation of text-to-image models on a multi-task benchmark. *arXiv preprint arXiv:2211.12112*, 2022. 4

[14] 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

[15] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image Super-Resolution via Iterative Refinement. *arXiv:2104.07636 [cs, eess]*, 2021. 1

[16] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 1, 4

[17] Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. *arXiv:1907.05600 [cs, stat]*, 2020. 1

[18] Sam Witteveen and Martin Andrews. Investigating prompt engineering in diffusion models, 2022. 1

[19] Xindi Wu, Dingli Yu, Yangsibo Huang, Olga Russakovsky, and Sanjeev Arora. Conceptmix: A compositional image generation benchmark with controllable difficulty. *arXiv preprint arXiv:2408.14339*, 2024. 1, 2

[20] Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024. 2

ConceptMix++: Leveling the Playing Field in Text-to-Image Benchmarking via Iterative Prompt Optimization

Supplementary Material

A. Additional details on prompt optimization module

A.1. Algorithm

Algorithm 1 outlines the complete prompt optimization framework:

Algorithm 1 Text2Image Grad

Require: Initial prompt p_0, diffusion model $\mathcal{D}$, VLM $\mathcal{V}$, LLM $\mathcal{U}_{\text{LLM}}$, maximum iterations T
Ensure: Optimized prompt p^*
  Initialize history table $\mathcal{H} \leftarrow \emptyset$
  Initialize best score $s_{\text{best}} \leftarrow -\infty$
  Initialize best prompt $p_{\text{best}} \leftarrow p_0$
  for $t = 0$ to T **do**
    Generate image $\mathcal{I}_t \leftarrow \mathcal{D}(p_t)$
    Evaluate image and generate feedback $(s_t, f_t) \leftarrow \mathcal{V}(\mathcal{I}_t)$
    Update history $\mathcal{H} \leftarrow \mathcal{H} \cup \{(p_t, s_t, f_t)\}$
    if $s_t > s_{\text{best}}$ **then**
      $s_{\text{best}} \leftarrow s_t$
      $p_{\text{best}} \leftarrow p_t$
    end if
    if $t == T$ **then**
      break
    end if
    Generate improved prompt $p_{t+1} \leftarrow \mathcal{U}_{\text{LLM}}(p_{\text{best}}, \mathcal{H})$
  end for
  return p_{best}

A.2. Analogy with traditional Gradient Descent

Traditional gradient descent optimization iteratively updates parameters using the gradient of the objective function according to the following update rule:

$$\theta_{t+1} = \theta_t - \eta \nabla f(\theta_t) \tag{1}$$

where θ_t represents the parameters at iteration t, η is the learning rate, and $\nabla f(\theta_t)$ is the gradient of the objective function f with respect to θ_t.

In the context of text-to-image generation, our objective is to maximize the score function $\mathcal{V}(\mathcal{D}(p))$, where p is the prompt, $\mathcal{D}$ is the diffusion model, and $\mathcal{V}$ is the VLM evaluation. However, we cannot directly compute the gradient $\nabla_p \mathcal{V}(\mathcal{D}(p))$ for several reasons:

- The prompt p exists in a discrete, non-Euclidean space rather than a continuous parameter space.
- The diffusion process $\mathcal{D}$ involves complex, non-differentiable stochastic sampling procedures.
- The VLM evaluation $\mathcal{V}$ is similarly complex and not directly differentiable with respect to its inputs.

To address these challenges, our prompt optimization framework draws inspiration from numerical optimization methods to approximate the gradient and update direction. The key insight is that we can view our optimization process through the lens of zeroth-order optimization methods, particularly finite-difference approximations of gradients.

Consider how traditional finite-difference methods approximate gradients:

$$\nabla f(\theta) \approx \frac{f(\theta + \delta) - f(\theta)}{\delta} \tag{2}$$

In our context, we generalize this approach to work with a collection of prompt-score pairs $\{(p_i, s_i)\}$ that represent different points in the prompt space. Each pair provides information about the evaluation function $\mathcal{V}(\mathcal{D}(p))$ at different locations.

Instead of just using point-wise differences, we leverage the entire history $\mathcal{H}$ of prompt-score-feedback tuples to approximate a more robust update direction. The LLM, acting as a sophisticated non-parametric estimator, analyzes this historical data to infer the direction in the prompt space that is most likely to increase the objective function.

Conceptually, the prompt update process at time step t can be represented as:

$$p_{t+1} = p_{\text{best}} + \Delta p_{\text{best}} \tag{3}$$

where

$$\Delta p_{\text{best}} \approx \mathcal{U}_{\text{LLM}}(p_{\text{best}}, \mathcal{H}) \tag{4}$$

and p_{best} is the best prompt at time step t based on scores.

Here, $\mathcal{U}_{\text{LLM}}$ serves as both the gradient approximator and the update direction determiner. By analyzing the relationship between previous prompts and their resulting scores, the LLM effectively constructs a local approximation of the prompt-score landscape and generates a new prompt that is likely to achieve a higher score.

A key distinction from standard gradient descent is our use of p_{best} rather than p_t as the base for updates. In the context of discrete prompt optimization, unlike continuous optimization where small steps can be taken with controlled learning rates, a single prompt update may result in significant performance degradation due to the stochasticity of diffusion models and the discrete nature of language. Therefore, we will need a more conservative strategy of updating

the prompt. By always starting from the best-performing prompt, we establish a reliable foundation for exploration.

Moreover, even iterations that do not improve upon p_{best} contribute valuable information to $\mathcal{H}$, enriching the LLM's understanding of the prompt-performance landscape. This accumulated knowledge enhances the accuracy of subsequent update steps, effectively implementing a form of trust-region optimization that balances exploitation and exploration.

This strategy mitigates the risk of divergence in the optimization process while still allowing for thorough exploration of the prompt space. Although this approach may theoretically limit escape from local optima, the high dimensionality of the prompt space and the relatively small number of iterations ($T = 5$) make this a favorable trade-off, prioritizing stable improvement over potentially unstable exploration.

A.3. Prompt update process

In our framework, we combine the gradient approximation and update steps into a single operation performed by an LLM. This design choice is motivated by the observation that separating these steps—first determining how to improve the prompt and then implementing those improvements—introduces unnecessary complexity and potential information loss.

By unifying these steps, we enable the LLM to reason holistically about the optimization process:

$$p_{t+1} = \mathcal{U}_{\text{LLM}}(p_{\text{best}}, \mathcal{H}) \tag{5}$$

The LLM receives the best-performing prompt so far and the complete history of previous prompt-score-feedback tuples. It then analyzes patterns in this data to identify what aspects of successful prompts contributed to their high scores and what aspects of unsuccessful prompts led to lower scores. Based on this analysis, it generates a new prompt that incorporates successful elements while addressing identified shortcomings.

This approach leverages the LLM's capabilities in several ways:

- **Pattern recognition**: The LLM can identify subtle patterns in the relationship between prompt characteristics and resulting scores.
- **Contextual understanding**: The LLM can interpret feedback in the context of specific prompts and images.
- **Generative capability**: The LLM can produce entirely new prompt formulations rather than being limited to predefined update rules.

A.4. Hyperparameters and example of optimizing ConceptMix

A.4.1. Hyperparameters

Following the TextGrad approach [20], we fix the number of iterations to 5 for all experiments, which provides a good balance between computational efficiency and optimization effectiveness. For our implementation of the framework, we use GPT-4o in two critical roles: first, as the VLM ($\mathcal{V}$) that evaluates generated images and provides feedback through concept-specific scoring; and second, as the LLM ($\mathcal{U}_{\text{LLM}}$) that generates improved prompts based on the accumulated history of previous iterations. This dual application of GPT-4o creates a unified optimization framework where both evaluation and improvement processes leverage the same multimodal understanding capabilities.

A.4.2. Example of optimizing ConceptMix

Here we provide a detailed example of the our prompt optimization process applied to a prompt from the ConceptMix dataset. This example illustrates our implementation of Algorithm 1 in practice. The visualization is shown in Figure 5.

Criteria We begin with the criteria:

Criteria C

- Does the image contain cow?
- Does the image contain roses?
- Is the rose tiny in size?
- Does the image contain exactly 4 cows?
- Do the cows have a glass texture?
- Are the cows black?

Initial Prompt Evaluation We then pick the initial prompt p_0 from the ConceptMix dataset:

Initial Prompt p_0

The image features four black cows with a glass texture. There is also one tiny rose present.

