Mirai: Autoregressive Visual Generation Needs Foresight

Supplementary Material

Table 7. **Alignment coefficient λ selection for Mirai-I**. All models are LlamaGen-B trained for 300 epochs.

Model	Schedule	λ (start $\rightarrow$ end)	FID↓
LlamaGen-B	–	–	5.34
+ Mirai-I	Const	$1 \rightarrow 1$	4.41
		$2 \rightarrow 2$	**4.34**
		$3 \rightarrow 3$	4.54
	Step	$2 \rightarrow 1$	4.59
	Cosine	$2 \rightarrow 0$	4.59

A. More Implementation Details

We adopt the AdamW optimizer [25] with a constant learning rate of 10^{-4}, using a batch size of 256, and enable cosine decay only for LlamaGen-XL experiments. $\beta 1$ = 0.9, $\beta 2$ = 0.95, weight decay = 0.05, gradient clipping of 1.0. The dropout is always 0.1 for the input token embedding, attention module, and FFN module. The class condition embedding dropout for classifier-free guidance is 0.1. All experiments are conducted on 8$\times$ NVIDIA A100 80GB GPUs with bfloat16 precision enabled.

B. More Component-Wise Analysis

Alignment Coefficient for Mirai-I. We investigate the impact of the alignment coefficient λ, which controls the relative strength of the foresight regularization, when using Mirai-I. We compare three scheduling strategies: constant schedule (Const), stepwise schedule (Step), and cosine-annealing schedule (Cosine). As summarized in Tab. 7, the constant schedule, which keeps λ fixed at 2 for the entire training, yields the best FID. This suggests that maintaining a relatively strong level of foresight regularization throughout is most beneficial for Mirai-I, stabilizing global structure through implicit foresight without preventing the AR transformer from learning fine-grained token prediction.

Foresight Encoder for LlamaGen-L with Mirai-I. We further study the choice of foresight encoder in Mirai-I when scaling up the AR backbone from LlamaGen-B to LlamaGen-L. As shown in Tab. 8, DINOv2-L outperforms DINOv2-B when used as the foresight encoder for LlamaGen-L. This indicates a scaling correspondence between the foresight encoder and the AR generator: while DINOv2-B provides the most learnable representation for the smaller size LlamaGen-B, the larger size LlamaGen-L benefits from the richer and more expressive features of DINOv2-L. Therefore, we adopt DINOv2-L as the default foresight encoder in Mirai-I for LlamaGen-L/XL.

Table 8. **The comparison of using different foresight encoders for Mirai-I on the larger size model.** All models are LlamaGen-L trained for 300 epochs.

Model	Target Repr.	Enc. Params.	FID↓
LlamaGen-L	–	–	3.73
+ Mirai-I	DINOv2-B	86M	3.47
	DINOv2-L	300M	**3.07**

Different Implementations of Projection Heads. We further compare two designs for the projection head ρ_k. The first is a lightweight three-layer MLP, which maps the AR hidden state to an intermediate projector space, applies a SiLU nonlinearity, repeats this Linear–SiLU block once more, and finally projects to the foresight dimension C. This design introduces about 7.34M parameters. The second variant replaces the simple MLP projector with a lightweight transformer-style block. It first applies LayerNorm followed by a 4-head self-attention layer, and adds the residual connection to preserve the original token information. A second LayerNorm–MLP block with expansion ratio 4.0 further refines token representations, followed by another residual connection. Finally, a linear projection maps the hidden dimension to C. This design introduces about 7.68M parameters and allows the projector to re-contextualize tokens before alignment.

As summarized in Tab. 9, the transformer projector consistently underperforms the simple MLP projector for both Mirai-I and Mirai-E, despite its higher capacity. We hypothesize that the transformer head tends to solve the foresight alignment objective within the projector itself. Due to self–attention, it can reconstruct foresight by mixing information across tokens even when the AR backbone representations are suboptimal, causing the foresight loss to be absorbed by the head rather than propagated as a strong constraint on the AR states. In contrast, the lightweight MLP performs strictly point–wise mapping, forcing each AR token representation to carry the necessary semantic signal, which leads to more effective regularization of the backbone and better generative performance.

