ATHENA: Adaptive Test-Time Steering for Improving Count Fidelity in Diffusion Models

Mohammad Shahab Sepehri* Asal Mehradfar* Berk Tinaz
Salman Avestimehr Mahdi Soltanolkotabi
Department of Electrical and Computer Engineering
University of Southern California, Los Angeles, CA, USA
{sepehri, mehradfa, tinaz, avestime, soltanol}@usc.edu

Abstract

Text-to-image diffusion models achieve high visual fidelity but surprisingly exhibit systematic failures in numerical control when prompts specify explicit object counts. To address this limitation, we introduce ATHENA, a model-agnostic, test-time adaptive steering framework that improves object count fidelity without modifying model architectures or requiring retraining. ATHENA leverages intermediate representations during sampling to estimate object counts and applies count-aware noise corrections early in the denoising process, steering the generation trajectory before structural errors become difficult to revise. We present three progressively more advanced variants of ATHENA that trade additional computation for improved numerical accuracy, ranging from static prompt-based steering to dynamically adjusted count-aware control. Experiments on established benchmarks and a new visually and semantically complex dataset show that ATHENA consistently improves count fidelity, particularly at higher target counts, while maintaining favorable accuracy–runtime trade-offs across multiple diffusion backbones. Our code and data are publicly available at https://github.com/MShahabSepehri/ATHENA.

1. Introduction

Text-to-image (T2I) diffusion models have gained huge popularity and become the dominant paradigm for high-quality image generation, driven by their ability to synthesize images from natural language prompts with strong visual realism, semantic alignment, and compositional expressiveness [13]. Despite their advances, diffusion models continue to exhibit systematic failures in numerical control when prompts specify explicit object counts [3, 12] (Figure 1).

*Equal contribution.

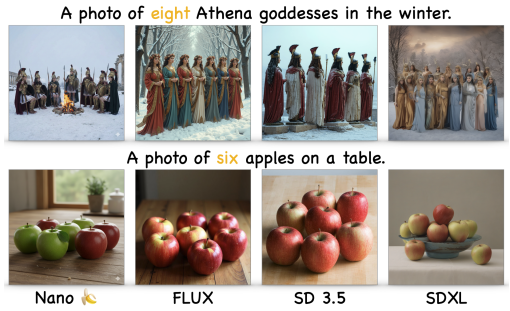


Figure 1. Count fidelity failures in T2I generative models.

Prior work attempts to improve count fidelity through architectural modifications or auxiliary layout models that detect and correct counting discrepancies [1, 7]. Other approaches enforce target counts via gradient-based guidance or detector feedback during diffusion sampling [4, 16, 17], while training-free alternatives modify attention maps or decompose generation into sequential sub-tasks [8, 9]. However, these approaches often require additional training, architectural changes, or computationally expensive optimization, limiting efficient test-time control of object counts.

In this work, we introduce ATHENA (Adaptive Trajectory Harmonization via Early Numerical Assessment), a test-time *adaptive* steering framework for improving object count fidelity in T2I diffusion models. ATHENA estimates object count from intermediate representations and uses this signal to adaptively modify the sampling trajectory via count-aware noise correction. ATHENA is model-agnostic, requires no architectural modifications or retraining, and can be applied across diverse diffusion backbones.

Contributions. Our main contributions are as follows:

- We propose ATHENA, a model-agnostic steering framework for improving object-count fidelity in T2I diffusion models that requires no architectural changes or retrain-

ing. ATHENA can be applied to arbitrary T2I models via simple forward-pass interventions and yields immediate improvements in numerical fidelity.

- We introduce the ATHENA dataset that complements existing counting datasets by targeting challenging object categories (e.g., accordion) and compositional prompt structures with relational constraints (e.g., next to a river) and object-level distractions (e.g., with a person).
- Through extensive experiments on three datasets, we demonstrate that ATHENA improves the numerical accuracy of the base diffusion models by up to 22% and outperforms baselines, while reducing memory usage by approximately $4\times$ and achieving up to $2.5\times$ faster image generation relative to the baselines.

2. Method

2.1. Problem Setup and Design Goals

We study test-time control of object count fidelity in diffusion-based T2I models. Our objective is to improve the accuracy with which the final generated image satisfies this target count. We assume access to intermediate latent or decoded representations during sampling, and all interventions are performed at test time without retraining or modifying the generator parameters.

2.2. Count Estimation Across Diffusion Steps

Diffusion-based T2I generation produces a sequence of intermediate states $\{z_t\}_{t=T}^{0}$, where z_T is pure noise and iterative denoising progressively transforms it into the final clean representation z_0, which decodes to the generated image.

At early steps, object structure is insufficiently formed for reliable counting. At very late steps, object instances are well defined and counting is accurate, but this requires completing the full denoising trajectory, increasing inference time. This trade-off creates an intermediate regime where object count can be estimated reliably without incurring the cost of full late-stage sampling (Sec. A).

Motivated by this observation, ATHENA performs count estimation at a fixed intermediate step t_{est}. Instead of decoding the intermediate state $z_{t_{\text{est}}}$ directly, we use the estimation of the final representation at that step, $\hat{z}_0^{(t_{\text{est}})}$, obtained from the standard denoising prediction. Decoding this estimate provides a reliable approximation of the final image while avoiding completion of the full diffusion process.

2.3. ATHENA: Test-Time Steering Framework

ATHENA is a test-time steering framework that improves object count fidelity in diffusion-based T2I generation, without modifying model architecture or requiring retraining. The framework estimates object count at an intermediate diffusion step and uses this signal to steer the sampling trajectory via prompt-based control, enabling corrective intervention before structural errors form. All components operate at inference time and are compatible with both deterministic and stochastic diffusion samplers.

2.3.1. Prompt-Based Steering Mechanism

ATHENA uses a lightweight, training-free steering mechanism that modifies the denoising trajectory via prompt conditioning. At diffusion step t, the denoiser $\epsilon_\theta(.)$ is evaluated twice on the current latent z_t: once with the original prompt p, producing $\epsilon_t \triangleq \epsilon_\theta(t, z_t, p)$, and once with a *control prompt* $\hat{p}$, producing $\hat{\epsilon}_t \triangleq \epsilon_\theta(t, z_t, \hat{p})$ (see Figure 2). These two predictions are combined to form a steered noise estimate

$$\tilde{\epsilon}_t = \epsilon_t + \gamma\,(\epsilon_t - \hat{\epsilon}_t), \tag{1}$$

where $\gamma \geq 0$ controls the steering strength. The sampler update is then applied using $\tilde{\epsilon}_t$ in place of ϵ_t to obtain z_{t-1}. We normalize $\tilde{\epsilon}_t$ to match the norm of ϵ_t, ensuring the steered latent remains within the denoiser's expected scale and preserves stable, in-distribution sampling.