After generating an image $\mathcal{I}_0 = \mathcal{D}(p_0)$ using our diffusion model, for each criterion $c_i \in C$ we score the image using GPT-4o as our VLM $\mathcal{V}$ with the following prompt:

VLM Score Prompt

Does the image contain cow? Respond 'Yes' or 'No'. [IMAGE]

For each criterion c_i at timestep t, we get a score $s_{t,i} = P(\mathcal{V}(c_i + \text{“Respond 'Yes' or 'No'.”}|\mathcal{I}) = \texttt{"Yes"})$ using

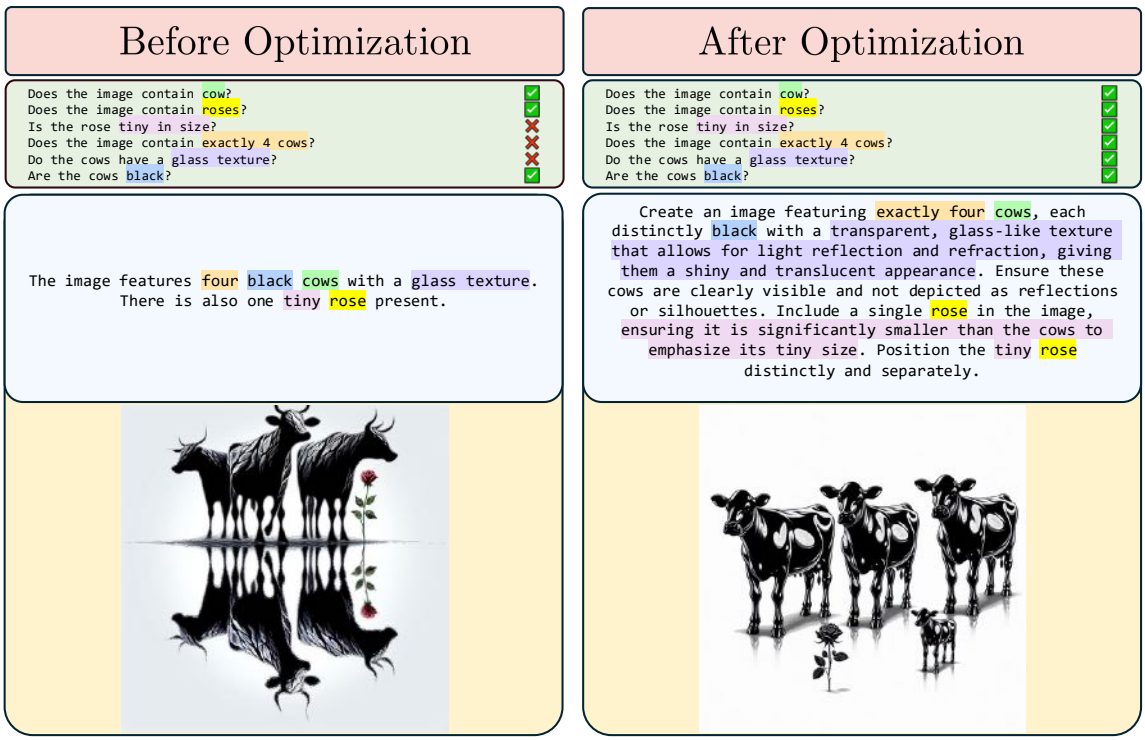


Figure 5. Example of the our prompt optimization process. Left: Before optimization, the image shows 3 black cows with reflections and a rose that is not tiny. The corresponding prompt satisfies only 3 out of 6 criteria. Right: After optimization, the image shows exactly 4 black cows with glass texture and a tiny rose, satisfying all 6 criteria.

the probability distribution of the first new token, and feedback $f_{t,i}$. The overall score s_t for the current timestep is calculated as the product of all individual criterion scores:

$$s_t = \prod_{i=1}^{k+1} s_{t,i} \tag{6}$$

where k is the number of criteria -1 (in this case, $k = 5$). Similarly, for each criterion $c_i \in C$, we get the feedback with the following prompt:

VLM Feedback Prompt

Does the image contain cow? If the answer is "No", please explain in one sentence the specific issue that prevents the image from satisfying the question; otherwise, just output that the image satisfies the question. [IMAGE]

The overall feedback f_t is the string produced by concatenating each criterion c_i and its corresponding feedback $f_{t,i}$ for all criteria. Here's an example of combined feedback for the initial prompt ($t = 0$):

VLM Feedback Results

Does the image contain cow? The image satisfies the question.
Does the image contain roses? The image satisfies the question.
Is the rose tiny in size? The rose is nearly as tall as the cows, which is not tiny in size.
Does the image contain exactly 4 cows? The image does not contain exactly 4 cows because it only shows 3 cows and their reflections.
Do the cows have a glass texture? No, the cows do not have a glass texture as they appear as solid, opaque silhouettes without transparency or reflective qualities.
Are the cows black? The image satisfies the question.

We then add the tuple (p_0, s_0, f_0) to the history table $\mathcal{H}$ and prompt the LLM to generate an improved prompt:

LLM Optimization Prompt

I'm trying to generate an image that matches specific requirements. Please create a concise description that will help the model generate an image that satisfies all the requirements and get a high product of scores. Here's the context:
1. Requirements: The image must satisfy these criteria (all should receive 'Yes' answers): {criteria}
2. History: Previous attempts sorted by performance (best to worst): {formatted history table}
Based on the requirements and previous attempts, please provide a new, improved description for the image generation model. The description should:
- Be specific to guide the image generation
- Address all the required elements from the questions
- Learn from previous attempts, especially what worked in higher-scoring versions

Now please give a concise description to help the model generate an image that meets all the requirements and gets the highest product of scores.

Iterative Optimization This prompt optimization process continues for a total of $T = 5$ iterations, with each new prompt p_t being evaluated to produce a score s_t and feedback f_t. The history table $\mathcal{H}$ is updated at each iteration, and the LLM uses this accumulated information to generate increasingly refined prompts.

Final Result After the iterative optimization process, we obtain our final optimized prompt p^*:

Final Optimized Prompt p^*

Create an image featuring exactly four cows, each distinctly black with a transparent, glass-like texture that allows for light reflection and refraction, giving them a shiny and translucent appearance. Ensure these cows are clearly visible and not depicted as reflections or silhouettes. Include a single rose in the image, ensuring it is significantly smaller than the cows to emphasize its tiny size. Position the tiny rose distinctly and separately from the cows.

This example demonstrates how our prompt optimization framework systematically improves the initial prompt through targeted feedback and iterative optimization, resulting in increasingly accurate representations of the desired concepts. At each stage, the algorithm leverages both visual evaluation and linguistic refinement capabilities of the LLM to navigate the complex prompt space effectively.

B. Model-wise performance charts

To better understand how prompt optimization reveals the true capabilities of text-to-image models, we present comprehensive performance comparisons across all tested diffusion models. Figure 6 visualizes both the *mean* score and *best-of-5* score metrics for each model before and after prompt optimization across varying complexity levels ($k = 1$ to $k = 7$).

The charts clearly illustrate several key findings:

- **Substantial Performance Gaps:** All models demonstrate significant improvements with optimized prompts, particularly at moderate complexity levels (k=3 to k=5). This confirms our hypothesis that conventional benchmarks using rigid prompts underestimate models' true visual synthesis capabilities.
- **Persistent Complexity Ceiling:** Despite optimization, all models show declining performance as complexity increases, revealing fundamental architectural limitations that better prompting alone cannot overcome.
- **Model-Specific Improvement Patterns:** While DALL·E 3 achieves the highest absolute performance, SD 3.5 shows the largest relative gains from optimization, suggesting greater headroom for prompt engineering in this architecture.
- **Best-of-5 vs. Average Metrics:** The gap between optimized and original prompts is consistently more pronounced in the best-of-5 metric, indicating that optimized prompts not only improve average performance but also enable higher ceiling capabilities when multiple generations are considered.

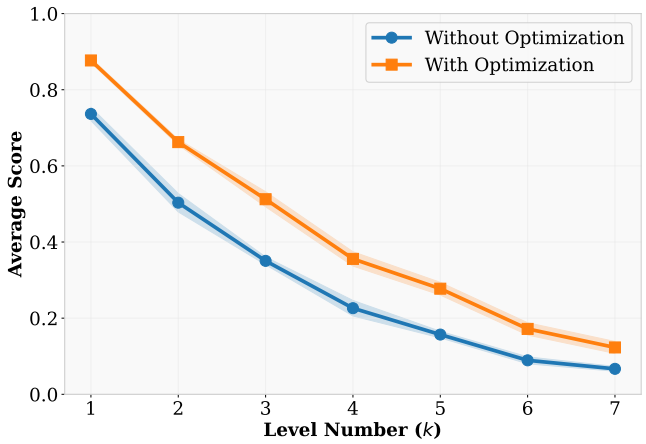
These visualizations highlight the importance of more flexible evaluation frameworks in benchmarking text-to-image models. The substantial performance differences between standard and optimized prompts across all models demonstrate that conventional rigid benchmarking approaches may significantly underrepresent actual model capabilities, particularly for certain architectures like SD 3.5. On the other hand, the persistent decline of improvement at higher complexity levels still reveals fundamental limitations that require architectural innovations.

