D2D: Detector-to-Differentiable Critic for Improved Numeracy in Text-to-Image Generation

Nobline Yoo*
University of Pennsylvania

Olga Russakovsky
Princeton University

Ye Zhu
LIX, CNRS, École Polytechnique, IP Paris

Abstract

Text-to-image (T2I) diffusion models have achieved strong performance in semantic alignment, yet they still struggle with generating the correct number of objects specified in prompts. Existing approaches typically incorporate auxiliary counting networks as external critics to enhance numeracy. However, since these critics must provide gradient guidance during generation, they are restricted to regression-based models that are inherently *differentiable*, thus excluding detector-based models with superior counting ability, whose count-via-enumeration nature is *non-differentiable*. To overcome this, we propose **Detector-to-Differentiable** (*D2D*), a novel framework that transforms non-differentiable detection models into differentiable critics, leveraging their superior counting ability to guide numeracy generation. Specifically, we design custom activation functions to convert detector logits into soft binary indicators, which are then used to optimize the noise prior at inference time with pre-trained T2I models. Our extensive experiments on SDXL-Turbo, SD-Turbo, and Pixart-DMD across four benchmarks of varying complexity (low-density, high-density, and multi-object scenarios) demonstrate consistent and substantial improvements in object counting accuracy (e.g., boosting up to 13.7% on D2D-Small, a 400-prompt, low-density benchmark), with minimal degradation in overall image quality and computational overhead. Project page: https://noblineY.github.io/d2d/.

1 Introduction

Open-sourced diffusion-based text-to-image generation models [6, 7, 1, 8, 9] have achieved promising performance in text-image semantic alignment, particularly with recent post-enhancement techniques such as fine-tuning [10, 11, 12, 13, 14, 15, 16] or sampling-based, inference strategies [17, 2, 18, 19]. But even with such advanced alignment techniques (e.g. ReNO [2] optimizes image alignment via human preference rewards), they continue to struggle at generating exact counts of objects, as illustrated in Fig. 1. Importantly, this challenge persists even in newer, larger models: FLUX-schnell [20] and SD3.5 Large Turbo [21] still score below 50% on the CoCoCount [3] benchmark (200 prompts with counts ≤ 10). Motivated by this, we tackle the challenge of accurate numeracy in generation with novel count-aware guidance.

Ingredients of numeracy optimization. Since vanilla T2I models are not explicitly trained to count, existing methods [4, 5] introduce auxiliary counting critics to provide additional inference-time supervision via gradients propagated through the generative backbone. To effectively guide generation, these critics must provide both an accurate estimation of the current object count and strong gradients for adding/removing objects as needed, which restricts current approaches to differentiable, regression-based models such as RCC [24] and CLIP-Count [25]. As a result, this

*Work mainly completed while at Princeton University and the summer prior to doctoral program. Correspondence to nobline@engineering.upenn.edu.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

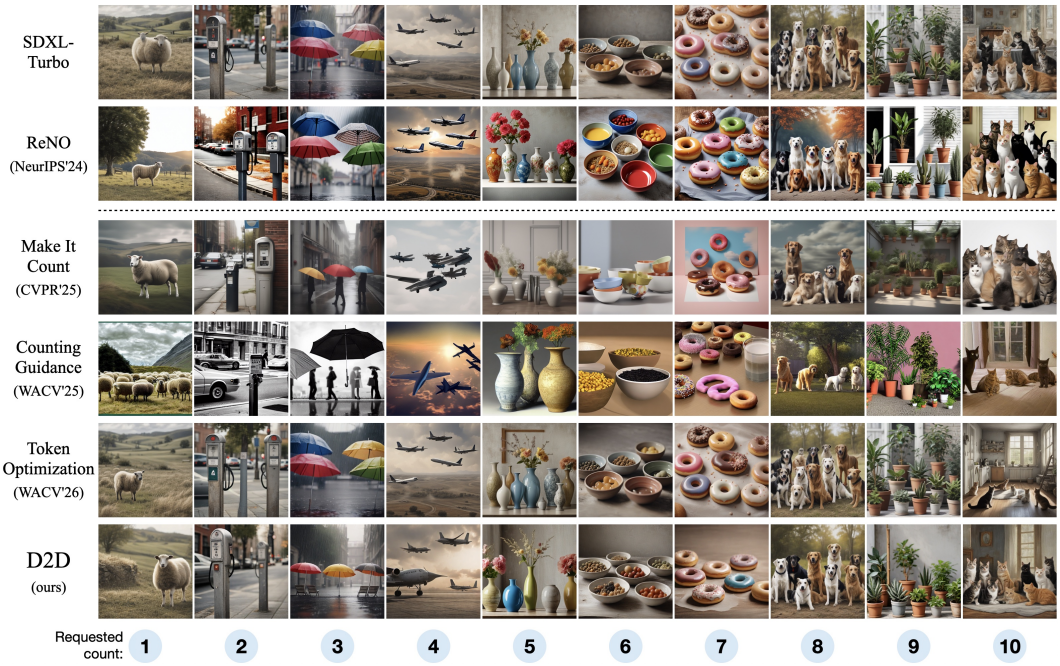
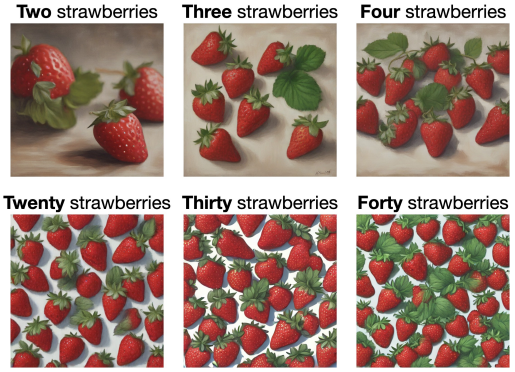
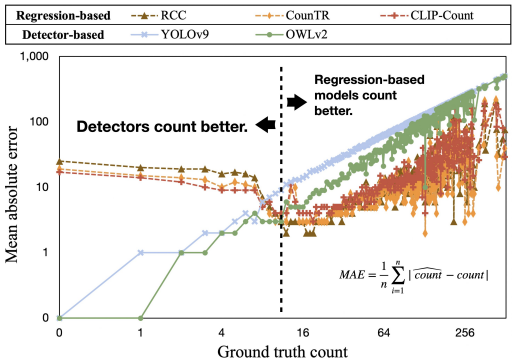


Figure 1: **Qualitative examples illustrating the count-correction ability of our detector-based critic on a variety of objects, counts 1-10.** SDXL-Turbo [1] is a base model with no post-enhancement. ReNO [2] is a generic semantic alignment method that exhibits limited performance in this setting. More recent methods (e.g., Make It Count [3], Counting Guidance [4], TokenOpt [5]), explicitly address count-correction. Our method proposes a new, effective way to leverage detectors for this challenging task. Prompt template: "A realistic photo of a scene with [count] [object class]."



(a) Low and high-density examples with incorrect numeracy, generated by SDXL-Turbo.



(b) Error by ground truth count on TallyQA [22] and FSC147 [23].

Figure 2: **The low-density setting is where incorrect numeracy is most noticeable and also where detectors count better than regression-based methods.** *But detectors are not differentiable, which precludes them from being used as critics for count correction.*

design excludes the possibility of using stronger detector-based count critics, as they count via bounding box enumeration. As illustrated in Fig. 2b, despite being non-differentiable, detectors outperform regression-based counterparts in low-density object scenarios important to the generative setting, due to their more advanced object localization ability. To compensate, prior works like Token Optimization (TokenOpt) [5] rely on a dual-model pipeline, using a regression-based model for the gradient signal and a detector for accurate counting, at the cost of processing the image through both models. Others [3] forego the use of counters, instead relying on layout-based methodologies at the cost of image quality. We propose resolving this bottleneck by directly converting existing object detectors into differentiable critics, allowing T2I diffusion models to benefit from stronger counting models for improved numeracy.

The Detector-to-Differentiable (*D2D*) recipe. The *D2D* framework builds on two key insights that set it apart from prior count correction methods [4, 5, 3]. First, instead of relying on regression-based counters, we design a high-curvature activation function that relaxes detector-produced discrete object counts into a smooth surrogate. Crucially, the shape of this curve is *carefully designed* to extract convergence-friendly gradients, as naive soft relaxations provide insufficient optimization signals. Second, to leverage our new gradient, unlike prior approaches that intervene at intermediate states or denoised predictions along the sampling trajectory, we instead optimize the initial noise using a lightweight test-time tunable module, the Latent Modifier Network (LMN), which learns a perturbation of the Gaussian latent while mixing it with the original noise to maintain proximity to the diffusion prior, thereby providing an implicit regularization effect. This backbone-agnostic design enables broader generalization of our method across diverse diffusion-based T2I architectures, including U-Net [26] and DiT [27].

We demonstrate the effectiveness of *D2D* via comprehensive experiments on various generative backbones (i.e., SDXL-Turbo [1], SD-Turbo [1], Pixart-DMD [9]) and multiple prompt benchmarks (i.e., CoCoCount [3], D2D-Small, D2D-Multi, D2D-Large), covering diverse numeracy scenarios, including single and multiple objects. *D2D* yields the highest numeracy across all multi-step and one-step baselines and benchmarks. On base model SDXL-Turbo, *D2D* effectively corrects 42% of under-generations (i.e., when the initial generation contains fewer than requested objects) and 40% of over-generations, nearly or more than $2\times$ the correction rate of ReNO [2] and TokenOpt [5]. In summary, our contributions are as follows:

- We highlight the importance of accurate numeracy in T2I generation and propose a framework to convert robust object detectors into differentiable critics for count-correction with a newly designed activation function, overcoming the bottleneck of having to rely on existing regression-based methods with limited accuracy.
- We reposition the count-correction problem within the initial noise optimization framework, leveraging structural priors in the Gaussian latent with mixing-based regularization that preserves proximity to the diffusion prior across models.
- Our method *D2D* outperforms previous one-step and multi-step count-correction methods by up to **13.7%** points (from 30% with Make It Count to 43.7% with *D2D* on D2D-Small), with minimal degradation in image quality (Fig. 1). On single-object prompts with counts less than or equal to 10, our method introduces less than or comparable computational overhead than baselines.
- When applied to a smaller model (SDXL-Turbo [1]), *D2D* outperforms larger models FLUX-schnell [20] and SD3.5 Large Turbo [21] by up to 9.5% (from FLUX-schnell's 46.0% to 55.5% using *D2D* with SDXL-Turbo on the CoCoCount benchmark).

2 Related Work

2.1 Generic alignment-enhancement methods

As noted in the literature [14, 11], the base log-likelihood objective of diffusion models is insufficient on its own to achieve state-of-the-art alignment. To address this, prior works optimize human preference scores via post-enhancement strategies ranging from fine-tuning the U-Net or text encoder [10, 15, 12, 13, 14, 16, 11] to inference-time, training-free sampling strategies that update the intermediate latents [18, 19]. A promising recent line of work [17, 2] proposes inference-time alignment via initial noise selection, motivated by the presence of semantic/structural priors in the initial noise [28] that influence the semantics/structure of the generated output consistently across diffusion models even with different backbones. But regardless of the specific optimization strategy, the problem remains that generic alignment objectives like human-preference scores are insufficient to steer the model towards count-correct distributions; there persists a significant gap relative to state-of-the-art count-correction methods like [3]. In our work, we specifically address the challenge of improving numeracy with a new formulation for the objective, as well as adopt initial noise optimization as the method of learning, for the ease with which it can be applied across different backbones and the ability to leverage optimized seeds to complement existing methods, as we demonstrate in experiments.

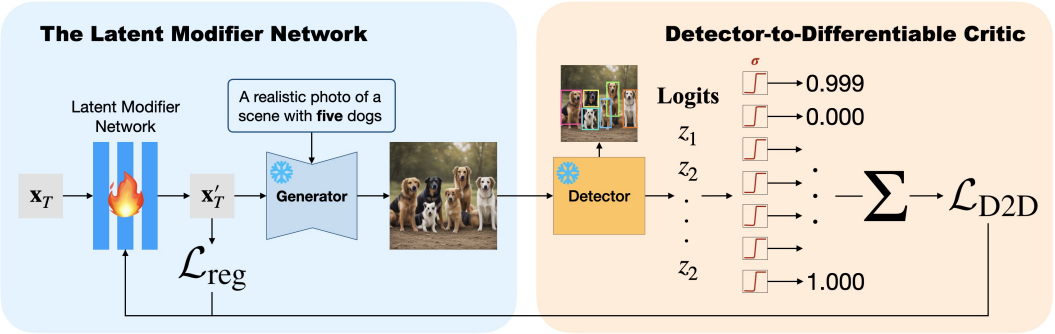


Figure 3: **The *D2D* pipeline for improving T2I numeracy.** *D2D* consists of two main components that work together to improve numeracy: our detector-based count critic guides the Latent Modifier Network (LMN) on how to transform the original initial noise $\mathbf{x}_T$ into a more optimal $\mathbf{x}'_T$. Our count critic uses sigmoid-derived activation functions to convert logits from the frozen detector into gradient signals, which are backpropagated through the frozen generator to update the LMN weights.

2.2 Numeracy correction methods

Existing count-correction methods leverage two main mechanisms at inference-time to correct count: (1) apply the gradient of external counting models to correct a tunable portion of the generation process, like Counting Guidance [4] and TokenOpt [5], or (2) use attention to control the layout of generated instances, like Make It Count [3]. Counting Guidance uses the RCC counting model [24] to optimize the predicted noises, and TokenOpt uses CLIP-Count [25] to optimize the embedding of a count token injected into the prompt as well as a detector to correct for CLIP-Count's overestimates, increasing the computational overhead (about 2-6$\times$ longer than *D2D* on average). Make It Count [3] is an SDXL-specific [6] method that uses self-attention features of the U-Net to extract masks of generated instances and cross-attention to enforce a corrected set of masks. These works are either limited by the need to rely on regression-based counters or manner in which they enforce structure at the cost of image quality, a phenomenon documented in [29, 5, 30] and noted in our experiments (Appendix J.1). Instead, *D2D* leverages a more robust *detector-based* critic that enables more effective correction in the low-density setting.

