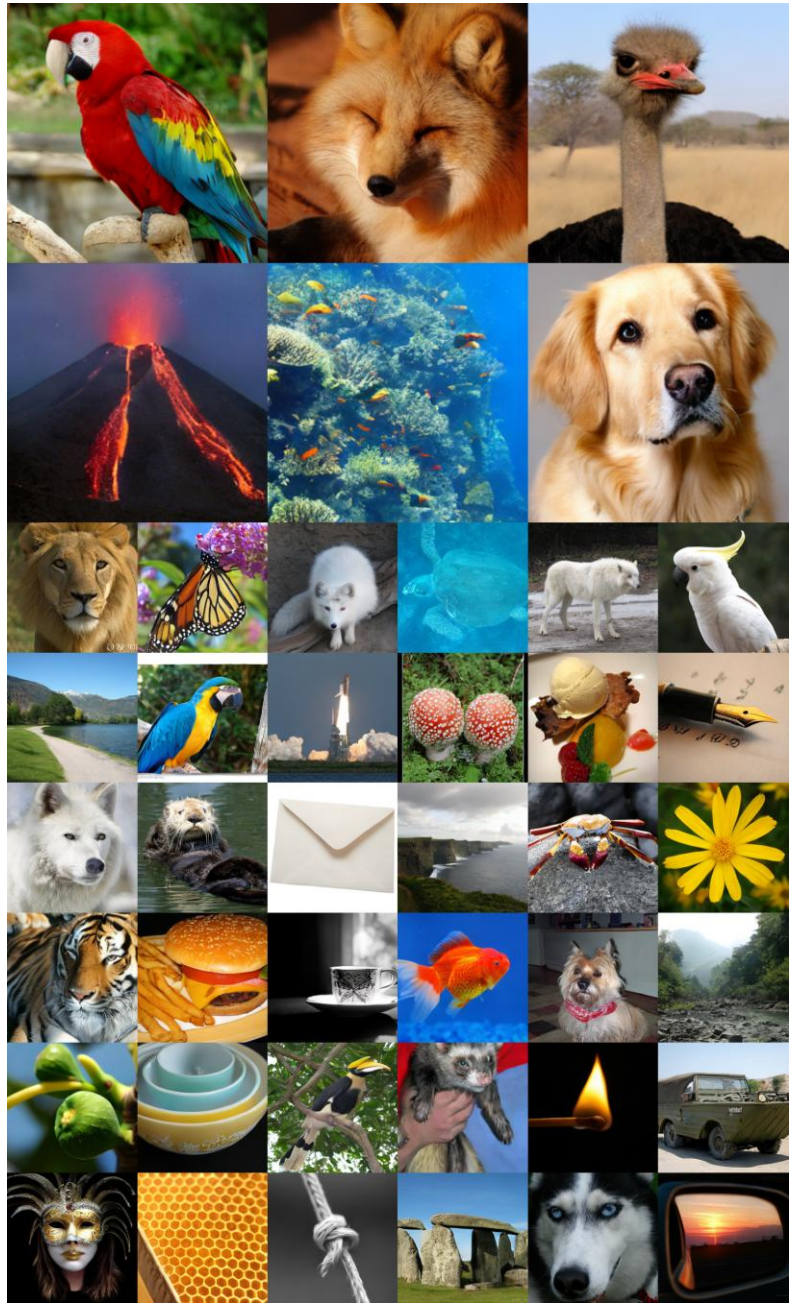
NeurIPS 2024

Autoregressive Image Generation without Vector Quantization

Tianhong Li (MIT), Yonglong Tian (Google DeppMind),
He Li (Tsinghua Univ.), Mingyang Deng, and Kaiming He



Presented by Hyokyung Choi

November 3, 2026

The bottlenecks in image AR

✓ **Can image AR keep continuous features and reduce sequential generation?**

- **VQ bottleneck** : Finite codes lose information. Good VQ tokenizers are difficult to train.
- **Sequential generation** : One token per step requires many Transformer calls.