Warm-up in Mirai-E. Mirai-E relies on the model's own EMA as the foresight encoder. However, in the early training stage, the EMA is not yet more stable than the AR model until 15 epochs. To avoid injecting unreliable foresight, we introduce a 15-epoch warm-up stage in which Mirai-E is trained only with the standard AR next token prediction, and explicit foresight is activated afterward. As shown in Tab. 10, adding warm-up significantly improves FID compared with applying foresight from the beginning, confirm-

ing the necessity of delaying explicit foresight alignment until the EMA becomes reliable.

Table 9. **The comparison of using different types of projection heads**. All models are LlamaGen-B trained for 80 epochs.

Model	Type	Head Params.	FID↓
LlamaGen-B	–	–	6.36
+ Mirai-I	MLP	7.34M	**4.77**
	transformer	7.68M	5.65
+ Mirai-E	MLP	7.34M	**5.22**
	transformer	7.68M	6.80

Table 10. **The comparison of whether using warm-up for Mirai-E**. All models are LlamaGen-B trained for 80 epochs.

Model	Warm-up	FID↓
LlamaGen-B	–	6.36
+ Mirai-E	no	8.32
	yes	**5.22**

Table 11. **Mirai at 384×384 Resolution.** The generated images are 384×384 and is resized to 256×256. All models are trained for 80 epochs. ↓ and ↑ indicate whether lower or higher values are better, respectively.

Model	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
LlamaGen-B	7.43	6.60	153.41	0.84	0.40
+ Mirai-I	4.91	6.41	192.28	0.83	0.47
+ Mirai-E	5.72	6.33	182.24	0.80	0.45

C. Mirai at Different Resolutions

To evaluate the scalability of Mirai beyond 256×256 resolution, we further apply foresight alignment to 384×384 image generation on ImageNet. For evaluation, the generated images are downsampled to 256×256. The results are summarized in Tab. 11. Both Mirai-I and Mirai-E consistently improve the baseline across multiple metrics. These results suggest that Mirai remains effective when scaling to higher resolutions.

D. Mirai on Larger Scale Models

We further scale Mirai to LlamaGen-XXL (1.4B). As shown in Tab. 12, foresight continues to yield improvements at the billion-parameter scale.

E. Mirai on Other AR Architectures

To further examine the generality of Mirai, we apply our foresight alignment strategy to a different AR architecture,

Table 12. **Mirai on Larger Scale Models.** All models are LlamaGen-XXL trained for 40 epochs. ↓ and ↑ indicate whether lower or higher values are better, respectively.

Model	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
LlamaGen-XXL	3.49	6.66	272.98	0.86	0.49
+ Mirai-I	3.04	6.73	284.19	0.84	0.53
+ Mirai-E	3.10	6.65	289.10	0.85	0.50

Table 13. **Mirai on another AR architecture.** All models are PAR-B trained for 80 epochs. ↓ and ↑ indicate whether lower or higher values are better, respectively.

Model	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
PAR-B	7.47	7.04	183.60	0.87	0.36
+ Mirai-I	5.59	7.01	201.42	0.84	0.44
+ Mirai-E	6.64	6.96	193.13	0.85	0.40

Table 14. **Mirai in low-resource and data settings.** All models are LlamaGen-S trained for 80 epochs on **1/10** ImageNet. ↓ and ↑ indicate whether lower or higher values are better, respectively.

Model	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
LlamaGen-S	47.81	10.39	24.31	0.43	0.45
+ Mirai-I	35.62	9.99	37.66	0.55	0.45
+ Mirai-E	41.84	9.35	29.60	0.49	0.45

Parallelized Autoregressive Visual Generation (PAR) [44]. Unlike LlamaGen, which follows a strictly sequential next-token decoding process, PAR parallelizes AR inference by predicting multiple tokens in each step while preserving the causal dependency structure. This presents a different modeling bias from sequential AR and therefore serves as a strong testbed for validating the universality of our method. As summarized in Tab. 13, both Mirai-I and Mirai-E consistently improve PAR across most major metrics. These results show that Mirai is not tailored to a specific AR architecture but generalizes to AR models with different paradigms, demonstrating that foresight-based alignment is broadly applicable for enhancing AR visual generation.

F. Mirai in Low-Resource and Limited-Data Settings

To evaluate Mirai in low-resource and Limited-data scenarios, we construct a smaller model based on LlamaGen-B by halving its parameters, which we denote as LlamaGen-S. We train LlamaGen-S with Mirai on a subset of ImageNet containing 100 images per class using a single A100 GPU. Results in Tab. 14 show that our method still provides improvements even in low-resource and Limited-data settings.

