Z-IMAGE ULTRA-TURBO: TOWARD HIGH-FIDELITY TWO-STEP IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-step diffusion distillation has become increasingly mature for 4–8-step generation, yet pushing further to 2 steps remains challenging. In this work, we introduce **Z-Image Ultra-Turbo**, a high-quality 2-step image generation model distilled from the 8-step Z-Image Turbo teacher. Our method addresses the central bottlenecks of the increased task difficulty and limited model capacity in 2-step generation through three simple but effective design choices tailored to this regime. First, we propose *Distribution-Aligned Adversarial Learning*, which uses teacher-generated images rather than external real images as real samples for GAN training, providing a more attainable and informative adversarial target. Second, we adopt *Step-Decoupled Parameterization*, assigning independent model parameters to the two denoising steps to better match their distinct capacity demands. Third, we perform *End-to-End Training with Iterative Regularization*, allowing the first step to receive gradients from final image quality while preserving a meaningful intermediate generation through an explicit step-1 loss. Together, these designs substantially narrow the quality gap between 2-step and 8-step generation in both qualitative and quantitative evaluations, highlighting the potential of carefully tailored distillation strategies for improving the quality–efficiency trade-off in few-step generation.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020b) have achieved remarkable success in image generation, producing outputs of exceptional quality and diversity. However, this performance comes at a significant computational cost: the iterative sampling process typically requires 40–100 neural network evaluations, creating a substantial barrier for deployment.

Few-step distillation has emerged as the dominant paradigm for addressing this bottleneck. Methods such as Distribution Matching Distillation (DMD) (Yin et al., 2024b; Luo et al., 2023; Liu et al., 2026), Consistency Models (Song et al., 2023), Progressive Distillation (Salimans & Ho, 2022), and adversarial approaches (Sauer et al., 2024; Lin et al., 2024) have provided important foundations for this line of work. Meanwhile, strong publicly released models such as Z-Image-Turbo (Team, 2025), Qwen-Image-Lightning (ModelTC, 2025), and FLUX.2[klein] (Labs, 2025) have demonstrated successful compression to 4–8 steps with minimal quality degradation. These results have established few-step generation as a mature and increasingly standardized stage in the model production pipeline.

Beyond 4–8 steps, which still leave room for efficiency gains, **2-step generation** occupies a unique position: it maximizes inference efficiency while retaining sufficient iterative structure to be leveraged, whereas single-step generation remains too challenging to produce satisfactory quality. The 2-step setting itself is not new: general few-step methods such as DMD and DMD2 (Yin et al., 2024b;a) can in principle be instantiated with two sampling steps, and recent work such as TwinFlow (Cheng et al., 2025) explicitly reports 2-step results. In practice, however, their quality in this regime remains limited, and naively reducing the step count of existing methods to 2 leads to severe degradation, suggesting that the 2-step regime presents fundamentally different challenges. Our goal is therefore not to introduce 2-step generation, but to substantially narrow its quality gap to the multi-step teacher.

Through extensive experimentation, we identify two core challenges that distinguish 2-step generation from conventional few-step distillation. The first is *optimization difficulty*. With only two function evaluations, each denoising step must cover a very large interval of the noise-to-data trajectory:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 1: Images generated by Z-Image Ultra-Turbo with only 2 steps. Best viewed with zoom.

the first step must transform pure noise into a meaningful intermediate state, while the second step must refine this state into a clean image. In this regime, directly imposing an overly distant target distribution can make training unstable and produce persistent artifacts. We therefore find that the choice of learning target is crucial: the target should be strong enough to improve perceptual quality, but also close enough to the student's attainable distribution to provide useful gradients.

The second challenge is *capacity under extreme step specialization*. In standard multi-step sampling, the same model is reused across many timesteps, and each evaluation performs a relatively local update. In contrast, the two steps of a 2-step generator play sharply different roles. This makes parameter sharing unusually restrictive: a single model must simultaneously solve two highly distinct and demanding subproblems. Moreover, because the first step determines the intermediate representation consumed by the second, optimizing the two steps independently can lead to suboptimal coordination. These observations suggest that successful 2-step distillation requires training objectives, parameterizations, and optimization procedures that are explicitly adapted to the trainability, capacity, and coordination constraints of this regime.

To address these challenges, we propose **Z-Image Ultra-Turbo**, a carefully distilled model derived from the original 8-step Z-Image Turbo and specialized for high-quality 2-step image generation. The core technical designs are as follows:

- To make the unusually difficult 2-step objective trainable, we employ adversarial training to align our 2-step model with *an existing few-step (8-step) teacher*. We deliberately depart from the prevailing practice, established by ADD and DMD2, of using external real images as the discriminator's real samples, and use the teacher's generations instead. Beyond making a different choice, we show in controlled comparisons that the two options differ systematically: external real images yield a persistently growing adversarial loss and recurring artifacts, whereas teacher-generated samples provide a stable and attainable target that substantially improves final quality.

We attribute this to the closer distributional alignment between the teacher's and student's outputs, which provides more informative gradient signals.

- To address the insufficient effective capacity of a shared model under extreme step specialization, we propose Step-Decoupled Parameterization, where the step-specific models are initialized from the same teacher weights but updated independently thereafter, effectively enlarging model capacity and reducing interference between the two sharply different denoising tasks.
- We introduce end-to-end training that treats the 2-step generation process as a fully differentiable pipeline, enabling the first step to receive gradients that directly optimize the final output quality. To preserve the pretrained model's iterative generation pattern, we further retain an explicit step-1 loss as iterative regularization. This improves capacity utilization while allowing the two steps to coordinate more flexibly.

