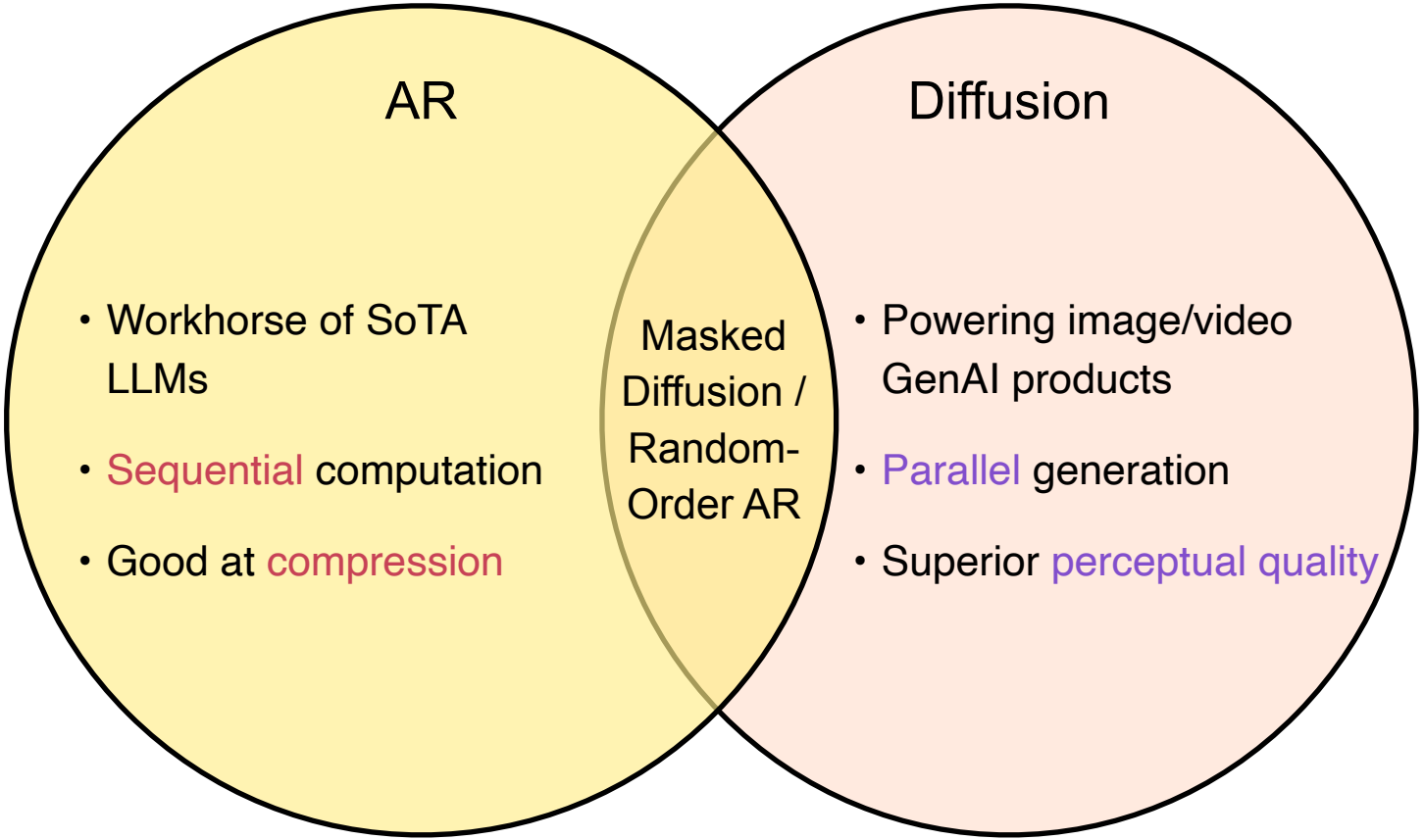
What Really Separates Autoregression & Diffusion?