2.3 Regression vs. detector-based counting models

Regression-based counting methods take an input image and estimate count on a continuous scale. Different variations allow for (1) exemplar-based (i.e., count the instances that look similar to the user-provided example), (2) zero-shot (i.e., count the most salient object), and (3) text-prompted counting (i.e., count the text-specified object). Designed to help count high-density images, where continuous-scale predictions are appropriate, they exhibit limited performance in low-density images [31], as illustrated in Fig. 2b. On the other hand, our *D2D* critic is derived from detectors which show robust performance given low-density images, which is critical to the generative setting (Fig. 2). Furthermore, *D2D* can be used to generate objects in the open set by leveraging *open-vocabulary* detectors, like OWLv2 [32], with minimal modification to detector architecture. In our work, we compare our critic against three regression-based counting methods: RCC [24] (zero-shot), CLIP-Count [25] (text-specified), and CounTR [33] (zero-shot).

3 The *D2D* Framework

Problem statement. Given a pre-trained, one-step T2I model G_θ and prompt p requesting N counts of an object of class C, our goal is to generate an image with exactly N counts of C.

Summary of approach. We propose a detector-based count critic that provides a more effective gradient signal. We then design a mixing method to use that signal to influence the generation process, by leveraging the structural priors in the initial latent which we modify to align with the gradient.

3.1 Detector-to-Differentiable critic

Detector $\mathcal{D}$ takes as inputs an object class C and image I and outputs a set of n bboxes $\{B_i | 1 \leq i \leq n\}$ and logits $\mathbf{z} = \{z_i | 1 \leq i \leq n\}$. A sigmoid $\sigma(z_i) = \frac{1}{1+e^{-z_i}}$ converts the logits into confidence scores between 0 and 1, filtering the salient bboxes with threshold τ, as follows: $\mathbf{B} = \{B_i | \sigma(z_i) \geq \tau\} = \{B_i | z_i \geq \tau_z\}$, where $\tau_z = \sigma^{-1}(\tau)$. The final count is $|\mathbf{B}|$. Our goal is to derive a gradient from $\mathcal{D}$ that can effectively increase/decrease $|\mathbf{B}|$ as needed. Our approach is to first, define a differentiable function $f : \mathbf{z} \in \mathbb{R}^n \mapsto \mathbb{N}$ that extracts the count from the logits $\mathbf{z}$, and second, transform f so its gradient is more amenable to convergence (a step we find crucial), arriving at critic $\mathcal{L}_{\text{D2D}}$.

Extract the count via f. Counting is discrete in nature, but we convert it into a continuous, differentiable one by borrowing techniques from logistic regression for binary classification, which optimizes the steepness and transition threshold of a sigmoid-curve for discrete 0/1 prediction. We convert each logit into a soft binary indicator of whether to "count" the corresponding bbox, by applying to each logit a steep sigmoid curve with transition threshold τ_z and steepness coefficient β, with the final differentiable count formulated as a sum of sigmoids: $f_{\beta,\tau_z}(\mathbf{z}) = \sum_{i=1}^{n} \sigma(\beta \cdot (z_i - \tau_z))$. The next challenge is to make this gradient-friendly.

Transform f to effectively handle over/under-generation. An effective critic provides a strong gradient signal above/below τ_z (our domain of interest) to push logits below or beyond the threshold as needed to erase/add objects in response to over/under-generation. To illustrate, in over-generated cases, the goal is to prioritize decreasing logits above τ and allowing some flexibility for those already below τ. However, by nature of its sigmoidal shape, f has significant plateauing (i.e., weak gradient signals) above and below τ. To improve the gradient steepness in our domain of interest, we scale each sigmoid output by the corresponding logit (Eq. 1), arriving at $\mathcal{L}_{\text{D2D}}$. Appendix T compares other scaling formulations. At inference, we use $\nabla\mathcal{L}_{\text{D2D}}$ to optimize the generated image.[2] Extending this to prompts with $m > 1$ object classes, we update Eq. 1 to correct each bbox's largest logit, while minimizing all others. Details in Appendix G.

$$\mathcal{L}_{\text{D2D}} = \begin{cases} \sum_{i=1}^{n} \sigma(\beta \cdot (z_i - \tau_z)) \cdot (z_i - \tau_z), & \text{if } f_{\beta,\tau_z} > N \\ \sum_{i=1}^{n} \sigma(\beta \cdot (\tau_z - z_i)) \cdot (\tau_z - z_i), & \text{if } f_{\beta,\tau_z} < N \end{cases} \tag{1}$$

3.2 The Latent Modifier Network (LMN)

Given our count critic, we now turn to the learning method used to optimize this objective. Motivated by the presence of meaningful priors in the initial noise, previous works [2, 28] have used generic alignment metrics to tune the initial noise directly. Building on this motivation, we propose the Latent Modifier Network (LMN), a test-time tunable module whose output is mixed with the original noise to determine the optimal initial noise and whose weights are steered using $\mathcal{L}_{\text{D2D}}$.

Given initial noise $\mathbf{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$, $\mathbf{x}_T \in \mathbb{R}^d$ and prompt p that requests N counts of an object of class C, one-step T2I model G_θ generates image I. Our goal is to find an optimal $\mathbf{x}_T^*$ that produces an image I^* with exactly N of the specified object. To achieve this, we introduce a tunable Latent Modifier Network (LMN) M_ϕ: a small, 3-layer perceptron, between the initial random latent and T2I model (Fig. 3), with input/output dimensions equal to that of the initial latent and whose output dictates how to update $\mathbf{x}_T$. As shown in Eq. 2, the new latent is a weighted sum of $\mathbf{x}_T$ and $M_\phi(\mathbf{x}_T)$, with mixing weight $w = 0.2$. Compared to tuning the initial latent directly, the LMN composes a relatively larger parameter space and introduces an additional form of regularization by preserving a consistent Gaussian component sourced from the original latent throughout all iterations. At inference-time, we tune ϕ using $\nabla\mathcal{L}_{\text{D2D}}$ with the goal of correcting the initial noise, and thereby the numeracy, as described in the following section.

$$\mathbf{x}'_T = w \cdot \mathbf{x}_T + (1 - w) \cdot M_\phi(\mathbf{x}_T). \tag{2}$$

3.3 Optimization

The goal is to find the optimal set of parameters ϕ that minimizes the error between the generated and requested count, as follows: $\phi^* = \arg\min_\phi |\mathcal{D}(G_\theta(\mathbf{x}'_T)) - N|$.

[2] Unless otherwise noted, we use f to perform early-stopping once the requested count is met.

Since detector $\mathcal{D}$ is non-differentiable, we leverage $\mathcal{L}_{\text{D2D}}$ to optimize ϕ iteratively, rendering our final update rule Eq. 3 with regularization term $\mathcal{L}_{\text{reg}}$, learning rate η, and weights α and λ.

$$\phi \Leftarrow \phi - \eta \nabla (\alpha \mathcal{L}_{\text{D2D}} + \lambda \mathcal{L}_{\text{reg}}). \tag{3}$$

We adaptively rescale the loss to address exploding gradients that may be encountered due to the large number of tunable parameters. During numeracy optimization, we apply a variant of the regularization term used in ReNO [2], using the negative log-likelihood of the norm of $\mathbf{x}'_T$ as follows: $\mathcal{L}'_{\text{reg}} = ||\mathbf{x}'_T||^2/2 - (d-1) \cdot \log(||\mathbf{x}'_T||)$. We use $\mathcal{L}_{\text{reg}} = (a\mathcal{L}'_{\text{reg}} + c)^{10}$, with scaling coefficient a and shift constant c. Full algorithm in Appendix E.

ϕ initialization and calibration. To give M_ϕ a good starting point (i.e., initialize the network's initial output distribution to Gaussian), we propose a short, pre-inference alignment stage to be performed one time per base model using only the regularization term. Specifically, we train M_ϕ on 100 different randomly sampled latents ($\mathbf{x}_T$) for 200 epochs each (Algorithm 2 in the appendix). At inference-time, given a new, randomly sampled $\mathbf{x}_T$ the network has never seen, we introduce a ~0.2-second calibration phase to allow the network to adapt to the new input, using only the regularization term. Afterward, we leverage both *D2D* and regularization terms.

4 Experiments and Analysis

4.1 Experimental setup

Benchmarks. Our main experimental setting of single-object, low-density prompts leverages two benchmarks, CoCoCount [3] and D2D-Small. D2D-Small is a set of 400 prompts created using 40 countable objects from COCO [34] with counts ranging from 1-10 and a prompt template adapted from [35]: "A realistic photo of a scene with [count] [object]." CoCoCount consists of 200 prompts from 20 COCO classes and requested counts roughly equally split among 2, 3, 4, 5, 7, and 10. Experiments on multi-object or high-density prompts were performed on D2D-Multi (400 prompts with two objects sampled from 40 countable COCO classes, with $N_1, N_2 \leq 10$, and following the template: "A realistic photo of a scene with [count] [object] and [count] [object]") and D2D-Large (similarly constructed with counts 11-20).

Base models. We apply *D2D* to three one-step models: SDXL-Turbo [1], SD-Turbo [1], and Pixart-DMD [9]. SDXL-Turbo and SD-Turbo, respectively distilled from SDXL [6] and SD2.1 [7], have U-Net backbones [26]. Pixart-DMD, distilled from Pixart-α [8], has a DiT backbone [27]. We also benchmark *D2D* against larger base models FLUX-schnell [20] and SD3.5 Large Turbo [21].

Comparison of numeracy enhancement methods. We compare *D2D* against count-correction baselines **(1) Make It Count** [3] (multi-step), which uses attention-based mechanisms to identify/correct object layout via updates to the intermediate latents, **(2) Counting Guidance** [4] (multi-step), which uses the auxiliary counting model RCC [24] to correct the predicted noises, and **(3) TokenOpt** [5], a one-step method which injects a count token into the prompt and tunes it using CLIP-Count [25]. We run each baseline following its original experiment setup. Importantly, Make It Count is an SDXL-based method; Counting Guidance (original experiments, codebase) is primarily centered on SD1.4; and TokenOpt is built on SDXL-Turbo, so we evaluate Make It Count on SDXL, Counting Guidance on SD1.4, and TokenOpt on SDXL-Turbo. Furthermore, Make It Count addresses the low-density, single-object setting and TokenOpt addresses the single-object setting, so we only evaluate Make It Count on CoCoCount and D2D-Small and TokenOpt on CoCoCount and D2D-Small/Large.

Measuring the impact of the *D2D* critic. We demonstrate *D2D* on detectors OWLv2 [32] (open-vocabulary, robust) and YOLOv9 [36] (high-throughput, trained on COCO [34] objects). We expect a small accuracy-cost tradeoff (i.e., OWLv2 enables superior numeracy with greater computational overhead; YOLOv9 yields slightly lower numeracy but faster inference). We further compare our detector-based critic against regression-based ones (i.e., RCC [24], CLIP-Count [25], CounTR [33]).

Measuring the impact of the Latent Modifier Network. The most relevant prior initial noise optimization work is ReNO [2], a framework for one-step T2I models that uses the combined gradient of multiple image quality and prompt-image alignment metrics (ImageReward [15], PickScore [37], HPSv2 [38], CLIPScore [39]) to optimize semantic alignment and image quality. Instead of tuning an LMN, ReNO directly modifies the initial latent over 20-50 iterations, with a regularization loss term to keep the noise within the initial distribution and gradient clipping to prevent gradient explosion.

Table 1: **Quantitative results.** *D2D* outperforms all baselines across all four benchmarks (CoCoCount [3] and D2D-Small/Multi/Large), even generalizing across detector variants OWLv2 [32] and YOLOv9 [36]. *D2D* with YOLOv9 on base model SDXL-Turbo is in ***bold italics*** to show that while it outperforms all baselines, it is second to using OWLv2. The higher-performing OWLv2 detector is used in all subsequent experiments on SD-Turbo and Pixart-DMD. Standard deviations indicate the significance of our findings. Base models with no post-enhancement highlighted in gray. Avg. over four seeds.

Method	CoCoCount	D2D-Small	D2D-Multi	D2D-Large
SDXL [6]	24.88 ±1.70	16.06 ±1.86	2.44 ±0.59	1.44 ±0.38
+ Make It Count [3]	46.75 ±2.10	30.00 ±1.93	—	—
SDXL-Turbo [1]	27.38 ±2.69	20.31 ±1.95	2.12 ±0.83	2.56 ±0.55
+ ReNO [2]	41.88 ±1.03	27.50 ±0.68	5.31 ±0.38	4.69 ±1.25
+ TokenOpt [5]	35.12 ±0.75	23.31 ±1.66	—	3.94 ±0.72
+ *D2D* w/ YOLOv9 (Ours)	***52.75*** ±1.55	***36.69*** ±2.40	***6.25*** ±1.77	***7.50*** ±1.06
+ *D2D* w/ OWLv2 (Ours)	**55.62** ±2.72	**43.69** ±2.36	**9.81** ±0.97	**9.94** ±1.57
SD2.1 [7]	32.75 ±1.32	24.75 ±2.85	4.81 ±1.23	2.94 ±0.75
SD1.4 [7]	27.62 ±4.11	16.69 ±2.59	2.81 ±0.31	2.12 ±0.32
+ Counting Guidance [4]	28.38 ±1.11	17.12 ±1.69	3.38 ±1.16	1.88 ±0.60
SD-Turbo [1]	20.88 ±3.07	15.31 ±0.87	2.56 ±0.83	3.19 ±1.18
+ ReNO [2]	43.38 ±3.47	32.06 ±0.99	8.94 ±1.76	4.25 ±1.14
+ *D2D* w/ OWLv2 (Ours)	**48.38** ±3.09	**39.44** ±2.37	**10.75** ±1.06	**11.44** ±1.98
Pixart-α [8]	19.62 ±1.03	14.00 ±1.08	1.31 ±0.75	1.81 ±0.66
Pixart-DMD [9]	38.12 ±2.32	27.88 ±1.51	6.25 ±0.46	3.19 ±0.62
+ ReNO [2]	44.75 ±1.44	37.25 ±1.70	9.44 ±0.75	4.75 ±0.74
+ *D2D* w/ OWLv2 (Ours)	**53.25** ±2.40	**41.25** ±2.81	**13.31** ±1.36	**7.62** ±1.18

Though its use of human-preference reward models does improve numeracy relative to the base model, there remains a gap between using such generic objectives and our count-tailored critic (Tab. 1). A key difference between our method and ReNO's is our introduction of the LMN, which provides an additional opportunity for regularization via mixing and expands the tunable parameter space. To assess the impact of introducing the LMN, we compare our initial noise optimization method with ReNO's, controlling for the loss by swapping out their human-preference models for our *D2D* critic.