❖ **Diffusion Loss addresses representation. MAR addresses generation.**

Main contribution

: AR models relationships between tokens.
Diffusion models the possible values of each continuous token.

Approach	Tokens	How token values are modeled
VQ-based AR / MaskGIT(Masked Generative Image Transformer)	Discrete	Code probabilities (cross-entropy)
GIVT(Generative Infinite-Vocabulary Transformers)	Continuous	Gaussian mixture for each token
Latent diffusion (DiT)	Continuous	Diffusion over the full latent image
This work: AR + MAR	**Continuous**	**Small diffusion head for each token, conditioned on Transformer output**

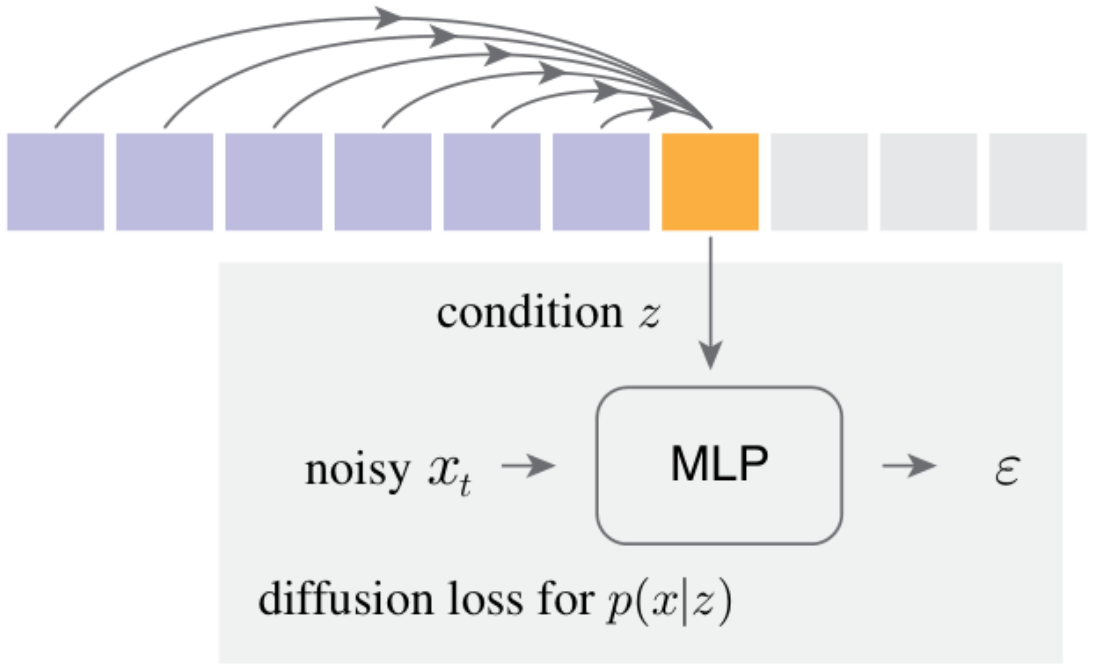
A distribution for each continuous token

✓ **Transformer**

Known tokens provide context z for the next position.

✓ **Small diffusion head**

Samples a continuous token from p(x | z).



❖ **The main model keeps the tokenizer and image decoder, and removes VQ.**

Diffusion Loss in one equation

$$\mathcal{L}(z, x) = \mathbb{E}_{\varepsilon,t}\left[\|\varepsilon - \varepsilon_\theta(x_t|t, z)\|^2\right].$$

Minimize the error between added noise and predicted noise.

1. Add noise to a real token at a random noise level t.
2. Predict the noise using the noisy token and context z.
3. Backpropagate through both the MLP and Transformer.

* Direct clean-token MSE gives a deterministic prediction. The tested baseline has FID > 100.

Sampling has two loops

64

MAR iterations

The Transformer computes context for the next token positions.

New tokens join the known context.