A Synthesis and Path Beyond

Jiaxin Shi
Meta Superintelligence Labs
2026/8/6 @Simons Institute

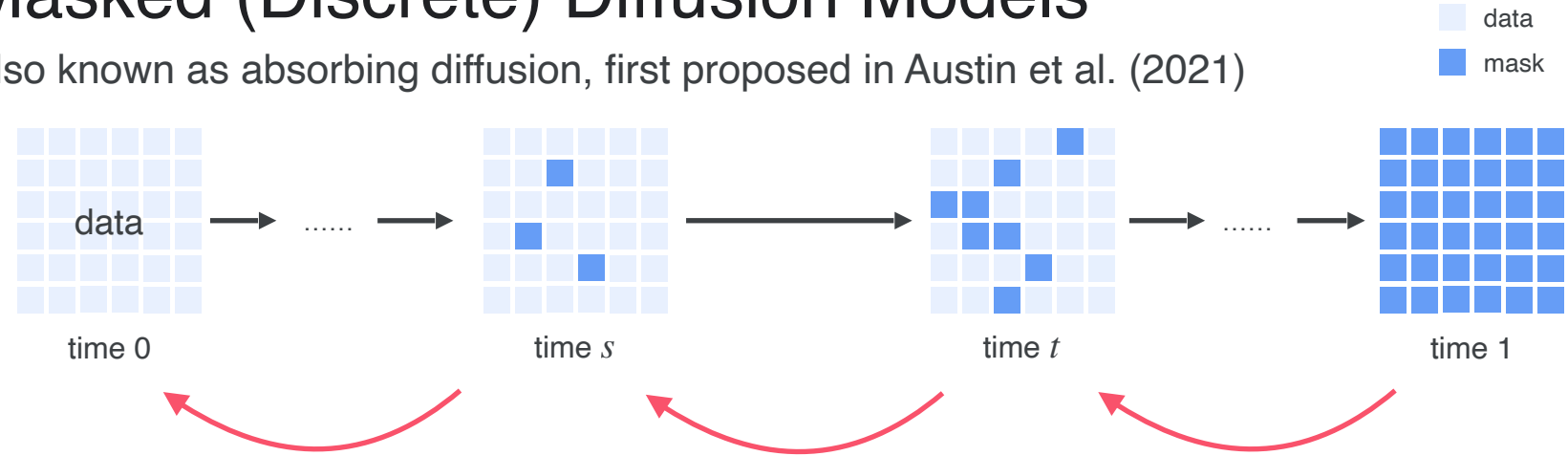
jiaxins.io

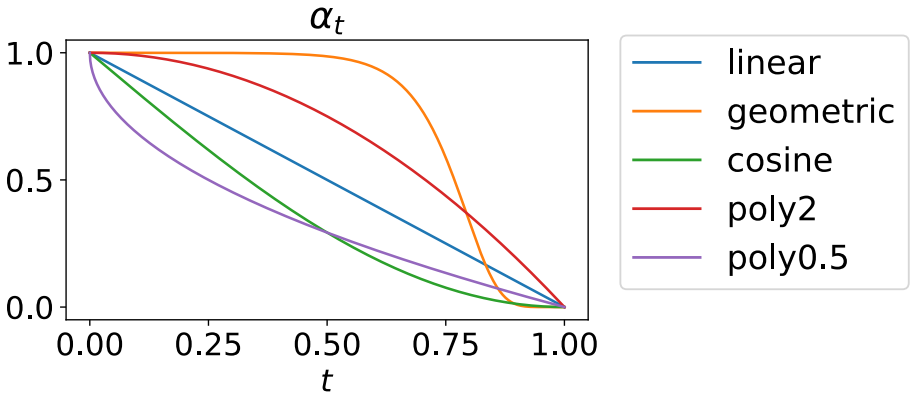
Autoregression vs. Diffusion



Masked (Discrete) Diffusion Models

Also known as absorbing diffusion, first proposed in Austin et al. (2021)

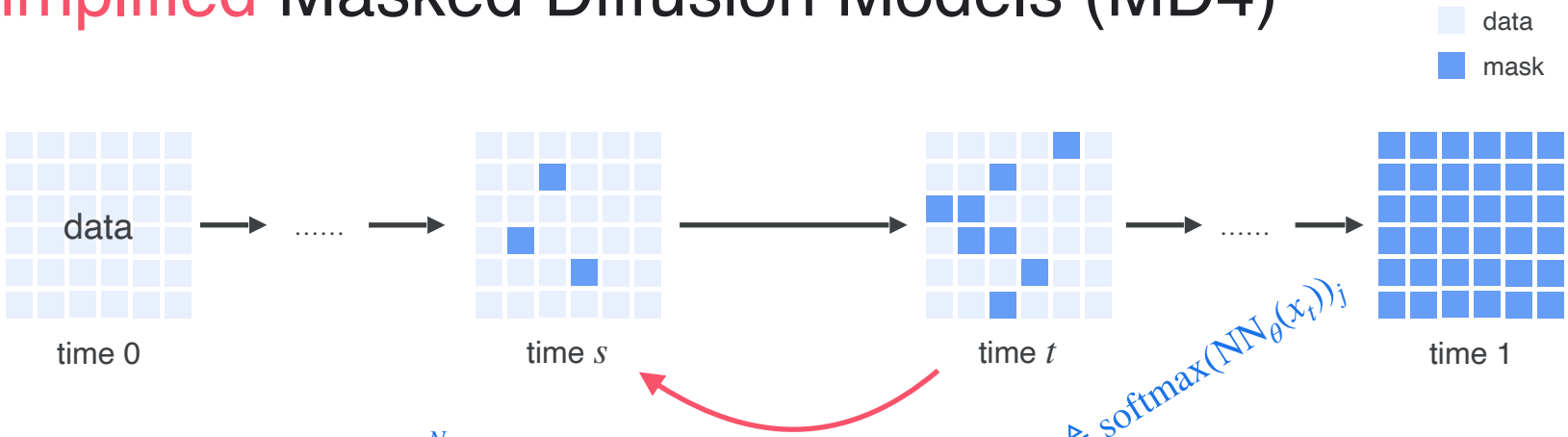




Masking schedule α_t: The expected proportion of unmasked elements at t

Austin et al. (2021). Structured denoising diffusion models in discrete state-spaces.

Simplified Masked Diffusion Models (MD4)



Reverse process $q(x_s \mid x_t) \approx \prod_{n=1}^{N} q(x_s^{(n)} \mid x_t)$ **as** $s \to t$

$$\begin{cases} \text{w/ prob. } \dfrac{\alpha_s - \alpha_t}{1 - \alpha_t} \mathbb{E}[x_0^{(n)} = j \mid x_t], \text{ unmask to state } j \\ \text{w/ prob. } \dfrac{1 - \alpha_s}{1 - \alpha_t}, \text{ remain masked} \end{cases}$$

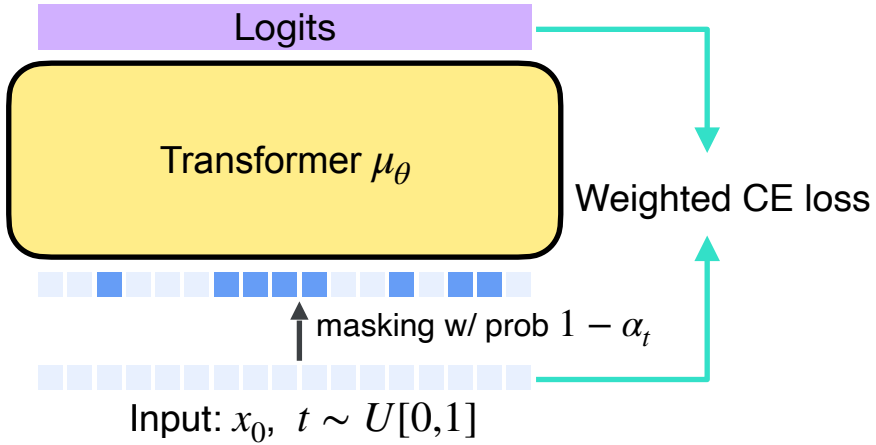
Shi et al. (2024). Simplified and Generalized Masked Diffusion for Discrete Data.
Concurrent work MDLM (Sahoo et al., 2024) and RADD (Ou et al., 2024) studied similar models

Simplified Masked Diffusion Models (MD4)