Evaluation. Following similar evaluation protocols [3, 4, 5], we use CountGD [40], a SOTA counting model built on detector GroundingDINO [41], to count the generated objects and compute the proportion of correct images (see Appendix Q for CountGD's counting accuracy compared to other regression/detector-based models). Like [2], we analyze image-quality/prompt alignment with human-preference-trained models (ImageReward [15], PickScore [37], HPSv2 [38]), and CLIPScore [39].

Implementation details. Our main experiments were completed on an L40. For Make It Count [3] which requires $>$ 50 GB, we used an A100. Although $\mathcal{L}_{\text{D2D}}$ is piecewise, optimization is generally stable due to several implementation choices. We use momentum, gradient clipping, and reduce the learning rate as needed when the soft count approaches the target (N) to help prevent oscillation near the decision boundary. If the count reaches N, we early-stop without further backpropagation. Average inference time is between 11 and 21 seconds. Additional details in Appendix F.

4.2 Numeracy improvements

Tab. 1 shows our main *D2D*-to-baseline comparisons. Baseline evaluations illustrate that though the prompt setting is relatively simple, generating accurate counts remains challenging. On numeracy, *D2D* consistently outperforms baselines across low-density, single-object, multi-object, and high-density prompts, across base models with U-Net and DiT backbones. On SDXL-Turbo, we demonstrate that performance boosts from *D2D* generalize across OWLv2 and YOLOv9 detector backbones (i.e., the detector used to compute $\mathcal{L}_{\text{D2D}}$), with a small accuracy-cost tradeoff as expected (Fig. 7). The robust OWLv2 detector yields higher numeracy with slightly more overhead, while the real-time YOLOv9 detector yields slightly lower (but still high) numeracy with faster inference (in all other experiments, we use the higher-performing OWLv2 backbone unless otherwise noted).

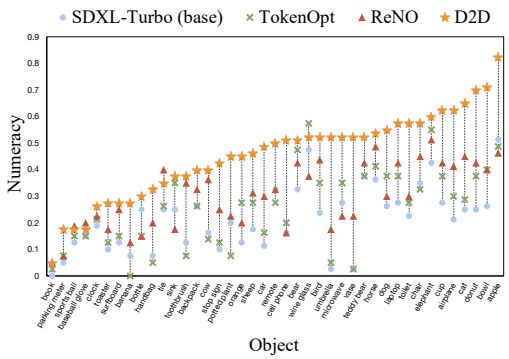


Figure 4: ***D2D* improves numeracy on the majority of the 41 objects in CoCoCount and D2D-Small.** Evaluated against ReNO [2] and TokenOpt [5] on base SDXL-Turbo. Avg. over four seeds.

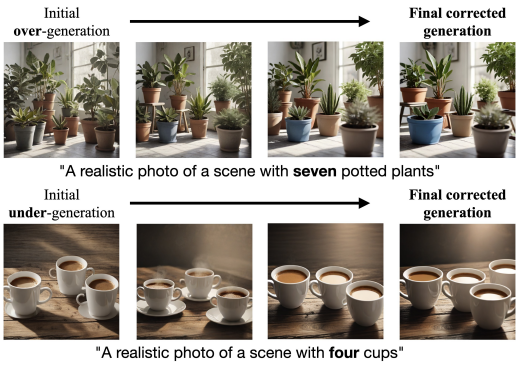


Figure 5: ***D2D* effectively corrects over and under-generation**. The initial image contains at least two more potted plants/one fewer cup than requested. *D2D* iteratively corrects them, arriving at seven potted plants/four cups, as requested.

Initial generation	Over	Under	Correct
TokenOpt [5]	13.28	25.24	69.92
ReNO [2]	23.32	25.11	62.19
D2D w/ OWLv2	**40.13**	**41.83**	**72.57**

Table 2: **Given the same initial conditions, *D2D* effectively corrects both over and under-generation.** We report the correction rate of initial over/under-generations, as well as the proportion of correct generations that were maintained. On SDXL-Turbo, across CoCoCount [3] and D2D-Small. Avg. over four seeds.

Method	Numeracy
FLUX-schnell [20] (1-step)	46.0
FLUX-schnell [20] (4-step)	47.5
SD3.5 Large Turbo [21] (4-step)	48.0
SDXL-Turbo [1]	30.0
SDXL-Turbo [1] + *D2D*	**55.5**

Table 3: ***D2D* on a weaker base model outperforms stronger base models.** Integrating *D2D* with a smaller model like SDXL-Turbo outperforms stronger base models such as SD3.5 Large Turbo and FLUX-schnell. On CoCoCount [3], seed=0.

Additionally, *D2D* effectively complements baselines, boosting numeracy across all four benchmarks when used in combination with TokenOpt or ReNO (Tab. 15 in appendix). For example, applying *D2D*-optimized seeds to TokenOpt improves numeracy by 13.63% points, relative to TokenOpt's baseline performance (from 35.12% to 48.75%) on CoCoCount.

Improved numeracy on multi-object/high-density prompts. *D2D* maintains relative improvement even in the more challenging multi-object/high-density settings. Nevertheless, the accuracy drop from low-density benchmarks to D2D-Large illustrates the remaining challenge of correctly generating large counts (e.g., base SDXL-Turbo: from 43.69% on D2D-Small to 9.94% on D2D-Large).

$\mathcal{L}_{\text{D2D}}$ effectively boosts numeracy across all classes. *D2D* improves numeracy over the base model across all 41 object categories in CoCoCount and D2D-Small (Fig. 4), spanning a large variety (e.g., apples, elephants, cars, etc.). Upon applying *D2D* to SDXL-Turbo, umbrella and vase are the two classes that see the most improvement, each jumping from 2.50% (SDXL-Turbo base) to 52.50% (*D2D*) accuracy. Wine glass and bottle, both of which are (semi)transparent objects, are among the classes that see the least improvement (47.50% to 52.50% and 25.00% to 30.00% accuracy, respectively), which may suggest a future direction where detectors are fine-tuned on more difficult classes, or similar, with the purpose of generating highly-tailored scenes of objects.

***D2D* best handles over and under-generation.** Tab. 2 breaks down results by the numeracy of the initial generation I, illustrating how well different methods are able to *correct* over/under-generation while *maintaining* the numeracy of already-correct images. We compare TokenOpt, ReNO, and *D2D* on base model SDXL-Turbo, across benchmarks CoCoCount and D2D-Small. *D2D* has the highest correction rate, correcting 40.13% of over-generations and 41.83% of under-generations, which is at least 16% points over the baselines, while maintaining 72.57% of already-correct generations, which is also more than both TokenOpt and ReNO. Fig. 5 illustrates *D2D*'s iterative correction process on two sample prompts, going from at least nine potted plants to the requested seven and from three cups to the requested four. Additional qualitative examples in Appendix R.

τ	0.1	0.2	0.5	0.8
CountGD [40]	51.50	**55.50**	43.50	32.50

β	1	10	100	300	400
CountGD [40]	43.00	40.00	52.00	**55.50**	52.50

Table 4: **Ablation study on key hyperparameters τ and β.** Detector threshold $\tau = 0.2$ is optimal. A lower τ (which counts low-confidence bboxes) and higher τ (which potentially discards actually-legitimate bboxes) result in drops in numeracy. Steepness coefficient $\beta = 300$ is optimal. Tested on CoCoCount [3], seed=0.

Count Critic	CoCoCount	D2D-Small	D2D-Multi	D2D-Large
RCC	37.75	26.38	—	4.25
CounTR	38.38	25.62	—	5.31
CLIP-Count	40.00	25.88	5.19	6.38
f (OWLv2)	32.00	20.75	3.06	3.38
$\mathcal{L}_{\text{D2D}}$ (OWLv2)	**55.62**	**43.69**	**9.81**	**9.94**

Table 5: **Comparison of count critics. Using f alone yields weak gradients and poor convergence, but $\mathcal{L}_{\text{D2D}}$'s logit-scaled form enables effective optimization and higher numeracy.** $\mathcal{L}_{\text{D2D}}$ outperforms RCC [24], CounTR [33], and CLIP-Count [25]. On SDXL-Turbo. Avg. over four seeds.

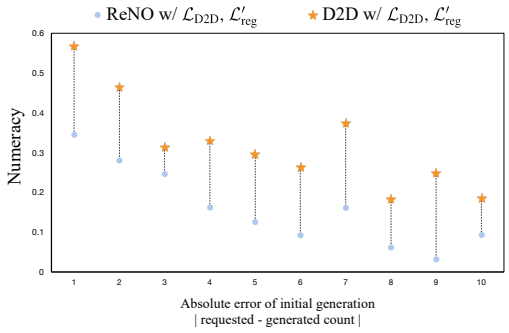


Figure 6: **The LMN consistently boosts numeracy.** Breakdown of numeracy by absolute error in the initial generation. To compare optimizing via LMN vs. direct tuning of the initial latent, we benchmark *D2D* against ReNO [2] using $\mathcal{L}_{\text{D2D}}$ and $\mathcal{L}'_{\text{reg}}$ for both, controlling for the number of iterations tuned. On SDXL-Turbo, across CoCoCount [3] and D2D-Small. Avg. over four seeds.

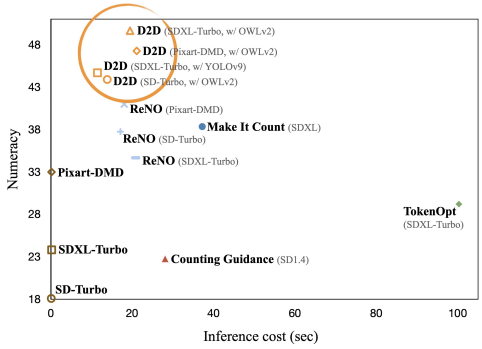


Figure 7: **Numeracy vs. inference cost by method.** Across base models (SDXL-Turbo, SD-Turbo, Pixart-DMD) and detectors (OWLv2 [32], YOLOv9 [36]), *D2D* scores in the top left (i.e. both high-numeracy and low-cost). *D2D* w/ YOLOv9 is even more compute-efficient than w/ OWLv2. Base model/detector in gray. Avg. on CoCoCount [3], D2D-Small, over four seeds.

***D2D* vs. larger models.** Tab. 3 compares the numeracy of newer, larger models like FLUX-schnell [20] and SD3.5 Large Turbo [21] against *D2D* on the smaller base model SDXL-Turbo [1]. Even on a weaker base model (30.0% accuracy), *D2D* (55.5%) yields a **+9.5%**, **+8%**, **+7.5% point improvement** over FLUX-schnell (1-step: 46.0%; 4-step: 47.5%) and SD3.5 Large Turbo (48.0%).

4.3 Analysis of *D2D* critic

***D2D* vs. regression-based counters.** Tab. 5 compares our critic against existing regression-based ones and confirms the importance of $\mathcal{L}_{\text{D2D}}$'s convergence-friendly form. Across all four benchmarks, our detector-based critic outperforms regression-based methods RCC, CLIP-Count, and CounTR on numeracy (e.g., ours reaches 55.62% when the max score reached by any regression-based one is 40% on CoCoCount). Notably, $\mathcal{L}_{\text{D2D}}$ outperforms the others even on the high-density benchmark D2D-Large, though regression-based methods outperform detectors in the non-generative, counting setting (Fig. 2b). Notably, f_{β,τ_z} yields the lowest numeracy, which indicates that though it composes the mathematical backbone of $\mathcal{L}_{\text{D2D}}$, its sigmoid saturation leads to weak gradients and poor convergence when used directly as the critic (Tab. 5). In contrast, $\mathcal{L}_{\text{D2D}}$'s logit-scaled formulation preserves gradient magnitude and improves convergence, making it crucial to *D2D*'s performance.

Impact of key hyperparameters τ and β. We report our key hyperparameter studies on values for τ (detector threshold) and β (steepness coefficient), conducted using base model SDXL-Turbo on benchmark CoCoCount (seed=0) on an A6000/L40. Results (Tab. 4) show that $\tau = 0.2$, $\beta = 300$ are optimal, which we use in all our main experiments.

4.4 Analysis of the Latent Modifier Network

Impact of the LMN on numeracy. Next, we assess the impact of introducing the LMN, a module whose output is mixed with the original noise to arrive at the optimal noise, by comparing our method with ReNO's, controlling for the optimization objectives used ($\mathcal{L}_{\text{D2D}}$, $\mathcal{L}'_{\text{reg}}$) and number of iterations tuned. Fig. 6 shows the LMN provides robust numeracy gains even as the magnitude of the initial error increases; numeracy jumps 10% points on CoCoCount and D2D-Small from 43.25% to 53.88% and from 32.00% to 42.44%, respectively (Tab. 12 in appendix).

Optimal balance between count correction and regularization. w is the mixing weight that determines the extent to which the original noise $\mathbf{x}_T$ should be preserved in the corrected noise $\mathbf{x}'_T$. When w is small (e.g., $w = 0$) the final noise consists largely of the count correction signal from M_ϕ, at the risk of steering away from the initial noise distribution. When w is large (e.g., $w = 1$), the initial noise is largely preserved due to heavy regularization (i.e., minimal count correction). The key is to find a w that is high enough to preserve image quality and low enough to improve numeracy. As shown in Tab. 23, $w = 0$ is numeracy-wise the best, followed by $w = 0.2$, but results in patchy visual artifacts, relative to $w = 0.2$ (Fig. 15). Hence, we use $w = 0.2$ in our main experiments.

***D2D* is both high-numeracy and low-cost.** Our method averages between 11 and 21 seconds in inference cost, compared to counting baselines, which average upwards of 28 to 100 seconds (Fig. 7). ImageReward, PickScore, HPSv2, and CLIPScore metrics in Fig. 9 show *D2D*'s image quality and overall prompt alignment is comparable to counting baselines and even surpasses multi-step baselines in many cases, including the layout control-based Make It Count (MIC). For example, SDXL-Turbo + *D2D* (OWLv2) yields ImageReward 0.51 (MIC: 0.30), PickScore 21.98 (MIC: 21.48), and HPSv2 0.28 (MIC: 0.26) on D2D-Small.