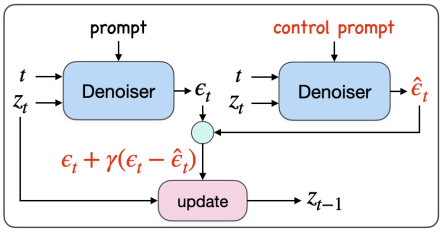


Figure 2. ATHENA steering block.

The steering operation in (1) is a controlled perturbation of the denoiser's prediction that biases the sampling trajectory away from behaviors induced by the control prompt. The difference term $(\epsilon_t - \hat{\epsilon}_t)$ isolates the directional change in predicted noise when replacing the original prompt p with $\hat{p}$, capturing the local influence of the control condition at step t. By adding a scaled version of this difference to ϵ_t, ATHENA reinforces directions that move the generation toward the target count while preserving semantic structure.

2.3.2. Control Prompt Construction

ATHENA induces steering by evaluating the denoiser under two textual conditionings: the original prompt p and a modified *control prompt* $\hat{p}$, whose effect on the denoiser output is formalized in Equation (1). The control prompt provides an alternative conditioning signal that encodes corrective intent purely at the prompt level, while remaining fully compatible with the pretrained generator.

The framework imposes minimal assumptions on the form of $\hat{p}$. In practice, $\hat{p}$ is constructed by modifying only the object-count specification in the original prompt p, while keeping all other semantic content unchanged. We consider two forms of control prompts: (i) a *count-agnostic* prompt obtained by removing explicit cardinality constraints, and

(ii) a *feedback-based* prompt in which the target count is replaced by an estimated count from an intermediate step.

By separating control prompt construction from the steering mechanism, ATHENA decouples how corrective information is encoded from how it influences the sampling trajectory. Different choices of $\hat{p}$ yield distinct instantiations of the framework, while sharing the same denoiser evaluations and sampler interaction. We describe these instantiations in the following subsection.

2.3.3. ATHENA Control Strategies

We present three instantiations of ATHENA that progressively increase adaptivity while preserving the same test-time, training-free steering mechanism. All strategies rely on the prompt-based steering operation and differ in whether and how intermediate generation signals are used to select the control prompt and steering strength. This progression moves from fixed, count-agnostic control to feedback-informed and adaptive steering, enabling increasingly targeted correction of counting errors.

ATHENA-Static. ATHENA-Static applies prompt-based steering using a fixed, count-agnostic control prompt. Given an original prompt p specifying a target count k, the control prompt $\hat{p}$ is constructed by removing the explicit cardinality constraint while preserving all other semantic content. No intermediate count estimation is performed, and steering is applied once from the initial diffusion step to a cutoff t_{steer}, after which standard diffusion proceeds unmodified.

This variant isolates the effect of prompt-level steering without feedback or adaptive adjustment. It incurs minimal computational overhead and serves as a baseline demonstrating that count fidelity can be influenced through prompt-based steering. Details are provided in Sec. B.1.

ATHENA-Feedback. While static steering can improve count fidelity, its effectiveness depends on how well a fixed control prompt aligns with the trajectory. ATHENA-Feedback addresses this limitation by incorporating a single intermediate count estimate to inform the control prompt.

Specifically, diffusion is first run without steering until an intermediate step t_{est}, where the partially denoised latent is decoded to approximate the object count using an object detector. If the estimated count differs from the target, generation is restarted from the same initial noise using a feedback control prompt constructed by replacing the original count in p with the observed count. Prompt-based steering is then applied once during early diffusion steps down to a cutoff t_{steer}, after which diffusion proceeds unmodified to completion. No additional count checks or parameter updates are performed. Algorithmic details are provided in Sec. B.2.

ATHENA-Adaptive. Although feedback steering makes the control prompt feedback-informed, its success depends on the steering strength γ. If γ is too small, steering may reduce the counting error without reaching the target; if γ is too large, it can overshoot the target and add or remove too many objects. ATHENA-Adaptive resolves this sensitivity through a single, direction-aware adjustment of γ based on intermediate feedback.

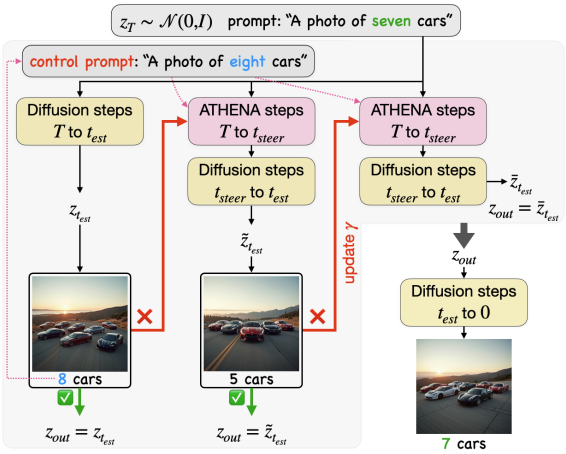


Figure 3. ATHENA-Adaptive pipeline.

The method first estimates the baseline count at t_{est} without steering. If it deviates from the target, prompt-based steering is applied once using the initial γ, as in the previous variant. The count is then re-estimated at t_{est}. If it matches the target, generation proceeds normally. Otherwise, the change in error indicates whether the steering was insufficient or overly aggressive. In the former case, γ is doubled; in the latter, it is halved. A final steered run is then performed with the adjusted γ until t_{steer}, after which diffusion continues unmodified to completion (Figure 3).

Importantly, ATHENA-Adaptive does not perform iterative optimization or repeated parameter tuning. The error direction observed after a single feedback step provides a signal for adjusting the steering magnitude. As a result, the method requires at most two steered trajectories. Pseudocode is provided in Sec. B.3.

3. Experiments

3.1. Experimental Setup

We evaluate ATHENA across multiple diffusion backbones and counting benchmarks. Experiments are conducted using three representative T2I diffusion models spanning different architectures and training regimes: SDXL [11, 14], SD 3.5 Large [2, 15], and FLUX.1-dev [5, 6]. We compare ATHENA against standard unsteered diffusion sampling as well as two baselines from prior work, CountGen [1] and Counting Guidance [4]. Intermediate and final object counts

Table 1. Quantitative counting performance across diffusion backbones and datasets. Accuracy (%) is reported as exact-match count accuracy, with the best result for each model–dataset pair shown in **bold**. Time denotes mean generation time per sample (seconds).