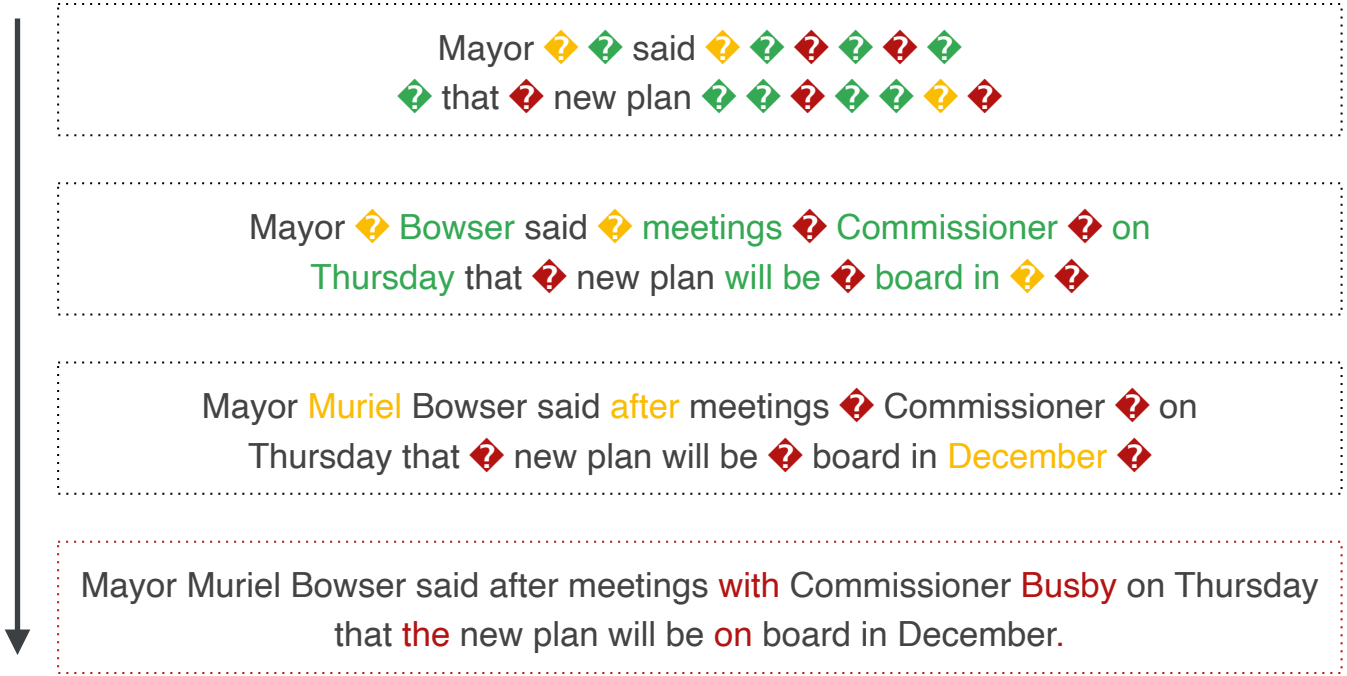
Continuous-time ELBO

$$\log p_\theta(x_0) \geq -\int_0^1 \frac{\alpha_t'}{1-\alpha_t} \mathbb{E}_{q(x_t|x_0)} \left[\sum_{n:x_t^{(n)}=m} (x_0^{(n)})^\top \log \mu_\theta^{(n)}(x_t, t) \right] \mathrm{d}t.$$

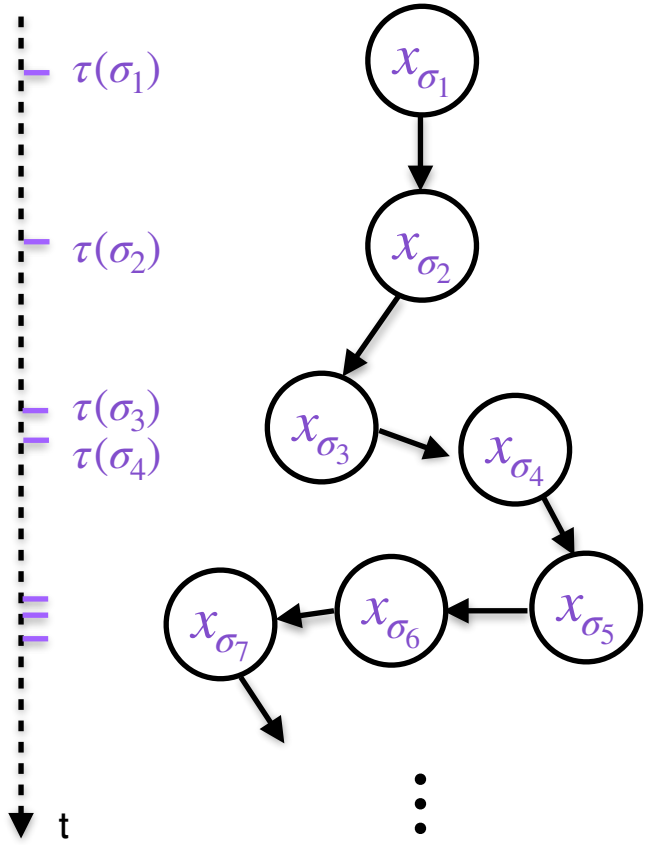
- Maximum likelihood is as simple as training an ensemble of BERTs
- Many popular diffusion LLMs are now based on masked diffusion and this objective (e.g., LLaDA, Dream)



Faster Generation via Parallel Unmasking



Masked Diffusion as Random-Order AR

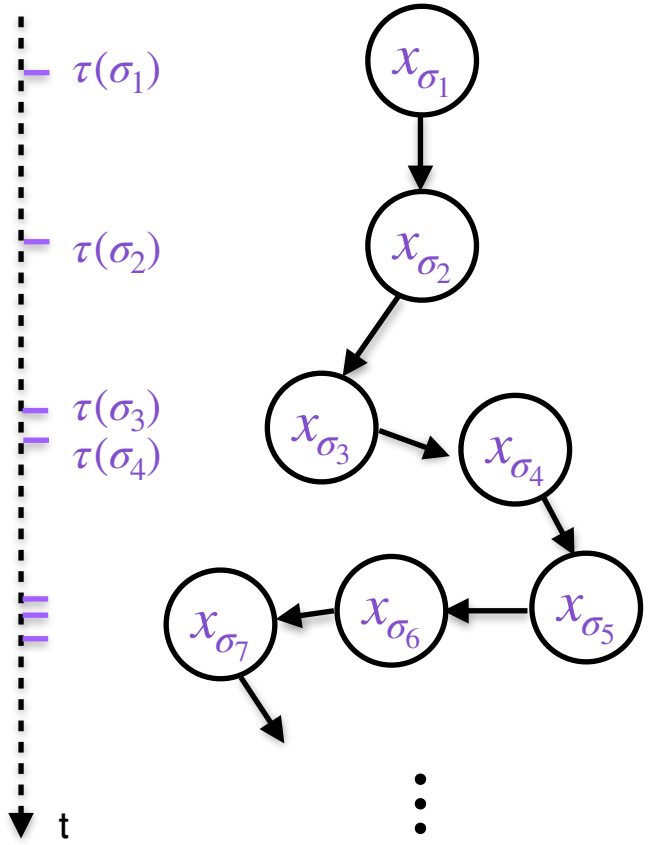


Consider the continuous-time reverse process

- $\tau(n)$: unmasking time of the n-th element
- there exists an ordering of the unmasking times

Uria, B. et al. (2014). A deep and tractable density estimator.
Hoogeboom et al. (2021). Autoregressive diffusion models.