5 Conclusion and Discussion

In this work, we address the challenge of correcting numeracy in generation. We identify a central limitation of previous methods, specifically their reliance on differentiable, regression-based counting models as critics. We propose *D2D*, a novel way to convert more robust detectors into differentiable count critics, which we then use to optimize the initial noise at inference-time to improve numeracy. Our method yields the highest numeracy across prompt scenarios, including low-density, single-object, multi-object, high-density settings, effectively correcting both over and under-generation, with minimal additions to temporal overhead (i.e., the additional inference time introduced beyond the base model's generation time) and minimal degradation in image quality.

Limitation and future directions. While our method exhibits significant improvements in numeracy, high-density scenarios remain challenging. Given regression-based methods are more appropriate in this setting, a future direction may explore how to adapt them into the generative setting. *D2D* is not a layout-control approach and so is limited in more fine-grained control (e.g., object placement). But future directions may explore using *D2D* to perform more complex tasks, such as object positioning and attribute binding, by leveraging detectors that can robustly work with prompts specifying objects and associated attributes.

Acknowledgements

This research was partially supported by an Amazon Research Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Amazon. YZ also acknowledges the support of the French National Research Agency (ANR) via the "MiMoGC" JCJC grant (ANR-26-CE23-7469). We are also grateful to Princeton Research Computing for compute resources.

References

[1] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 87–103, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73016-0.

[2] Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. ReNO: Enhancing one-step text-to-image models through reward-based noise optimization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 125487–125519. Curran Associates, Inc., 2024.

[3] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13242–13251, June 2025.

[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, 2025. doi: 10.1109/WACV61041.2025.00097.

[5] Oz Zafar, Yuval Cohen, Lior Wolf, and Idan Schwartz. Detection-driven object count optimization for text-to-image diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1885–1894, March 2026.

[6] 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. In *The Twelfth International Conference on Learning Representations*, 2024.

[7] 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 (CVPR)*, pages 10684–10695, June 2022.

[8] Junsong Chen, Jincheng YU, Chongjian GE, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

[9] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-Σ: Weak-to-strong training of diffusion transformer for 4K text-to-image generation. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 74–91, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73411-3.

[10] Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *The Twelfth International Conference on Learning Representations*, 2024.

[11] Chaofeng Chen, Annan Wang, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Enhancing diffusion models with text-encoder reinforcement learning. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 182–198, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72698-9.

[12] Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8941–8951, June 2024.

[13] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8228–8238, June 2024.

[14] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[15] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 15903–15935. Curran Associates, Inc., 2023.

[16] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 79858–79885. Curran Associates, Inc., 2023.

[17] Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7280–7290, October 2023.

[18] Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. CFG++: Manifold-constrained classifier free guidance for diffusion models. *CoRR*, abs/2406.08070, 2024.

[19] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Trans. Graph.*, 42(4), July 2023. ISSN 0730-0301. doi: 10.1145/3592116.

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

[21] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=FPnUhsQJ5B`.

[22] Manoj Acharya, Kushal Kafle, and Christopher Kanan. TallyQA: Answering complex counting questions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8076–8084, Jul. 2019. doi: 10.1609/aaai.v33i01.33018076.

[23] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3394–3403, June 2021.

[24] Michael Hobley and Victor Prisacariu. Learning to count anything: Reference-less class-agnostic counting with weak supervision. *arXiv preprint arXiv:2205.10203*, 2022.

[25] Ruixiang Jiang, Lingbo Liu, and Changwen Chen. CLIP-Count: Towards text-guided zero-shot object counting. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pages 4535–4545, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701085. doi: 10.1145/3581783.3611789.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[27] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, October 2023.

[28] Ruoyu Wang, Huayang Huang, Ye Zhu, Olga Russakovsky, and Yu Wu. The silent assistant: NoiseQuery as implicit guidance for goal-driven image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17618–17628, October 2025.

[29] Anh-Dung Dinh, Daochang Liu, and Chang Xu. Rethinking conditional diffusion sampling with progressive guidance. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 42285–42297. Curran Associates, Inc., 2023.

[30] Zakaria Patel and Kirill Serkh. Enhancing image layout control with loss-guided diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3916–3924, 2025. doi: 10.1109/WACV61041.2025.00385.

[31] Ruisu Zhang, Yicong Chen, and Kangwook Lee. Improving CLIP counting accuracy via parameter-efficient fine-tuning. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

[32] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 72983–73007. Curran Associates, Inc., 2023.

[33] Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. CounTR: Transformer-based generalised visual counting. In *British Machine Vision Conference (BMVC)*, 2022.

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

[35] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. LLM-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. Featured Certification.

[36] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. YOLOv9: Learning what you want to learn using programmable gradient information. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXXI*, pages 1–21, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72750-4. doi: 10.1007/978-3-031-72751-1_1.

[37] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-Pic: An open dataset of user preferences for text-to-image generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 36652–36663. Curran Associates, Inc., 2023.

[38] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

[39] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595.

[40] Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. CountGD: Multi-modal open-world counting. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 48810–48837. Curran Associates, Inc., 2024.

[41] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Computer Vision – ECCV 2024*, pages 38–55, 2025.

[42] Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. SANA-Sprint: One-step diffusion with continuous-time consistency distillation. *arXiv preprint arXiv:2503.09641*, 2025.

[43] 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 (ICCV)*, pages 16765–16775, October 2025.

[44] Tianyi Wei, Dongdong Chen, Yifan Zhou, and Xingang Pan. Enhancing MMDiT-based text-to-image models for similar subject generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

[45] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G. Hauptmann. DecideNet: Counting varying density crowds through attention guided detection and density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[46] Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*, chapter Reliability, pages 277–360. SAGE Publications, Inc., Thousand Oaks, California, fourth edition, 2019. doi: 10.4135/9781071878781.n13.

[47] Zhang Xiaofeng, Aaron Courville, Michal Drozdzal, and Adriana Romero-Soriano. The intricate dance of prompt complexity, quality, diversity and consistency in T2I models. In *International Conference on Learning Representations*, volume 2026, pages 50001–50038, 2026.

[48] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[49] Dan Friedman and Adji Bousso Dieng. The Vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=g97OHbQyk1`.

[50] Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Inference-time alignment of diffusion models with direct noise optimization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=JpbqiD7n9r.

Appendix

Appendix A gives a high-level summary comparing detection vs. regression-based numeracy critics in the generative setting. Appendix B validates our method via human pilot study. Appendix C compares *D2D* to a roughly compute-matched, best-of-N baseline. Appendix D gives additional details and discussion on the numeracy-speed tradeoff. Appendix E details the algorithms for pre-inference alignment of the Latent Modifier Network (LMN) and inference-time, numeracy optimization. Appendix F describes additional implementation details, including hyperparameters used for the LMN architecture (Appendix F.1), core *D2D* formulation (Appendix F.2), and optimization (Appendix F.3). Appendix G discusses the implementation details of extending *D2D* to multi-object scenarios. Appendix H applies *D2D* to FLUX-schnell [20], a larger base model, and SANA-Sprint [42], a recent one-step T2I model. Appendix I shows quantitatively how the LMN improves numeracy. Appendix J shows *D2D* has image quality and overall prompt alignment comparable to counting baselines, even surpassing multi-step baselines in many cases (Appendix J.1). Also included are per-object fidelity results (Appendix J.2). Appendix K shows the performance improvements that result from using *D2D* to complement existing baselines. Appendix L shows a breakdown of numeracy by total density on D2D-Multi. Appendix M includes further deep-dive analysis on the impact of object size, detector confidence, and requested count on numeracy. Appendix N shows how *D2D* handles overlapping detections. Appendix O shows *D2D*'s numeracy improvements over the more recent YOLO-COUNT [43] work. Appendix P compares *D2D* with method EnMMDiT [44]. Appendix Q shows CountGD's baseline counting performance, relative to other regression/detector-based counters. Additional qualitative results are illustrated in Appendix R, including on more free-form prompts (Appendix R.1). Appendix S includes results from additional ablation experiments, including studies on the value of the mixing weight w used to mix the LMN output with the original noise (Appendix S.1); variants of the LMN architecture (Appendix S.2); the necessity of pre-inference alignment (Appendix S.3) and inference-time calibration (Appendix S.4) of M_ϕ; and regularization term variants (Appendix S.5). Appendix T compares alternative formulations of the *D2D* loss. Appendix U shows a sample numeracy optimization trajectory for a single prompt, comparing the raw values of the *D2D* loss $\mathcal{L}_{\text{D2D}}$ and f over iterations. Appendix V details methods to preprocess prompts for selective activation of the *D2D* method. Appendix W demonstrates an extension of the *D2D* critic for multi-step diffusion models, using an existing multi-step-oriented initial noise optimization framework. Appendix X presents analysis on the impact of *D2D* on the diversity of generated images. Appendix Y describes broader impacts. Appendix Z describes our LLM usage.

A Detectors vs. regression-based numeracy critics

Based on our analysis of the baseline counting accuracies of detectors compared to regression-based models (Fig. 2b), we observe that detectors are more robust, as also observed in previous works [31, 45]. Furthermore, this performance boost generalizes to the generative setting, because accurate count understanding is one of the key components of numeracy optimization. As for the question of why detectors are more robust in the first place, we refer to existing work [45], which indicates this may be due to detectors' more advanced localization ability. In the low-density setting, objects are more likely to be sparsely separated, so it becomes essential to localize each instance in order to obtain a count; detectors count explicitly by localizing instances first, whereas regression-based methods capture the density of the crowd of instances, generally without specific localization objectives. In the high-density setting, when object instances are too small and potentially occluded, localization may become more difficult, thereby reducing the accuracy of detector counts.

B Human pilot study

To validate *D2D* performance, we conducted additional human evaluation and multi-evaluator agreement studies, which confirm *D2D*'s performance boosts. During model development, we employed different detectors for *D2D*'s critic (OWLv2) and evaluator (CountGD) to avoid potential critic-evaluator leak. Human evaluation and automated YOLOv9/CountGD-based metrics agree that *D2D* outperforms counting baselines (Tab. 6).

Table 6: ***D2D* outperforms counting baselines,** as validated by human evaluators and YOLOv9/CountGD. On 50 randomly selected prompts from D2D-Small.

Method	Human	YOLOv9 [36]	CountGD [40]
SDXL [6]	0.10	0.10	0.10
SDXL-Turbo [1]	0.24	0.20	0.22
TokenOpt [5] (SDXL-Turbo)	0.26	0.24	0.22
Counting Guidance [4] (SD1.4)	0.12	0.16	0.24
Make It Count [3] (SDXL)	0.32	0.38	0.34
D2D w/ OWLv2 (SDXL-Turbo)	**0.40**	**0.40**	**0.42**

We conducted the pilot study with 22 volunteers from the author's network using 50 randomly selected prompts from D2D-Small across the six models in the table above (300 total images), with each image evaluated by at least two different participants and the rounded average of their responses used as the final score. Each volunteer was given a questionnaire of 30 randomly shuffled questions, each with an image and a short, free-response "How many [object class] are there in the image?" Importantly, we did not disclose which baseline generated each image. The collected data consists of images and corresponding counts; the images carry minimal risk. Below is the prompt we used:

Thank you so much for agreeing to participate in this short questionnaire. I really appreciate you! Below are 30 questions. Each question presents an image and a short prompt: "How many <object> are there in the image?" Based on the image, answer the question. Below is an example, to illustrate. How many backpacks are there in the image? [image] Example user response: 6. Continue to the next section to begin answering the questions. All answers must be integers, in numerical format. So 15 is a valid answer, but fifteen is not. 0 is also a possible answer. Emphasis: a single integer response for each question - no additional text. When the answer seems unclear, please make your best effort. Thank you!

Our method yields the lowest error between generated and requested counts, as computed from human scores (Tab. 7).

Table 7: ***D2D* yields the lowest mean absolute error.** On 50 randomly selected prompts from D2D-Small.

Method	(Human) mean |generated - requested count|
SDXL [6]	6.00
SDXL-Turbo [1]	3.42
TokenOpt [5] (SDXL-Turbo)	3.22
Counting Guidance [4] (SD1.4)	3.34
Make It Count [3] (SDXL)	3.88
D2D w/ OWLv2 (SDXL-Turbo)	**1.76**

Using Krippendorff's α [46] with the interval metric to measure inter-rater agreement (maximum value of 1), we compute a score of 0.924, indicating strong agreement.

C *D2D* vs. compute-matched best-of-N baseline

We additionally run experiments on an A100 (80 GB), comparing *D2D* against a roughly compute-matched, best-of-N baseline on CoCoCount and D2D-Small, using the OWLv2 detector to select the best-of-N image. Tab. 8 shows *D2D* outperforms best-of-N, with N=45 for CoCoCount and N=70 for D2D-Small.

Table 8: ***D2D* beats best-of-N baseline.** Numeracy followed by inference cost in seconds and compute-matched value of N for the baseline in parentheses. On SDXL-Turbo, seed=0, using A100 GPU.

Method	CoCoCount [3]	D2D-Small
Best-of-N	54.0 (22 sec, N=45)	33.5 (35 sec, N=70)
D2D w/ OWLv2	**59.0** (22 sec)	**41.0** (34 sec)

D *D2D* introduces minimal inference time overhead

Improving generation capabilities beyond the base model introduces tradeoffs, as seen in other T2I settings, such as quality–speed and quality/alignment-diversity [47]. *D2D* achieves this improvement with minimal overhead compared to existing count-correction methods, as shown in Tab. 9 below.

Table 9: ***D2D* introduces substantially less overhead than existing count-correction methods.** Base model inference time and additional overhead in seconds. On D2D-Small. Avg. over four seeds.

Method	Base inference time (s)	Additional overhead (s)
TokenOpt [5]	0.16 (SDXL-Turbo)	102.94
Make It Count [3]	7.68 (SDXL)	34.60
Counting Guidance [4]	1.52 (SD1.4)	26.87
D2D	0.16 (SDXL-Turbo)	**21.55**

Moreover, for applications where count correctness matters, this additional cost can be a worthwhile tradeoff: unlike subjective image quality metrics, numeracy is often a correctness criterion. Finally, *D2D* is backbone-flexible and does not require retraining, allowing it to directly benefit from future improvements in diffusion inference efficiency and faster generation pipelines.