With these designs, our model substantially narrows the gap between 2-step and 8-step generation, preserving most of the teacher's visual quality and benchmark performance while reducing inference to only two denoising steps.

2 Related Work

Diffusion Model Acceleration. Training-free acceleration includes advanced ODE solvers such as DDIM (Song et al., 2020a), DPM-Solver (Lu et al., 2022), and UniPC (Zhao et al., 2023), as well as model-level techniques such as pruning (Fang et al., 2023), quantization (He et al., 2023; Shang et al., 2023), and caching (Liu et al., 2025; Ma et al., 2024). Step distillation, which trains a student generator to reproduce a teacher's sampling behavior with fewer evaluations, is closest to our setting.

Few-Step Distillation. Progressive Distillation (Salimans & Ho, 2022) pioneered a curriculum-based approach that halves the step count iteratively. Consistency Models (Song et al., 2023) and their variants (Wang et al., 2024; Lu & Song, 2024; Ren et al., 2024) enforce self-consistency along the ODE trajectory to enable direct mapping to the trajectory endpoint. Distribution Matching Distillation (DMD) (Yin et al., 2024b;a) minimizes the distributional divergence between student and teacher outputs. Decoupled DMD (Liu et al., 2026) clarifies the working mechanism of DMD as a CFG Augmentation engine and a Distribution Matching regularizer, enabling principled schedule design and achieving strong 4–8 step results. Other notable approaches include InstaFlow (Liu et al., 2023), Rectified Flow (Liu et al., 2022), and Moment Matching Distillation (Salimans et al., 2024). TwinFlow (Cheng et al., 2025) folds the generator and the real/fake score estimators into a single time-conditioned, parameter-shared network, a simplification that improves memory efficiency but is orthogonal to the capacity and optimization bottlenecks of the 2-step regime.

GAN-Based Distillation. Adversarial Diffusion Distillation (ADD) (Sauer et al., 2024) introduced adversarial training to the distillation pipeline, using a pretrained visual feature extractor with lightweight discriminator heads. SDXL-Lightning (Lin et al., 2024) combines progressive distillation with adversarial objectives. DMD2 (Yin et al., 2024a) augments the DMD loss with a GAN loss, forming a classic distillation combination that we also adopt in this work. DMD (Yin et al., 2024b) does not train a discriminator; its teacher-generated noise–image pairs only serve an auxiliary regression loss. Following ADD and DMD2, external real images have become the default real samples for the discriminator. Adv-KD (Mekonnen et al., 2024) uses teacher outputs as discriminator positives, but in a different setting that trains a GAN student from scratch, and it does not examine how this choice compares with external real images. We instead identify, under controlled few-step distillation, a systematic stability gap between the two options.

Timestep-Specialized Parameters. Assigning different parameters to different stages of the denoising trajectory has precedent in multi-step diffusion: eDiff-I (Balaji et al., 2022) and ERNIE-ViLG 2.0 (Feng et al., 2023) use expert denoisers for different noise intervals, SDXL (Podell et al., 2023) adds a separate refiner, Wan2.2 (Wan Team, 2025) switches between high- and low-noise experts, and Ganjdanesh et al. (2024) automatically select intervals and sub-networks for efficient experts. Whereas these methods specialize parameters across many steps, the 2-step regime pushes task divergence and capacity scarcity to the extreme while requiring only two parameter sets, making full per-step decoupling both especially beneficial and practical.

3 PRELIMINARY

3.1 FLOW MATCHING

We adopt the flow matching framework (Lipman et al., 2022) throughout this paper. We define $t = 0$ as pure noise and $t = 1$ as clean data. Given a data sample $x_1 \sim p_{\text{data}}$ and noise $\epsilon \sim \mathcal{N}(0, I)$, the forward process constructs an intermediate noisy sample:

$$x_t = t \cdot x_1 + (1 - t) \cdot \epsilon, \quad t \in [0, 1]. \tag{1}$$

A neural network $v_\phi(x_t, t)$ is trained to predict the velocity field that transports the noise distribution to the data distribution. During inference, generation proceeds by integrating the learned velocity field from $t = 0$ to $t = 1$ using a numerical ODE solver, typically requiring many discretization steps for high quality.

3.2 DISTRIBUTION MATCHING DISTILLATION

Distribution Matching Distillation (DMD) (Yin et al., 2024b) trains a few-step student generator G_θ by minimizing the integral KL divergence between the student and teacher distributions:

$$\mathcal{L}_{\text{IKL}}(p_{\text{real}}, p_{\text{fake}}) = \int_0^1 \mathbb{KL}(p_{\text{real},\tau} \| p_{\text{fake},\tau})\, d\tau. \tag{2}$$

In practice, its gradient is estimated with a frozen "real" score model and a concurrently trained "fake" score model:

$$\nabla_\theta \mathcal{L}_{\text{DMD}} = \mathbb{E}_{z_t,\tau,x_\tau}\left[-\left(s^{\text{real}}_{\text{cfg}}(x_\tau) - s^{\text{fake}}_{\text{cond}}(x_\tau)\right) \frac{\partial G_\theta(z_t)}{\partial \theta} \right], \tag{3}$$

where x_τ is obtained by renoising $G_\theta(z_t)$ to noise level τ. This practical objective differs from the theoretical estimator, which should have used $s^{\text{real}}_{\text{cond}}$ instead of the CFG-guided score $s^{\text{real}}_{\text{cfg}}$. Despite this mismatch, CFG is crucial for strong performance in large-scale text-to-image distillation.

