DC-AR: Efficient Masked Autoregressive Image Generation with Deep Compression Hybrid Tokenizer

Supplementary Material

A. Appendix

We provide additional information and results in the appendix, as outlined below:

- Appendix A.1: Ethics Statement, discussing about how we prevent the misuse of DC-AR.
- Appendix A.2: Implementation Details, including the training hyper-parameters for tokenizer and generator, inference hyper-parameters for generator.
- Appendix A.3: Additional text-to-image generation of DC-AR and other popular methods.
- Appendix A.4: Qualitative comparison between DC-AR and the discrete-only baseline.
- Appendix A.5: Additional experiments to help clarfiy the advantages of DC-AR.

A.1. Ethics Statement

The misuse of generative AI for creating NSFW (not safe for work) content continues to be a critical concern within the community. To address this, we have integrated DC-AR with ShieldGemma-2B [72], a robust LLM-based safety check model. In our implementation, user prompts are first evaluated by the safety check model to detect NSFW content, including harmful, abusive, hateful, sexually explicit, or otherwise inappropriate material targeting individuals or protected groups. If a prompt is deemed safe, it is forwarded to DC-AR for image generation. If not, the prompt is rejected and replaced with a default prompt ("A red heart"). Through rigorous testing, we have demonstrated that our safety check model effectively filters out NSFW prompts under strict thresholds, ensuring that our pipeline does not produce harmful AI-generated content.

A.2. Implementation Details

Tab. 7 and Tab. 8 present the hyper-parameters used for training the tokenizer and generator, respectively. For image generation, we employ the following sampling hyper-parameters: a randomized temperature of 4.5, a CFG (Classifier-Free Guidance) scale of 4.5, a constant CFG schedule, 12 sampling steps for discrete tokens, and 20 diffusion steps for residual tokens.

Hyper-parameters	Configuration
optimizer	Adam
β_1	0.9
β_2	0.95
discriminator loss weight	0.5
perceptual loss weight	1.0
L_1 loss weight	0.0
L_2 loss weight	1.0
weight decay	0.0
learning rate	1e-4
lr schedule	constant
batch size	128
training epochs (continuous warm-up)	10
training epochs (discrete learning)	40
training epochs (alternate fine-tuning)	10

Table 7. Training hyper-parameters for our tokenizer.

Hyper-parameters	Configuration
optimizer	Adamw
β_1	0.9
β_2	0.96
condition dropout	0.1
attention dropout	0.1
cross-entropy loss weight	1.0
diffusion loss weight	1.0
weight decay	0.03
learning rate	1e-4
lr schedule	cosine
batch size (256×256)	1024
batch size (512×512)	1024
training steps (256×256)	200K
training steps (512×512)	50K

Table 8. Training hyper-parameters for our generator.