E Algorithms

Algorithm 1 Inference

Input: Prompt p specifying N of object of class C; pre-trained Latent Modifier Network M_ϕ; latent dimension d; weight w, diffusion model G_θ; minimum # calibration iterations t_{min}; threshold value specifying "good enough" regularization τ_{reg}; counter f_{β,τ_z} and critic $\mathcal{L}_{\text{D2D}}$; Stage 1 (Calibration) learning rate η_{calib} and loss weight λ_{calib}; Stage 2 numeracy optimization learning rate η and loss weights α and λ; # tuning steps K.
Output: Optimal noise $\mathbf{x}_T^*$.

Stage 1: Calibrate M_ϕ to newly sampled $\mathbf{x}_T$.
resample $\leftarrow$`True`
while resample **do**
  Sample $\mathbf{x}_T \in \mathbb{R}^d \sim \mathcal{N}(0, \boldsymbol{I})$
  for $1 \leq t \leq K$ **do**
    $\mathbf{x}'_T = w \cdot \mathbf{x}_T + (1-w) \cdot M_\phi(\mathbf{x}_T)$
    Compute $\mathcal{L} = \lambda_{\text{calib}} \mathcal{L}'_{\text{reg}}$
    if $t \geq t_{\text{min}}$ and $\mathcal{L} \leq \tau_{\text{reg}}$ **then**
      resample $\leftarrow$ `False`
      break
    else
      $\phi \Leftarrow \phi - \eta_{\text{calib}} \nabla \mathcal{L}$
    end if
  end for
end while

Stage 2: Optimize numeracy.
for $t \leq$ epoch $\leq K$ **do**
  Compute $\mathcal{L}_{\text{reg}}$
  $I = G_\theta(\mathbf{x}'_T, p)$
  Compute f_{β,τ_z} and $\mathcal{L}_{\text{D2D}}$
  return if $f_{\beta,\tau_z} = N$ {if I is optimal, stop early}
  $\phi \Leftarrow \phi - \eta \nabla (\alpha \mathcal{L}_{\text{D2D}} + \lambda \mathcal{L}_{\text{reg}})$
  $\mathbf{x}'_T = w \cdot \mathbf{x}_T + (1-w) \cdot M_\phi(\mathbf{x}_T)$
end for

Algorithm 2 Pre-inference alignment stage (done once per model)

Input: Latent Modifier Network M_ϕ, initialized with random weights; latent dimension d; weight w; learning rate η, loss weight λ.
Output: Pre-trained M_ϕ with output aligned to Gaussian distribution.
Set seed = 1.
repeat 100 **times**
  Sample $\mathbf{x}_T \in \mathbb{R}^d \sim \mathcal{N}(0, \boldsymbol{I})$
  for $1 \leq$ epoch ≤ 200 **do**
    $\mathbf{x}'_T = w \cdot \mathbf{x}_T + (1-w) \cdot M_\phi(\mathbf{x}_T)$
    Compute $\mathcal{L} = \lambda \mathcal{L}'_{\text{reg}}$
    $\phi \Leftarrow \phi - \eta \nabla \mathcal{L}$
  end for
end

F Additional implementation details

F.1 Architecture of the Latent Modifier Network

The LMN is a three-layer perceptron, with input/output layers of size d (dimension of initial latent, $\mathbf{x}_T \in \mathbb{R}^d$), and two hidden layers of size 100 each. On base models SDXL-Turbo, SD-Turbo, and Pixart-DMD, the LMN has 3,303,384 tunable parameters.

F.2 *D2D* formulation

The core *D2D* hyperparameters are the detector threshold τ and steepness coefficient β, which determine the transition threshold and curvature of the sigmoid underlying $\mathcal{L}_{\text{D2D}}$, and the mixing weight w, which determines the ratio with which to combine the LMN output and original noise to obtain the optimal noise. We use $\tau = 0.2$, $\beta = 300$, and $w = 0.2$. Studies on hyperparameters τ and β are reported in Tab. 4 in Sec. 4.3 of the main text. Experiments on w are reported in Tab. 23 in Appendix S.1. Unlike the OWLv2 critic, for YOLOv9 we apply NMS filtering (threshold=0.7) prior to count summation.

F.3 Optimization

During pre-inference alignment and inference-time calibration, when we optimize only the regularization term, we use $\mathcal{L}'_{\text{reg}}$. At inference-time, during numeracy optimization, we use $\mathcal{L}_{\text{reg}}$ in conjunction with $\mathcal{L}_{\text{D2D}}$. Appendix S.5 reports ablation studies on using $\mathcal{L}'_{\text{reg}}$ vs. $\mathcal{L}_{\text{reg}}$ during numeracy optimization.

In the pre-inference alignment stage, we use learning rate $\eta = 10^{-4}$ and loss weight $\lambda = 0.01$. At inference-time (stage 1: calibration), we use minimum number of iterations $t_{\text{min}} = 70$; "good enough" threshold for determining when calibration is done $\tau_{\text{reg}} = -712.8$, learning rate $\eta_{\text{calib}} = 10^{-3}$, and loss weight $\lambda_{\text{calib}} = 0.01$. During stage 2 (numeracy optimization), we use learning rate $\eta = 5 \times 10^{-4}$ and loss weights $\alpha = 5$ and $\lambda = 10^{-4}$. During this stage, we use count-based, adaptive learning rate scheduling of η to ease steady convergence when the generated count approximates the requested count; we also adaptively rescale (i.e., increase) λ to counteract larger deviations from Gaussian (i.e., divergence).

For *D2D* experimental results with Pixart-DMD on benchmark D2D-Large, we use the base detector OWLv2, instead of f_{β,τ_z}, to perform early-stopping. For single-object scenarios, we set $K = 200$, except for *D2D* with Pixart-DMD on D2D-Large, for which we use $K = 400$. For the multi-object scenario, we use $K = 400$.

G Multi-object $\mathcal{L}_{\text{D2D}}$

In extending *D2D* to prompts with $m > 1$ object classes $\{C_j, 1 \leq j \leq m\}$, we have to note that every predicted bbox B_i now comes with m corresponding scores $z_i^{(1)}, z_i^{(2)}, ..., z_i^{(m)}$, the max of which, z_i^{max}, indicates its corresponding label. Only the bboxes where $z_i^{\text{max}} > \tau_z$ are counted. To correct over/under-generation, our approach then is to focus on appropriately adjusting the largest score z_i^{max} of each bbox, while minimizing all other (i.e., non-max) scores. This means minimizing the z_i^{max} of bboxes that correspond to over-generated classes and maximizing the z_i^{max} of those that correspond to under/correctly generated classes (to increase/maintain the count). We use roughly the same formulation as Eq. 1, with z_i replaced by the specific per-class, per-bbox logits z_i^j that need to be minimized or maximized (Eq. 4).

If $\mathbb{C}_{\geq}$ is the set of classes that are under-generated or correctly generated, then

$$\mathbb{S}_{\text{bbox}} = \{(i,j) | 1 \leq i \leq n \wedge j = (\arg\max_{j'} z_i^{j'}) \in \mathbb{C}_{\geq}\}$$

refers to the set of the bboxes corresponding to the under/correctly generated classes, along with their max logits. We compute the loss $\mathcal{L}_{\text{D2D}}$ like so:

$$\mathcal{L}_{\text{D2D}}^{\text{multi}} = \sum_{(i,j) \in \mathbb{S}_{\text{bbox}}} \sigma(\beta \cdot (\tau_z - z_i^j)) \cdot (\tau_z - z_i^j) + \sum_{(i,j) \notin \mathbb{S}_{\text{bbox}}} \sigma(\beta \cdot (z_i^j - \tau_z)) \cdot (z_i^j - \tau_z) \quad (4)$$

H *D2D* extensions on FLUX-schnell [20] and SANA-Sprint [42]

Table 10: ***D2D* also yields numeracy gains on FLUX-schnell, a larger base model.** On CoCo-Count [3], seed=0.

Method	Numeracy
FLUX-schnell [20] (1-step)	46.0
FLUX-schnell [20] (4-step)	47.5
FLUX-schnell [20] (1-step) + *D2D*	**57.0**

Table 11: ***D2D* also yields numeracy gains on SANA-Sprint, a recent advancement in one-step T2I.** Evaluated on 50 randomly sampled prompts from CoCoCount [3], seed=0.

Method	Numeracy
SANA-Sprint [42]	42.0
SANA-Sprint [42] + *D2D*	**62.0**

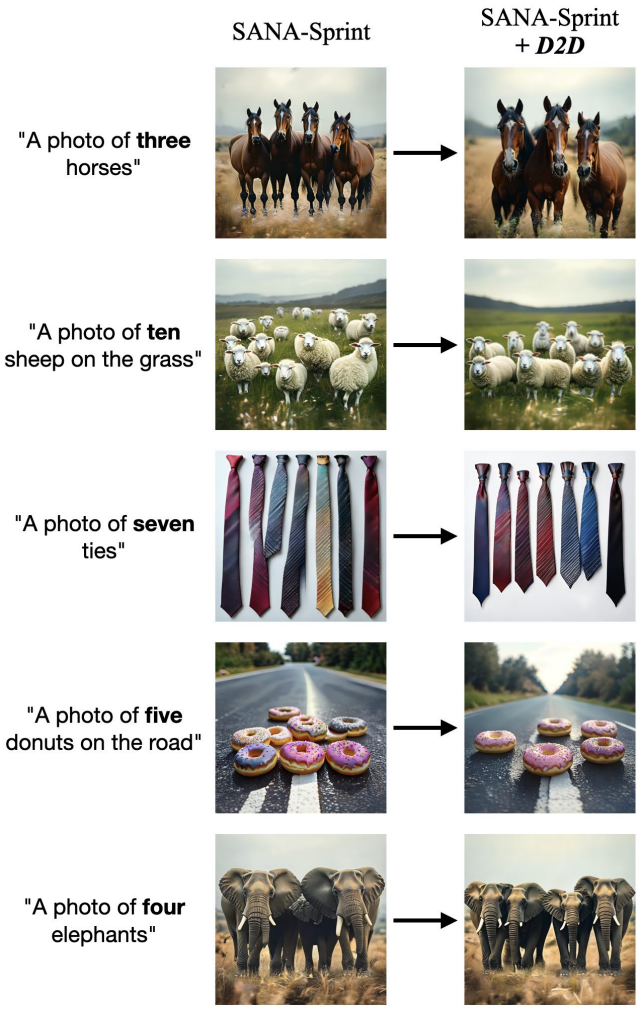


Figure 8: Qualitative results of *D2D* on SANA-Sprint [42].

I The Latent Modifier Network effectively corrects numeracy

Table 12: **The LMN boosts numeracy.** We compare *D2D* against ReNO [2] using $\mathcal{L}_{\text{D2D}}$ and $\mathcal{L}'_{\text{reg}}$ for both, controlling for the number of iterations tuned. We note boosts in numeracy, with comparable image quality. On CoCoCount [3] and D2D-Small. Base model SDXL-Turbo. Avg. over four seeds.

Method	CountGD [40] ↑		ImageReward [15] ↑		PickScore [37] ↑		HPSv2 [38] ↑		CLIPScore [39] ↑	
	CoCoCount	D2D-Small	CoCoCount	D2D-Small	CoCoCount	D2D-Small	CoCoCount	D2D-Small	CoCoCount	D2D-Small
ReNO [2] w/ $\mathcal{L}_{\text{D2D}}$, $\mathcal{L}'_{\text{reg}}$	43.25	32.00	1.04	0.45	23.25	21.98	0.296	0.281	32.81	31.79
D2D w/ $\mathcal{L}_{\text{D2D}}$, $\mathcal{L}'_{\text{reg}}$	**53.88**	**42.44**	1.08	0.52	23.28	21.99	0.299	0.282	32.77	31.71

J *D2D* on image quality and overall prompt alignment

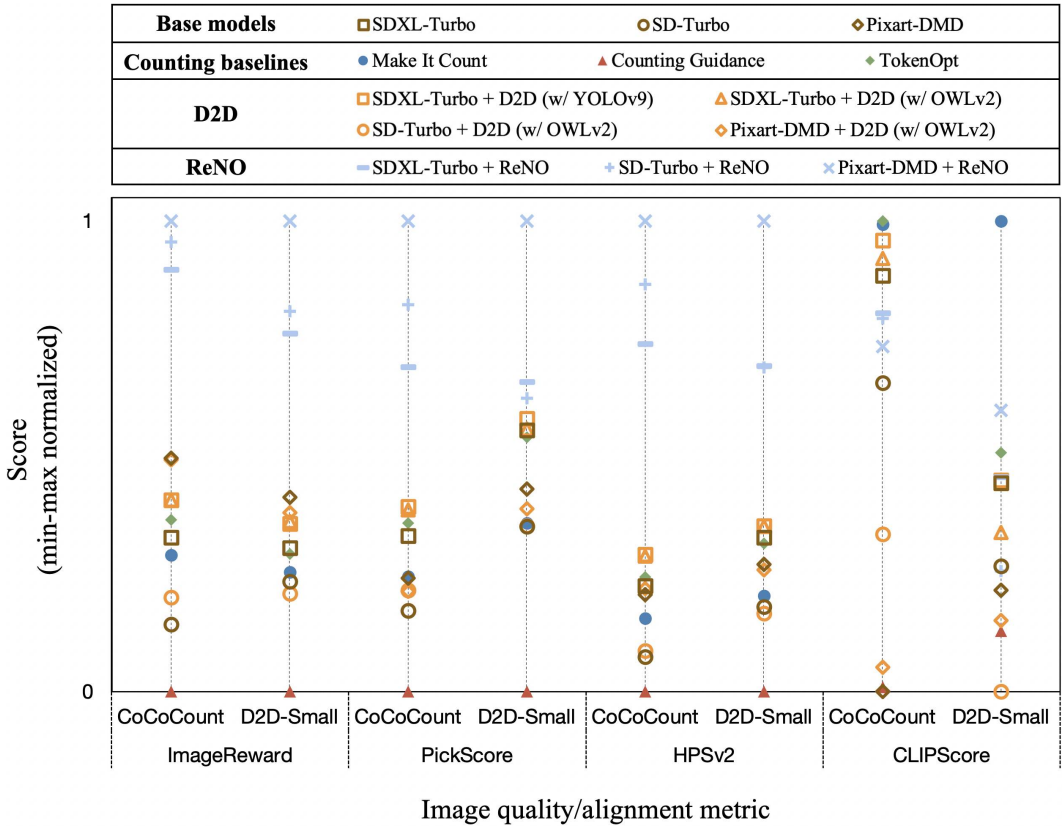


Figure 9: **Image quality/alignment scores** (ImageReward [15], PickScore [37], HPSv2 [38], CLIP-Score [39]) by method. Aside from ReNO [2], which often scores highest (it specifically optimizes those metrics), *D2D* is comparable to counting baselines. Min-max normalized.