C. Category-wise capability analysis

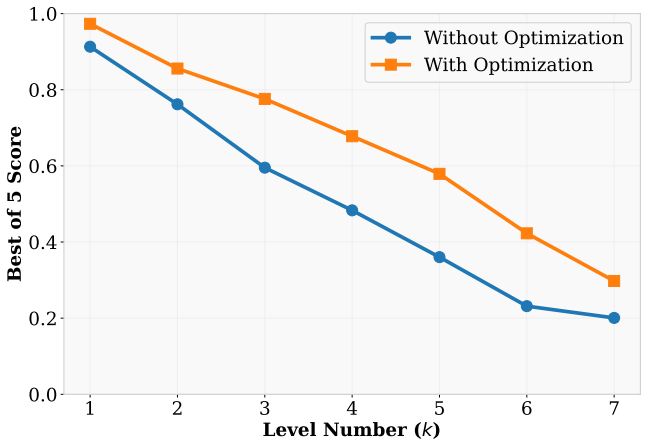
C.1. Heatmap analysis across models

Figure 7, Figure 8, and Figure 9 present heatmap visualizations of improvements across all visual concept categories for DALL·E 3, Stable Diffusion 3.5, and Playground v2.5, respectively. These visualizations reveal important patterns in how prompt optimization affects each model's performance across different visual categories.

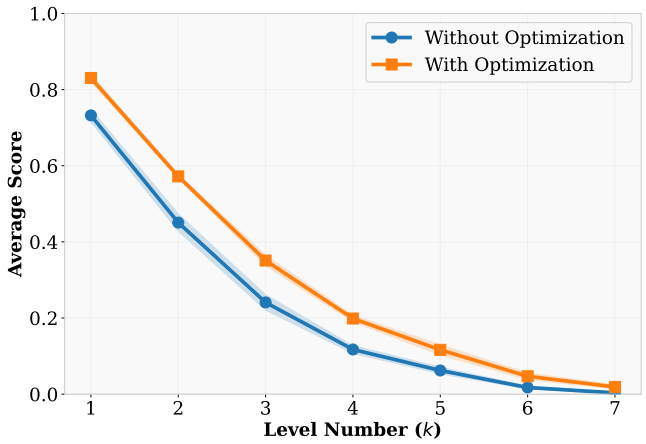
Similar to DALL·E 3 and Stable Diffusion 3.5, Playground v2.5 shows significant overall improvements



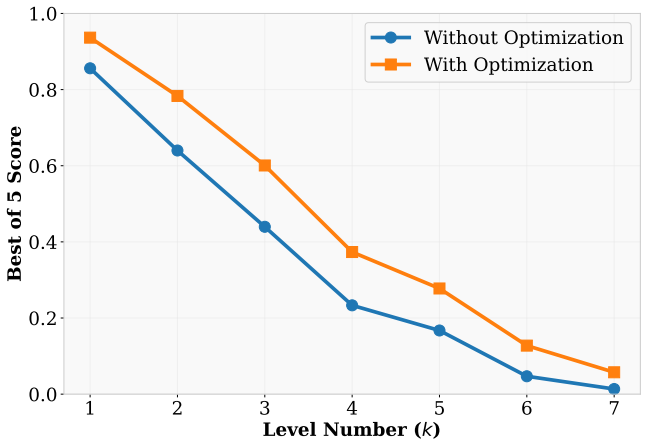
(a) SD 3.5: Average Score



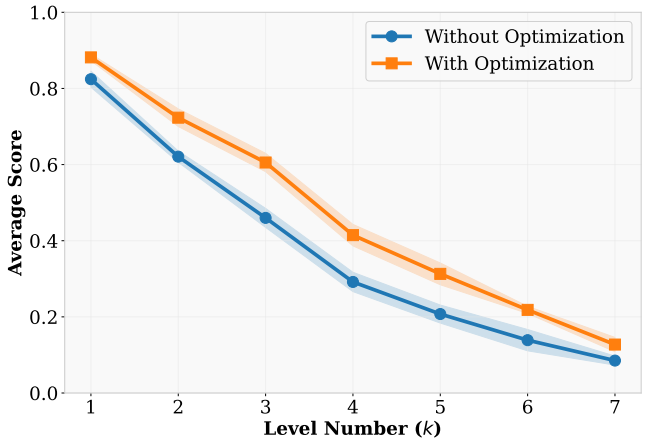
(b) SD 3.5: Best-of-5 Score



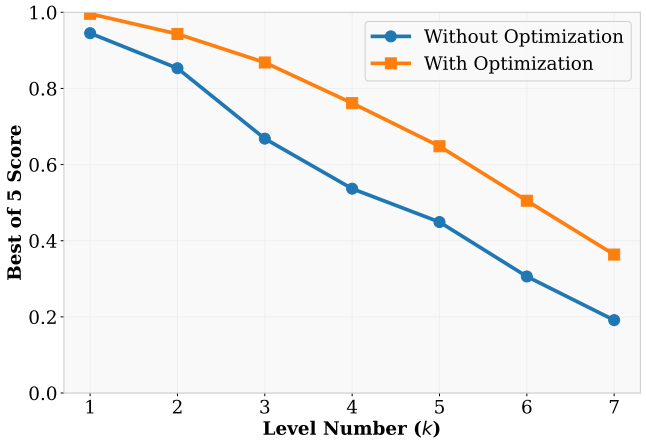
(c) Playground v2.5: Average Score



(d) Playground v2.5: Best-of-5 Score



(e) DALL·E 3: Average Score



(f) DALL·E 3: Best-of-5 Score

Figure 6. Performance comparison across models and complexity levels, showing original vs. optimized prompt performance. Left column: Average Score; Right column: Best-of-5 Score. Top row: Stable Diffusion 3.5; Middle row: Playground v2.5; Bottom row: DALL·E 3.

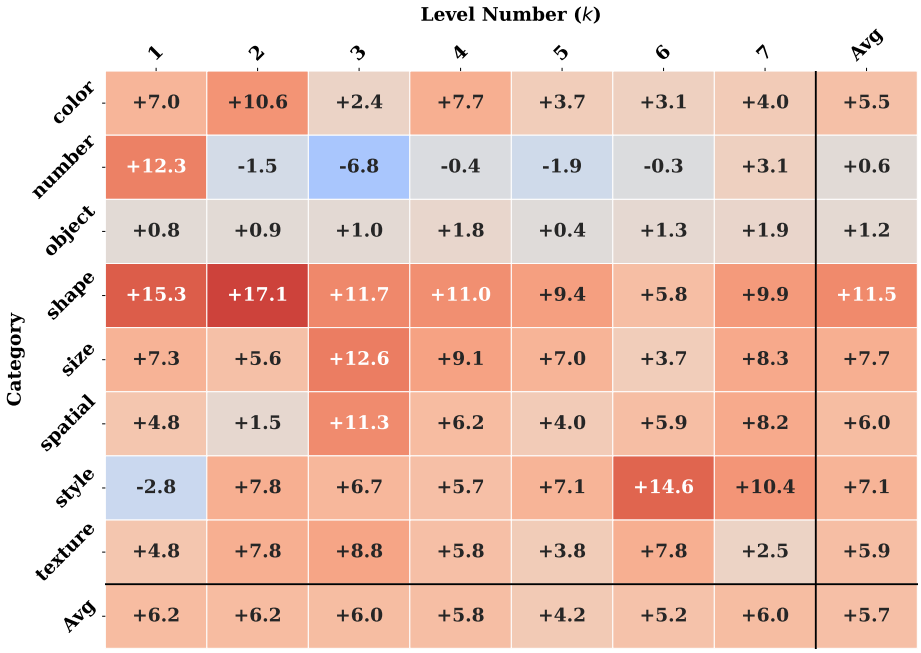


Figure 7. Heatmap showing improvement magnitude (optimized - original) for DALL·E 3 across visual concept categories and complexity levels. Darker red colors indicate larger capability gaps, revealing where prompt optimization most significantly improves performance, while blue indicates negative improvement.