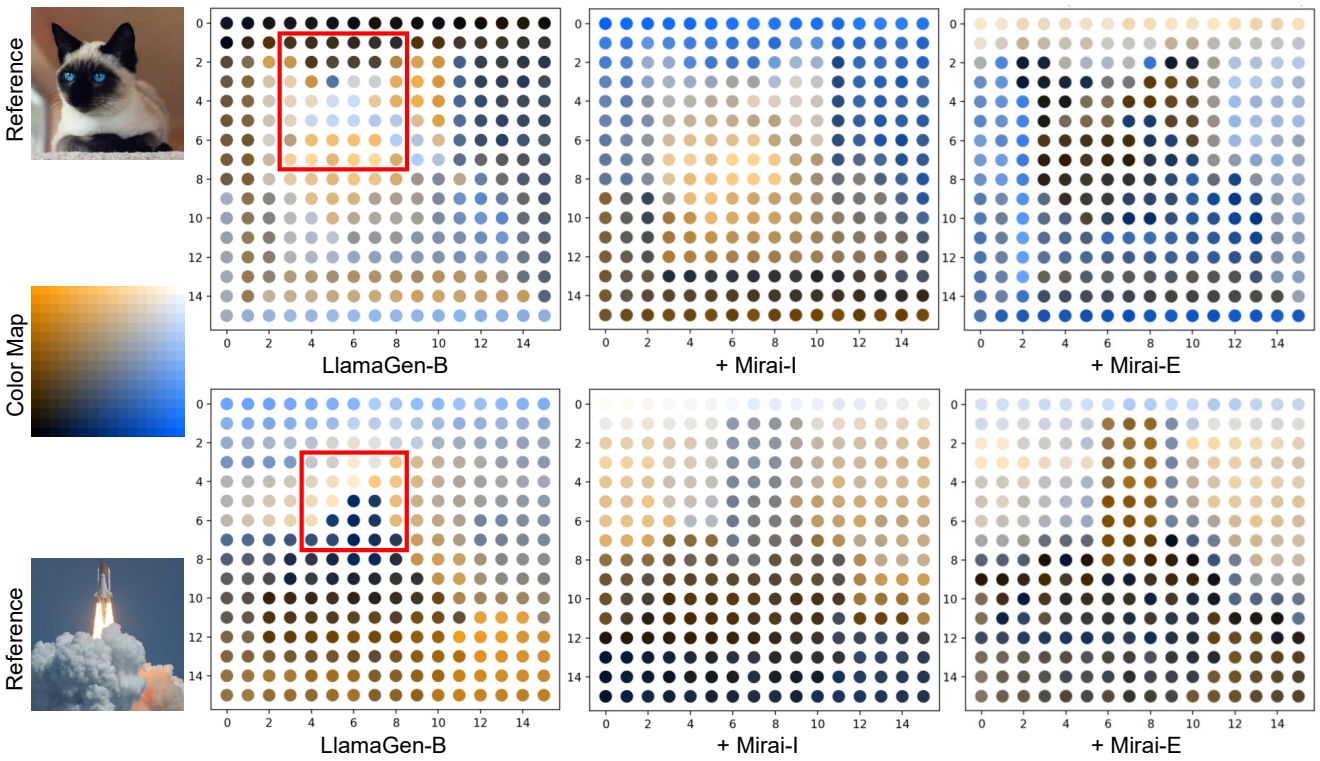

Figure 9. **More results for visualization of layer-8 internal representations on the 2D token grid.** Each token's 2D t-SNE [27] embedding is mapped to a color (with the Color Map at the middle left) and plotted at its original grid location. Smooth color fields indicate 2D-structured representations; the red rectangle in LlamaGen-B highlights abrupt color changes where spatial structure breaks down.

G. More Internal Representation Visualization.

We provide more internal representation visualization results by computing a t-SNE [27] embedding of all internal representation tokens at the 8th layer for one image, then mapping each token's t-SNE coordinate to a color and plotting it back at its original location on the image grid. If nearby tokens in the image share similar features, colors vary smoothly in space; if the representation ignores 2D structure, colors appear scrambled. In Fig. 9, compared to LlamaGen-B, both Mirai-I and Mirai-E produce smoother, more spatially coherent color fields that align with object and background regions, indicating stronger 2D organization of internal representations.