Model	Method	CoCoCount				CoCoCount-E				ATHENA Dataset			
		Acc (↑)	MAE (↓)	RMSE (↓)	Time (↓)	Acc (↑)	MAE (↓)	RMSE (↓)	Time (↓)	Acc (↑)	MAE (↓)	RMSE (↓)	Time (↓)
FLUX.1-dev	Unsteered	58.4	0.98	1.96	45.8	46.5	1.12	1.93	22.6	39.4	1.78	2.96	45.5
	ATHENA-Static	**73.3**	0.56	1.35	54.8	58.5	0.84	1.73	27.4	48.9	1.31	2.44	54.7
	ATHENA-Feedback	71.4	0.67	1.72	56.0	58.3	0.98	2.00	29.8	52.2	1.38	2.60	60.8
	ATHENA-Adaptive	70.2	0.65	1.63	64.8	**62.2**	0.85	1.90	35.1	**53.6**	1.40	2.72	72.5
SD 3.5 Large	Unsteered	58.4	0.78	1.50	43.9	44.8	0.98	1.56	24.1	38.9	1.55	2.54	43.8
	ATHENA-Static	68.3	0.70	1.57	53.4	52.9	0.97	1.78	29.5	46.7	1.40	2.52	52.7
	ATHENA-Feedback	71.4	0.63	1.51	54.9	59.1	0.80	1.77	32.3	51.1	1.31	2.44	58.7
	ATHENA-Adaptive	**78.3**	0.41	1.05	62.0	**65.6**	0.73	1.60	38.1	**56.1**	1.16	2.32	70.4
SDXL	Unsteered	31.7	2.47	4.92	9.4	26.5	3.09	5.85	5.3	19.7	4.02	7.94	9.4
	CountGen	50.3	1.90	4.60	44.1	41.7	2.04	4.68	29.2	29.4	2.70	5.45	55.7
	ATHENA-Static	39.8	1.98	3.85	11.2	31.8	2.40	4.19	6.3	27.5	2.61	4.62	11.2
	ATHENA-Feedback	46.6	1.67	3.59	15.0	36.4	2.24	4.93	8.8	27.2	2.71	4.96	15.8
	ATHENA-Adaptive	**54.0**	1.50	3.47	19.4	**45.5**	1.93	4.52	11.6	**34.7**	2.34	3.90	21.2
SD 1.4	Counting Guidance	28.6	2.19	3.62	19.9	19.4	2.80	4.27	11.1	10.8	3.69	4.85	19.6

are estimated using the pretrained open-vocabulary detector GroundingDINO [10], applied only at test time with the same configuration across all methods and datasets. Evaluation is performed on three counting benchmarks of increasing difficulty: *CoCoCount* [1], its extended version *CoCoCount-E*, and the ATHENA dataset, our newly introduced dataset containing prompts with challenging counting conditions and visual distractions. ATHENA dataset features prompts generated by a large language model to combine multiple constraints within prompts. Additional details about the datasets and configurations are provided in Secs. C and D.

3.2. Quantitative Results

Table 1 reports counting accuracy and runtime across methods and datasets. ATHENA-Adaptive consistently improves exact-count accuracy over unsteered generation, with gains ranging from 11.8% to 22.3%. It outperforms static and feedback variants in nearly all settings, highlighting the importance of direction-aware adaptive steering for robust count control. In addition, ATHENA provides favorable efficiency, achieving higher accuracy than CountGen while being over $2.5\times$ faster and using roughly $4\times$ less memory. Figure 4 analyzes accuracy as a function of target count on CoCoCount-E (SDXL). While performance drops for all methods as counts increase, ATHENA-Adaptive maintains higher accuracy, achieving around 80% for two to three objects and nearly doubling unsteered accuracy at larger counts. Additional results are provided in Sec. E.1.

3.3. Qualitative Results

Figure 5 compares baselines, and ATHENA across diverse prompts. ATHENA improves counting accuracy while preserving scene structure and visual fidelity, avoiding artifacts such as object blending or layout distortion. In contrast, existing methods often alter scene semantics or degrade image quality when enforcing counts, especially in complex settings. Among the variants, ATHENA-Adaptive performs most reliably, with failures mainly due to inaccurate approximations. Additional examples are provided in Sec. E.2.

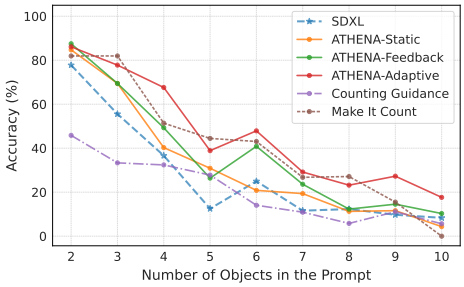


Figure 4. CoCoCount-E with SDXL backbone.

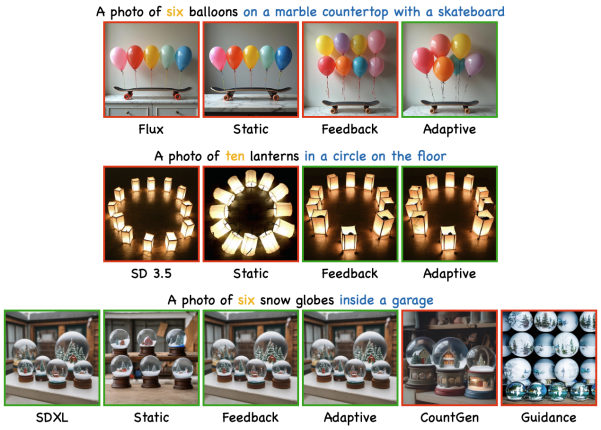


Figure 5. Qualitative results on complex, distractor-rich prompts. *Guidance* denotes Counting Guidance. Green borders indicate correct counts; red borders indicate incorrect counts.

4. Conclusion

We introduced ATHENA, a model-agnostic test-time steering framework for improving object counts in T2I diffusion

models without retraining or architectural changes. By estimating counts at an intermediate diffusion step and applying early prompt-based steering, ATHENA corrects numerical errors before structural mistakes become fixed. Across three diffusion backbones, ATHENA-Adaptive improves accuracy by up to 22.3%, achieves close to 80% accuracy for small target counts, and consistently outperforms prior baselines. These gains come with efficiency: ATHENA is nearly $2.5\times$ faster than CountGen, achieves almost twice the accuracy of Counting Guidance at similar runtime, and requires roughly $4\times$ less memory on SDXL. Overall, ATHENA is an efficient inference-time method for enforcing discrete numerical constraints in diffusion image generation.

Acknowledgements

We sincerely thank Willie Neiswanger and Justin Cho for their insightful feedback and valuable guidance. This work was partially supported by AWS credits through an Amazon Faculty Research Award, a NAIRR Pilot Award, and generous funding by Coefficient Giving. M. Soltanolkotabi and M. S. Sepehri were supported by the USC-Capital One Center for Responsible AI and Decision Making in Finance (CREDIF) Fellowship. M. Soltanolkotabi is also supported by the Packard Fellowship in Science and Engineering, a Sloan Research Fellowship in Mathematics, NSF CAREER Award #1846369, DARPA FastNICS program, NSF CIF Awards #1813877 and #2008443, and NIH Award DP2LM014564-01.