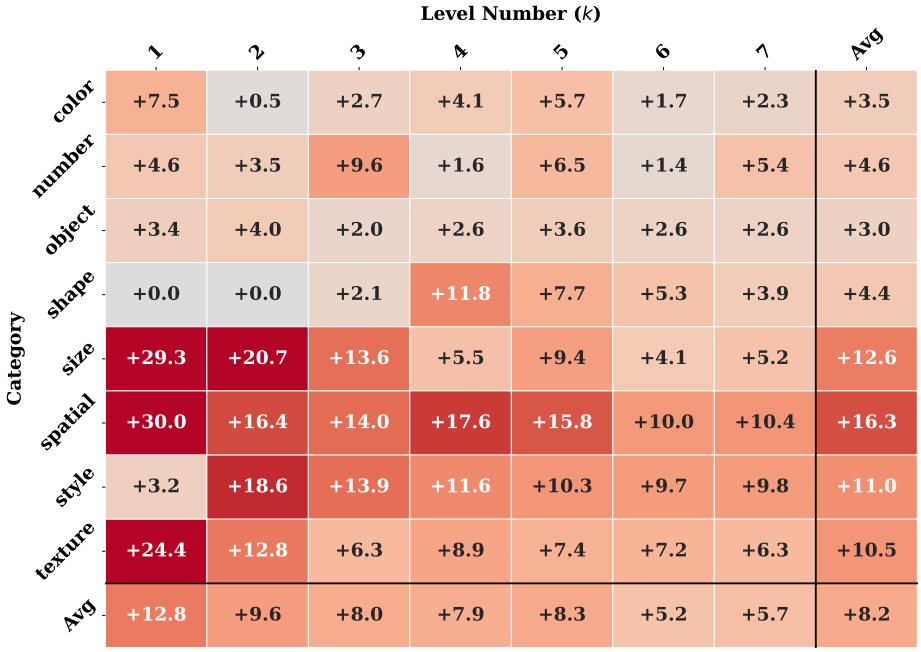


Figure 8. Heatmap showing improvement magnitude (optimized - original) for Stable Diffusion 3.5 across visual concept categories and complexity levels. Darker red colors indicate larger capability gaps, revealing where prompt optimization most significantly improves performance.

through prompt optimization (+7.1% average), but with unique category-specific strengths. Texture show the strongest consistent gains (+11.9% average), while object category shows the least improvement (+1.9% average), in-

Figure 9. Heatmap showing improvement magnitude (optimized - original) for Playground v2.5 across visual concept categories and complexity levels. Darker red colors indicate larger capability gaps, revealing where prompt optimization most significantly improves performance, while blue indicates negative improvement.

dicating that original prompts were already relatively effective for simple object generation.

C.2. Radar graph analysis by complexity level

To better visualize how each model responds to prompt optimization across different complexity levels, Figures 10 through 16 present radar graphs comparing the three models at each complexity level k from 1 to 7.

D. Cross-model prompt transferability

To investigate whether the prompts optimized for one model can effectively generalize to another, we conduct transferability experiments. For each pair of models (M_{src}, M_{tgt}), we:

1. Run our prompt optimization framework using M_{src} as the backbone diffusion model to obtain optimized prompts p^*_{src}
2. Evaluate these transferred prompts p^*_{src} on target model M_{tgt}
3. Compare the performance against both the original prompt p_0 and M_{tgt}'s self-optimized prompts p^*_{tgt}

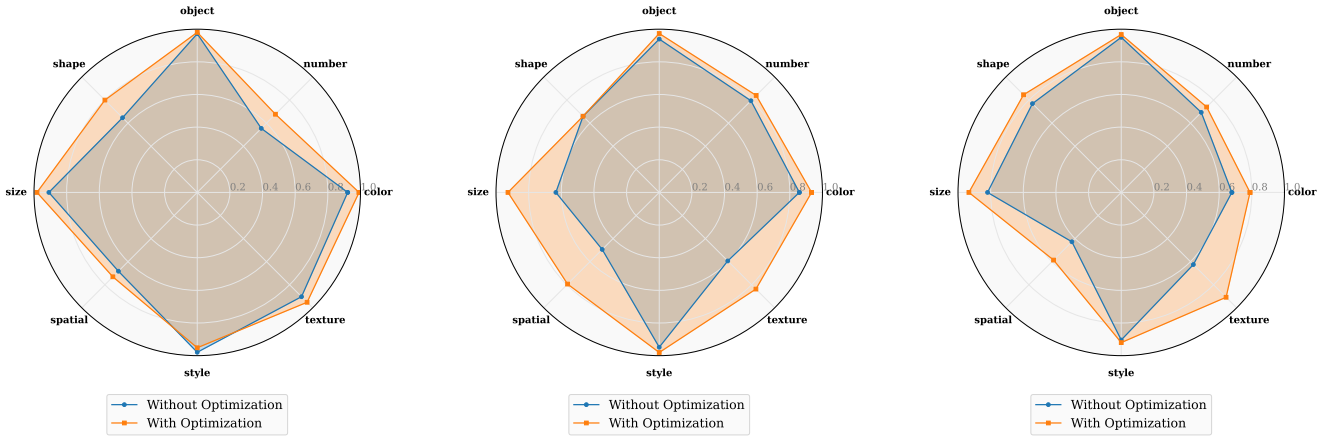
Our results demonstrate transferability between models, with transferred prompts achieving better performance than original prompts in most cases. This finding also has important practical implications. It suggests that practitioners can leverage a cost-effective optimization workflow: (1) optimize prompts using more accessible or computationally efficient models, and (2) apply these enhanced prompts to more powerful models for final generation. This approach significantly reduces the computational overhead and API costs associated with prompt optimization.

D.1. Model-specific transfer patterns

The effectiveness of prompt transfer varies depending on the source and target models. Some key observations:

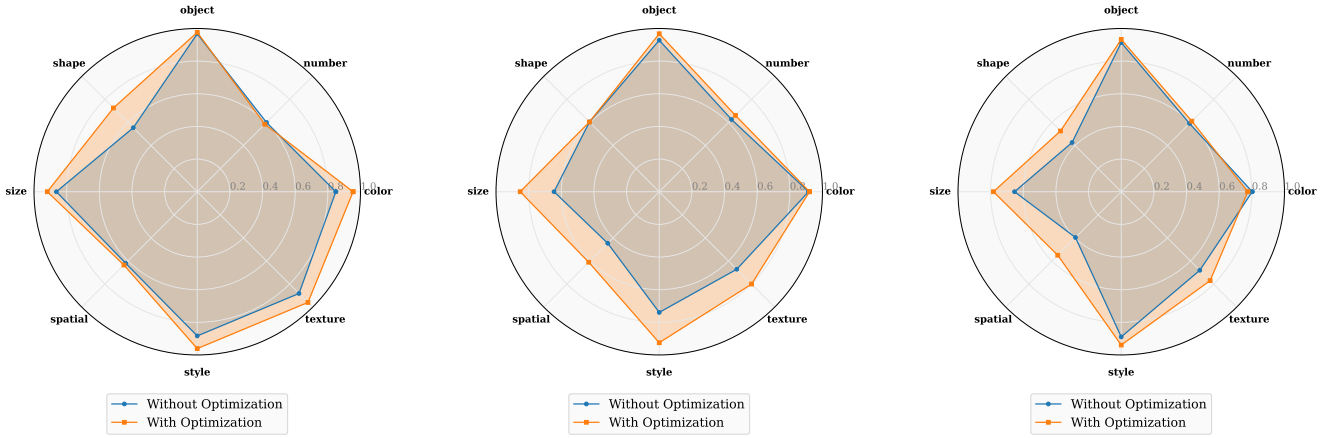
- **SD 3.5 → DALL·E 3:** Prompts optimized for Stable Diffusion 3.5 transfer effectively to DALL·E 3, achieving approximately 85% of the performance gain of DALL·E 3's self-optimized prompts.
- **Playground v2.5 → DALL·E 3:** Similarly, Playground-optimized prompts transfer well to DALL·E 3, suggesting these models share similar prompt understanding mechanisms.
- **Asymmetric Transfer:** Interestingly, transfer effectiveness is not always symmetric. Prompts optimized for DALL·E 3 transfer less effectively to other models, suggesting it may have developed more specialized prompt understanding capabilities.

These transfer patterns provide insights into the shared conceptual understanding across different model architectures. The transferability suggests that different diffusion models may learn similar representations of visual concepts and share underlying prompt preferences, enabling the remarkable transferability observed in our experiments.



(a) DALL·E 3 (b) Stable Diffusion 3.5 (c) Playground v2.5

Figure 10. Radar graphs showing category improvements for complexity level $k = 1$ across all three models.



(a) DALL·E 3 (b) Stable Diffusion 3.5 (c) Playground v2.5

Figure 11. Radar graphs showing category improvements for complexity level $k = 2$ across all three models.