H. Evaluation Metrics

We strictly follow the ADM suite [7] for evaluation and adopt their released reference batches to ensure fair comparison. We use 8× NVIDIA A100 80GB GPUs for evaluation with a batch size of 256 and enable bfloat16 for faster sampling. Below, we briefly summarize the evaluation metrics used in our experiments:

- **FID** [15] quantifies the distributional discrepancy between real and generated samples by comparing Inception-V3 [39] features under the Gaussian assumption.
- **sFID** [28] extends FID by using intermediate spatial features of Inception-V3, making the metric sensitive to spatial structure in generated images.
- **IS** [36] measures both image quality and class diversity by evaluating the KL divergence between the marginal label distribution and the conditional label distribution obtained from the Inception-V3 classifier.
- **Precision & Recall** [20] separately evaluate the realism of generated samples (precision) and the coverage of the real data manifold (recall).

I. FLOPs

We estimate the training compute in floating point operations (FLOPs) for LlamaGen-B and Mirai on a per-image basis, counting only dense matrix multiplications in the transformer and projection heads while ignoring cheaper element-wise operations (e.g., LayerNorm, activations, softmax). The results are summarized in Tab. 15. Relative to the LlamaGen-B baseline, Mirai-I increases the per-image training compute by only 6.6%, while Mirai-E

Table 15. **Per-image training FLOPs and relative compute overhead.** FLOPs measures the per-image training cost; Compute Overhead measures the percentage increase in per-image computational cost introduced by Mirai relative to LlamaGen-B.

Model	FLOPs	Compute Overhead (%)
LlamaGen-B	1.70×10^{11}	–
+ Mirai-I	1.81×10^{11}	6.6%
+ Mirai-E	2.35×10^{11}	38.2%

increases it by 38.2%. To fairly compare training efficiency, we combine the per-image FLOP factors with convergence speed. Empirically, LlamaGen-B requires 400 epochs to reach an FID, while Mirai-I and Mirai-E converge in only 40 and 80 epochs, respectively. On an epoch basis, this corresponds to 10× and 5× faster convergence. After accounting for the FLOPs, Mirai-I achieves a 9.4× reduction in total training compute to reach the same FID, and Mirai-E achieves a 3.6× reduction.

J. Methods for Comparison

We briefly describe the models used in the system-level comparison.

• **BigGAN** [2] A class-conditional large-scale GAN that jointly scales the generator and discriminator in both capacity and resolution. Strong spectral normalization, hinge loss, and architectural improvements, *e.g.*, shared embeddings, enable competitive FID and IS on ImageNet.

• **GigaGAN** [18] A high-capacity adversarial generator trained at hundreds of millions of parameters with multi-scale training and perceptual as well as CLIP-based losses. Extensive augmentation and optimization heuristics further improve GAN fidelity and IS.

• **LDM-4** [35] A latent diffusion model trained and sampled in a VAE latent space with 4× downsampling, reducing computational cost while retaining strong perceptual quality. The model employs classifier-free guidance and decodes latents back to pixel space.

• **DiT-XL** [31] A pure-transformer diffusion backbone that uses AdaLN-Zero conditioning and large-batch training to achieve stable scalability. The architecture demonstrates that ViT-style blocks can replace U-Nets for high-resolution diffusion.

• **SiT-XL** [26] A continuous-time, flow-style reformulation of DiT that simplifies training schedules and objectives. The model achieves higher throughput while preserving or improving image quality relative to discrete-time diffusion transformers.

• **MaskGIT** [3] A non-AR masked-token predictor that performs iterative parallel decoding. A bidirectional transformer fills masked codes in a few refinement steps, offering faster generation than strictly causal AR models.

Figure 10. **Failure cases with 16 foresight tokens.** Objects merge into the background or neighboring ones.

• **RCG (cond.)** [23] Representation-Conditioned Generation that drives an image generator using self-supervised visual features instead of human labels. In our setting, RCG operates on top of a MaskGIT-style parallel decoder with additional class conditioning to refine masked-token synthesis.

• **VAR-d12/d16/d20** [40] Visual Autoregressive modeling that redefines AR learning as coarse-to-fine next-scale prediction: each higher-resolution token map is generated in parallel, conditioned on all previous scales with a block-wise causal mask. Depth tags, d12/d16/d20, denote the number of transformer layers.