Decoupled DMD (Liu et al., 2026) explains this mismatch by decomposing the CFG-guided score difference into two terms:

$$s^{\text{real}}_{\text{cfg}}(x_\tau) - s^{\text{fake}}_{\text{cond}}(x_\tau) = \alpha\left(s^{\text{real}}_{\text{cond}}(x_\tau) - s^{\text{real}}_{\text{uncond}}(x_\tau)\right) + \left(s^{\text{real}}_{\text{cond}}(x_\tau) - s^{\text{fake}}_{\text{cond}}(x_\tau)\right) = \alpha\Delta_{\text{CA}} + \Delta_{\text{DM}}. \tag{4}$$

Here α is the CFG scale. Decoupled DMD shows that *CFG Augmentation* (CA) is the main engine for few-step conversion, while *Distribution Matching* (DM) mainly regularizes training and suppresses artifacts. This insight inspires principled schedule design that improves 4–8 step distillation and underlies the strong 8-step generation capability of Z-Image-Turbo.

3.3 GAN IN DISTILLATION

GAN objectives are widely used in diffusion distillation (Sauer et al., 2024; Yin et al., 2024a). A discriminator D distinguishes "real" from "fake" images and provides an adversarial gradient to the generator:

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}_{x\sim p_{\text{real}}}[\log D(x)] + \mathbb{E}_{x\sim p_{\text{fake}}}[\log(1 - D(x))]. \tag{5}$$

A common practice is to freeze a cloned multi-step diffusion model as the discriminator backbone and train lightweight discriminator heads on top. We follow this architecture.

4 METHOD

4.1 OVERVIEW

Our approach follows a two-phase pipeline:

Phase 1: Few-Step Teacher Preparation. We assume access to an 8-step teacher model obtained through established distillation techniques. We use Z-Image-Turbo (Team, 2025), a 6B single-stream diffusion transformer distilled with Decoupled DMD (Liu et al., 2026) and DMDR (Jiang et al., 2026). The production of such few-step models is well-studied and increasingly standardized; we treat this phase as a given prerequisite.



Figure 2: Adversarial training with different real-sample sources. From top to bottom and left to right: 8-step samples generated by Z-Image-Turbo, 2-step results trained with external real images, and 2-step results trained with 8-step teacher-generated images (our adopted setting).

Phase 2: Ultra-Turbo Distillation. Starting from the 8-step teacher, we distill a 2-step generator through three synergistic techniques: Distribution-Aligned Adversarial Learning (§4.2), Step-Decoupled Parameterization (§4.3), and End-to-End Training with Iterative Regularization (§4.4). The overall training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{GAN}} + \lambda \mathcal{L}_{\text{DMD}}, \tag{6}$$

where $\mathcal{L}_{\text{GAN}}$ is the distribution-aligned adversarial loss and $\mathcal{L}_{\text{DMD}}$ provides complementary augmentation and regularization. Following the insights from Decoupled DMD, we set the renoising schedule as $\tau_{\text{CA}} = \tau_{\text{DM}} > t$, i.e., the renoising timestep is constrained to be *cleaner* than the input timestep. We do not employ the Decoupled-Hybrid schedule ($\tau_{\text{CA}} > t, \tau_{\text{DM}} \in [0, 1]$), because in our setting the adversarial objective already provides the anti-artifact effect typically supplied by the full-range DM term. Using a shared constrained schedule also avoids the additional score-model evaluation required by Decoupled-Hybrid. We refer readers to Sec. 4.3 of Liu et al. (2026) for details.

4.2 Distribution-Aligned Adversarial Learning

A key design choice in GAN-based distillation is the source of real samples for the discriminator. Since ADD (Sauer et al., 2024) and DMD2 (Yin et al., 2024a), the prevailing practice has been to use images from an external high-quality dataset. We deliberately depart from this default and use the **8-step teacher model's generated outputs** as real samples instead. Our point is not merely that a different choice is possible, but that the two options exhibit a systematic difference in the 2-step regime.

Fig. 2 and Fig. 3(a) compare the two options under an otherwise identical setup. Visually, training with external real images leads to systematic and pronounced artifacts, whereas teacher-generated real samples yield clean, high-quality results. The training dynamics show the same contrast: with teacher outputs as real samples, the generator's GAN loss first rises as the discriminator strengthens and then plateaus as the generator closes the gap; with external real images, the loss is substantially higher and keeps growing throughout training, indicating a divide the student cannot bridge.

This advantage is non-obvious: since real images are "more real" by construction, one might expect them to provide a stronger target distribution. We attribute the opposite outcome to distributional alignment: the 8-step teacher's outputs are much closer to the 2-step student's attainable distribution than external real images. With external images, the discriminator can rely on differences inherent to diffusion outputs versus natural photographs, such as texture and frequency statistics, rather than on differences that reflect generation quality. These deeply rooted differences are hard to eliminate during post-training, and forcing the student to close them can disrupt knowledge already encoded in the model. Once such persistent cues suffice to separate real from fake samples, the discriminator also has less incentive to expose more useful failure modes. Finally, since the 8-step teacher samples quickly, generating these real samples offline incurs only manageable cost.