E. Ablation study on optimization iteration count

A key hyperparameter in our Prompt Optimization framework is the number of optimization iterations T. While more iterations intuitively seem beneficial, they also increase computational cost and potentially risk overfitting to the evaluation metric. To investigate the impact of this parameter, we conducted an ablation study on DALL·E 3 with varying numbers of iterations: $T \in \{0, 1, 2, 3, 4, 5, 10, 15\}$, where $T = 0$ means we use the original prompt. For this experiment, we fix $k = 4$.

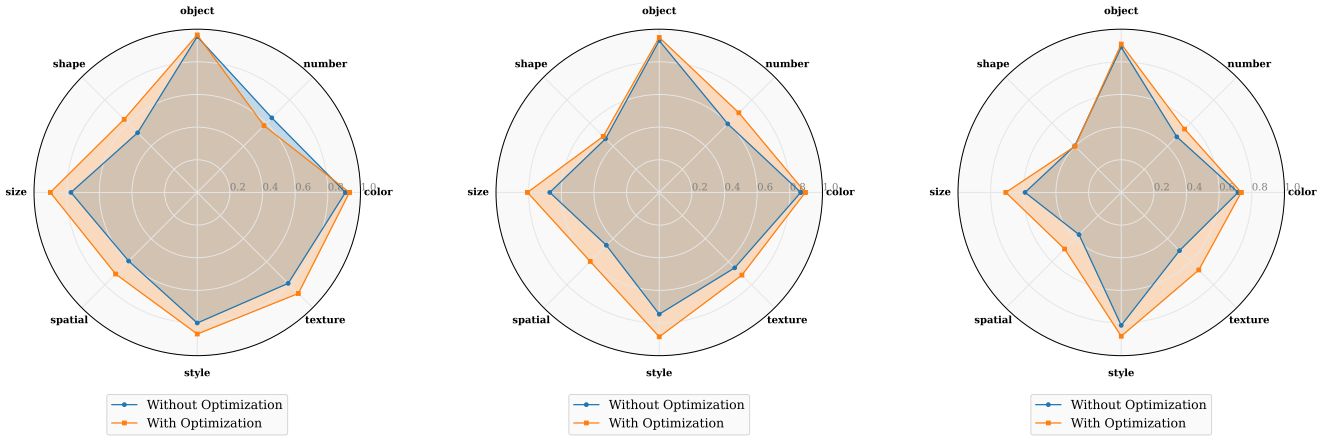
From Figure 23, we can tell that higher iteration number doesn't necessarily give us better prompt. This counterintuitive finding can be attributed to several factors:

1. **Inherent Stochasticity:** Diffusion models like DALL·E 3 have inherent randomness in their generation process. A prompt that produces high-quality images during optimization may not consistently yield the same quality during test time, even with the same sampling parameters.
2. **Overfitting to Specific Instances:** Later iterations may overfit to the specific random seed or generation parameters used during optimization, reducing generalizability to new generation instances.

Based on these results, we think $T = 5$ is indeed a choice that balances performance gains with computational efficiency for most applications. This finding is particularly valuable for deployment scenarios where optimization time is a concern.

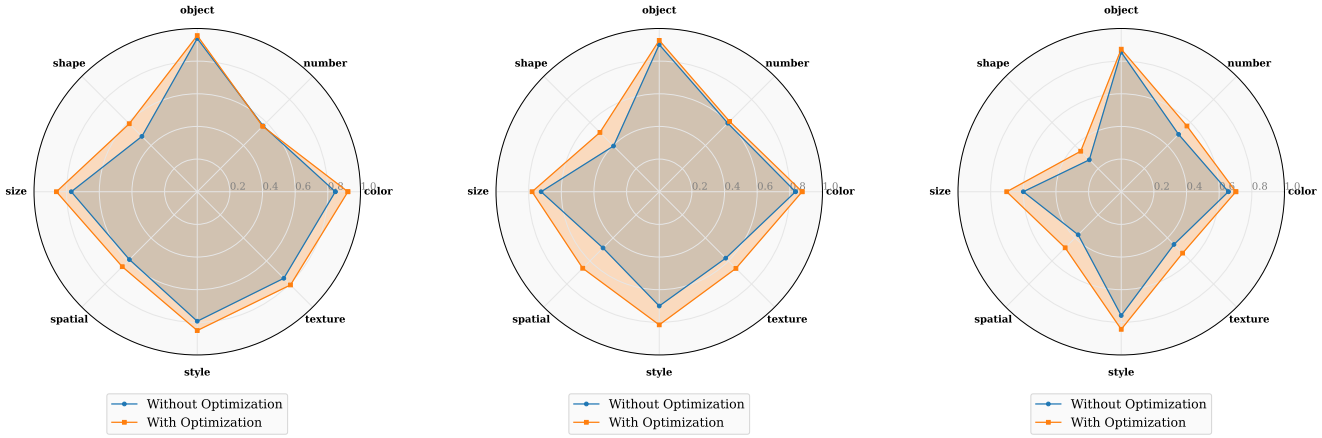
F. Ablation study on computational budget

An important practical consideration for prompt optimization is the computational budget, particularly the number of image generations required. In our main experiments, we



(a) DALL·E 3 (b) Stable Diffusion 3.5 (c) Playground v2.5

Figure 12. Radar graphs showing category improvements for complexity level $k = 3$ across all three models.



(a) DALL·E 3 (b) Stable Diffusion 3.5 (c) Playground v2.5

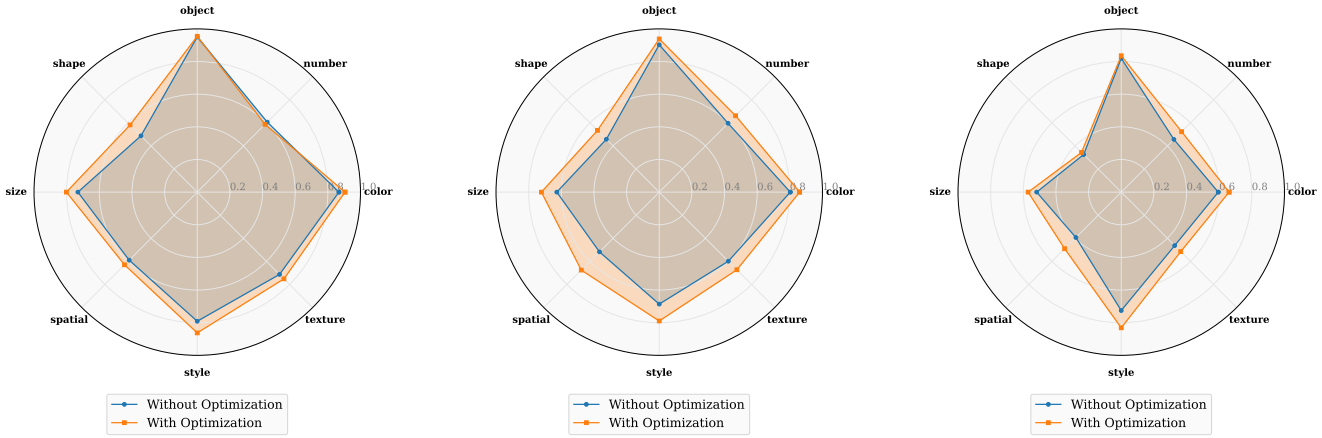
Figure 13. Radar graphs showing category improvements for complexity level $k = 4$ across all three models.

compared the performance of optimized prompts (which required 1 initial image + 5 optimization iterations + 5 final test generations) against baseline approaches that generated 5 images with the original prompt. This comparison, while demonstrating the effectiveness of our framework, does not account for the additional computational cost of the optimization process itself. To address this concern, we conduct a controlled computational budget experiment where both the baseline and our method are limited to exactly 5 image generations in total. For the baseline, we maintain the same approach of generating 5 images with the original prompt and selecting the best one. For our prompt optimization approach, we modify the procedure as follows:

1. We generate an initial image with the original prompt p_0
2. We run 4 iterations of our optimization process, generating one image with each improved prompt p_1, p_2, p_3, p_4
3. We select the best image from these 5 generations $(p_0, p_1, p_2, p_3, p_4)$ based on our evaluation metric