100

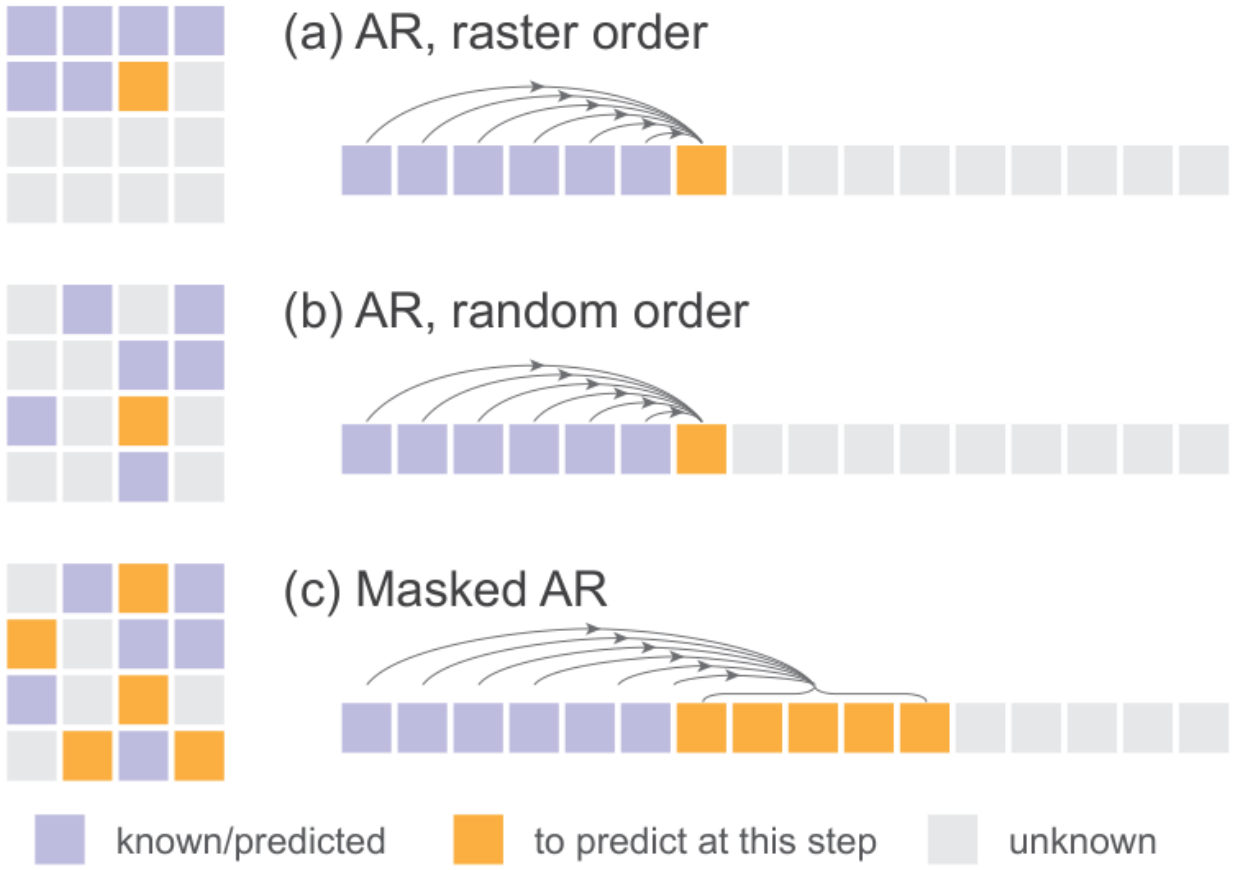
Diffusion steps

The small MLP samples token values while context z stays fixed.

Selected tokens sample in parallel.

* When all tokens are ready, the image decoder turns them into an image.