4.3 Step-Decoupled Parameterization

Each denoising step can be viewed as a different generation task. In a many-step sampler, each step covers only a small interval of the trajectory and thus carries a light local burden, whereas few-step models must cover much larger intervals per step. This specialization reaches its extreme at 2 steps:

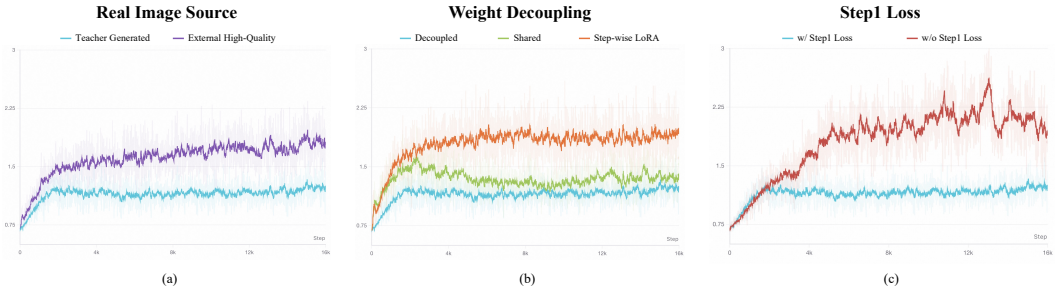

Figure 3: Generator GAN loss curves under different training settings.

the first step must construct a meaningful intermediate from near-pure noise, while the second must turn it into a clean image.

This raises a natural question: does model capacity become a bottleneck in the 2-step regime? Favorably, unlike a many-step sampler, the student only needs to handle two distinct tasks, which makes per-step parameter isolation feasible. We therefore initialize both steps' weights from the 8-step teacher and train them independently, effectively doubling the capacity dedicated to 2-step generation.

This simple design yields non-trivial improvements. As shown in Fig. 3(b), the step-2 generator GAN loss decreases substantially after decoupling. GAN loss is not a reliable metric across different settings, but its stable difference is informative in this controlled comparison, where only the parameterization changes. We also tested a weaker form of decoupling, in which the two steps share the backbone but use step-specific LoRA (Hu et al., 2022) modules. Its final GAN loss is even higher than that of a shared model, and its benchmark results are mixed (Tab. 1): it improves some generic metrics but remains clearly inferior to full decoupling on the most capacity-demanding, text-oriented ones. This suggests that the bottleneck involves both model capacity and multi-task interference: step-specific low-rank residuals partially reduce interference, but the shared backbone must still represent two substantially different denoising maps, whereas full decoupling alleviates both limitations.

Parameter decoupling doubles parameter storage but not per-step computation: each step still evaluates a single 6B model, so inference costs about a quarter of the 8-step teacher. Large-scale serving can pipeline the two models across devices with nearly unchanged throughput, which is almost costless for large models that already require distributed deployment. For smaller models deployed on device, however, the increased memory cost is a genuine drawback. Nevertheless, we view this cost as necessary for stronger 2-step quality and leave a better quality–storage balance to future work.

4.4 End-to-End Training with Iterative Regularization

End-to-end training becomes feasible at 2 steps. For multi-step models (>4 steps), end-to-end training through the entire generation chain is desirable but impractical because of prohibitive memory consumption and gradient instability. At 2 steps, the entire gradient path—from the initial noise through both step models to the final output and loss—is short and tractable, making full gradient tracking feasible.

End-to-end training provides two key advantages. The first is *direct optimization of the first step for final quality.* Each step determines compositional elements that later steps preserve; with only its local objective, the first step may make locally reasonable but globally suboptimal choices. End-to-end gradients let it resolve issues that originate in step 1 but only manifest after step 2. The second is *flexible resource coordination*: treating both steps as one system lets them implicitly coordinate their division of labor, improving capacity utilization.

4.4.1 The Necessity of Step-1 Loss

A natural question arises: if we optimize end-to-end for the final output, is a separate loss on the first step's own prediction still necessary? Note that this step-1 loss is applied to the first model's clean-image prediction, obtained by extrapolating its velocity to $t = 1$, rather than to the noisy



Figure 4: Visual comparison showing degraded quality when step-1 loss is removed.

intermediate state passed to step 2, because both the DMD and adversarial losses are defined on clean images; the intermediate state itself is optimized through the end-to-end gradient. Our initial hypothesis was that removing this constraint would free the model from redundant requirements and allow full capacity allocation to final quality.

Contrary to this intuition, experiments reveal that removing the step-1 loss causes the step-2 generator's GAN loss to surge, accompanied by visible degradation in generation quality (Fig. 4) and benchmark results (Tab. 1). The step-1 loss is therefore essential for stable and high-quality training.

We explain this behavior from a transfer-learning perspective. Final performance depends on (1) learning capacity for the downstream task and (2) the ability to leverage knowledge accumulated during pretraining. Diffusion models have a deeply ingrained *iterative nature*: each step produces a meaningful intermediate that the next step builds upon. Removing the step-1 loss increases flexibility (factor 1) but disrupts this pattern (factor 2): as shown in Fig. 4, the intermediate output degenerates into a low-quality representation that is poorly aligned with the inductive biases encoded in the pretrained weights. An explicit step-1 loss helps the model *preserve its familiar iterative generation mode*, enabling more effective transfer; in this regime, transfer efficiency matters more than unconstrained learning flexibility.