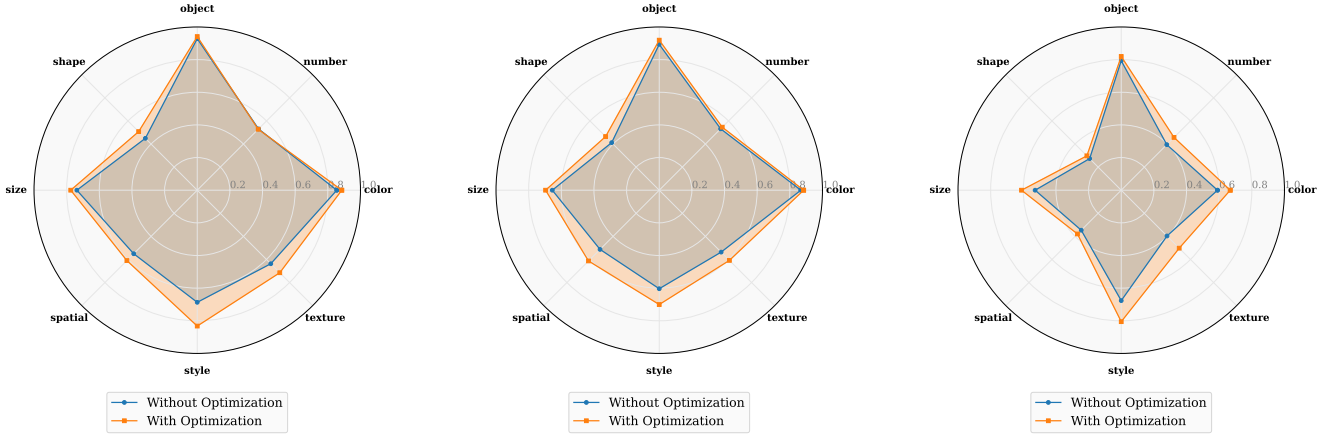
This approach maintains strict budget parity between the methods, with both generating exactly 5 images. The key difference is that our approach generates images from a sequence of progressively optimized prompts, while the baseline generates multiple images from the same initial prompt.

As shown in Figures 24, 25, and 26, even with the same computational budget, our prompt optimization approach consistently outperforms the baseline across all complexity levels for all three models. This highlights another practical application of prompt optimization framework beyond benchmarking model capabilities: improving resource-constrained generation. When users have a fixed compute budget (e.g., limited to 5 DALL·E 3 API calls) and specific image criteria to meet, our framework can efficiently allocate these resources by progressively updating prompts based on previous generations, rather than repeatedly sampling from the same initial prompt.



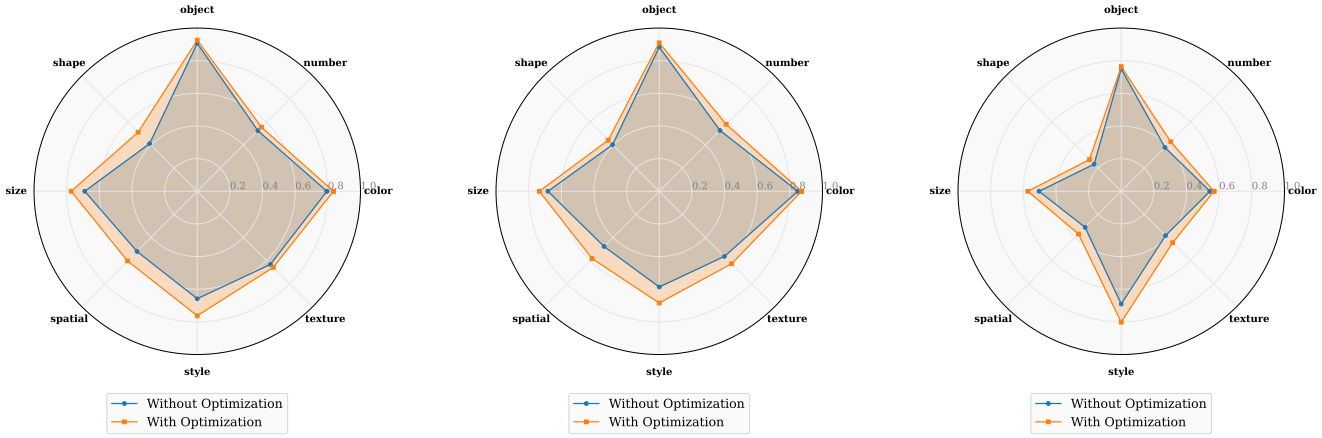
(a) DALL·E 3 (b) Stable Diffusion 3.5 (c) Playground v2.5

Figure 14. Radar graphs showing category improvements for complexity level $k = 5$ across all three models.



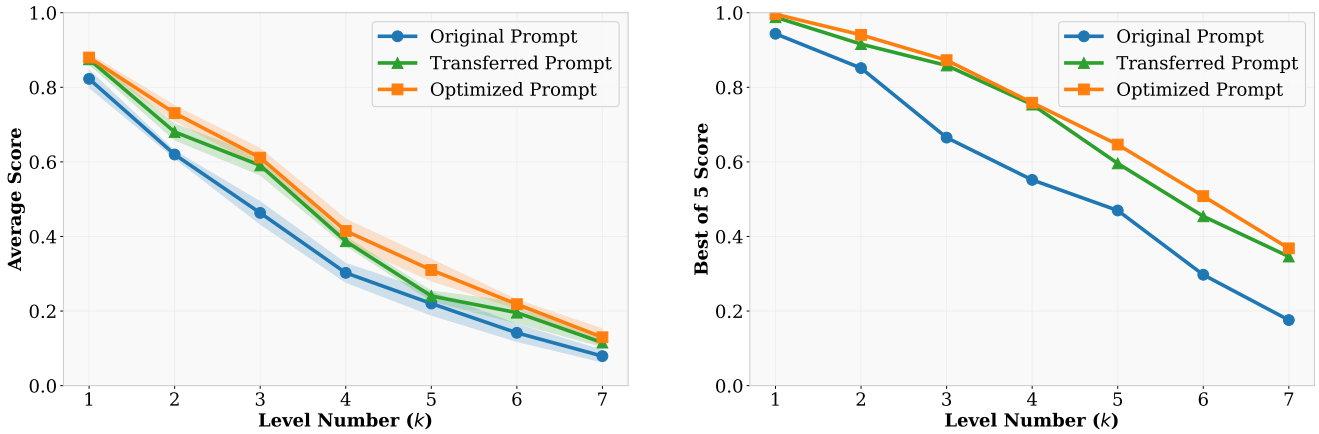
(a) DALL·E 3 (b) Stable Diffusion 3.5 (c) Playground v2.5

Figure 15. Radar graphs showing category improvements for complexity level $k = 6$ across all three models.



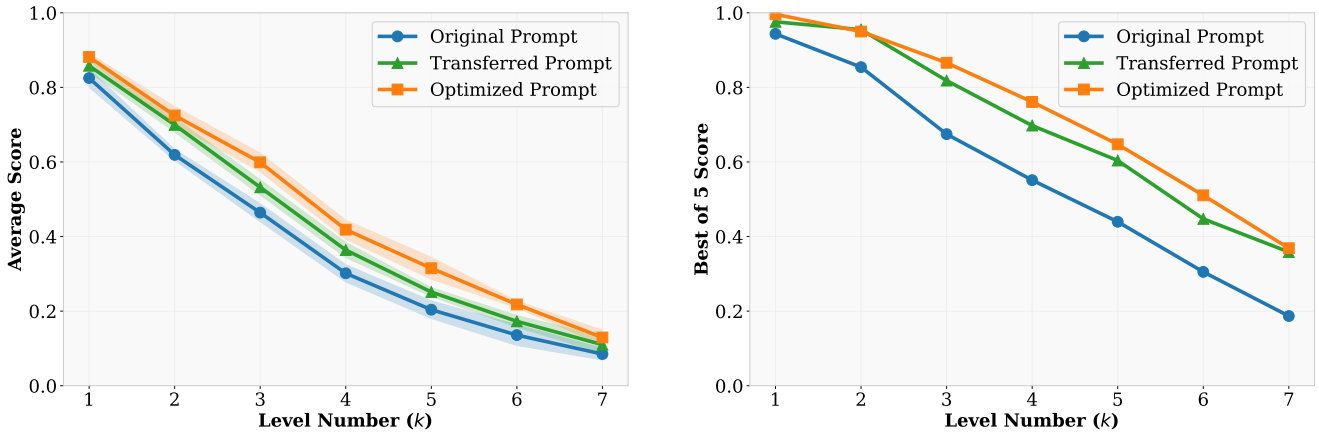
(a) DALL·E 3 (b) Stable Diffusion 3.5 (c) Playground v2.5

Figure 16. Radar graphs showing category improvements for complexity level $k = 7$ across all three models.