References

[1] Lital Binyamin, Yoad Tewel, Hilit Segev, Eran Hirsch, Royi Rassin, and Gal Chechik. Make It Count: Text-to-Image Generation with an Accurate Number of Objects. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13242–13251, 2025. 1, 3, 4, 10, 14

[2] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 3

[3] Haosheng Gan, Berk Tinaz, Mohammad Shahab Sepehri, Zalan Fabian, and Mahdi Soltanolkotabi. ConceptMix++: Leveling the Playing Field in Text-to-Image Benchmarking via Iterative Prompt Optimization. *Generative Models for Computer Vision Workshop @ CVPR*, 2025. 1

[4] Wonjun Kang, Kevin Galim, Hyung Il Koo, and Nam Ik Cho. Counting guidance for high fidelity text-to-image synthesis. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 899–908. IEEE, 2025. 1, 3, 14

[5] Black Forest Labs. FLUX. https://github.com/black-forest-labs/flux, 2024. 3

[6] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. FLUX. 1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space. *arXiv preprint arXiv:2506.15742*, 2025. 3

[7] Joohyeon Lee, Jin-Seop Lee, and Jee-Hyong Lee. CountCluster: Training-Free Object Quantity Guidance with Cross-Attention Map Clustering for Text-to-Image Generation. *arXiv preprint arXiv:2508.10710*, 2025. 1

[8] Sen Li, Ruochen Wang, Cho-Jui Hsieh, Minhao Cheng, and Tianyi Zhou. Mulan: Multimodal-llm agent for progressive and interactive multi-object diffusion. *arXiv preprint arXiv:2402.12741*, 2024. 1

[9] Yanyu Li, Pencheng Wan, Liang Han, Yaowei Wang, Liqiang Nie, and Min Zhang. CountDiffusion: Text-to-Image Synthesis with Training-Free Counting-Guidance Diffusion. *arXiv preprint arXiv:2505.04347*, 2025. 1

[10] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 4

[11] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 3

[12] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 1

[13] 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

[14] Stability AI. Stable Diffusion XL - Base 1.0. https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0, 2023. 3

[15] Stability AI. Stable Diffusion 3.5 - Large. https://huggingface.co/stabilityai/stable-diffusion-3.5-large, 2024. 3

[16] Oz Zafar, Lior Wolf, and Idan Schwartz. Detection-Driven Object Count Optimization for Text-to-Image Diffusion Models. *arXiv preprint arXiv:2408.11721*, 2025. 1

[17] Guanning Zeng, Xiang Zhang, Zirui Wang, Haiyang Xu, Zeyuan Chen, Bingnan Li, and Zhuowen Tu. YOLO-Count: Differentiable Object Counting for Text-to-Image Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16765–16775, 2025. 1

A. Object Count Estimation During Diffusion Process

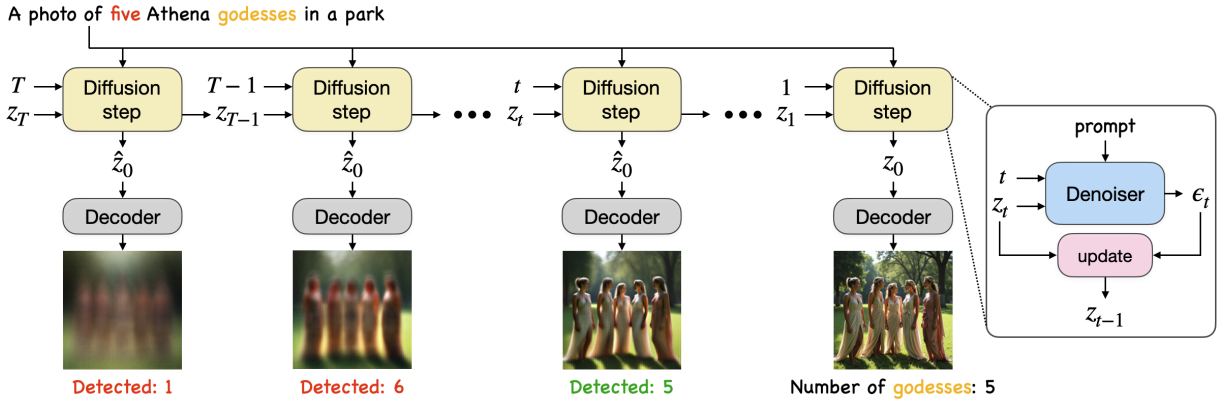


Figure 6. Count estimation across diffusion steps. At early sampling steps, object structure is weakly formed and decoded images are too noisy for reliable count estimation. At later steps, object instances become well defined, and counting is reliable, but requires additional diffusion steps and a higher inference cost. This trade-off motivates estimating object count at an intermediate diffusion step.

B. Algorithmic Details of ATHENA

B.1. ATHENA-Static

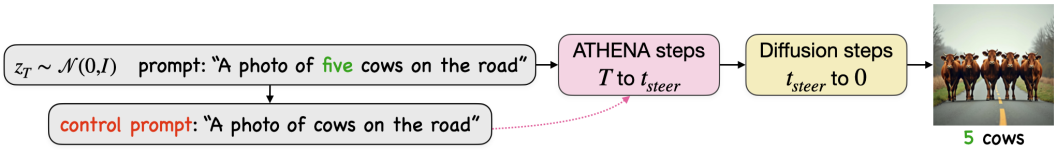


Figure 7. ATHENA-Static pipeline. Starting from the same initial noise, prompt-based steering is applied using a fixed, count-agnostic control prompt from diffusion step T down to a cutoff t_{steer}. Standard diffusion then proceeds unmodified to completion, yielding improved count fidelity without intermediate feedback or adaptive adjustment.

Algorithm 1 DIFFUSION-STEPS

Require: Denoiser: ϵ_θ, sampling operator: $\mathcal{S}_t$, text prompt: p, starting step: t_s, end step: t_e, initial latent: z_{t_s}

for $t = t_s$ to $t_e + 1$ **do**
  $\epsilon_t = \epsilon_\theta(t, z_t, p)$
  $z_{t-1} = \mathcal{S}_t(z_t, \epsilon_t)$
end for
Return: z_{t_e}

Algorithm 2 ATHENA-STEPS

Require: Denoiser: ϵ_θ, sampling operator: $\mathcal{S}_t$, text prompt: p, control prompt: $\hat{p}$, starting step: t_s, end step: t_e, initial latent: z_{t_s}, steering strength: $\gamma > 0$

for $t = t_s$ to $t_e + 1$ **do**
  $\epsilon_t = \epsilon_\theta(t, z_t, p)$
  $\hat{\epsilon}_t = \epsilon_\theta(t, z_t, \hat{p})$
  $\epsilon_{steer} = \epsilon_t + \gamma(\epsilon_t - \hat{\epsilon}_t)$
  $\epsilon_{steer} = \frac{\|\epsilon_t\|}{\|\epsilon_{steer}\|}\epsilon_{steer}$ {Matching the norm of the new update with the original update}
  $z_{t-1} = \mathcal{S}_t(z_t, \epsilon_{steer})$
end for
Return: z_{t_e}