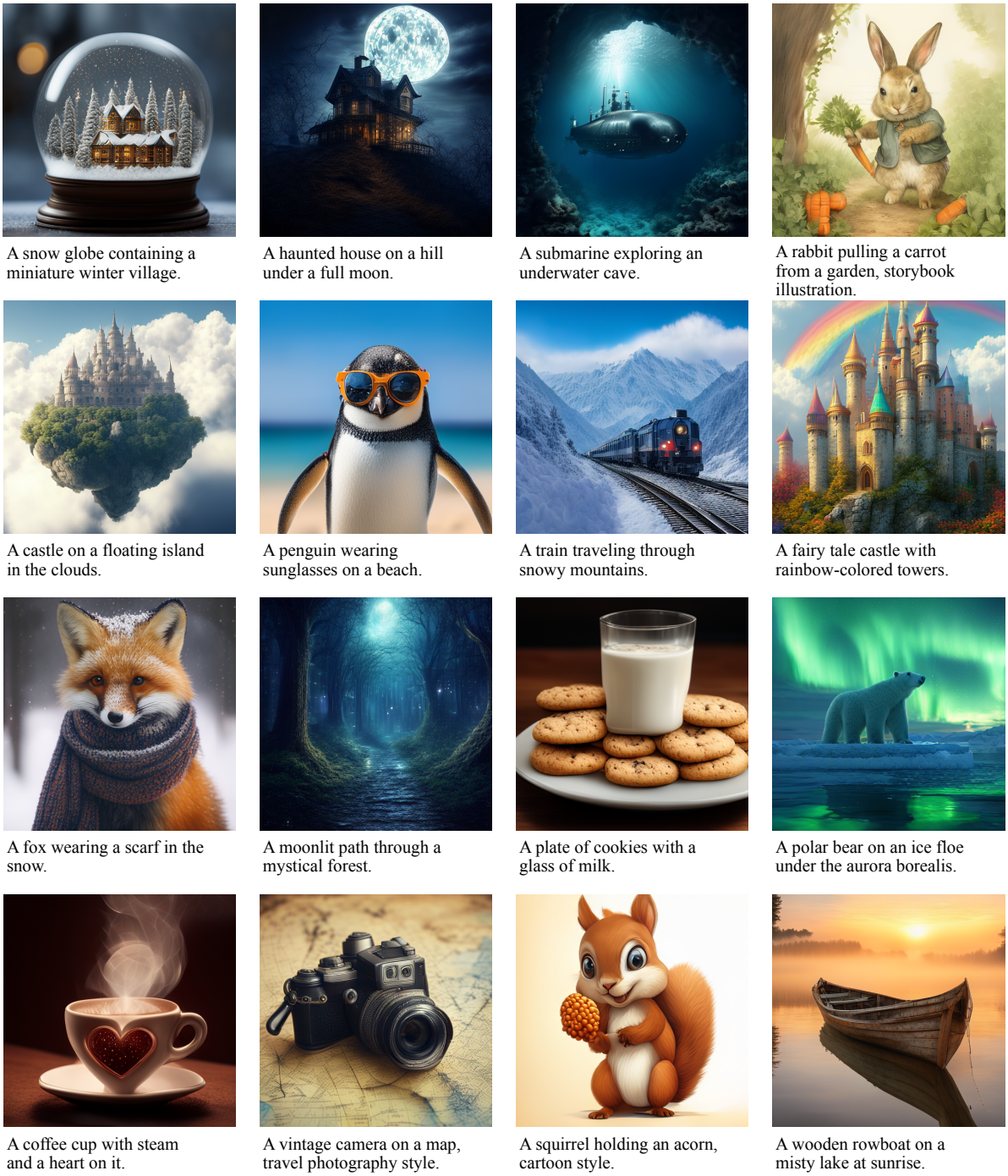


Figure 7. **Additional text-to-image generation results of DC-AR.**

A.3. Additional Text-to-image Examples

Fig. 7 and Fig. 8 includes more qualitative examples of text-to-image generation results of DC-AR.

A close-up photo of a honeycomb with bees actively working, golden honey visible in cells, wings a blur of movement.

Underwater tea party with mermaids and sea turtles, coral reef in background.

Tiny house inside a terrarium, miniature garden with working lights, tilt-shift photography.

A close-up photo of a lotus flower emerging from muddy water, perfect pink petals opening toward sunlight, water droplets visible.

A 4k dslr image of a lemur wearing a red magician hat and a blue coat performing magic tricks with cards in a garden.

Robot barista making coffee in a steampunk café, brass pipes and gears visible.

A photo of a bonsai tree in a handcrafted ceramic pot, perfectly pruned, sitting by a window with rain droplets visible.

A silverback gorilla sitting thoughtfully in misty mountain forest, massive hands gently examining a small flower, rain-dampened fur glistening.

Dragon made of constellation stars flying across night sky, over mountain landscape.

Astronaut discovering alien flowers on distant planet, sci-fi concept art, dramatic lighting.

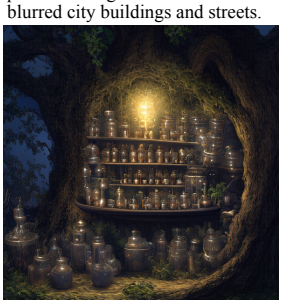
A close-up photo of frost patterns forming intricate crystalline structures on a red maple leaf, backlit by early morning sun.

A close-up photo of a woman. She wore a blue coat. She has blue eyes and blond hair, and wears a pair of earrings. Behind are blurred city buildings and streets.

Post-apocalyptic greenhouse preserving Earth's last plant species, tended by robots, with the ruined cityscape visible through cracked glass panels.

Cosmic lighthouse keeper's cottage surrounded by aurora waves, collecting stardust in glass jars, with a telescope tracking wandering celestial bodies.

Crystalline city floating among clouds, connected by rainbow bridges, with inhabitants riding winged creatures between iridescent spires.



Witch's apothecary nestled in a hollow tree, filled with bubbling potions, sentient plants, and familiars organizing ingredients by moonlight.