J.1 *D2D* exhibits minimal degradation in image quality, compared to layout-control method Make It Count

Tab. 13 reports the image quality of layout control-based, count-correction method Make It Count [3], its base model SDXL, and *D2D*, along with base model SDXL-Turbo. Make It Count exhibits some tradeoffs between numeracy and image quality, yielding slightly lower PickScore and HPSv2, compared to its base model SDXL (e.g., on D2D-Small: PickScore: **21.70** (SDXL), 21.48 (MIC); HPSv2: **0.272** (SDXL), 0.264 (MIC)).

D2D offers minimal image quality degradation (and sometimes better image quality) than base model SDXL-Turbo (e.g., on D2D-Small: ImageReward: 0.40 (SDXL-Turbo), **0.51** (*D2D* w/ OWLv2); HPSv2: 0.279 (SDXL-Turbo), **0.282** (*D2D* w/ OWLv2)).

Table 13: **D2D yields minimal degradation in image quality and alignment, with minimal computational overhead, compared to layout control-based, multi-step method Make It Count.** Base models with no post-enhancement highlighted in gray. On CoCoCount [3] and D2D-Small. Avg. over four seeds.

Method	ImageReward ↑		PickScore ↑		HPSv2 ↑		CLIPScore ↑		Inference Time (s)
	CoCoCount	D2D-Small	CoCoCount	D2D-Small	CoCoCount	D2D-Small	CoCoCount	D2D-Small	
SDXL [6]	0.87	**0.31**	**22.99**	**21.70**	**0.290**	**0.272**	32.91	32.58	7.69
+ Make It Count [3]	**0.91**	0.30	22.83	21.48	0.286	0.264	**32.92**	**32.96**	37.16
SDXL-Turbo [1]	0.96	0.40	23.12	21.98	0.293	0.279	32.80	31.89	0.16
+ *D2D* (w/ OWLv2)	**1.06**	**0.51**	23.30	21.98	**0.300**	**0.282**	32.84	31.69	19.42
+ *D2D* (w/ YOLOv9)	**1.06**	0.50	**23.33**	**22.04**	**0.300**	**0.282**	**32.88**	**31.90**	11.39

J.2 Per-object fidelity

We compute the PickScore (quality check) and CLIPScore (alignment score) per detected object instance to measure per-object quality in addition to overall image quality. *D2D* object fidelity is comparable with baselines.

Table 14: ***D2D* object fidelity is comparable with baselines.** PickScore and CLIPScore computed per detected object instance. On D2D-Small, across four seeds.

Method	PickScore [37]	CLIPScore [39]
SD1.4 [7]	19.29	28.26
+ Counting Guidance [4]	19.04	27.51
SDXL [6]	19.51	28.80
+ Make It Count [3]	19.80	28.55
SDXL-Turbo [1]	19.62	28.41
+ TokenOpt [5]	19.65	28.45
+ *D2D*	19.72	28.61

K *D2D* yields boosts in numeracy, in complement with baselines

Table 15: ***D2D* additionally yields improvements, in combination with T2I counting/enhancement baselines.** Standard deviations indicate the significance of our findings. Avg. over four seeds.

Base Model	Method	CoCoCount [3]	D2D-Small	D2D-Multi	D2D-Large
SDXL-Turbo [1]	TokenOpt [5]	35.12±0.75	23.31±1.66	—	3.94±0.72
	TokenOpt [5] + *D2D*	**48.75**±2.40	**34.00**±1.70	—	**8.56**±1.28
	ReNO [2]	41.88±1.03	27.50±0.68	5.31±0.38	4.69±1.25
	ReNO [2] + *D2D*	**54.38**±2.17	**41.25**±2.45	**9.69**±0.38	**10.38**±0.72
SD-Turbo [1]	ReNO [2]	43.38±3.47	32.06±0.99	8.94±1.76	4.25±1.14
	ReNO [2] + *D2D*	**53.62**±2.02	**42.44**±2.82	**11.62**±1.30	**10.19**±1.77
Pixart-DMD [9]	ReNO [2]	44.75±1.44	37.25±1.70	9.44±0.75	4.75±0.74
	ReNO [2] + *D2D*	**52.50**±2.27	**40.62**±2.37	**12.00**±0.68	**8.12**±0.92

L Density-based differential performance persists in D2D-Multi

Table 16: **Quantitative results.** *D2D* yields higher performance on low total-density prompts than high total-density ones. Avg. over four seeds.

Method	D2D-Multi	
	Low Total Density	High Total Density
SDXL [6]	3.08	0.50
Make It Count [3]	—	—
SDXL-Turbo [1]	2.75	0.25
ReNO [2]	6.67	1.25
TokenOpt [5]	—	—
D2D w/ OWLv2 (Ours)	**12.08**	**3.00**
D2D w/ YOLOv9 (Ours)	***7.83***	***1.50***
SD2.1 [7]	6.17	0.75
SD1.4 [7]	3.75	0.00
Counting Guidance [4]	4.42	0.25
SD-Turbo [1]	3.00	1.25
ReNO [2]	11.58	1.00
D2D w/ OWLv2 (Ours)	**13.42**	**2.75**
Pixart-α [8]	1.67	0.25
Pixart-DMD [9]	8.00	1.00
ReNO [2]	12.25	1.00
D2D w/ OWLv2 (Ours)	**17.17**	**1.75**

M Analysis of high-density performance

We further analyze *D2D*'s performance in the high-density case by reporting numeracy, binned by object size, detector confidence, total count, and object class on D2D-Large/Multi. We use the mean absolute error $|\text{generated} - \text{requested count}|$ as the evaluation metric. Unless otherwise noted, the bin ranges are followed by the error in parentheses, as follows: [bin start, bin end] (error $\downarrow$).

Tab. 17, 18, and 19 show the performance breakdown by quartiles. As expected, error decreases with larger objects, higher confidence, and lower total count. For higher counts, small object size remains a bottleneck, in particular for object classes that naturally appear in "herds" such as cows and sheep. Potential mitigation strategies could include adopting a bbox loss to explicitly increase object sizes at test-time and improving training sample diversity for the generative model (e.g. individual sheep, flocks).

Table 17: **Error vs. object size. The error decreases as object size increases. *D2D* tends to increase object size.** We bin the performance by object size (quartiles), calculated as $\frac{\text{instance size}}{\text{image size}} \times 100$. Error binned by object size quartile. Bin ranges in brackets, error ($\downarrow$) in parentheses. Base model SDXL-Turbo, across four seeds.

Method	Quartile 1	Quartile 2	Quartile 3	Quartile 4
D2D-Large				
Base	$[0.3, 2.6]$ (**52.2**)	$(2.6, 4.5]$ (**35.2**)	$(4.5, 6.9]$ (**21.2**)	$(6.9, 92.4]$ (**13.3**)
+ *D2D*	$[0.1, 3.5]$ (**21.1**)	$(3.5, 5.2]$ (**17.9**)	$(5.2, 7.7]$ (**11.7**)	$(7.7, 93.6]$ (**12.0**)
D2D-Multi				
Base	$[0.4, 3.9]$ (**19.8**)	$(3.9, 6.6]$ (**8.2**)	$(6.6, 10.3]$ (**5.1**)	$(10.3, 54.3]$ (**3.5**)
+ *D2D*	$[0.9, 5.3]$ (**9.2**)	$(5.3, 8.3]$ (**7.4**)	$(8.3, 12.5]$ (**4.5**)	$(12.5, 70.2]$ (**3.1**)

Table 18: **Error vs. detector confidence. The error decreases with higher detector confidence. *D2D* tends to increase overall detector confidence.** Error binned by detector confidence quartile. Bin ranges in brackets, error (↓) in parentheses. Base model SDXL-Turbo, across four seeds.

Method	Quartile 1	Quartile 2	Quartile 3	Quartile 4
D2D-Large				
Base	$[0.20, 0.38]$ (**70.3**)	$(0.38, 0.45]$ (**26.8**)	$(0.45, 0.51]$ (**14.2**)	$(0.51, 0.80]$ (**10.4**)
+ *D2D*	$[0.22, 0.40]$ (**35.3**)	$(0.40, 0.48]$ (**18.8**)	$(0.48, 0.55]$ (**5.6**)	$(0.55, 0.78]$ (**3.1**)
D2D-Multi				
Base	$[0.25, 0.42]$ (**18.7**)	$(0.42, 0.50]$ (**8.1**)	$(0.50, 0.57]$ (**5.8**)	$(0.57, 0.88]$ (**4.0**)
+ *D2D*	$[0.24, 0.44]$ (**14.6**)	$(0.44, 0.52]$ (**4.3**)	$(0.52, 0.59]$ (**3.2**)	$(0.59, 0.85]$ (**2.1**)

Table 19: **Error vs. total requested count. The error increases with higher total requested counts**, likely due to smaller object instances, which aligns with the observations above. Base model error and *D2D* error (↓) binned by total requested count. Base SDXL-Turbo, across four seeds.

D2D-Large		
Requested count	Base	+ *D2D*
$[11, 13]$	16.8	**8.2**
$(13, 15]$	32.4	**16.4**
$(15, 18]$	34.7	**18.2**
$(18, 20]$	42.5	**22.4**

D2D-Multi		
Requested count	Base	+ *D2D*
$[2, 4]$	4.2	**2.2**
$(4, 5]$	6.0	**3.3**
$(5, 10]$	9.1	**5.7**
$(10, 19]$	17.6	**13.0**

Object categories with highest error. Sheep, book and cow are among the top four most difficult categories for *D2D* on both D2D-Large and D2D-Multi, which may be explained by their tendency to occur in large groups (i.e., flock of sheep, shelf of books, etc).

N How *D2D* handles overlapping detections

To analyze how *D2D* handles overlapping detections, we compare bbox overlap before and after *D2D* correction in the high-density setting, where overlap is more likely. We define an image's overall "overlap score" as the global overlap-to-union ratio: $\frac{(\sum_i \text{Area}(B_i)) - \text{Area}(\bigcup_i B_i)}{\text{Area}(\bigcup_i B_i)}$. Furthermore, the mean absolute error of an image is: $|\text{requested} - \text{generated count}|$.

Using base model SDXL-Turbo and prompt set D2D-Large, we bin the initial generations into quartiles based on their overlap scores. For each quartile, we report the mean initial error, and after applying *D2D*, the new mean overlap score and error. Tab. 20 details the results.

Table 20: ***D2D* reduces error across all initial overlap levels.** Error and overlap before/after *D2D*, binned by initial overlap quartile. On D2D-Large, base model SDXL-Turbo, across four seeds.

Initial overlap bin	Initial mean abs. error	Post-*D2D* mean overlap	Post-*D2D* mean abs. error
$[0.00, 0.10]$	12.19	0.07	**7.47**
$(0.10, 0.22]$	18.17	0.16	**12.88**
$(0.22, 0.83]$	56.46	0.20	**28.40**
$(0.83, 1.33]$	34.88	0.32	**13.98**

D2D reduces error across all initial overlap ranges. In cases of high initial overlap (e.g., the third and fourth quartiles), the post-D2D overlap drops below the initial overlap range, suggesting that *D2D* actively reduces overlap.

Because over-generation is particularly prevalent in high-count settings (e.g., 78% of the initial images from above were over-generations), *D2D* works to delete objects and reduces overlap.

O *D2D* vs. YOLO-COUNT [43]

We include comparisons against a more recent work, YOLO-COUNT [43]. A key distinction between the two methods is that *D2D* leverages detector logits out-of-the-box, without any additional detector or counter training, making it easily adaptable to newly developed detectors.

We evaluated YOLO-COUNT using their two pretrained checkpoints (on FSC147 [23] and LVIS v1.0 [48]) on CoCoCount (seed = 0) and found that LVIS (47.5%) performed stronger than FSC (40.5%). We replace the strongest *D2D* critic with YOLO-COUNT's strongest checkpoint and compare performance on CoCoCount and D2D-Small/Multi/Large (averaged over four seeds). *D2D* outperforms YOLO-COUNT across all four benchmarks on SDXL-Turbo.

Table 21: ***D2D* vs. YOLO-COUNT on SDXL-Turbo.** *D2D* outperforms YOLO-COUNT across all four benchmarks on SDXL-Turbo. Standard deviations indicate the significance of our findings. Avg. over four seeds.

Method	CoCoCount [3]	D2D-Small	D2D-Multi	D2D-Large
YOLO-COUNT [43]	44.38 ± 3.25	30.44 ± 2.69	5.69 ± 3.02	8.06 ± 0.90
D2D	**55.62 ± 2.72**	**43.69 ± 2.36**	**9.81 ± 0.97**	**9.94 ± 1.57**

P *D2D* vs. EnMMDiT [44]

We additionally compare our method against EnMMDiT [44], a relevant work. The two methods address different aspects of generation: *D2D* directly optimizes object counts using a detector-based differentiable critic, whereas EnMMDiT addresses subject mixing and neglect by leveraging a suite of losses that correct attention activation in earlier layers, text encoder inconsistency, and overlapping attentions of subjects.

We report numeracy for EnMMDiT (base model SD3 [21]) and *D2D* (base model SDXL-Turbo) using CoCoCount and a subset of 100 prompts from D2D-Multi. Standard deviations reported across four seeds.

Table 22: **Although EnMMDiT builds on SD3 (a substantially stronger base model with nearly 2× numeracy over SDXL-Turbo), *D2D* achieves comparable performance with EnMMDiT, as seen by the overlapping performance range.** On CoCoCount and 100-prompt subset of D2D-Multi. Standard deviations noted. Across four seeds.

Method	CoCoCount [3]	D2D-Multi (subset)
SD3 [21]	54.38 ± 5.07	17.00 ± 3.83
+ EnMMDiT [44]	**58.62 ± 1.31**	**20.75 ± 4.99**
SDXL-Turbo [1]	27.38 ± 2.69	3.50 ± 1.73
+ *D2D*	**55.62 ± 2.72**	**15.25 ± 2.63**

Q CountGD in comparison with other counters

CountGD performs well in low and high-density settings.