Algorithm 3 ATHENA-STATIC

Require: Denoiser: ϵ_θ, sampling operator: $\mathcal{S}_t$, text prompt: p, target count: k, steering steps: t_{steer}, total steps: T, steering strength: $\gamma > 0$, decoder $\text{Dec}(\cdot)$

$z_T \sim \mathcal{N}(0, I)$
$\hat{p} \leftarrow$ remove k from p
$z_{t_{steer}} =$ **ATHENA-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, \hat{p}, t_s = T, t_e = t_{steer}, z_T, \gamma > 0)$
$z_0 =$ **Diffusion-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, t_s = t_{steer}, t_e = 0, z_{t_{steer}})$
$x = \text{Dec}(z_0)$
Return: x

B.2. ATHENA-Feedback

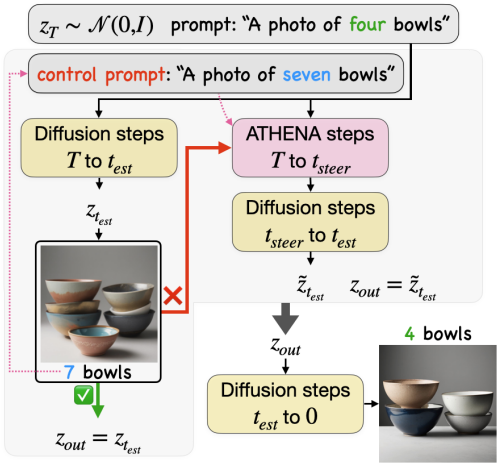


Figure 8. ATHENA-Feedback pipeline. An intermediate count is estimated at step t_{est} without steering. If the estimate differs from the target, generation is restarted from the same initial noise using a feedback control prompt, and a single early-stage steering phase is applied before completing diffusion.

Algorithm 4 ATHENA-FEEDBACK

Require: Denoiser: ϵ_θ, sampling operator: $\mathcal{S}_t$, reconstruction $\mathcal{D}_t$, text prompt: p, target count: k, steering steps: t_{steer}, estimation step t_{est}, total steps: T, steering strength: $\gamma > 0$, decoder $\text{Dec}(\cdot)$, counter $\text{Count}(\cdot)$

$z_T \sim \mathcal{N}(0, I)$
$\hat{p} \leftarrow$ remove k from p
$z_{t_{est}} =$ **Diffusion-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, t_s = T, t_e = t_{est}, z_T)$
$\hat{z}_0 = \mathcal{D}_t(z_{t_{est}})$
$\hat{x} = \text{Dec}(\hat{z}_0)$
$c = \text{Count}(\hat{x})$
if $c = k$ **then**
  $z_{out} = z_{t_{est}}$
else
  $\hat{p} \leftarrow$ replace k in p with c
  $z_{t_{steer}} =$ **ATHENA-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, \hat{p}, t_s = T, t_e = t_{steer}, z_T, \gamma > 0)$
  $\tilde{z}_{t_{est}} =$ **Diffusion-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, t_s = t_{steer}, t_e = t_{est}, z_{t_{est}})$
  $z_{out} = \tilde{z}_{t_{est}}$
end if
$z_0 =$ **Diffusion-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, t_s = t_{est}, t_e = 0, z_{out})$
$x = \text{Dec}(z_0)$
Return: x

B.3. ATHENA-Adaptive

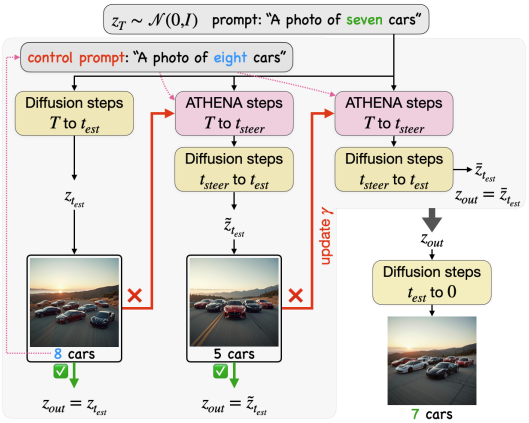


Figure 9. ATHENA-Adaptive pipeline. The method estimates object count at an intermediate diffusion step, applies early-stage prompt-based steering, adaptively adjusts the steering strength once based on the observed error direction, and completes generation using diffusion.

Algorithm 5 ATHENA-ADAPTIVE

Require: Denoiser: ϵ_θ, sampling operator: $\mathcal{S}_t$, reconstruction $\mathcal{D}_t$, text prompt: p, target count: k, steering steps: t_{steer}, estimation step t_{est}, total steps: T, steering strength: $\gamma > 0$, decoder $\text{Dec}(\cdot)$, counter $\text{Count}(\cdot)$

$z_T \sim \mathcal{N}(0, I)$
$\hat{p} \leftarrow$ remove k from p
$z_{t_{est}} =$ **Diffusion-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, t_s = T, t_e = t_{est}, z_T)$
$\hat{z}_0 = \mathcal{D}_t(z_{t_{est}})$
$\hat{x} = \text{Dec}(\hat{z}_0)$
$c_1 = \text{Count}(\hat{x})$
if $c_1 = k$ **then**
  $z_{out} = z_{t_{est}}$
else
  $\hat{p} \leftarrow$ replace k in p with c
  $z_{t_{steer}} =$ **ATHENA-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, \hat{p}, t_s = T, t_e = t_{steer}, z_T, \gamma > 0)$
  $\tilde{z}_{t_{est}} =$ **Diffusion-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, t_s = t_{steer}, t_e = t_{est}, z_{t_{est}})$
  $\hat{z}_0 = \mathcal{D}_t(\tilde{z}_{t_{est}})$
  $\hat{x} = \text{Dec}(\hat{z}_0)$
  $c_2 = \text{Count}(\hat{x})$
  if $c_2 = k$ **then**
    $z_{out} = \tilde{z}_{t_{est}}$
  else
    if $(c_1 \leq c_2 < k) \vee (k < c_2 \leq c_1)$ **then**
      {Case of not adding/removing enough objects}
      $\gamma \leftarrow 2 \times \gamma$
    else
      {Case of adding/removing too many objects}
      $\gamma \leftarrow \frac{\gamma}{2}$
    end if
    $z_{t_{steer}} =$ **ATHENA-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, \hat{p}, t_s = T, t_e = t_{steer}, z_T, \gamma > 0)$
    $\tilde{z}_{t_{est}} =$ **Diffusion-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, t_s = t_{steer}, t_e = t_{est}, z_{t_{est}})$
    $z_{out} = \tilde{z}_{t_{est}}$
  end if
end if
$z_0 =$ **Diffusion-Steps** $(\epsilon_\theta, \mathcal{S}_t, p, t_s = t_{est}, t_e = 0, z_{out})$
$x = \text{Dec}(z_0)$
Return: x

C. Dataset Construction and Statistics

We conduct experiments on three counting benchmarks of increasing complexity. *CoCoCount* [1] consists of simple prompts with explicit numerical constraints and minimal scene context. To avoid bias from repeated prompts in the original benchmark, we use a deduplicated version of the dataset in which prompts with identical text are retained once, yielding 161 unique prompts with target counts from 2 to 10 (see Sec. C). While well-suited for controlled evaluation, CoCoCount offers limited diversity in prompt structure and scene complexity.

