CycleVAR: Repurposing Autoregressive Model for Unsupervised One-Step Image Translation

Supplementary Material

1. Multi-Step Serial Generation

Algorithm 1 Multi-Step Serial Generation

```
Require: Source features $\{F_k\}_{k=1}^{K}$, class condition $t$, class embedding of $s$
Ensure: Final output $\hat{H}_K$
Initialize $\hat{H}_0 \leftarrow s$
for $k = 1$ to $K$ do
    $\tilde{G}_{k-1} \leftarrow$ VARTransformer$(\hat{H}_{k-1}, (\hat{H}_0, \dots, \hat{H}_{k-1}), t)$
    $\hat{R}_{k-1} \leftarrow$ Quantization$(\tilde{G}_{k-1})$
    $\hat{F}_k \leftarrow a \cdot (\hat{R}_{k-1} + \hat{H}_{k-1}) + (1-a) \cdot F_k$
    if $k < K$ then
        $\hat{H}_k \leftarrow$ Upsample$(\hat{F}_k, (h_{k+1}, w_{k+1}))$
    else
        $\hat{H}_k \leftarrow \hat{F}_k$
    end if
end for
return $\hat{H}_K$
```

The complete process of Multi-Step Serial Generation is shown in Algorithm 1.

2. CycleVAR w/ Infinity

Instead of using the original vector quantizer of vanilla VAR, Infinity uses a dimension-independent bitwise quantizer, which is implemented by LFQ and BSQ. The input feature $f^{(i,j)}$ is quantized to $q^{(i,j)}$ as follows:

$$q^{(i,j)} = \mathcal{Q}_{sign}(f^{(i,j)}) = \begin{cases} \text{sign}(f^{(i,j)}) & \text{if LFQ} \\ \frac{1}{\sqrt{d}}\text{sign}(\frac{f^{(i,j)}}{|f^{(i,j)}|}) & \text{if BSQ} \end{cases} \tag{17}$$

where $\text{sign}(\cdot)$ is the piecewise function that extracts the sign of a real number. $V = 2^d$ for the bitwise quantizer. Instead of using the index-wise classifier like VAR, the casual VAR transformer of Infinity predicts features' labels with d binary classifiers in parallel to predict whether each dimension of $\tilde{f}^{(i,j)}$ is positive or negative.

When implementing Softmax Relaxed Quantization with the bitwise quantizer, a comparable method is employed: The non-differentiable sampling operation based on logits after the classifier is substituted with softmax applied to the prediction of each binary classifier.

3. Training Details

We employed two models with the "next-scale" prediction paradigm as baselines: the class-conditional image generation model VAR [44] and the text-conditional generation model Infinity [8], with Infinity being an extension of VAR.

CycleVAR w/ VAR. When translating horse to zebra, we assign the class label of zebra as t for input to the casual VAR Transformer. In contrast, the casual VAR Transformer incorporates the horse class label as condition t.

CycleVAR w/ Infinity. In CycleVAR w/ Infinity, the text prompt serves as the condition t. The associated text in the horse ↔ zebra dataset is "picture of a horse" and "picture of a zebra." Likewise, in the day ↔ night dataset, the text is "driving in the day" and "driving in the night." As for the Anime Scene Dataset, the text corresponds to "a Makoto Shinkai style landscape" and "a photo of a real landscape."

4. Inference Time

Table 6. Inference Time Comparison. The unit of time is seconds.

Method	Time	Method	Time
CycleGAN	0.004	SDEdit	1.900
CUT	0.004	Plug&Play	6.300
		Pix2pix-Zero	14.20
CycleGAN-Turbo	0.080	Cycle-Diffusion	3.500
CycleVAR w/ VAR	0.030	DDIB	3.900
CycleVAR w/ Infinity	0.110	InstructPix2Pix	4.200

We also present a comparative analysis of inference times across various methods, with all evaluations conducted on an Nvidia H100 80G GPU. The results are summarized in Table 6. Traditional GAN-based approaches (blue rows) are characterized by high inference speeds and small model sizes, with approximately 12 Million active parameters. However, this efficiency is typically achieved at the expense of lower generation quality. In contrast, methods based on non-distilled Stable Diffusion (gray rows) exhibit substantially slower performance due to their iterative denoising process. These models are larger; for instance, Stable Diffusion 1.4 and 1.5 are built upon a 1.1 Billion-parameter architecture, while Stable Diffusion 2.1 utilizes a 1.3 Billion-parameter model. CycleGAN-Turbo (yellow row) achieves one-step inference by applying LoRA to a distilled SD-Turbo 2.1 model, utilizing 1.1 Billion relevant parameters. Our proposed method, CycleVAR (green rows), demonstrates a more favorable balance between efficiency and performance. By directly fine-tuning original autoregressive models, CycleVAR achieves rapid inference using the 420 Million- parameter VAR-310M and maintains