Masked Diffusion as Random-Order AR



Consider the continuous-time reverse process

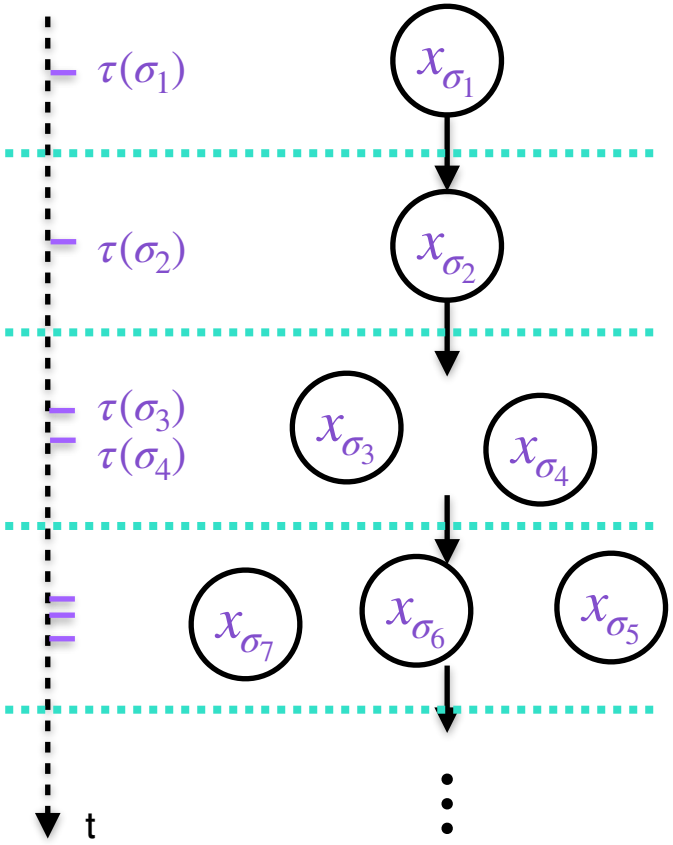
- $\tau(n)$: unmasking time of the n-th element
- there exists an ordering of the unmasking times

CDF of unmasking times:

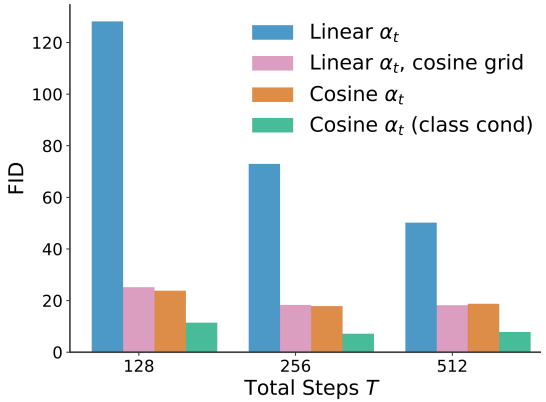
$$P(\tau(n) \leq t) = P(x_t^{(n)} = m) = 1 - \alpha_t$$

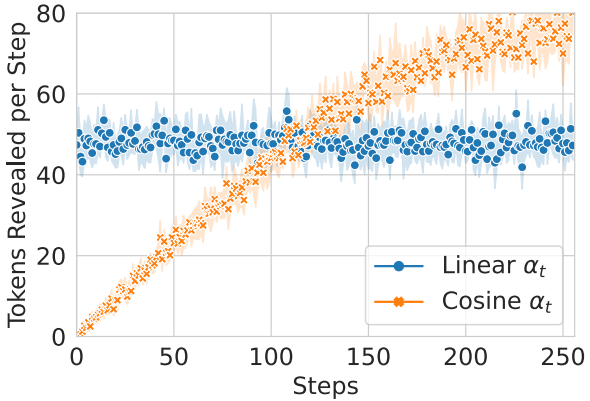
- When α_t is global, all $\tau(n)$s share the same distribution, the unmasking order is uniform across all permutations
- The equivalence is no longer true for generalized masked diffusion (Shi et al. 2024)

Masked Diffusion as Random-Order AR



- In continuous time (training), all choices of α_t lead to a random-order AR model
- In discrete time (sampling), shape of α_t determines the parallel sampling bandwidth





Information Profile

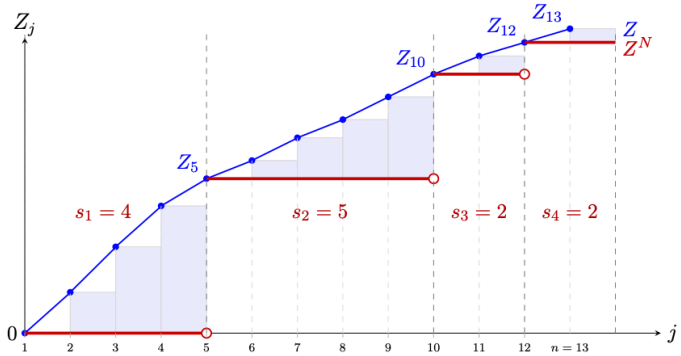
- Sampling error: $\mathrm{KL}(p_{\text{data}}(x) \| p_{\text{sampl}}(x)) \leq \epsilon_{\text{learn}} + \epsilon_{\text{disc}}$
- The discretization error ϵ_{disc} depends on the data distribution's "information profile":

$$f(i) = \mathbb{E}_{p_{\text{data}}(x), \sigma \sim \text{Unif}}[\log p_{\text{data}}(x_{\sigma_{i+1}} \mid x_{\sigma_{\leq i}})], \quad i = 0, \ldots, N-1$$

- Interpretation: how avg. predictability of remaining dimensions grows as we unmask more

$$\epsilon_{\text{disc}} = \mathbb{E}_{\nu(a)} \left[\sum_{0=1}^{N-1} f(i) - \sum_{k=0}^{K-1} (a_{k+1} - a_k) f(a_k) \right]$$

- The result shows we need a slow-to-fast schedule (Shi et al., 2024) for a concave information profile.