To enable systematic analysis across target cardinalities, we construct *CoCoCount-E*, an extended benchmark in which a filtered subset of CoCoCount prompts is instantiated with target counts from 2 to 10, yielding 648 distinct prompts with consistent structure and a balanced count distribution. We cap the maximum target count at 10, as prior work [1] reports substantial degradation beyond this range, which we also observe empirically.

We further introduce the ATHENA dataset, a new benchmark designed to evaluate object counting under progressively challenging prompt conditions. The dataset consists of 360 prompts organized into four levels of increasing complexity, with target counts ranging from 2 to 10 (see Tab. 2). Prompts are generated using a large language model to systematically combine multiple constraints within a single instruction, ensuring controlled diversity across difficulty levels.

Table 2. Structure of the ATHENA dataset with four levels of increasing prompt complexity.

Level	**Description**	**Example Prompt**
L1	Hard object categories	"nine microphones"
L2	+ Scene context	"eight sneakers at the edge of a pond"
L3	+ Distractor objects	"five jars beside a river with a person"
L4	+ Relational constraints	"seven dumbbells lined up along a sidewalk"

C.1. CoCoCount and CoCoCount-E

The CoCoCount dataset [1] consists of text prompts with explicit numerical constraints, targeting object counts from 2 to 10. The original release includes prompts that differ only by random generation seed, as well as prompts with identical semantic structure but different target counts. As a result, the dataset contains repeated prompt content and heterogeneous count distributions.

For evaluation, we construct a deduplicated version of CoCoCount by retaining a single instance of each unique prompt text. This results in 161 distinct prompts, which we use as the CoCoCount benchmark in all experiments.

To enable controlled analysis across target cardinalities, we further construct *CoCoCount-E*, an extended benchmark derived from CoCoCount. Starting from a filtered subset of 72 unique prompt templates, we systematically instantiate each prompt with target counts from 2 to 10 while keeping all other prompt content fixed. This yields 648 prompts with consistent structure and a balanced count distribution.

Following prior work [1], we restrict the target count range to 2–10, as generation quality and counting accuracy degrade substantially beyond this range. We observe the same trend across all evaluated diffusion backbones.

Table 3. Dataset statistics for CoCoCount and CoCoCount-E.

Dataset	**Unique Prompts**	**Count Range**	**Total Prompts**
CoCoCount (deduplicated)	161	2–10	161
CoCoCount-E	72	2–10	648

C.2. ATHENA Dataset Generation

The ATHENA dataset is generated by querying a large language model (GPT-5.2) using the prompt shown below. This prompt instructs the model to produce object-counting instructions with controlled variations in object category, scene context, distractors, and relational constraints, while sampling target counts between 2 and 10. Applying this prompt yields a structured collection of prompts organized into four difficulty levels, with consistent formatting and systematic increases in complexity. Finally, all generated samples are manually inspected to ensure quality.

Task Definition

```
Generate a challenging object-counting evaluation dataset for text-to-image diffusion
↪  models. The dataset is intended to evaluate count fidelity only, under increasingly
↪  difficult prompt conditions.

You are given:
- A JSON file containing the original CoCoCount dataset (attached).
- The specifications below.

You must generate a new dataset that:
- Does NOT reuse any object categories appearing in CoCoCount for Level 1.
- Uses clear, unambiguous noun-phrase prompts suitable for automatic counting.
- Contains no duplicate prompts.
- Contains exactly one counted object category per prompt.
- All prompts must begin with: "A photo of ..."
```

Dataset Output Format

```
- Output a single JSON file as a list of entries.
- Each entry must contain the following required fields.

Required JSON template (exact keys required):
```

```
{
  "id": <unique integer>,
  "prompt": <string>,                 // full prompt, starts with "A photo of"
  "object": <string>,                 // singular object name
  "object_plural": <string>,          // plural form used in the prompt
  "number": <string>,                 // number in words (e.g., "three")
  "int_number": <integer>,                 // numeric count (e.g., 3)

  "level": <integer>,                 // 1, 2, 3, or 4
  "difficulty_tag": <string>,         // short descriptor (e.g., "hard-object",
  ↪  "scene", "distractor", "relation")

  "scene": <string | null>,           // scene phrase if present (e.g., "on the
  ↪  ground")
  "distractor": <string | null>,      // uncounted object if present
  "relation": <string | null>,        // relational phrase if present

  "source": "ATHENA"
}
```

Detector Compatibility Constraint

```
All object categories MUST be reliably detectable by open-vocabulary grounding models
(e.g., Grounding DINO) under standard confidence thresholds.

Objects must:
- be visually salient at typical image resolutions,
- have strong and common visual-text associations,
- be distinguishable without fine-grained or microscopic detail,
commonly appear in natural images rather than technical diagrams or product-only photos.

Avoid objects that are:
- extremely small or thin,
- rare, niche, or domain-specific,
- visually indistinguishable without context,
- typically embedded inside other objects.
```

Count Constraints

```
- Valid counts: 2 - 10
- Counts should be approximately balanced
- "number" must be the word form of "count"
- "object_plural" must match the prompt text exactly
```

Difficulty Levels

Level 1: Hard Object Categories

```
- Object categories must NOT appear in CoCoCount.
- Objects should be visually challenging for counting (e.g., instance ambiguity,
↪  moderate occlusion, reflective or deformable surfaces), but must remain clearly
↪  recognizable as distinct object instances by open-vocabulary detection models.
- No verbs
- No scene
- No distractors
```

Prompt format:

```
A photo of <number> <object_plural>
```

Example pattern: *A photo of seven kites*

Level 2: Hard Objects + Scene Context

```
- Same object constraints as Level 1.
- Add a scene phrase.
- Scene must not introduce additional countable objects.
- No verbs
- No distractors
```

Prompt format:

```
A photo of <number> <object_plural> <scene>
```

Example pattern: *A photo of six lanterns in a temple hall*

Level 3: Counted Object + Semantic Distractor

```
- One counted object category.
- One uncounted distractor (no number specified).
- No verbs
- Avoid ambiguity about which object is counted.
```

Prompt format:

```
A photo of <number> <object_plural> <scene> with <distractor>
```

Example pattern: *A photo of five apples on a table with a dog*

Level 4: Relational Language

```
- One counted object category.
- Introduce spatial or relational structure.
- No second counted object.
- Relational phrasing may include light verb-based constructions if unavoidable.
```