4.4.2 Important Implementation Details

Our implementation keeps the end-to-end signal while avoiding the peak memory of a fully connected two-step graph. After the first-step velocity prediction, [first-step local loss] and [second-step $\rightarrow$ final loss] form two independent branches. We first backpropagate the second branch through a detached clone of the first-step prediction, store the resulting gradient, and then inject it into the first-step model via an *inherit loss* added to step 1's local loss, so the two branches never occupy memory simultaneously. We scale the inherited term by 0.1 to prevent gradient explosion. With per-block gradient checkpointing and FSDP, memory stays within a practical range; Appendix A gives the pseudo-code.

5 Experiments

Qualitative Results. Fig. 1 presents representative images generated by Z-Image Ultra-Turbo using only two inference steps. Despite this extremely compressed sampling budget, the model produces rich details, sharp textures, and strong text-rendering ability. In particular, it preserves most of the image quality and photorealistic appearance for which the original Z-Image model is known. Fig. 5 further compares our method against two important baselines: the original 8-step Z-Image Turbo model and TwinFlow (Cheng et al., 2025), a recent distillation method with an official 2-step checkpoint. Visually, both Z-Image Ultra-Turbo and TwinFlow retain basic generation capability, but our model shows clear advantages in three aspects: better preservation of global coherence and realistic style, more faithful reproduction of high-frequency details, and fewer systematic artifacts. The difference is especially pronounced in text generation, a challenging and discriminative dimension for ultra-few-step compression: TwinFlow exhibits substantial degradation, whereas our model maintains considerably higher text quality.



Figure 5: Qualitative comparison among 8-step Z-Image-Turbo, TwinFlow, and our Z-Image Ultra-Turbo. Our method achieves a better quality-efficiency trade-off.

Table 1: Overall performance comparison across benchmarks.

Idx	Model	NFE	OneIG	GenEval	DPG-Bench	LongText-CN	LongText-EN
Component Ablation							
①	Baseline	2	51.72	**77.50**	84.74	87.13	83.29
②	① + Teacher as Real	2	51.89	72.72	84.86	87.16	86.03
③	② + Decoupling Weight	2	52.15	74.67	85.39	89.27	85.81
④	③ + End-to-End Training (Ours)	2	**52.50**	75.70	**85.86**	**91.62**	**89.88**
Weight Decoupling Ablation							
④	Ours	2	**52.50**	75.70	85.86	**91.62**	**89.88**
⑤	④ w/ Shared Weight	2	50.67	73.62	85.51	87.64	81.14
⑥	④ w/ Per-Step LoRA	2	50.96	**76.10**	**86.58**	80.71	76.90
Step-1 Loss Ablation							
④	Ours	2	**52.50**	**75.70**	**85.86**	**91.62**	**89.88**
⑦	④ w/o Step-1 Loss	2	50.16	71.02	83.98	84.49	82.07
Comparative							
④	Ours	2	**52.50**	75.70	85.86	**91.62**	**89.88**
⑧	Twin Flow	2	51.38	72.41	**85.98**	78.65	71.99
⑨	DMD2	2	50.70	76.12	85.40	85.30	80.99
⑩	Z-Image-Turbo	2	50.94	**76.53**	85.78	78.48	72.89
Reference							
⑪	Z-Image-Turbo	8	52.84	75.01	84.86	92.56	91.74

Quantitative Results. We evaluate on four standard benchmarks: DPGBench (Hu et al., 2024), GenEval (Ghosh et al., 2023), OneIGBench (Chang et al., 2025), and LongTextBench (Geng et al., 2025). All prompts are taken directly from the official benchmark sets without prompt enhancement. The main results are summarized in Table 1. Since OneIGBench provides the most comprehensive coverage, we report its detailed breakdown in Table 2; detailed results for the remaining benchmarks are provided in the appendix.

Which metrics reflect 2-step fidelity? Not every benchmark is equally informative in our setting. GenEval and DPG-Bench are prompt-following metrics: they probe the presence, count, color, and spatial relations of major objects. As Fig. 5 shows, existing 2-step methods already preserve such coarse structure reasonably well; where they fall short is mid/high-frequency content, i.e., fine detail, texture, and legible text. A simple control confirms that these two benchmarks are blind to this failure mode: running the unmodified 8-step Z-Image-Turbo for only 2 steps (⑩) *raises* GenEval (76.53 vs. 75.01) and DPG-Bench (85.78 vs. 84.86) over its 8-step counterpart (⑪), although its samples are visibly much worse. Coarse layout is fixed within the first one or two denoising steps, so reducing the step count barely affects what these benchmarks measure. Their variation is also dominated by

Table 2: Detailed results on OneIG benchmark.