Lavenant & Zanella (2025). Error bounds and optimal schedules for masked diffusions with factorized approximations.
Chen, S., Cong, K., & Li, J. (2025). Optimal inference schedules for masked diffusion models.

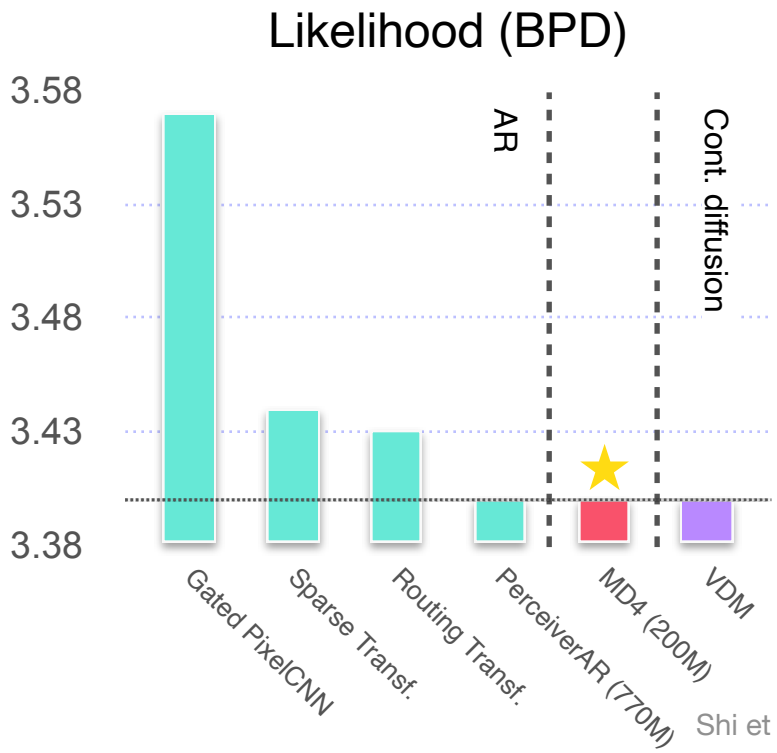
Our understanding so far..

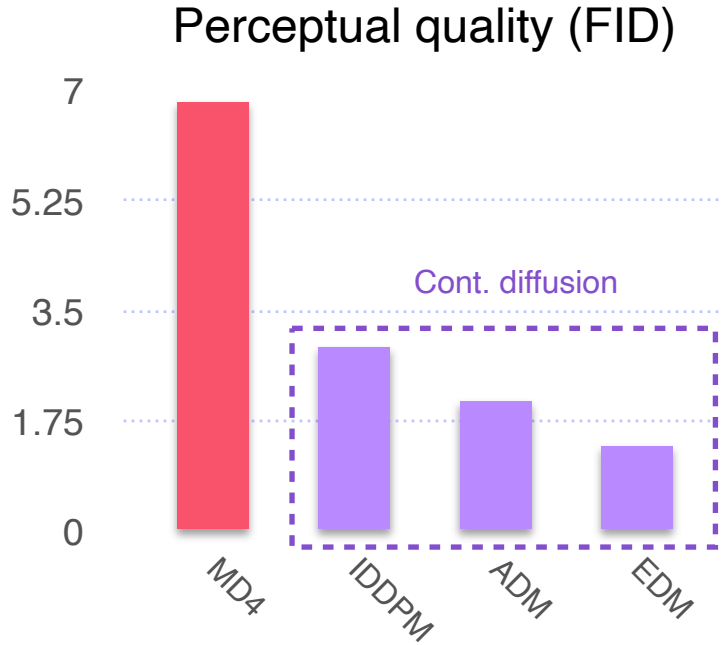
- Parallelism in masked diffusion models (MDMs): an artifact of discretization
- Parallelism in AR: came from increasing predictability using observed information

- **If parallelism is not unique to diffusion, what about perceptual quality?**
- If this advantage is fundamental, MDM/random-order AR should NOT be able to match continuous diffusion in sample quality.

Can MDM/random-order AR match continuous diffusion in perceptual quality?

- **Seems no:** MD4 reports likelihood comparable to strong continuous diffusion, but perceptual quality (as measured by FID) is still behind





Shi et al. (2024). Simplified and Generalized Masked Diffusion for Discrete Data.

Demystifying Diffusion Objectives: Reweighted Losses are Better Variational Bounds

Jiaxin Shi[1] and Michalis K. Titsias[1]

[1]Google DeepMind
{jiaxins,mtitsias}@google.com

arxiv.org/abs/2511.19664

Abstract

We derive a new theoretical interpretation of the reweighted losses that are widely used for training diffusion models. Our method is based on constructing a cascade of time-dependent variational lower bounds on the data log-likelihood, that provably improves upon the standard evidence lower bound and results in reduced data-model KL-divergences. Combining such bounds gives rise to reweighted objectives that can be applied to any generative diffusion model including both continuous Gaussian diffusion and masked (discrete) diffusion models. Then, we showcase this framework in masked diffusion and report significant improvements over previous training losses in pixel-space image modeling, approaching sample quality comparable to continuous diffusion models. Our results also provide a theoretical justification for the simple weighting scheme widely used in masked image models.