Prompt format:

```
A photo of <number> <object_plural> <relation>
```

Example pattern: *A photo of eight chairs arranged around a round table*

Strict Constraints

```
- Do NOT reuse Level-1 object categories from CoCoCount.
- Do NOT include multiple counted objects.
- Do NOT use vague quantifiers ("several", "many").
- Do NOT use negation-based counting.
- Do NOT generate prompts requiring subjective interpretation.
- Do NOT generate near-duplicate prompts.
- Ensure "scene", "distractor", and "relation" are null when not applicable.
```

Dataset Size

```
Generate exactly 360 prompts, distributed as:
- Level 1: 120
- Level 2: 100
- Level 3: 100
- Level 4: 40
```

Output Requirements

```
- Output valid JSON only
- All required fields must be present
- Prompts must be natural, fluent, and concise
- Share the dataset link when you finish generating it
```

D. Experimetal Setup

D.1. Baselines

We compare ATHENA against unsteered diffusion sampling and two representative baselines from prior work: *CountGen* [1] and *Counting Guidance* [4]. Further details about the baselines are provided in Sec. D. Both baselines rely on model-specific components and are therefore not backbone-agnostic. As a result, we evaluate each method on the backbone it was originally designed for: CountGen on SDXL and Counting Guidance on SD 1.4. Although Counting Guidance is evaluated on a different backbone, we include it as a reference due to its comparable inference-time overhead to ATHENA-Adaptive on SDXL, enabling a meaningful comparison of accuracy–runtime trade-offs.

D.2. Hyperparameter Settings

All ATHENA hyperparameters, including the estimation step t_{est}, steering horizon t_{steer}, and steering strength γ, are tuned exclusively on the CoCoCount dataset. Hyperparameter tuning is conducted separately for each diffusion backbone. Once selected, the resulting parameters are fixed and reused without modification for CoCoCount-E and the ATHENA dataset.

All diffusion models are evaluated using a fixed random seed (23) for image generation to ensure reproducibility. Table 4 summarizes the hyperparameter settings used for ATHENA across diffusion backbones and steering variants.

Table 4. Hyperparameter settings used for ATHENA across diffusion backbones and steering variants.

Parameter	**Diffusion Backbone**								
	FLUX.1-dev			**SD 3.5 Large**			**SDXL**		
	Static	Feedback	Adaptive	Static	Feedback	Adaptive	Static	Feedback	Adaptive
t_{est}	–	20	20	–	20	20	–	30	30
t_{steer}	10	5	5	10	5	5	10	10	10
γ	4	4	4	4	4	4	5	5	5

D.2.1. Computational Setup

All experiments are conducted on a single GPU per run. We use NVIDIA RTX A6000 GPUs (48 GB) for CoCoCount and the ATHENA dataset, and NVIDIA A100 SXM4 GPUs (40 GB) for CoCoCount-E. Within each dataset, the same GPU type and random seed are used across all models and methods to ensure fair comparisons. Absolute generation times may differ across datasets due to hardware differences; accordingly, we focus on relative accuracy–runtime trade-offs with fixed hardware.

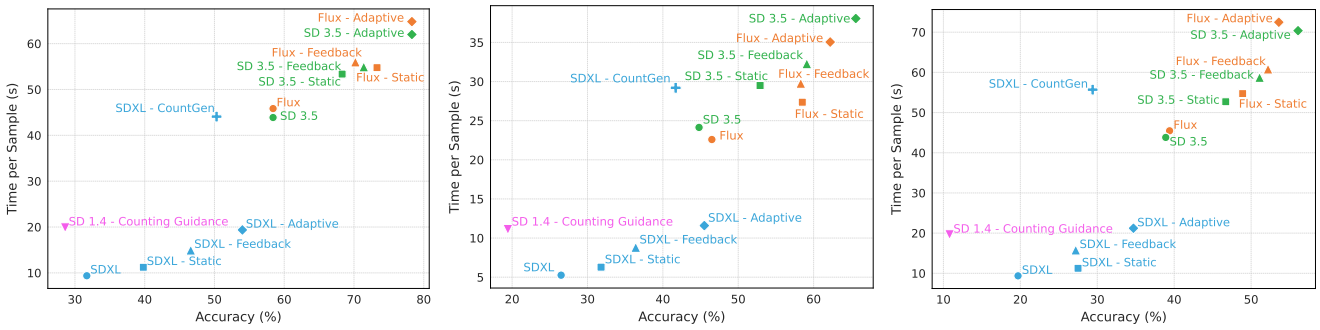
Notably, CountGen does not support target counts above nine; for such prompts, we report unsteered sampling. Counting Guidance does not support multi-word object categories; these are concatenated with underscores.

E. Additional Experiments

E.1. Quantitative Results

E.1.1. Efficiency

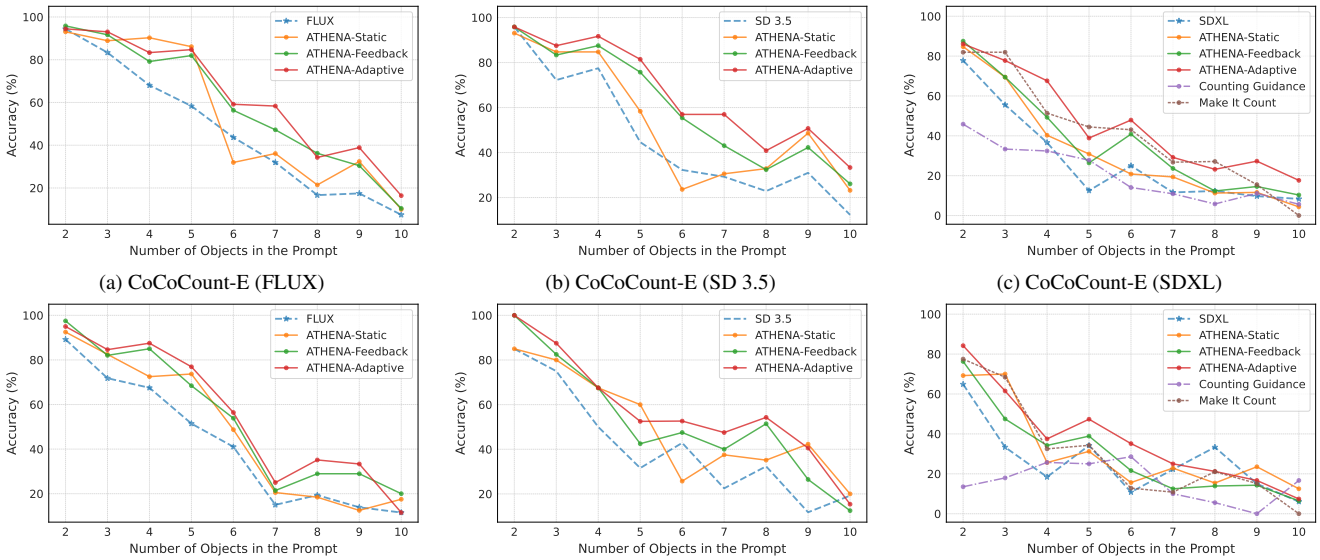
Accuracy improvements are achieved with favorable efficiency trade-offs. As shown in Table 1 and Figure 10c, ATHENA-Adaptive attains higher accuracy than CountGen while being over $2.5\times$ faster on the ATHENA dataset. At a comparable runtime to Counting Guidance, ATHENA-Adaptive on SDXL achieves nearly $2\times$ higher counting accuracy. In terms of memory usage, CountGen requires 47.5 GB, Counting Guidance uses 17.2 GB, while ATHENA-Adaptive on SDXL requires only 11.9 GB. Thus, ATHENA achieves higher accuracy than prior baselines with approximately $4\times$ lower memory usage than CountGen, highlighting its efficiency as a lightweight, test-time control method.