Masked autoregression predicts token groups



✓ **Random order**

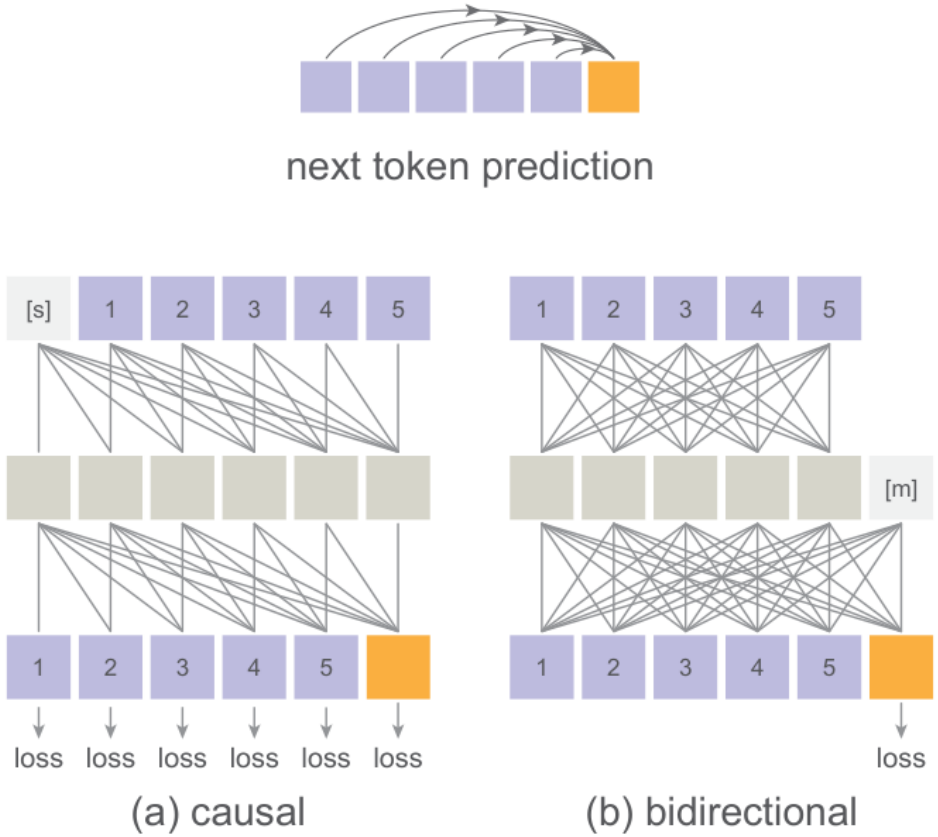
Each image uses a random token order.

✓ **Group prediction**

Several tokens share the known context at each iteration.

* Next-set-of-tokens prediction preserves the autoregressive dependency.

Bidirectional attention keeps targets hidden



✓ **Known tokens can attend to each other.**

The encoder sees only known values.
Training masks 70–100% of tokens.

Mask tokens identify unknown positions.
Loss applies only to the hidden targets.

Training

1) Convert a real Image into continuous tokens
2) Randomly mask some tokens
3) Use the MAR Transformer with bidirectional attention to compute z for masked tokens
4) Add noise to real tokens and predict the noise using the Diffusion MLP and z
5) Update the MLP and MAR Transformer using Diffusion Loss

Inference

1) Start with all image tokens masked
2) Use the trained MAR Transformer with bidirectional attention to compute z for the next tokens
3) Generate continuous tokens using diffusion sampling conditioned on z
4) Repeat 2-3 until all tokens are generated
5) Convert the tokens into an image using the image decoder