Continuous Diffusion Loss

Negative ELBO (cont-time limit)

Derived from maximizing the log-likelihood of data.

$$\mathscr{L}_\infty(\mathbf{x}) = \frac{1}{2}\mathbb{E}_{t\sim U[0,1],\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left[\lambda'(t)\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t)\|^2\right]$$

Reweighted Loss

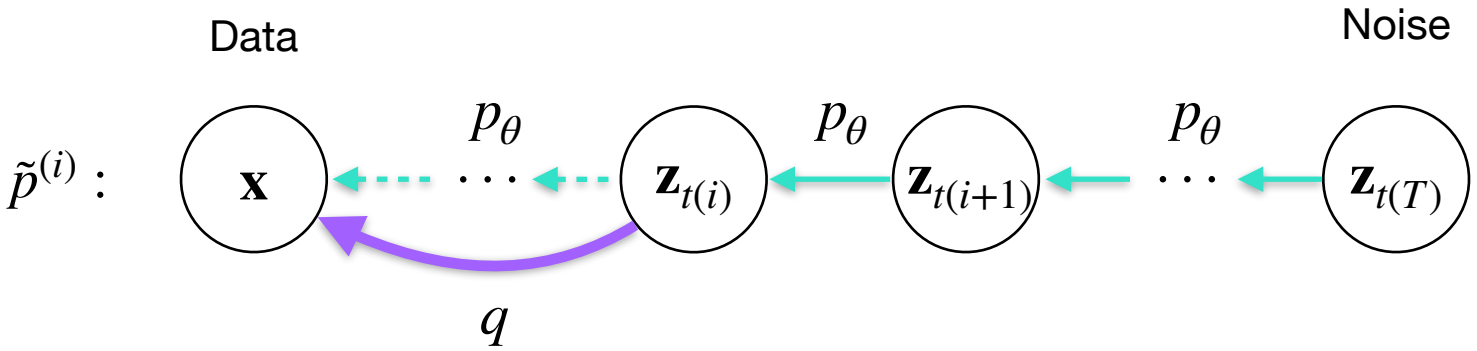
Practically a "reweighted" version is used

$$\mathscr{L}_\infty(\mathbf{x}) = \frac{1}{2}\mathbb{E}_{t\sim U[0,1],\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left[\tilde{w}(t)\lambda'(t)\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t)\|^2\right]$$

Name	Parameterization	$\lambda(t)$	$\hat{w}(\lambda)$	$\tilde{w}(t)$
EDM	mean prediction	$F^{-1}_{\mathcal{N}(2.4,2.4^2)}(1-t)$	$p_{\mathcal{N}(2.4,2.4^2)}(\lambda)\frac{e^{-\lambda}+0.5^2}{0.5^2}$	$w(\lambda(t))$
IDDPM	ϵ prediction	$-2\log\tan(\frac{\pi}{2}t)$	$\operatorname{sech}(\frac{\lambda}{2})$	$2\sin(\frac{\pi}{2}t)\cos(\frac{\pi}{2}t)$
Sigmoid	ϵ prediction	$-2\log\tan(\frac{\pi}{2}t)$	$\operatorname{sigmoid}(-\lambda+k)$	$\frac{1}{1+e^{-k}\tan(\frac{\pi}{2}t)^{-2}}$
FM	velocity prediction	$2\log\frac{1-t}{t}$	$e^{-\frac{\lambda}{2}}$	$\frac{t}{1-t}$

Kingma, D., & Gao, R. (2023). Understanding diffusion objectives as the ELBO with simple data augmentation.

Diffusion Models with “Optimal Decoders”



- Consider a modified diffusion model that replaces the learned denoiser with the ground-truth reverse transition $q(\mathbf{x}\,|\,\mathbf{z}_{t(i)}) \propto q(\mathbf{z}_{t(i)}\,|\,\mathbf{x})q(\mathbf{x})$ for the final i reverse steps.
- Define a sequence of such models $\tilde{p}^{(i)}$ indexed by $i = 1, \cdots, T$

Claim 1: More “optimal” reverse steps lead to better ELBOs

Let $\mathcal{L}^{(i)}(\mathbf{x})$ be the ELBO of the model of $\tilde{p}^{(i)}$

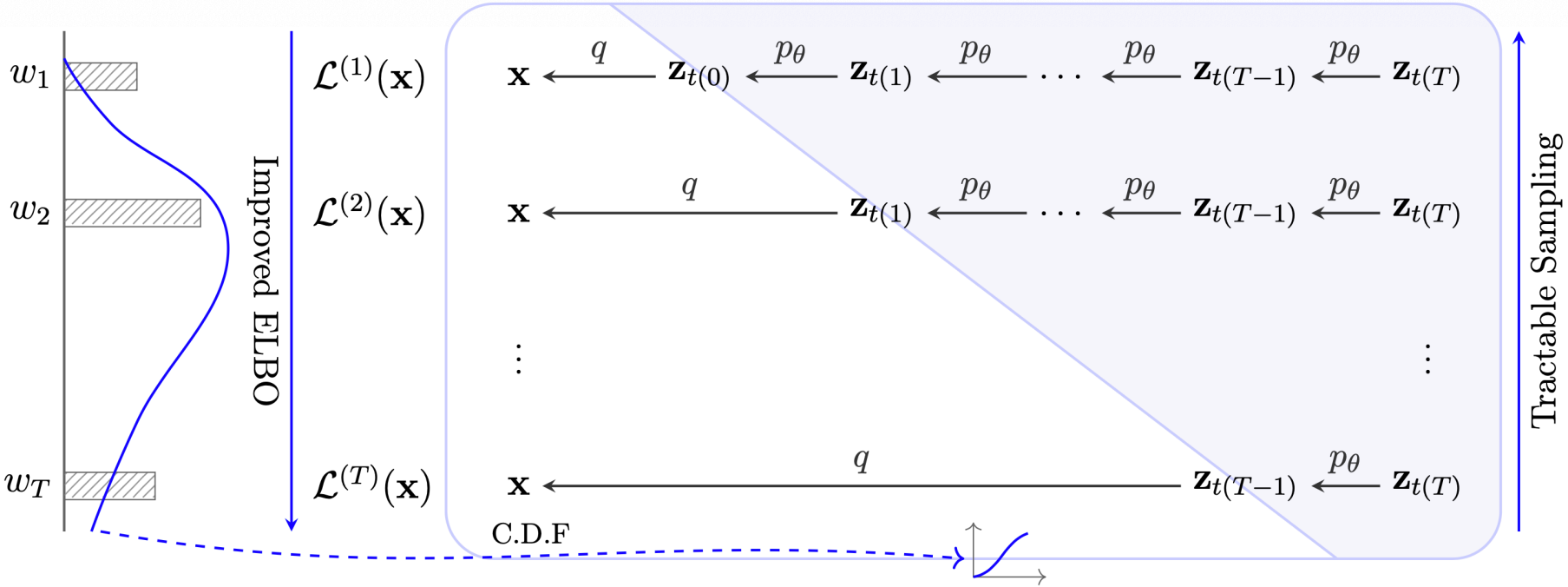
$$\mathcal{L}^{(i)}(\mathbf{x}) \triangleq \mathbb{E}_{q(\mathbf{z}_{s(i)}|\mathbf{x})}[\log q(\mathbf{x}|\mathbf{z}_{s(i)})] - \mathrm{KL}(q(\mathbf{z}_{t(T)}|\mathbf{x})\|p(\mathbf{z}_{t(T)})) - \sum_{j=i}^{T} \mathbb{E}_{q(\mathbf{z}_{t(i)}|\mathbf{x})}\left[\mathrm{KL}(q(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}, \mathbf{x})\|p(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}))\right]$$