(a) CoCoCount (b) CoCoCount-E (c) ATHENA dataset

Figure 10. Accuracy–runtime trade-offs across datasets. Colors denote the diffusion backbone, while marker shapes indicate the method. ATHENA-Adaptive consistently achieves higher accuracy at comparable or lower runtime than prior baselines across all benchmarks.

E.1.2. Accuracy vs. Object Count



(a) CoCoCount-E (FLUX) (b) CoCoCount-E (SD 3.5) (c) CoCoCount-E (SDXL)

(d) ATHENA dataset (FLUX) (e) ATHENA dataset (SD 3.5) (f) ATHENA dataset (SDXL)

Figure 11. Counting accuracy versus target object count across datasets and diffusion backbones. ATHENA-Adaptive consistently maintains higher accuracy as the target count increases, demonstrating improved robustness to increasing counting difficulty.

E.2. Qualitative Results

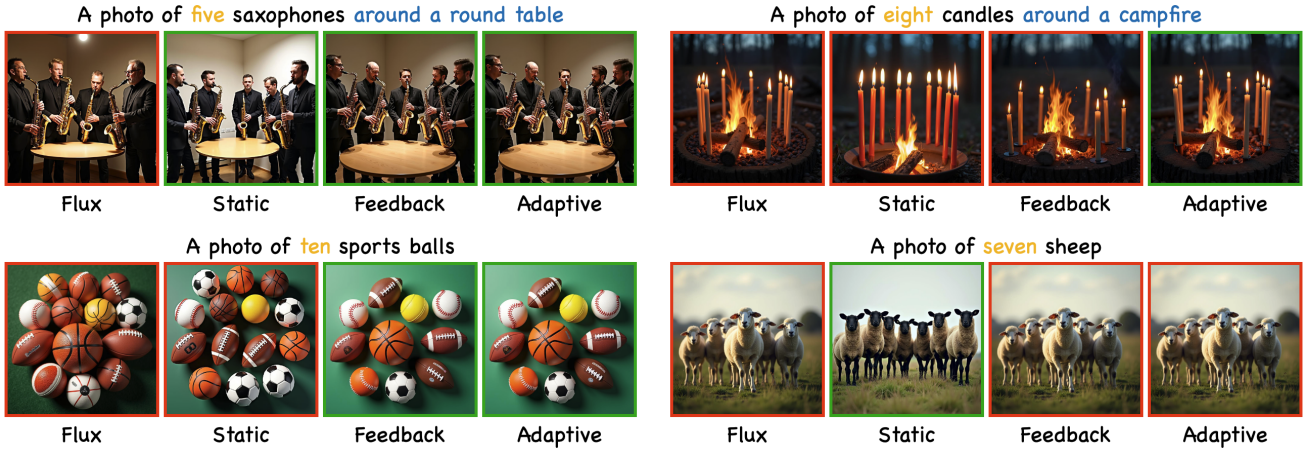


Figure 12. Additional qualitative results on relational and multi-object prompts. Results are shown for unsteered generation (FLUX) and the three ATHENA variants. Green borders indicate correct object counts, while red borders denote counting errors. ATHENA improves count fidelity across diverse settings (e.g., grouping, circular layouts, and natural scenes) while preserving scene structure and visual coherence, with the adaptive variant yielding the most consistent corrections.

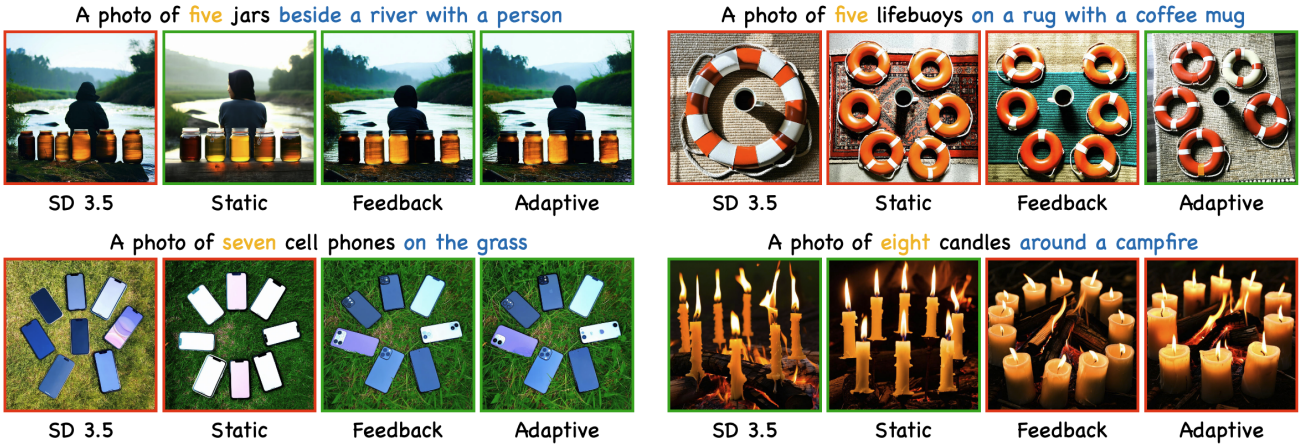


Figure 13. Additional qualitative results on relational and multi-object prompts. Results are shown for unsteered generation (SD 3.5) and the three ATHENA variants. Green borders indicate correct object counts, while red borders denote counting errors. ATHENA improves count fidelity across diverse settings (e.g., grouping, circular layouts, and natural scenes) while preserving scene structure and visual coherence, with the adaptive variant yielding the most consistent corrections.

Figure 14. Additional qualitative results on relational and multi-object prompts with the SDXL backbone. Results include unsteered generation, CountGen, Counting Guidance (*Guidance*), and the three ATHENA variants. Green borders indicate correct object counts, while red borders denote counting errors. Prior baselines frequently fail to enforce correct counts or introduce visual artifacts, whereas ATHENA improves count fidelity while preserving scene structure and visual coherence, with the adaptive variant yielding the most consistent corrections.