Experimental setup

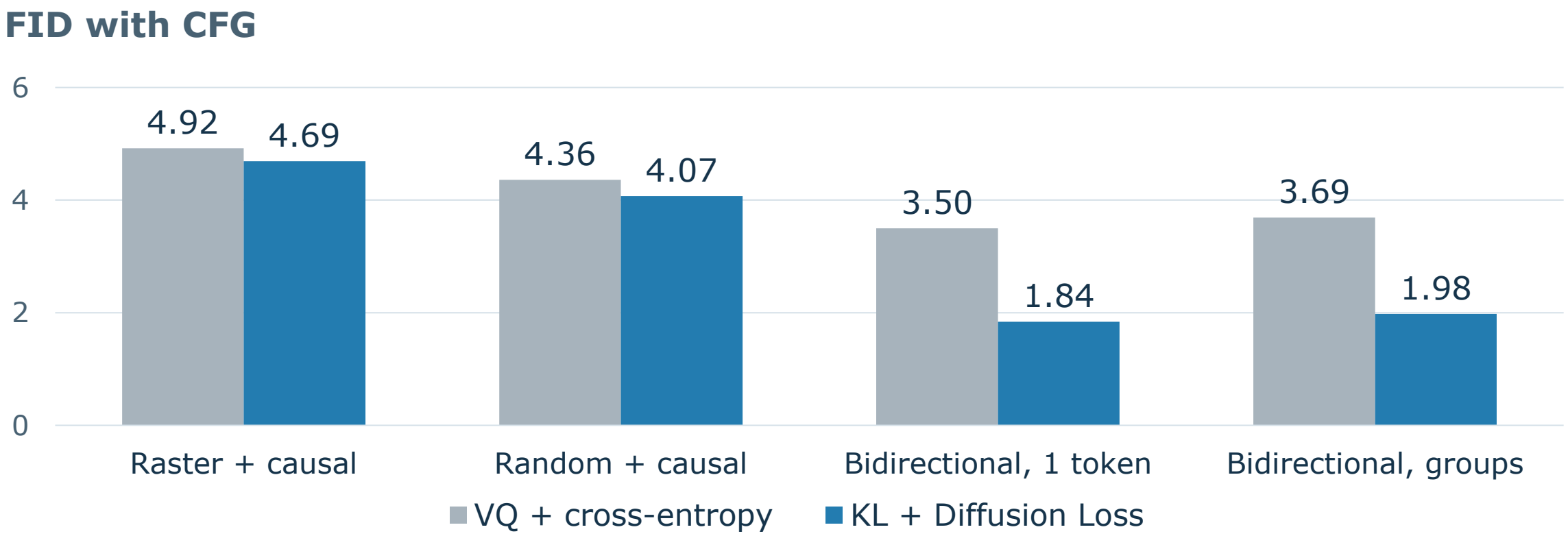
Class-conditional ImageNet, 256 × 256

KL-16 tokenizer: 16 × 16 = 256 continuous tokens

Comparison	Training	Generation
Ablation studies	400 epochs	Usually 64 MAR steps
Final MAR benchmarks	800 epochs	256 AR steps

*** Evaluation Metric : FID**
*** CFG(Classifier-Free Guidance) strengthens the class condition**

Diffusion Loss helps most with MAR



This comparison changes both the tokenizer and loss.