As we increase i, the ELBO improves

$$\mathbb{E}_{q(\mathbf{x})}[\mathcal{L}^{(i+1)}(\mathbf{x})] \geq \mathbb{E}_{q(\mathbf{x})}[\mathcal{L}^{(i)}(\mathbf{x})]$$

Since $\mathrm{KL}(q(\mathbf{x})\|\tilde{p}^{(i)}(\mathbf{x})) = -\mathbb{E}_{q(\mathbf{x})}[\log \tilde{p}^{(i)}(\mathbf{x})] + C \leq -\mathbb{E}_{q(\mathbf{x})}[\mathcal{L}^{(i)}(\mathbf{x})] + C$, this implies that incorporating an additional optimal reverse step results in a smaller upper bound on data-model KL divergence

Claim 2: Common diffusion objectives are a weighted sum of the improved ELBOs



Diffusion objectives: $\mathcal{L}^{\tilde{w}}(\mathbf{x}) = \lim_{T\to\infty} \sum_{i=1}^{T} w_i \mathcal{L}^{(i)}(\mathbf{x}) = \int_0^1 \tilde{w}(t) \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x})} \left[L_{\text{denoise}}(\mathbf{z}_t, \mathbf{x}, t) \right] dt + \text{C}$

Reweighted Loss for Masked Diffusion

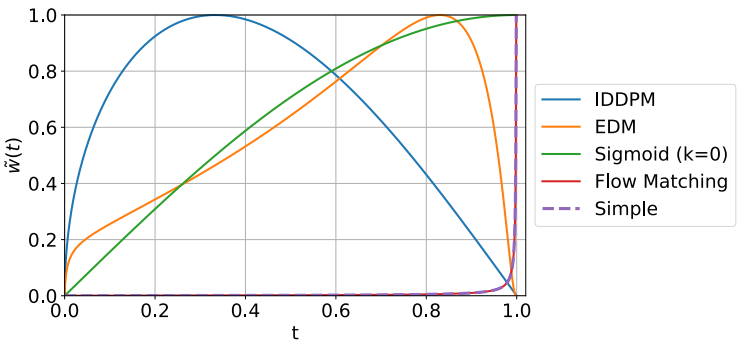
- The framework is general and can be applied to any diffusion processes
- Applying it to masked diffusion yields:

$$\mathscr{L}^{\tilde{w}}(\mathbf{x}) = -\int_0^1 \tilde{w}(t)\frac{\alpha_t'}{1-\alpha_t}\mathbb{E}_{q(\mathbf{z}_t|\mathbf{x})}\left[\delta_{\mathbf{z}_t,m}\mathbf{x}^\top \log \mu_\theta(\mathbf{z}_t)\right] dt$$

How to choose $\tilde{w}(t)$?

- Map common continuous diffusion weightings to masked diffusion
- Match in log-SNR (λ) space instead of t space to maintain schedule invariance

Name	$\lambda(t)$	$\hat{w}(\lambda)$	$\tilde{w}(t)$
EDM	$\log \frac{\alpha_t}{1-\alpha_t}$	$p_{\mathcal{N}(2.4,2.4^2)}(\lambda)\frac{e^{-\lambda}+0.5^2}{0.5^2}$	$w(\lambda(t))$
IDDPM		$\operatorname{sech}(\frac{\lambda}{2})$	$2\sqrt{\alpha_t(1-\alpha_t)}$
Sigmoid		$\operatorname{sigmoid}(-\lambda+k)$	$\frac{1-\alpha_t}{1-(1-e^{-k})\alpha_t}$
FM		$e^{-\frac{\lambda}{2}}$	$\sqrt{\frac{1-\alpha_t}{\alpha_t}}$
Simple		-	$-\frac{1-\alpha_t}{\alpha_t'}$

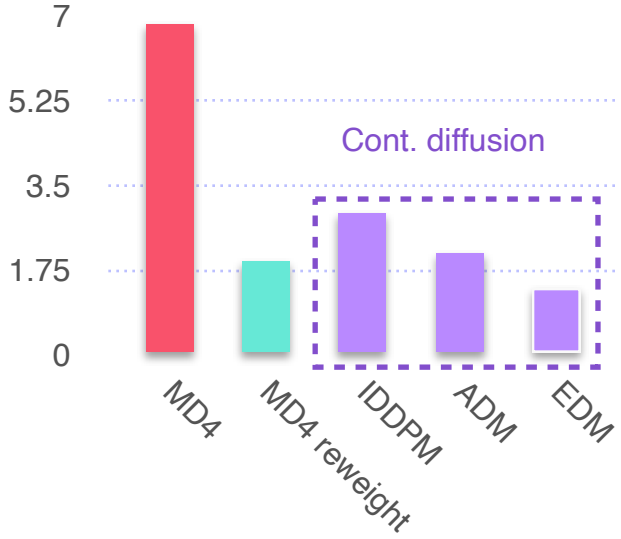


Impact of Weighting Functions

Pixel-space generation, ImageNet 64x64

Sample (324M Params, FID 1.92)

Method	#Params	FID (↓)	IS (↑)
Gaussian Diffusion			
IDDPM (Nichol and Dhariwal, 2021)		2.92	
ADM (Dhariwal and Nichol, 2021)	296M	2.07	
EDM (Karras et al., 2022)	296M	**1.36**	
VDM++ (Kingma and Gao, 2023)	296M	1.43	63.7
Masked Image Models			
MAR (Li et al., 2025)	479M	2.93	
FractalMAR (Li et al., 2025)		2.72	
Masked Diffusion			
MD4 (ELBO)	204M	6.84	30.3
Weighting:			
- IDDPM (non-monotonic)	204M	11.14	22.9
- EDM (nearly-monotonic)	204M	4.42	37.3
- Sigmoid ($k = 0$)	204M	3.91	40.1
- FM	204M	3.43	43.3
- Simple	204M	**2.96**	**46.7**
- Simple	324M	**1.92**	**57.9**