Idx	Model	NFE	Overall	Alignment	Text	Diversity	Style	Reasoning
Component Ablation								
①	Baseline	2	51.72	85.10	94.79	**13.14**	36.93	28.65
②	① + Teacher as Real	2	51.89	85.00	96.46	12.70	36.14	29.16
③	② + Decoupling Weight	2	52.15	84.98	96.49	13.05	37.18	29.07
④	③ + End-to-End Training (Ours)	2	**52.50**	**85.65**	**97.09**	11.75	**37.87**	**30.12**
Weight Decoupling Ablation								
④	Ours	2	**52.50**	**85.65**	**97.09**	11.75	**37.87**	**30.12**
⑤	④ w/ Shared Weight	2	50.67	84.68	92.35	11.10	36.79	28.44
⑥	④ w/ Per-Step LoRA	2	50.96	85.61	91.20	**12.54**	37.32	28.12
Step-1 Loss Ablation								
④	Ours	2	**52.50**	**85.65**	**97.09**	**11.75**	**37.87**	**30.12**
⑦	④ w/o Step-1 Loss	2	50.16	82.70	94.69	11.46	33.67	28.27
Comparative								
④	Ours	2	**52.50**	**85.65**	**97.09**	11.75	37.87	**30.12**
⑧	Twin Flow	2	51.38	85.26	90.61	**15.49**	**38.03**	27.52
⑨	DMD2	2	50.70	85.21	90.82	12.75	36.77	27.93
⑩	Z-Image-Turbo	2	50.94	83.92	92.85	13.85	36.25	27.86
Reference								
⑪	Z-Image-Turbo	8	52.84	84.03	99.32	13.85	36.81	30.19

non-essential factors: across our eleven settings GenEval has a standard deviation of only about 1.9, whereas merely rewriting its short prompts into longer, more detailed captions raises the 8-step model from about 75 to 82. *We therefore do not treat small fluctuations on GenEval and DPG-Bench as indicators of image quality in the 2-step regime*, but as a safeguard against losing coarse semantic alignment, on which our model stays on par with the teacher (75.70 vs. 75.01; 85.86 vs. 84.86).

Text rendering, by contrast, becomes an especially meaningful dimension in this setting—not because text rendering is our goal per se, but because legible text requires exactly the high-frequency fidelity that 2-step generation struggles to retain, making it a sensitive probe of the central challenge that separates 2-step methods. Accordingly, the same control pair differs sharply here (LongText-CN/EN: 78.48/72.89 vs. 92.56/91.74; OneIG-Text: 92.85 vs. 99.32). On this dimension, our model improves substantially over existing 2-step methods, e.g., LongText-EN 89.88 vs. 71.99 for TwinFlow, 80.99 for our reimplementation of DMD2, and 72.89 for direct 2-step inference, and comes much closer to the 8-step teacher (91.74). This matches the sharper details and text we observe in Fig. 5, and a blind pairwise human study (3 annotators, 450 judgments on matched prompt/seed pairs; Appendix B.2) confirms the ranking: annotators prefer our model over TwinFlow and DMD2 in 82% and 74% of judgments, respectively, and rate it tied with or better than the 8-step teacher in 52%.

Ablation summary. The three components jointly improve the text-sensitive metrics (①→④). The comparison between ③ and ⑤ suggests that end-to-end training relies heavily on weight decoupling, likely because jointly optimized steps demand stronger functional specialization. Per-step LoRA (⑥) still falls short of full decoupling on text generation, supporting a capacity bottleneck, and removing the step-1 loss (⑦) degrades every benchmark (Fig. 4).

6 Conclusion and Limitations

In this work, we introduce **Z-Image Ultra-Turbo**, a 2-step image generation model distilled from the 8-step Z-Image Turbo teacher. By combining Distribution-Aligned Adversarial Learning, Step-Decoupled Parameterization, and End-to-End Training with Iterative Regularization, our method addresses the stability, capacity, and knowledge-preservation challenges of the 2-step regime. The main limitation is additional parameter storage. Besides, while our model substantially narrows the gap between 2-step and 8-step generation, challenging cases such as dense text rendering, secondary objects, and complex-scene generation remain less reliable than the 8-step teacher, and, as is common for distribution-matching distillation, sample diversity is somewhat reduced (Table 2). Finally, our end-to-end training framework may provide a useful foundation for reinforcement learning-based optimization of few-step generators, but we leave this direction to future work.

AI USE STATEMENT

We used generative AI tools only for language editing and polishing of the text and for formatting adjustments. All AI-assisted edits were reviewed by the authors, and we take full responsibility for the final content of this work, including text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

The paper describes the full algorithmic design, including the training objective (Sec. 4.1), step-decoupled parameterization (Sec. 4.3), and the memory-efficient inherited-gradient implementation (Sec. 4.4.2, with pseudo-code in Appendix A). Appendix B.1 specifies the optimizer, learning rate, Adam betas, weight decay, update ratio, number of iterations, EMA, global batch size, teacher-sample preparation, loss weights, and the compute setup (16 H100 GPUs, approximately 80 hours of training). All evaluations use the official prompt sets of four public benchmarks without prompt enhancement. Code and model assets will be released upon acceptance.

REFERENCES

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Jingjing Chang, Yixiao Fang, Peng Xing, Shuhan Wu, Wei Cheng, Rui Wang, Xianfang Zeng, Gang Yu, and Hai-Bao Chen. Oneig-bench: Omni-dimensional nuanced evaluation for image generation. *arXiv preprint arXiv:2506.07977*, 2025.

Zhenglin Cheng, Peng Sun, Jianguo Li, and Tao Lin. Twinflow: Realizing one-step generation on large models with self-adversarial flows. *arXiv preprint arXiv:2512.05150*, 2025.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models, 2023. URL https://arxiv.org/abs/2305.10924.

Zhida Feng, Zhenyu Zhang, Xintong Yu, Yewei Fang, Lanxin Li, Xuyi Chen, Yuxiang Lu, Jiaxiang Liu, Weichong Yin, Shikun Feng, Yu Sun, Li Chen, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE-ViLG 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. In *CVPR*, 2023.