Figure 8. **Additional text-to-image generation results of DC-AR.**

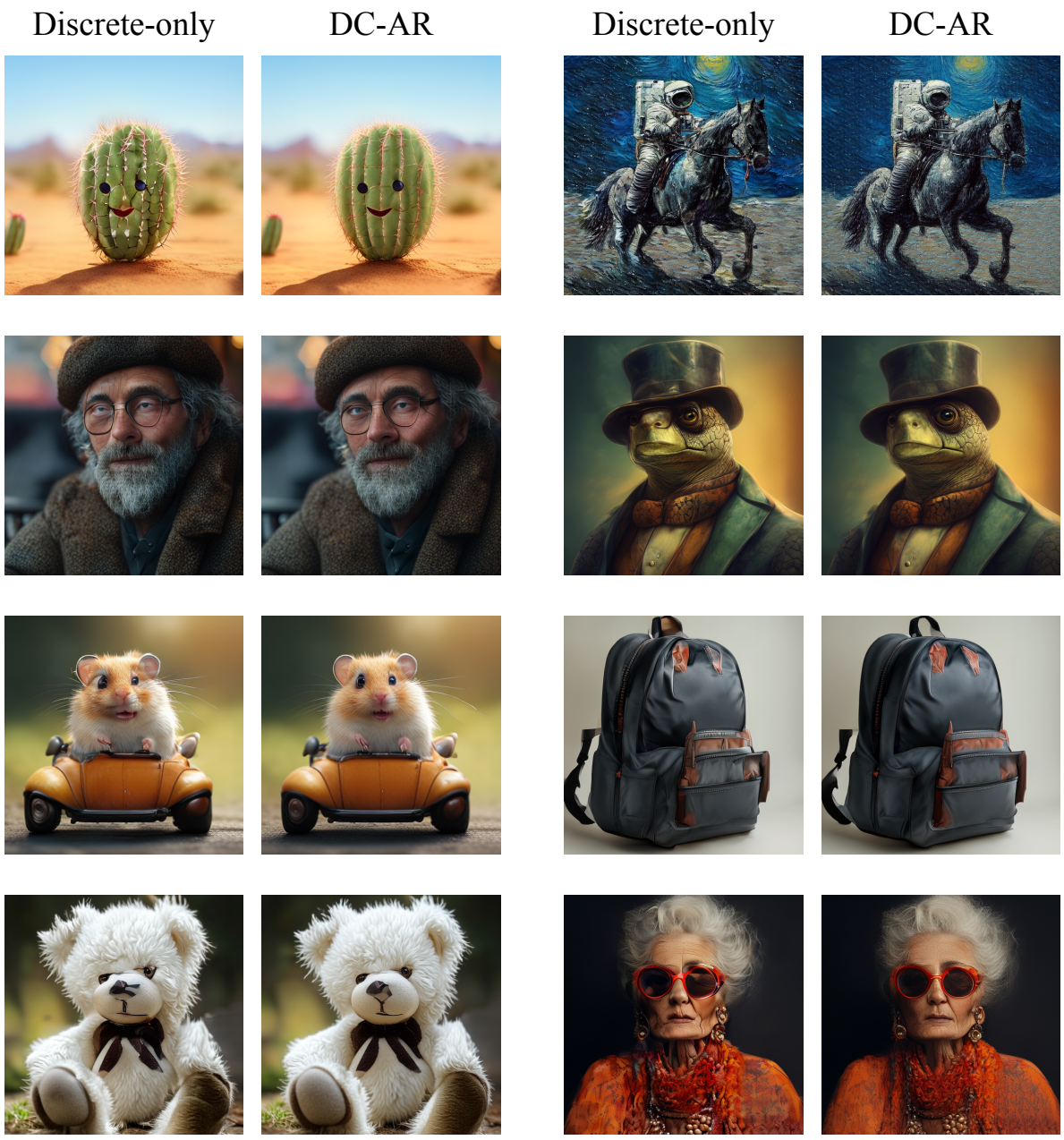


Figure 9. **Qualitative Comparison: Images Generated by DC-AR vs. the Discrete-Only Baseline.** For each pair of images, the left image is produced by the discrete-only baseline, while the right image is generated by DC-AR.

A.4. Qualitative Comparison of DC-AR and discrete-only baseline.

We present qualitative comparison examples of images generated by DC-AR and the discrete-only baseline. From these examples, it is evident that the diffusion head and residual tokens significantly enhance image refinement, particularly in capturing fine details such as eyes and textures.

A.5. Additional Experimental Results.

In this section, we provide some other experiments related to DC-AR.

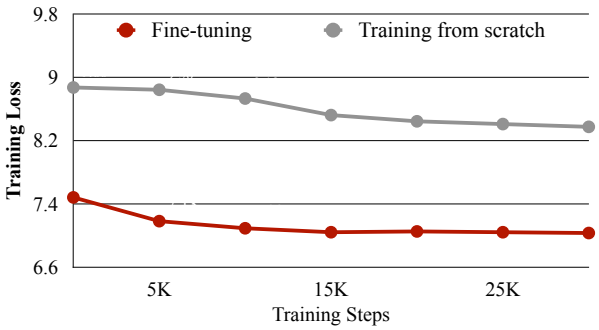


Figure 10. The resolution generalizability of DC-HT allows us to train a 512×512 model by fine-tuning from a pre-trained 256×256 model, achieving significantly faster convergence compared to training from scratch.