* Tokenizer reconstruction FID: VQ-16 5.87 KL-16 1.43

Random order and full attention drive gains

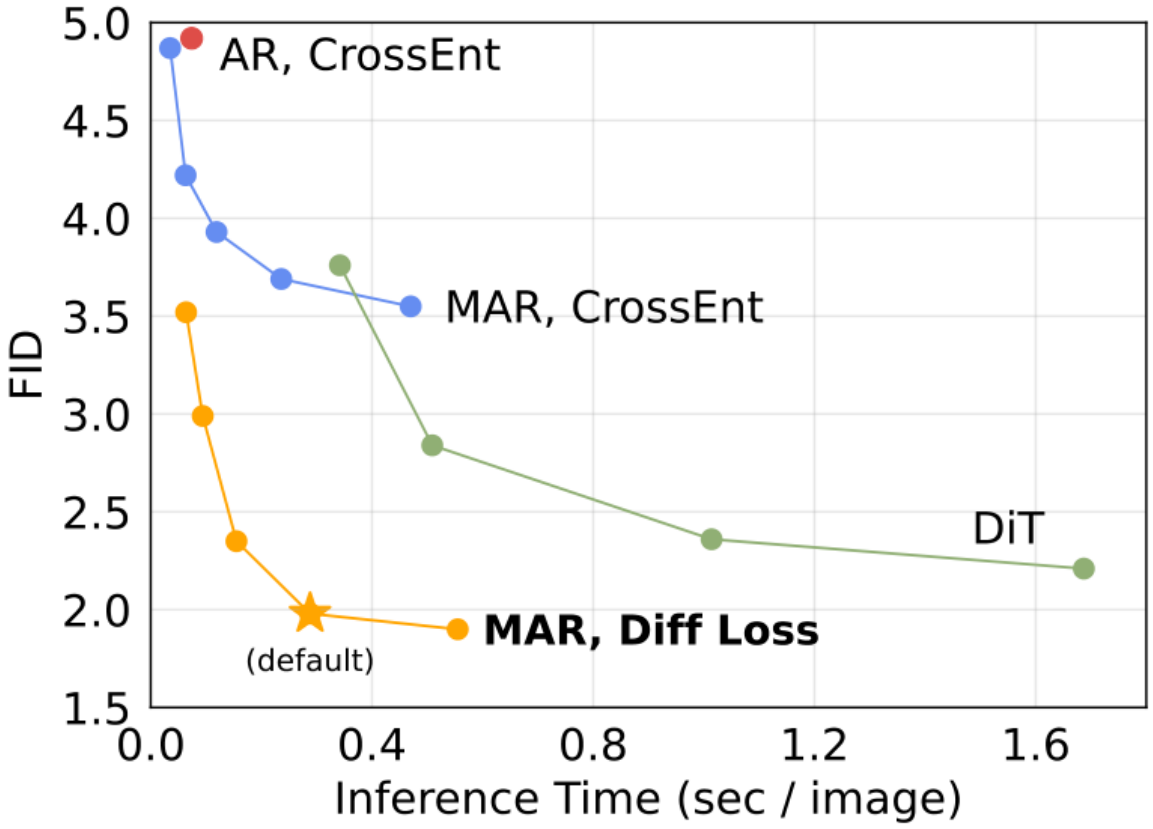
Same KL-16 tokenizer and Diffusion Loss, without CFG

Token order	Attention	Tokens per step	FID
Raster	Causal	1	19.23
Random	Causal	1	13.07
Random	Bidirectional	1	3.43
Random	Bidirectional	Multiple	3.50

Full attention gives the largest gain in this ablation.

Group prediction gives fewer iterations with a small quality cost.

A favorable speed–quality trade-off



FID 1.97

0.288 seconds / image

: Default MAR-L setting
with classifier-free guidance

(using Small diffusion sampler
about 10% of total runtime)

One A100 GPU, batch size 256. Batched time per image, not single-image latency.

Strong benchmark results at larger scale

//Selected ImageNet 256 × 256 results with CFG

Model	Parameters	FID ↓	IS ↑
DiT-XL/2 (Diffusion Transformer)	675M	2.27	278.2
MAGVIT-v2 (Masked Generative Video Transformer)	307M	1.78	319.4
MDTv2-XL/2 (Masked Diffusion Transformer)	676M	1.58	314.7
MAR-B	208M	2.31	281.7
MAR-L	479M	1.78	296.0
MAR-H	943M	1.55	303.7

*** MAR: 800 epochs and 256 AR steps, with larger denoising heads**

Strengths and Limitations

✓ Strengths

No vector quantization
Efficient generation
Flexible design

✓ Limitations

Tokenizer dependence
Visible artifacts
Limited evaluation



Takeaways and Questions for Discussion

✓ Takeaways

1. Autoregressive image generation can work well without vector quantization.

2. MAR combines prediction of token groups with a small diffusion model for each token.

Takeaways and Questions for Discussion

✓ Questions

1. Why is such a small MLP enough? Does this mean the per-token distribution given z is close to unimodal?
2. If every token is now generated by diffusion, is this really an autoregressive model or a diffusion model with a transformer deciding the order, and does that distinction matter?
3. Why does generating multiple tokens at once slightly reduce image quality compared to generating one token at a time?
4. How does diffusion loss perform in other continuous domains? Why would it perform well there compared to quantization in those domains?
5. Within a MAR step, does sampling tokens independently cause incoherence? How does it scale with the number of tokens predicted per step?

Thank you!