Alireza Ganjdanesh, Yan Kang, Yuchen Liu, Richard Zhang, Zhe Lin, and Heng Huang. Mixture of efficient diffusion experts through automatic interval and sub-network selection. In *ECCV*, 2024.

Zigang Geng, Yibing Wang, Yeyao Ma, Chen Li, Yongming Rao, Shuyang Gu, Zhao Zhong, Qinglin Lu, Han Hu, Xiaosong Zhang, et al. X-omni: Reinforcement learning makes discrete autoregressive image generative models great again. *arXiv preprint arXiv:2507.22058*, 2025.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36:13237–13249, 2023.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, David Liu, Changsheng Lu, Zhen Li, Bo Zhang, Mengmeng Wang, Steven Hoi, Peng Gao, and Harry Yang. Distribution matching distillation meets reinforcement learning, 2026. URL https://arxiv.org/abs/2511.13649.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Dongyang Liu, Peng Gao, David Liu, Ruoyi Du, Zhen Li, Qilong Wu, Xin Jin, Sihan Cao, Shifeng Zhang, Steven HOI, and Hongsheng Li. Decoupled DMD: CFG augmentation as the spear, distribution matching as the shield. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jBztvOiCKE.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7353–7363, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15762–15772, 2024.

Kidist Amde Mekonnen, Nicola Dall'Asen, and Paolo Rota. Adv-kd: Adversarial knowledge distillation for faster diffusion sampling. *arXiv preprint arXiv:2405.20675*, 2024.

ModelTC. Qwen-image-lightning, 2025. URL https://github.com/ModelTC/LightX2V-Qwen-Image-Lightning.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *Advances in Neural Information Processing Systems*, 37:117340–117362, 2024.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep distillation of diffusion models via moment matching. *Advances in Neural Information Processing Systems*, 37: 36046–36070, 2024.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1972–1981, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

Z-Image Team. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Wan Team. Wan2.2: Open and advanced large-scale video generative models. https://github.com/Wan-Video/Wan2.2, 2025.

Fu-Yun Wang, Zhaoyang Huang, Alexander Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency models. *Advances in neural information processing systems*, 37:83951–84009, 2024.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024b.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

A PSEUDO-CODE FOR MEMORY-EFFICIENT TRAINING

```
def naive_generator_training(step0_model, step1_model,
                             step0_stride, step1_stride,
                             caption, noise):
    assert step0_stride + step1_stride == 1.0

    step0_v_prediction = step0_model(
        noise, caption, time_cond=0.0)
    step0_x_prediction = noise + step0_v_prediction * 1.0

    step1_input = noise + step0_v_prediction * step0_stride
    step1_v_prediction = step1_model(
        step1_input, caption, time_cond=step0_stride)
    step1_x_prediction = step1_input + \
        step1_v_prediction * step1_stride

    # Loss computation also requires teacher-generated images
    # and other training states, omitted with ... for simplicity.
    step0_loss = GAN_LOSS(step0_x_prediction, ...) + \
                 DMD_LOSS(step0_x_prediction, ...)
    step1_loss = GAN_LOSS(step1_x_prediction, ...) + \
                 DMD_LOSS(step1_x_prediction, ...)

    (step0_loss + step1_loss).backward()

def memory_efficient_generator_training(
        step0_model, step1_model, step0_stride, step1_stride,
        caption, noise, inherit_weight=0.1):
    assert step0_stride + step1_stride == 1.0

    step0_v_prediction = step0_model(
        noise, caption, time_cond=0.0)

    # First handle the path from step0_v_prediction to step1_loss.
    # detach() cuts the graph here; the gradient is later
    # passed back to step0_model through inherit_loss.
    step0_v_detached = step0_v_prediction.clone().detach()
    step0_v_detached.requires_grad_(True)

    step1_input = noise + step0_v_detached * step0_stride
    step1_v_prediction = step1_model(
        step1_input, caption, time_cond=step0_stride)
    step1_x_prediction = step1_input + \
        step1_v_prediction * step1_stride

    step1_loss = GAN_LOSS(step1_x_prediction, ...) + \
                 DMD_LOSS(step1_x_prediction, ...)
    step1_loss.backward()

    # Then handle step0's own loss and inherited step1 gradient.
    step0_x_prediction = noise + step0_v_prediction * 1.0
    inherit_loss = torch.sum(
        step0_v_prediction * step0_v_detached.grad.detach())

    step0_loss = GAN_LOSS(step0_x_prediction, ...) + \
                 DMD_LOSS(step0_x_prediction, ...)

    # Directly passing the full gradient to step0_model can
    # lead to gradient explosion; inherit_weight modulates it.
    total_step0_loss = step0_loss + inherit_weight * inherit_loss
    total_step0_loss.backward()
```

Listing 1: Pseudo-code for the naive two-step generator update and our memory-efficient implementation with inherited gradients.

B More Experimental Details

B.1 Experiment Settings

For all trainable models, we use the Adam optimizer with a learning rate of 1×10^{-5}, $\beta_1 = 0.0$, $\beta_2 = 0.9$, and no weight decay. Following the TTUR strategy in DMD2 (Yin et al., 2024a), we update the generator and the guidance model with a frequency ratio of 1:5. The full training runs for 20,000 iterations, and we maintain an exponential moving average of the generator parameters with a decay rate of 0.99.