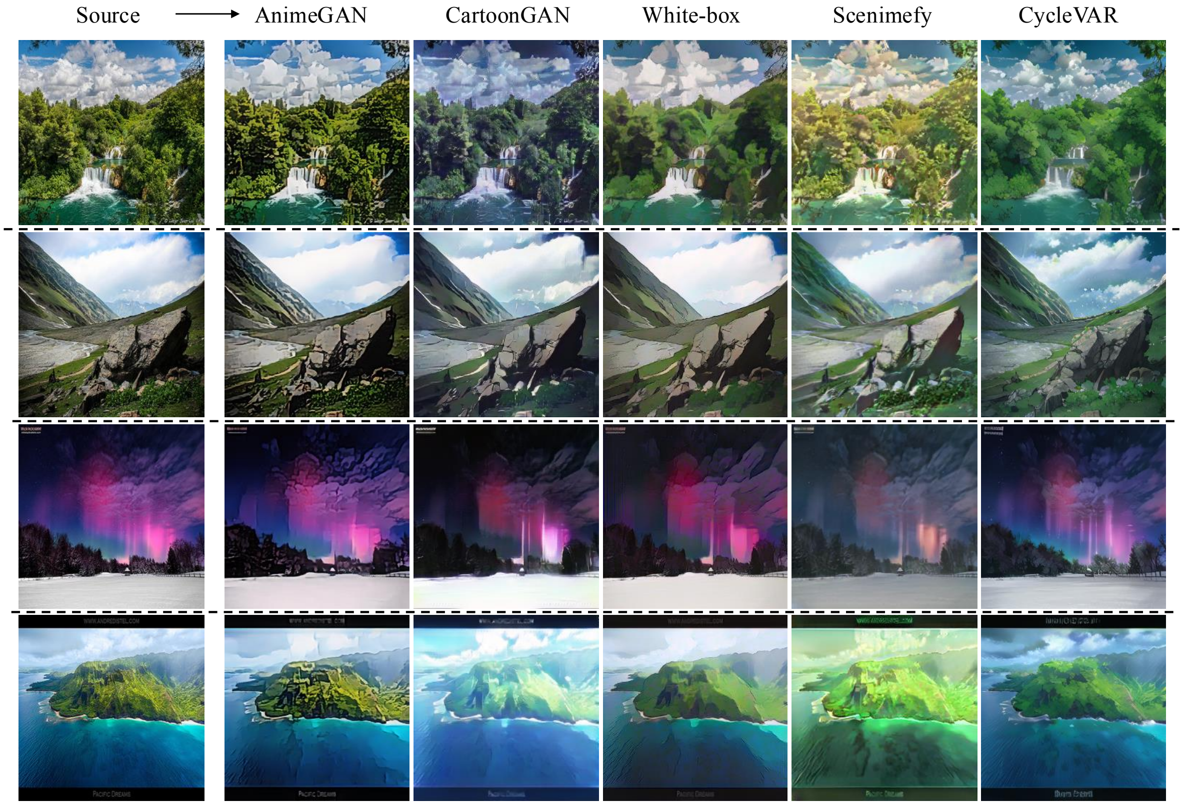

Figure 7. Qualitative comparison on Anime Scene Dataset. We compare parallel one-step CycleVAR w/ Infinity with other state-of-the-art baselines designed for image translation in anime.

comparable speed with the larger Infinity-2B, which has 2.3 Billion parameters for causal VAR Transformer. This highlights CycleVAR's ability to deliver high-quality results while remaining computationally efficient, representing a significant advancement over existing trade-offs between speed and fidelity.

5. Qualitative and Quantitative Results

Our method produces clearer images with richer texture details while capturing the light and dark color characteristics reminiscent of Shinkai's style, as shown in Figure 7. The images generated by AnimeGAN focus too much on the edges, resulting in excessive contrast and a lack of Shinkai's distinctive style. The overall color scheme of CartoonGAN's images tends to be either too washed out or too dark. Conversely, White-box-generated images exhibit basic cartoon color characteristics, but the details are blurred and lack finer granularity. Scenimefy preserves more details than White-box; however, the images still appear blurry and lack the delicate presentation of light spots, failing to evoke a romantic and fantastical atmosphere.

For the user preference study in Table 2, ten sets of images are shuffled and anonymously distributed to participants, who will select the best result from five methods in each set according to different criteria. We summarized the results from 46 participants to calculate the average preference scores.