• **VQGAN** [11] An AR generator that operates over discrete codes obtained from a VQGAN tokenizer and reconstructs images through its decoder. The final fidelity is constrained by the reconstruction capability of the tokenizer since the AR head only predicts token sequences.

• **ViT-VQGAN** [46] A VQGAN variant that replaces CNN modules in the tokenizer and decoder with ViT-style components, improving reconstruction fidelity and narrowing the performance gap between token reconstruction and AR generation.

• **RQTransformer** [22] An AR model over residual vector-quantized tokens produced by multiple stacked codebooks. Each codebook is predicted sequentially, progressively refining quantization residuals and supporting higher-fidelity generation.

K. Limitation

Injecting excessive foresight into Mirai-E can lead to failure cases, as shown in Fig. 10. In particular, when the number of foresight tokens increases to 16, generated objects may partially blend into the background or merge with nearby objects. This suggests that excessive foresight may over-constrain the representation and compromise locality. Reducing the amount of foresight mitigates this issue. A promising direction for future work is to explore how to better exploit richer foresight signals without sacrificing local fidelity.

L. Ethical Considerations

This work improves the training of AR generation models using foresight. All experiments are conducted on the publicly available ImageNet dataset, and no private or sensitive data are used. Similar to other generative models, our ap-

proach could potentially be misused to synthesize misleading visual content. We encourage responsible use of generative technologies and acknowledge that dataset biases may be inherited from the training data.

M. More Qualitative Results

Below, we show additional uncurated generation results on ImageNet 256×256 from the LlamaGen-XL + Mirai in Fig. 11, Fig. 12, Fig. 13, Fig. 14, Fig. 15 and Fig. 16. We use classifier-free guidance with scale 1.75.

Figure 11. 256×256 LlamaGen-XL + Mirai-I samples. Classifier-free guidance scale = 1.75. Class label = "golden retriever" (207).

Figure 12. 256×256 LlamaGen-XL + Mirai-E samples. Classifier-free guidance scale = 1.75. Class label = "golden retriever" (207).

Figure 13. 256×256 LlamaGen-XL + Mirai-I samples. Classifier-free guidance scale = 1.75. Class label = “sport car” (817).

Figure 14. 256×256 LlamaGen-XL + Mirai-E samples. Classifier-free guidance scale = 1.75. Class label = “sport car” (817).

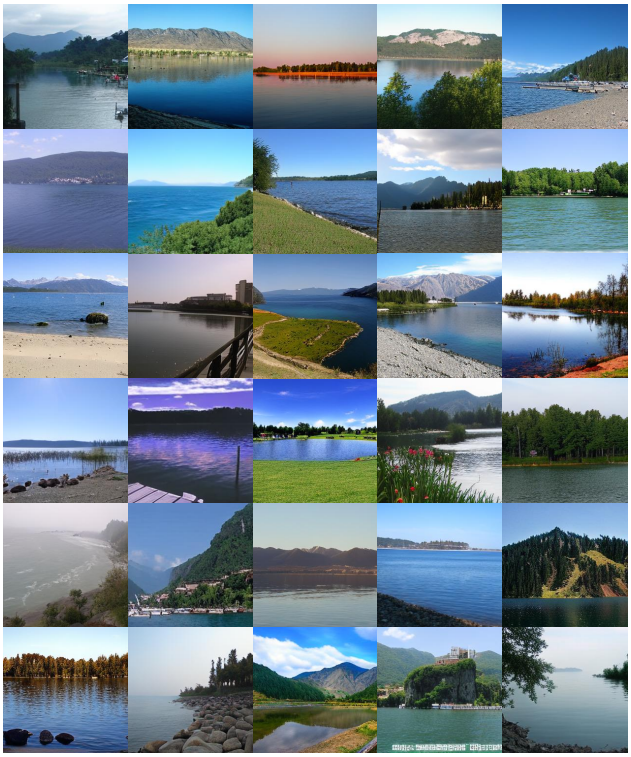

Figure 15. 256×256 LlamaGen-XL + Mirai-I samples. Classifier-free guidance scale = 1.75. Class label = “lake shore” (975).

Figure 16. 256×256 LlamaGen-XL + Mirai-E samples. Classifier-free guidance scale = 1.75. Class label = “lake shore” (975).