Before training, we pre-generate samples from the 8-step teacher and use them as the teacher distribution for subsequent training. All experiments are conducted on 16 H100 GPUs with a global batch size of 64. Under our implementation, the complete training process takes approximately 80 hours.

For the loss weights, we set both the DMD loss on the generator and the corresponding diffusion loss on the guidance model to 1×10^{-2}. The GAN generator loss and discriminator loss are both weighted by 1×10^{-3}, and the inherit loss weight is set to 0.1.

B.2 Human Preference Study

Because the central differences among 2-step models lie in fidelity, text legibility, and artifacts that automatic metrics capture incompletely, we conducted a randomized pairwise human evaluation with three annotators. Each annotator judged 150 image pairs: 50 comparing our model with TwinFlow (2 steps), 50 with DMD2 (2 steps), and 50 with the 8-step Z-Image-Turbo teacher. The two images in each pair were generated from the same prompt, random seed, and initial noise; since all compared models derive from the same Z-Image model family, this matched setup yields broadly similar content and makes differences in fine detail, texture, and artifacts easier to observe. Comparison order and left/right placement were randomized, model names were hidden, and annotators selected the better image or indicated a tie, yielding 450 judgments in total. Table 3 reports the results from the perspective of our model. Annotators clearly prefer our model over both 2-step baselines. The 8-step teacher remains preferred overall, confirming that a quality gap remains, yet our 2-step model is rated tied with or better than the teacher in 52% of judgments.

Table 3: Pairwise human preference study (450 judgments from three annotators), reported from the perspective of our model.

Comparison	Win	Tie	Loss
Ours vs. TwinFlow (2 steps)	123	12	15
Ours vs. DMD2 (2 steps)	111	15	24
Ours vs. Z-Image-Turbo (8 steps)	24	54	72

Table 4: Detailed results on DPG-Bench benchmark.

Idx	Model	NFE	overall	attribute	global	relation	other	entity
Component Ablation								
①	Baseline	2	84.74	89.08	83.28	93.62	86.40	91.10
②	① + Teacher as Real	2	84.86	88.90	82.98	93.73	87.60	91.11
③	② + Decoupling Weight	2	85.39	**90.47**	**91.32**	91.03	**89.83**	91.58
④	③ + End-to-End Training (Ours)	2	**85.86**	89.30	83.89	**94.58**	89.60	**91.84**
Weight Decoupling Ablation								
④	Ours	2	85.86	**89.30**	83.89	94.58	**89.60**	91.84
⑤	④ w/ Shared Weight	2	85.51	88.88	82.07	94.00	89.20	91.31
⑥	④ w/ Per-Step LoRA	2	**86.58**	89.26	**84.80**	**94.89**	86.80	**92.37**
Step-1 Loss Ablation								
④	Ours	2	**85.86**	**89.30**	**83.89**	**94.58**	**89.60**	**91.84**
⑦	④ w/o Step-1 Loss	2	83.98	88.28	80.85	93.08	86.40	90.34
Comparative								
④	Ours	2	85.86	**89.30**	83.89	**94.58**	**89.60**	**91.84**
⑧	Twin Flow	2	**85.98**	89.02	81.46	94.16	87.60	91.83
⑨	DMD2	2	85.40	88.88	83.59	93.81	85.60	91.10
⑩	Z-Image-Turbo	2	85.78	88.78	**84.19**	94.16	**89.60**	91.73
Reference								
⑪	Z-Image-Turbo	8	84.86	90.14	91.29	92.16	88.68	89.59

Table 5: Detailed results on GenEval benchmark.

Idx	Model	NFE	overall	two_object	color_attr	position	counting	colors	single_object
Component Ablation									
①	Baseline	2	**77.50**	**88.64**	**67.75**	**51.25**	**70.94**	**86.44**	**100.00**
②	① + Teacher as Real	2	72.72	87.63	61.50	43.50	57.81	85.90	**100.00**
③	② + Decoupling Weight	2	74.67	87.12	62.75	46.00	68.12	84.04	**100.00**
④	③ + End-to-End Training (Ours)	2	75.70	87.63	64.50	48.75	69.06	84.57	99.69
Weight Decoupling Ablation									
④	Ours	2	75.70	87.63	**64.50**	48.75	69.06	84.57	99.69
⑤	④ w/ Shared Weight	2	73.62	84.09	61.25	44.00	70.00	82.98	99.38
⑥	④ w/ Per-Step LoRA	2	**76.10**	**90.40**	59.25	**50.00**	**71.56**	**85.37**	**100.00**
Step-1 Loss Ablation									
④	Ours	2	**75.70**	**87.63**	**64.50**	**48.75**	**69.06**	**84.57**	**99.69**
⑦	④ w/o Step-1 Loss	2	71.02	82.83	61.25	45.50	54.69	82.18	**99.69**
Comparative									
④	Ours	2	75.70	87.63	64.50	48.75	69.06	84.57	99.69
⑧	Twin Flow	2	72.41	87.12	54.75	40.00	67.81	85.11	99.69
⑨	DMD2	2	76.12	**90.40**	**65.50**	49.00	67.50	84.31	**100.00**
⑩	Z-Image-Turbo	2	**76.53**	89.39	57.75	**50.50**	**74.38**	**87.77**	99.38
Reference									
⑪	Z-Image-Turbo	8	75.01	88.89	58.75	45.75	71.88	85.11	99.69