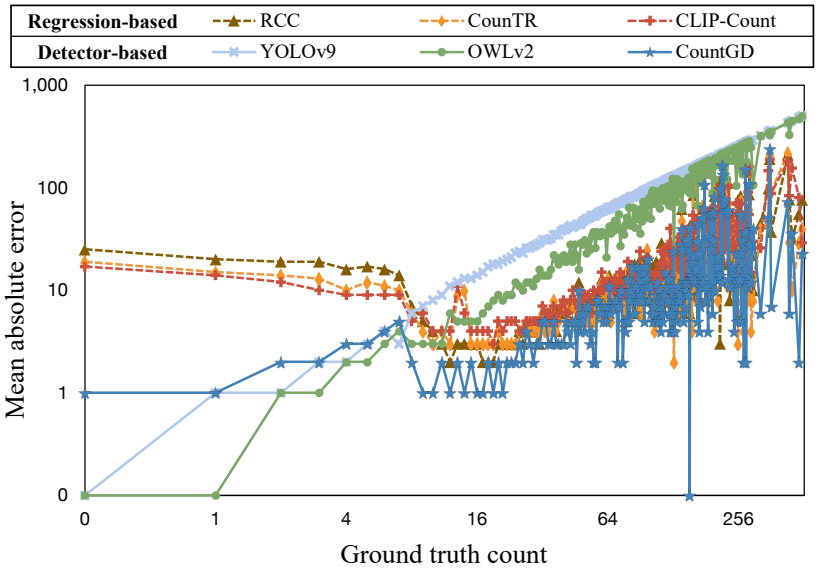


Figure 10: **CountGD** is a state-of-the-art counter, based on GroundingDINO.

R Additional qualitative results

Additional qualitative results on prompts from benchmarks CoCoCount [3] and D2D-Small shown.



Figure 11: **Qualitative examples** from count correction methods on base model SDXL-Turbo, along with SDXL.

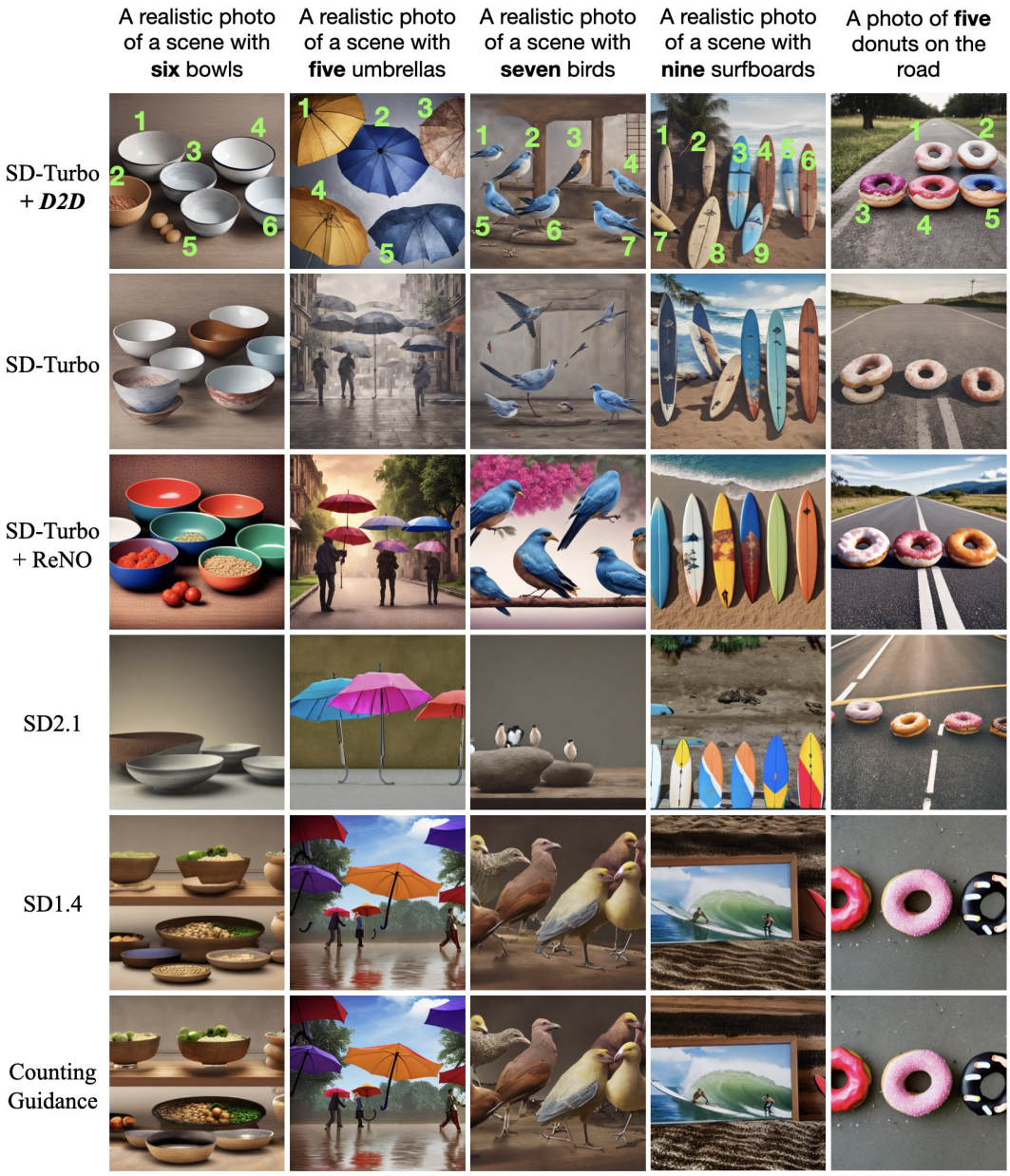


Figure 12: **Qualitative examples** from count correction methods on base model SD-Turbo, along with SD1.4, SD2.1, and Counting Guidance [4].

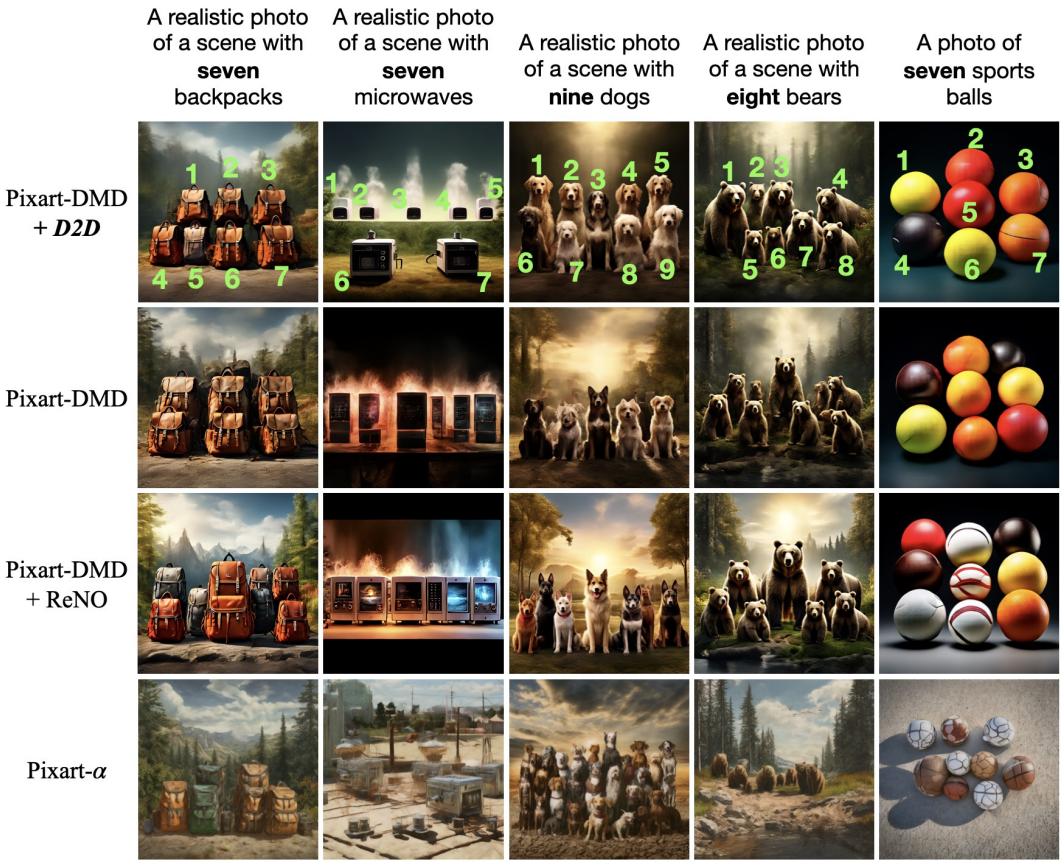


Figure 13: **Qualitative examples** from count correction methods on base model Pixart-DMD, along with Pixart-α.

R.1 *D2D* on free-form prompts

D2D as a method just requires the user to provide the prompt along with the desired (object, count) pair. While our quantitative evaluation focuses on prompts that follow structured formats, we provide additional qualitative figures on more free-form prompts.



Figure 14: **Qualitative examples** of *D2D* on more free-form prompts using base model SANA-Sprint [42].

S Additional ablations

S.1 w = 0.2 is optimal for numeracy and quality

w is the weight used to determine how to mix the LMN output with the original noise. $w = 0$ is numeracy-wise the best, followed by $w = 0.2$, but results in patchy visual artifacts, relative to $w = 0.2$ (Fig. 15). Hence, the hyperparameter value we use in our main experiments is $w = 0.2$.

Table 23: **In terms of just numeracy, $w = 0$ is optimal, followed by $w = 0.2$.** Base model SDXL-Turbo, on CoCoCount [3], seed = 0.

	w			
	0.0	0.2	0.5	0.8
Numeracy	**62.50**	55.50	37.50	48.00

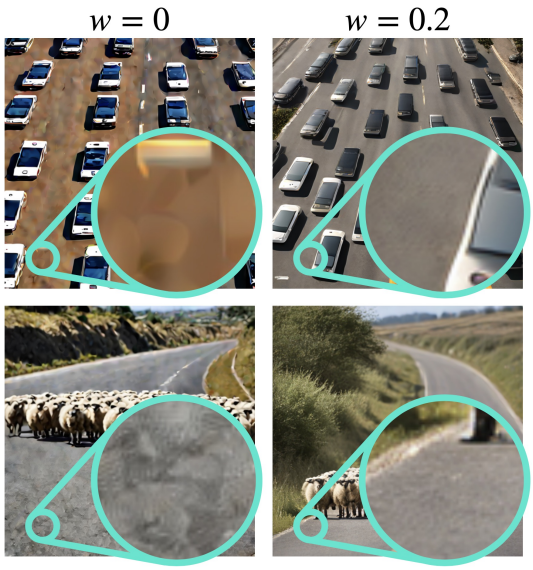


Figure 15: **$w = 0$ results in patchy visual artifacts, relative to $w = 0.2$.** So, we use $w = 0.2$.

S.2 LMN architecture

For the choice of 3-layer MLP, we perform additional ablation studies and present the results in Tab. 24. Between MLP architectures, the 3-layer, 100-width and 200-width variants perform the best, but the former is more parameter-efficient. We also try using a 6-layer CNN (with adjusted kernels and channels to approximate the parameter count of the 3-layer, 100-width MLP), which yielded a slight 1% boost in numeracy. Our results demonstrate that even a small and simple MLP works quite well as an LMN architecture.

Table 24: **Even a small and simple 3-layer, 100-width MLP works effectively for boosting numeracy.** On CoCoCount [3]. On SDXL-Turbo, seed=0.

LMN Architecture	Numeracy
3-layers, width 100	55.5
2-layers, width 100	52.5
4-layers, width 100	52.0
3-layers, width 50	54.5
3-layer, width 200	55.5
CNN (approximate parameter-count match)	56.5

S.3 Pre-inference alignment of M_ϕ

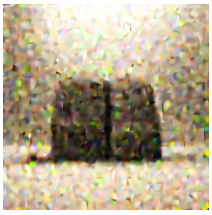
We conduct experiments without the pre-inference alignment stage and report below (Tab. 25). We find that without pre-inference alignment, it generally takes an additional 20-30 steps during inference time to align the network to Gaussian outputs, which effectively reduces the number of steps available to perform numeracy optimization, because we cap the max number of steps.

Table 25: **Pre-inference alignment frees up more inference-time steps to be dedicated to numeracy optimization,** yielding higher numeracy. On CoCoCount [3], using SDXL-Turbo, seed=0.

Method	Numeracy
w/o pre-inference alignment	47.0
w/ pre-inference alignment	**55.5**

S.4 Inference-time calibration of M_ϕ

At inference-time, given a new $\mathbf{x}_T$, we allow a few iterations to calibrate M_ϕ's output to the Gaussian distribution. Fig. 16 shows alignment is crucial to maintaining high image quality.



(a) Without alignment.



(b) With alignment.

Figure 16: **Inference-time alignment of M_ϕ is necessary.** Prompt: "A realistic photo of a scene with one backpack."

S.5 Regularization

Our regularization term (Eq. 6) is a variant of the one ReNO [2] uses (Eq. 5) that penalizes larger deviations from Gaussian more and smaller deviations less, to allow enough flexibility in the vicinity of the initial distribution to accommodate updates specified by the count critic. We scale $\mathcal{L}'_{\text{reg}}$ by coefficient a and shift by constant c, which we take to the power of 10.

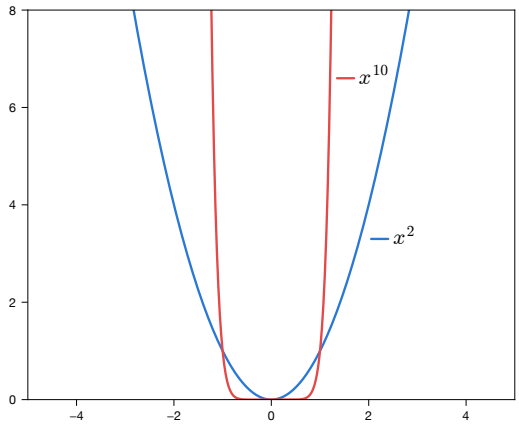


Figure 17: **Exponentiating by the power of 10 sharpens the gradients far away from the minimum while flattening the gradient closer to the minimum**, effectively allowing the optimization process to focus more on numeracy optimization when the latent is already well-regularized.

$$\mathcal{L}'_{\text{reg}} = \frac{||\mathbf{x}'_T||^2}{2} - (d-1) \cdot \log(||\mathbf{x}'_T||), \tag{5}$$

$$\mathcal{L}_{\text{reg}} = (a\mathcal{L}'_{\text{reg}} + c)^{10}. \tag{6}$$

We set coefficient a to 0.03. Shift constant c is set to 2139 to offset the optimal value of $a\mathcal{L}'_{\text{reg}}$, which we find to be -2139 in practice.