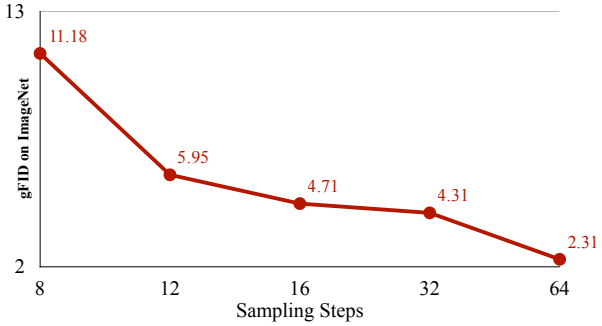


Figure 11. **gFID Results on ImageNet 256×256 for MAR-B at Different Sampling Steps.** MAR-B requires 64 sampling steps to achieve its best performance, significantly lagging behind our method, which attains optimal performance in just 12 steps.

Training Loss Curve: Fine-Tuning vs. Training from Scratch. Fig. 10 illustrates the training loss curves for fine-tuning and training from scratch on 512×512 models over the first 30K steps. It is evident that fine-tuning from a pre-trained 256×256 model enables the 512×512 model to converge significantly faster than training from scratch.

Sampling Step Requirements for MAR. Our primary motivation for adopting a hybrid generation framework, rather than following MAR's paradigm of exclusively using continuous tokens, stems from the observation that MAR typically requires a large number of steps to achieve optimal performance. This is demonstrated in Fig. 6, where we evaluate the official MAR-B model for class-conditional image generation on ImageNet at 256×256 resolution. Despite the image token sequence length being 256, MAR-B requires 64 steps to reach its optimal performance, resulting in a substantial inference cost. In contrast, DC-AR achieves optimal performance in just 12 steps, making it significantly more efficient during sampling.

A.6. Discussion of DC-AR and Related Works.

As a novel autoregressive image generation framework, DC-AR draws inspiration from several related works in the field while introducing significant innovations that distinguish it from each of them.

Difference with MaskGen [31]. Both MaskGen and DC-AR adopt the masked autoregressive generation paradigm for text-to-image generation and employ an image tokenizer with a high compression ratio for efficient generation. However, their technical approaches to building the tokenizer and generator differ substantially. On the tokenizer side, MaskGen follows the recent trend of using a 1D compact tokenizer to achieve a high compression ratio. However, a major limitation of such 1D tokenizers is their lack of generalizability across different resolutions. Consequently, MaskGen must train separate tokenizers and generators from scratch for each resolution, leading to significantly higher training costs, especially for resolutions of 512×512 or higher. In contrast, DC-AR utilizes a single tokenizer trained on 256×256 images for all resolutions and fine-tunes the generator for higher resolutions from a pre-trained low-resolution model, resulting in much greater efficiency. On the generator side, MaskGen combines the MaskGIT paradigm for discrete token generation with the MAR paradigm for continuous token generation. In contrast, DC-AR introduces a novel hybrid generation framework that leverages the superior representation capability of continuous tokens while maintaining the high inference speed of discrete tokens.

Difference with HART [53]. HART proposes the idea of hybrid tokenization, using a transformer model to generate discrete tokens and a lightweight diffusion head to generate continuous tokens. While DC-AR inherits these ideas, it adapts them in a fundamentally different setting. HART follows the VAR paradigm, which generates images through progressive next-scale refinement. In contrast, DC-AR adopts the MaskGIT paradigm, which generates images through progressive unmasking. Although the VAR paradigm is widely recognized for its high generation quality and speed, we believe the MaskGIT paradigm offers unique advantages, including fewer tokens (VAR requires additional tokens due to its multi-scale tokenization design) and a natural suitability for image editing tasks. Building on this foundation, DC-AR introduces novel methods, such as a single-scale hybrid tokenizer with a 32× compression ratio (via our three-stage adaptation strategy) and an efficient hybrid generation framework that extends MaskGIT (via our discrete token-dominated generation pipeline). Notably, in the results section, we do not include comparisons with VAR-based methods, as we aim to focus the discussion

on how DC-AR advances the MaskGIT paradigm. In future work, we plan to explore adapting our approach to the VAR paradigm to design even more effective generation frameworks.