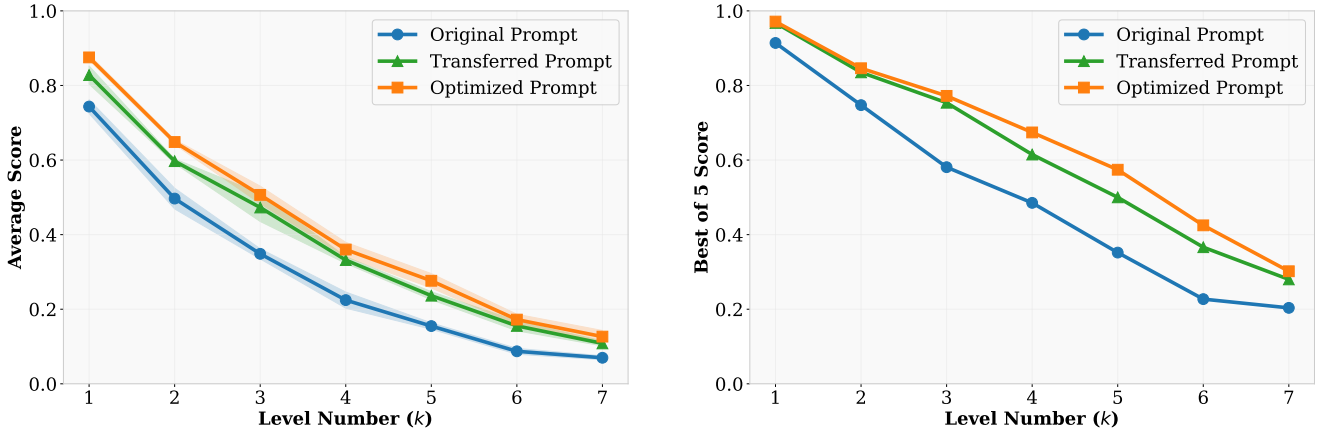
(a) Average Score (b) Best-of-5 Score

Figure 17. DALL·E 3 performance with SD 3.5 optimized prompts compared to original prompts and self-optimized prompts.



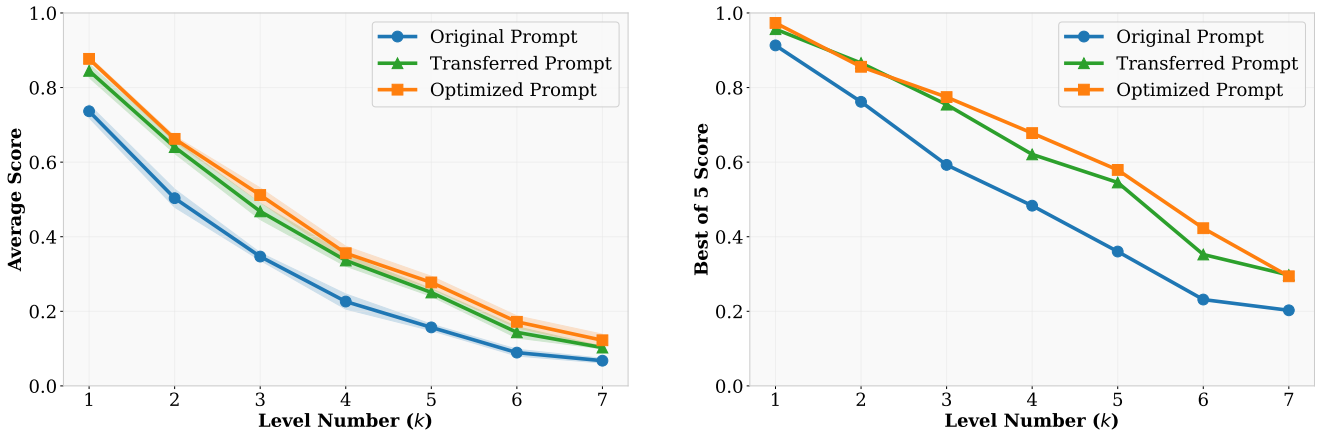
(a) Average Score (b) Best-of-5 Score

Figure 18. DALL·E 3 performance with Playground v2.5 optimized prompts compared to original prompts and self-optimized prompts.



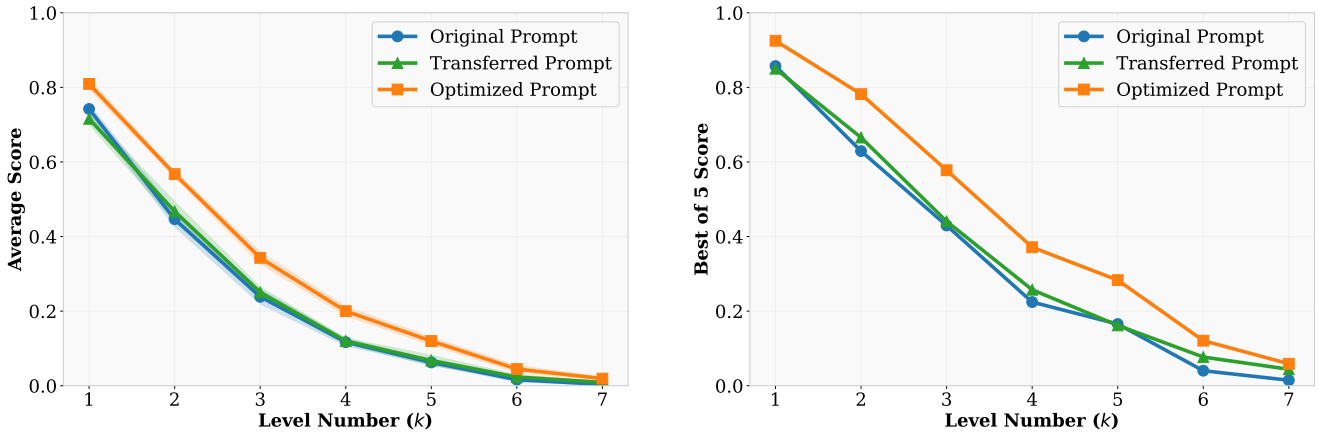
(a) Average Score (b) Best-of-5 Score

Figure 19. Stable Diffusion 3.5 performance with DALL·E 3 optimized prompts compared to original prompts and self-optimized prompts.



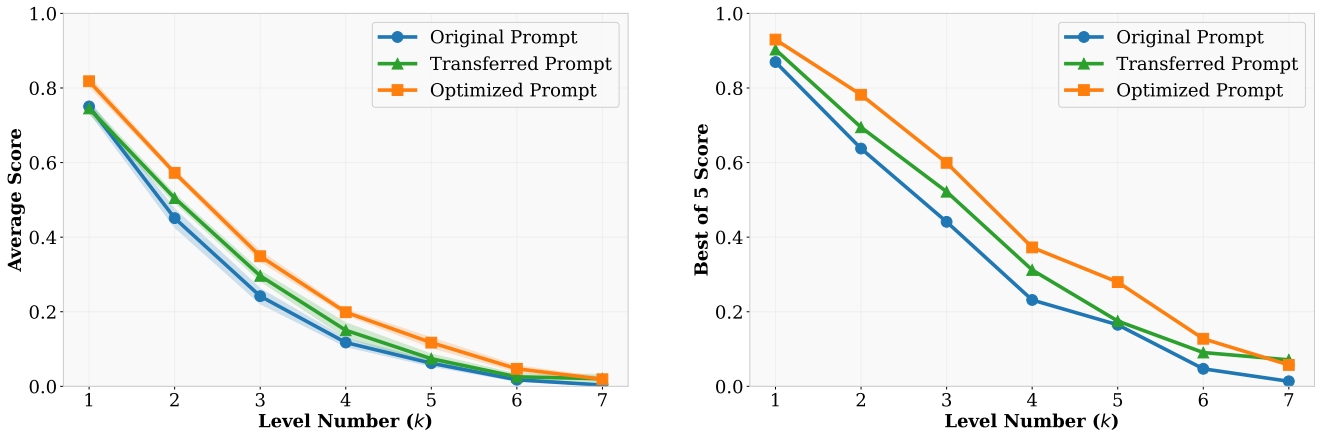
(a) Average Score (b) Best-of-5 Score

Figure 20. Stable Diffusion 3.5 performance with Playground v2.5 optimized prompts compared to original prompts and self-optimized prompts.