Our ablation analysis confirms that this variant of the regularization leads to overall numeracy improvements across both CoCoCount and D2D-Small (Tab. 26), indicating our regularization objective allows for effective numeracy correction, even while enforcing a heavier penalty on non-Gaussian initial noises.

Table 26: **Our variant of regularization is beneficial to numeracy optimization.** $\mathcal{L}_{\text{reg}}$, which penalizes larger deviations from Gaussian more and smaller deviations less, yields higher numeracy on CoCoCount and D2D-Small benchmarks. On SDXL-Turbo. Avg. over four seeds.

Method	CoCoCount [3]	D2D-Small
D2D w/ $\mathcal{L}'_{\text{reg}}$	53.88	42.44
D2D w/ $\mathcal{L}_{\text{reg}}$	**55.62**	**43.69**

Table 27: **Our variant of regularization does not significantly impact diversity**, as measured by Vendi score [49], on CoCoCount [3] and D2D-Small. On SDXL-Turbo. Across four seeds.

Method	Vendi
D2D w/ $\mathcal{L}'_{\text{reg}}$	1.82
D2D w/ $\mathcal{L}_{\text{reg}}$	1.82

T *D2D* scaling formulation

Table 28: **Our *D2D* loss formulation is effective.** We compare our formulation against several alternatives. Ours yields the highest numeracy with fastest inference. On CoCoCount [3]. On SDXL-Turbo, seed=0.

Scaling formulation	Numeracy	Inference time (s)
$\sigma(\beta(z-\tau))\cdot(z-\tau)$ (ours)	55.50	17.10
$e^{z-\tau}$	52.00	56.28
$e^{\sigma(\beta(z-\tau))}$	31.50	32.24
$z-\tau$	50.50	53.23
$\ln(1+e^{z-\tau})$	55.00	20.09
$(z-\tau)\cdot e^{z-\tau}$	49.00	72.72
$(z-\tau)\cdot\ln(1+e^{z-\tau})$	54.00	22.31
$\sigma(\beta(z-\tau))\cdot e^{z-\tau}$	50.50	63.96
$\sigma(\beta(z-\tau))\cdot\ln(1+e^{z-\tau})$	55.00	18.65

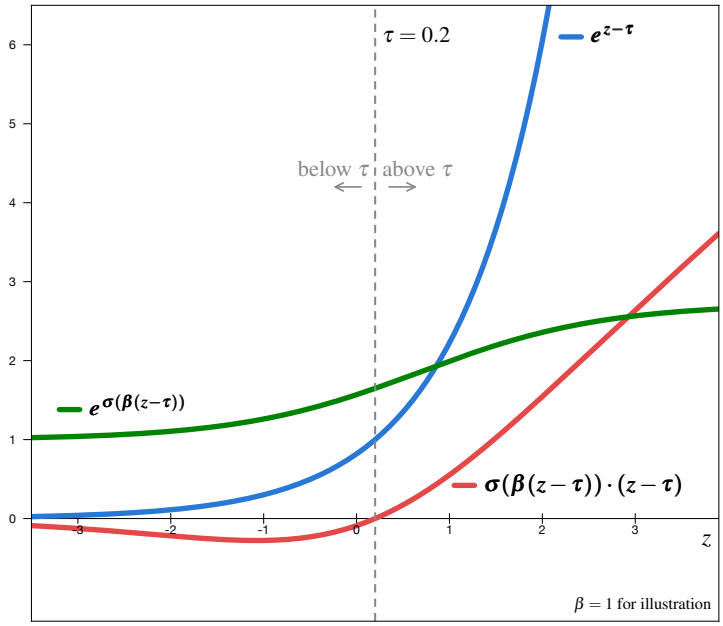


Figure 18: **Design of *D2D* loss.** We want to design a fully-smooth loss that will effectively update each bbox's logit in order to bring the total count closer to the requested count. To illustrate, if the total generated count $>$ requested count, then the overall objective is to bring the logits down. However, practically speaking, on the level of individual logits, this means focusing on decreasing the logits that are above the τ and allowing some flexibility for logits that are already below τ so that the optimization can focus on the $>\tau$ logits. In other words, we want a strong gradient signal above τ and little signal below. This plots the curves of the first three scaling formulations from the table above, illustrating how they behave above and below τ. Our formulation (in red) has a strong signal above and weak signal below τ. $e^{z-\tau}$ still maintains a significant gradient below τ. $e^{\sigma(\beta(z-\tau))}$ still shows plateauing above τ.

U Sample numeracy optimization trajectory

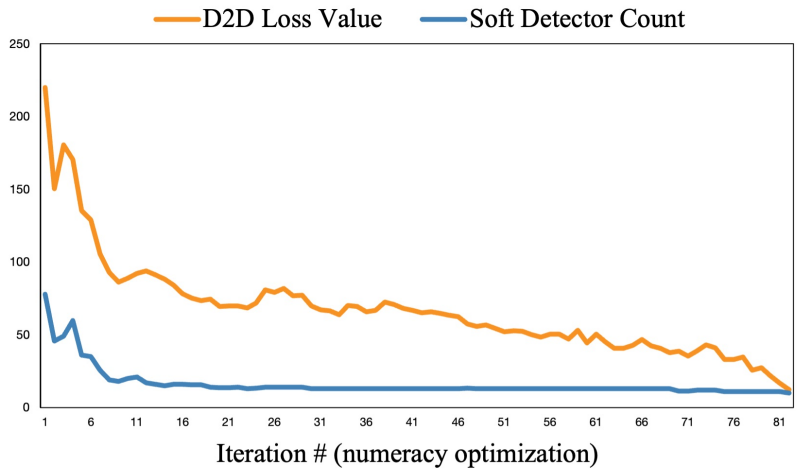


Figure 19: **Sample trajectory of $\mathcal{L}_{\text{D2D}}$** and the soft detector count f over numeracy optimization steps for the prompt "A photo of ten sheep on the grass" from CoCoCount [3]. Though the two curves do not have the same raw value, their coarse trends do correlate.

V Selective *D2D* activation

D2D focuses on numerical prompts, but not all prompts are numeric in nature. Furthermore, prompts may lack concrete numerical targets, as in the prompt "A few cups." From the perspective of user intention, using words, like "few" or "many," that lack clear numerical grounding may indicate lack of strict numerical target in mind. So, it is reasonable to allow some flexibility in mapping those words to numbers. At test-time, we can leverage an LLM to determine whether a prompt is a numerical one and, if so, to extract the (object, requested count) pairs. Below is a sample context for this task.

In-context prompting for *D2D* selective activation

You are an expert prompt analyzer. Given a text-to-image prompt, you extract two things:

1. Boolean indicator of whether the prompt specifies a count of objects. Examples include "Six dogs," "Some cats." It is not a countable prompt if all the objects are uncountable nouns. For the prompt to be a counting prompt, at least one object must be countable.
2. For each **countable** object and requested count, extract this information (object in singular form, object in form seen in the prompt, requested count in number form). So, for prompts with multiple objects and corresponding requested counts, you would extract a list like this [(object in singular form, object in form seen in the prompt, requested count), (object in singular form, object in form seen in the prompt, requested count), ...].

##########
Examples:

Prompt: A photo of seven dogs.
Think: Prompt requests seven dogs, and dog is countable, so it is a counting prompt. Seven is 7 in number form. The object form seen in the prompt is dogs. The corresponding singular form is dog.
Response: True, [("dog", "dogs", 7)]

Prompt: A photo of seven dogs and two cats.
Think: Prompt requests seven dogs and two cats. In this case, both dog and cat are countable, so it is a counting prompt. Seven is 7 in number form, and two is 2. The object form seen in the prompt is dogs and cats. The corresponding singular forms are dog and cat.
Response: True, [("dog", "dogs", 7), ("cat", "cats", 2)]

Prompt: A photo of some cups.
Think: Prompt requests some cups; "some" indicates a small number, so it is countable. Let's just choose a reasonable number between 1 and 5, feel free to randomize within this range. Let's choose 3 this time. The object form seen in the prompt is cups. The corresponding singular form is cup.
Response: True, [("cup", "cups", 3)]

Formatted transcript of selected LLM responses

Prompt: A realistic photo of a scene with nine pineapples.
ChatGPT: Think: The prompt specifies "nine pineapples." Pineapple is a countable noun, so this is a counting prompt. The number "nine" corresponds to 9. The object form seen in the prompt is "pineapples," and the singular form is "pineapple."

Response:
True, [("pineapple", "pineapples", 9)]

Prompt: A photo of some water.
ChatGPT: Think: The prompt specifies "some water." Water is an uncountable noun, so it cannot be treated as a countable object. There is no countable object requested here.

Response:
False, []

Prompt: A few apples.
ChatGPT: Think: The prompt specifies "a few apples." Apple is a countable noun, so this is a counting prompt. "A few" typically corresponds to 3. The object form seen in the prompt is "apples," and the singular form is "apple."

Response:
True, [("apple", "apples", 3)]

Prompt: A magnificent landscape with lots of sunshine, animals everywhere, and plenty of greenery.
ChatGPT: Think: The prompt mentions "animals," but it does not specify a number or quantity for any countable object. All quantities are vague ("lots," "plenty"), so there is no explicit count requested.

Response:
False, []

Prompt: A bright outdoor farmers' market captured in a natural documentary style: show 4 wooden tables arranged in two rows, each table displaying 11 neatly stacked pumpkins. Behind the tables stand 5 vendors, each with 3 small hanging lanterns attached to their stall. On the left, include 3 crates, each filled with 6 fresh apples, all clearly visible. In front of the crates, show 2 children crouched down examining 9 colorful marbles scattered on the ground. On the right side of the scene, place 2 bicycles, each with 2 woven baskets, and inside each basket include 4 lemons. Add 5 small fabric flags strung across the walkway above. Ensure all objects are visible, not overlapping, and easy to count. *(prompt generated separately by ChatGPT).*
ChatGPT: True, [("table", "tables", 4), ("pumpkin", "pumpkins", 11), ("vendor", "vendors", 5), ("lantern", "lanterns", 3), ("crate", "crates", 3), ("apple", "apples", 6), ("child", "children", 2), ("marble", "marbles", 9), ("bicycle", "bicycles", 2), ("basket", "baskets", 2), ("lemon", "lemons", 4), ("flag", "flags", 5)]

W Extension to multi-step models

We show that the *D2D* critic can be extended to multi-step diffusion models as well. We demonstrate this by applying *D2D* to the multi-step initial noise framework proposed in Direct Noise Optimization (DNO) [50] (see Fig. 20). However, there is a significant tradeoff in latency, as convergence took approximately 7 minutes for the prompt "A realistic photo of a scene with five cups."

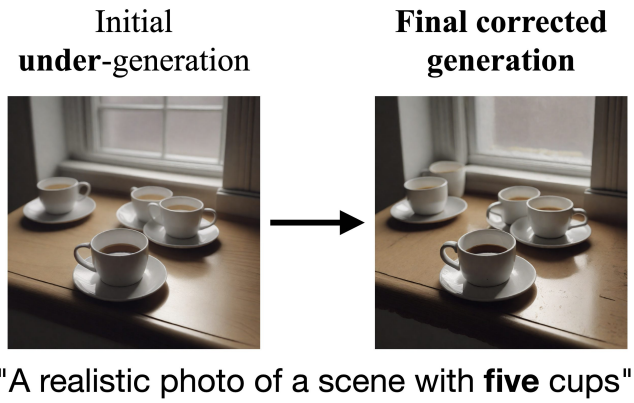


Figure 20: **Example of numeracy correction using DNO [50] with our *D2D* critic.** This took 12 optimization steps (approximately 7 minutes).

X Diversity analysis of generated images

Table 29: ***D2D* does not significantly impact diversity compared to corresponding base models.** This aligns with prior work [28] which found that selecting higher-quality initial noises does not significantly impact diversity. We compute Vendi scores across images generated from different random seeds for the same prompt, and average the scores across prompts, using Inception-based embeddings. Standard deviations noted. On CoCoCount [3] and D2D-Small, using base models SDXL-Turbo, SD-Turbo, and Pixart-DMD. Across four seeds.

Method	Vendi [49]
SDXL-Turbo [1]	1.76 ± 0.23
SDXL-Turbo [1] + *D2D*	1.82 ± 0.31
SD-Turbo [1]	1.87 ± 0.25
SD-Turbo [1] + *D2D*	1.98 ± 0.33
Pixart-DMD [9]	1.76 ± 0.26
Pixart-DMD [9] + *D2D*	1.82 ± 0.32

Y Broader impacts

As a text-to-image pipeline, our method has many practical downstream uses, so it is necessary to exercise caution in deployment. Our method offers the advantage of stronger numeracy, which may be desirable in applications where users need to generate specific counts of objects. Our model may inherit biases of pre-trained diffusion base models and detectors. We suggest using strict NSFW filters and building more robust detectors.

Z Use of Large Language Models (LLMs)

We used LLMs to help with word choice for clarity and debugging.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the methods and experiments sections justify our claims, with additional details in the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, we discuss limitations in our experiments section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: Our work does not present new theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper and appendix provide fine-grained implementation details. We will release our code and benchmarks.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, we will release our code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting is detailed in the experiments section, with more specifics included in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For different experiments, we report the number of seeds we averaged results over (often four seeds, with one seed for some ablations).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details on the compute setting (GPU type and hence, the corresponding GPU VRAM, and inference cost) in our experiments section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, our work conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We discuss safeguards in the appendix.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The related works, models, and datasets are explicitly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Aside from our code, which we will release with appropriate licenses, there are no new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We conducted a questionnaire-based human evaluation of generated images. The participant instructions and additional details of the study protocol are provided in the appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: The study consisted of a non-biomedical questionnaire evaluating generated images and was conducted in accordance with applicable local regulations and institutional ethics requirements.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Our core method does not involve LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.