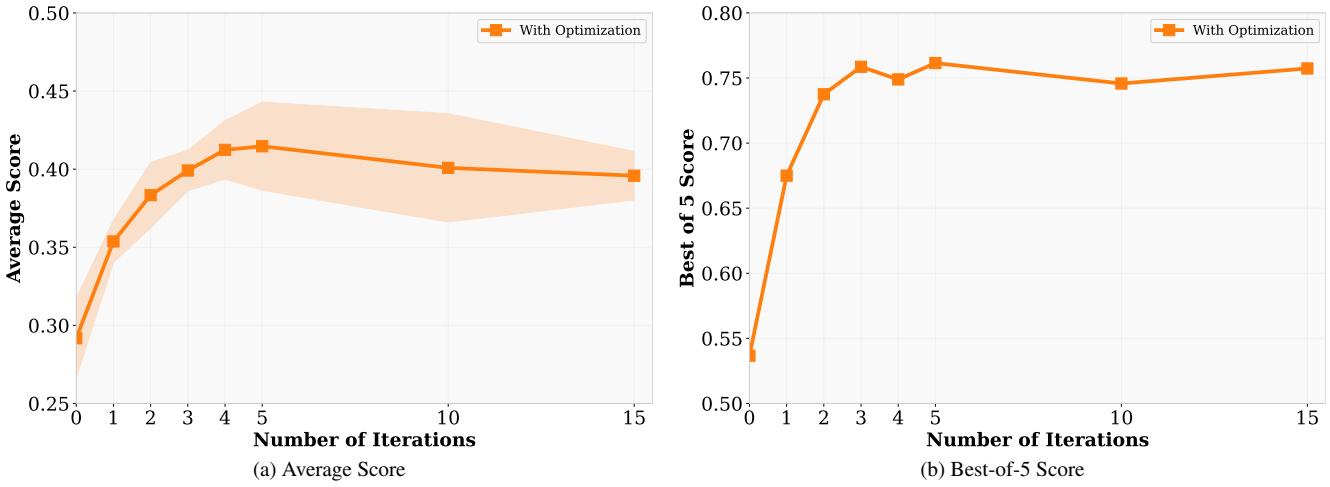
(a) Average Score (b) Best-of-5 Score

Figure 21. Playground v2.5 performance with DALL·E 3 optimized prompts compared to original prompts and self-optimized prompts.



(a) Average Score (b) Best-of-5 Score

Figure 22. Playground v2.5 performance with Stable Diffusion 3.5 optimized prompts compared to original prompts and self-optimized prompts.



(a) Average Score (b) Best-of-5 Score

Figure 23. Performance comparison of different iteration numbers (T) in our prompt optimization framework on DALL·E 3 with fixed complexity $k = 4$. The x-axis represents the number of iterations, while the y-axis shows scores.

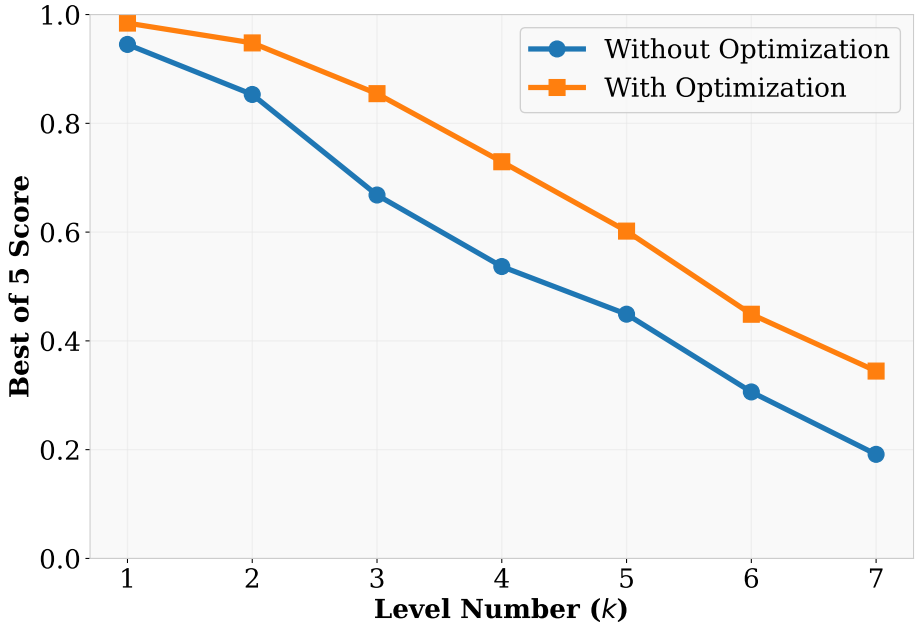


Figure 24. Comparison of DALL·E 3 performance under equal computational budget (5 images). The graph shows Best-of-5 Score for the baseline approach (5 images with the original prompt) versus our approach (5 images with progressively optimized prompts) across different complexity levels ($k = 1$ to $k = 7$).

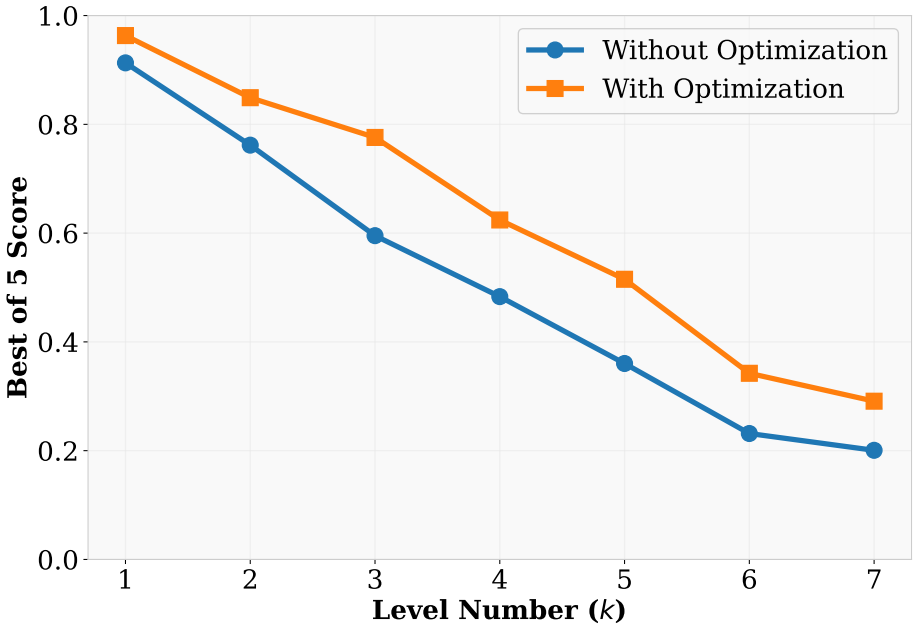


Figure 25. Comparison of Stable Diffusion 3.5 performance under equal computational budget (5 images). The graph shows Best-of-5 Score for the baseline approach (5 images with the original prompt) versus our approach (5 images with progressively optimized prompts) across different complexity levels ($k = 1$ to $k = 7$).

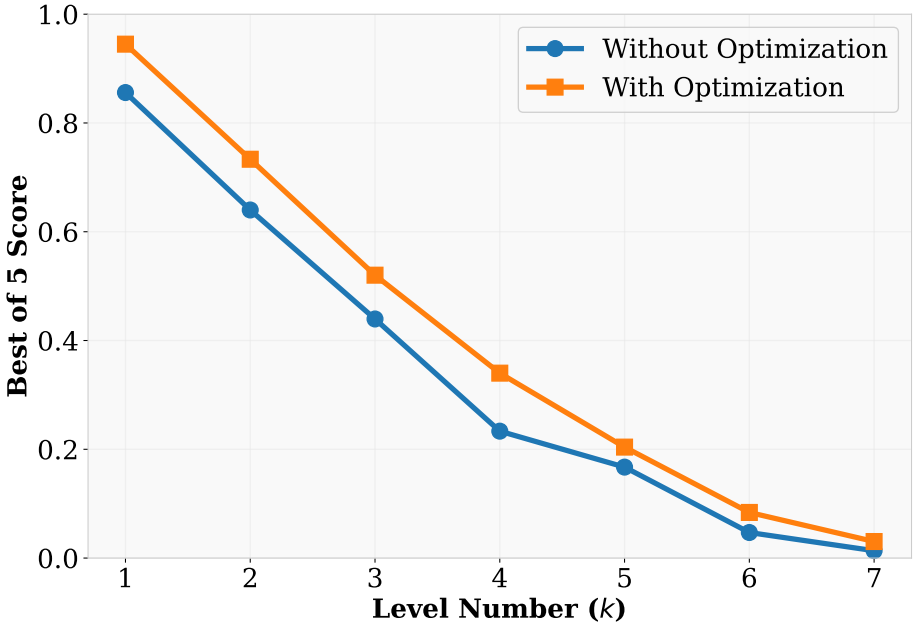


Figure 26. Comparison of Playground v2.5 performance under equal computational budget (5 images). The graph shows Best-of-5 Score for the baseline approach (5 images with the original prompt) versus our approach (5 images with progressively optimized prompts) across different complexity levels ($k = 1$ to $k = 7$).