Beyond (fixed-order) autoregression and (random-order) diffusion

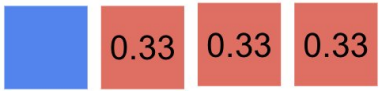
Learning-Order Autoregressive Models

Distribution of the next position to generate:

Fixed-Order

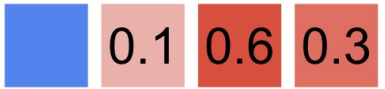
e.g., left-to-right in AR LLMs

Any-Order (Masked Diffusion)



Randomly select next position

Learning-Order



Learn a policy to select next position given partial generation

Order policy

$$p_\theta(y) = \sum_\sigma p_\theta(y, \sigma) = \sum_\sigma p_\theta(\sigma_i \mid \sigma_{<i}, y_{\sigma_{<i}}) p_\theta(y_{\sigma_i} \mid y_{\sigma_{<i}})$$

σ_i: the index of the i-th element to generate in sequence y

Wang, Shi, Heess, Gretton, & Titsias (2025). Learning-order autoregressive models with application to molecular graph generation.

The Importance of Insertion

- When the generation order is non-monotonic, model has to guess both the value and the absolute position. The latter is very difficult.

```
if b == '<':
    depth += 1
elif [M] [M] [M] [M]
    depth -= 1
```

If the model choose to generate `depth -= 1` because it's easier, it has to know how many masks to keep before it

Wu et al. (2026). Dreamon: Diffusion language models for code infilling beyond fixed-size canvas.

The Insertion Process

Start with empty sequence $y = ()$

For each step i:

- Sample where to insert:
  $z_i \in \{0, \ldots, |y|\} \sim p_\theta(z \mid y)$
- Sample what to insert:
  $x_i \in V \sim p_\theta(x \mid y, z_i)$
- $y \leftarrow \text{insert}(y, z_i, x_i)$ if not terminate

$$
\begin{aligned}
y^0 &\quad () \\
y^1 &\quad (x_1) \\
y^2 &\quad (x_1, x_2) \\
y^3 &\quad (x_3, x_1, x_2) \\
y^4 &\quad (x_3, x_4, x_1, x_2) \\
&\quad \vdots
\end{aligned}
\qquad
\begin{aligned}
z_1 &= 0 \\
z_2 &= 1 \\
z_3 &= 0 \\
z_4 &= 1
\end{aligned}
$$

Zhang, Wang, Gretton, Ying, van Dijk, Titsias, & Shi (2016). Variational Learning for Insertion-based Generation.

Variational Learning for Insertion-based Generation

- Exact likelihood is intractable. Standard variational learning maximize the ELBO:

$$\log p(y) - \mathrm{KL}(q(\{z_i\}_{i=1}^{L} \mid y) \| p(\{z_i\}_{i=1}^{L} \mid y))$$

- **Challenge**: hard to specify the distribution over z: different sequences of z that lead to the same y do not live in a fixed coordinate system.
- **Solution**: Use permutation-insertion bijection to reparameterize the likelihood:

$$z_i = f(\sigma_{\leq i}) \triangleq 1 + |\{j \in \sigma_{<i} : j < \sigma_i\}|$$

$$p(y) = \sum_{\sigma} p(y_{\sigma_i} \mid y_{\sigma_{<i}}, f(\sigma_{\leq i})) p(f(\sigma_{\leq i}) \mid y_{\sigma_{<i}}) = \sum_{\sigma} p(y, \sigma)$$

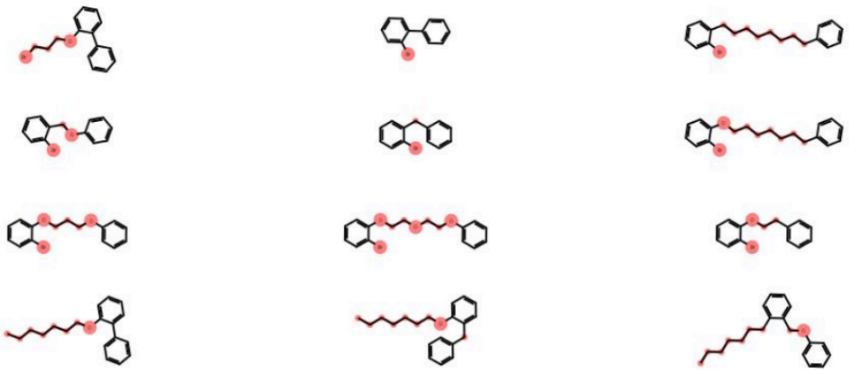
- Permutation ELBO:

$$\log p_\phi(y) \geq E_{q(\sigma|y)} \left[\log \frac{p(y, \sigma)}{q(\sigma \mid y)} \right]$$

- A further decomposition allows gradient-based training with low variance (Zhang et al., 2026)

Zhang, Wang, Gretton, Ying, van Dijk, Titsias, & Shi (2016). Variational Learning for Insertion-based Generation.

Insertion-based Generative Models for Molecular Strings



Model	V. ↑	V.U. ↑	V.U.N. ↑	KL div ↑	FCD ↑
Training set	100.0	100.0	0.0	99.9	92.8
LSTM†	95.9	95.9	87.5	99.1	**91.3**
VAE†	87.0	86.9	84.7	98.2	86.3
AAE†	82.2	82.2	82.1	88.6	52.9
FO-ARM (Transformer)	**98.5**	**98.3**	80.4	**99.4**	90.3
AO-ARM	83.1	83.1	80.7	93.1	69.3
LO-ARM	89.7	89.4	81.6	96.1	83.7
AO-IP	88.2	88.2	87.7	95.3	79.1
IP (policy-based term.)	97.2	97.0	94.9	97.1	88.2
IP (classifier-based term.)	97.4	97.3	**95.6**	97.2	89.2

1: Start of generation

2: Building molecule structure

3: Only inserting atoms

4:Termination

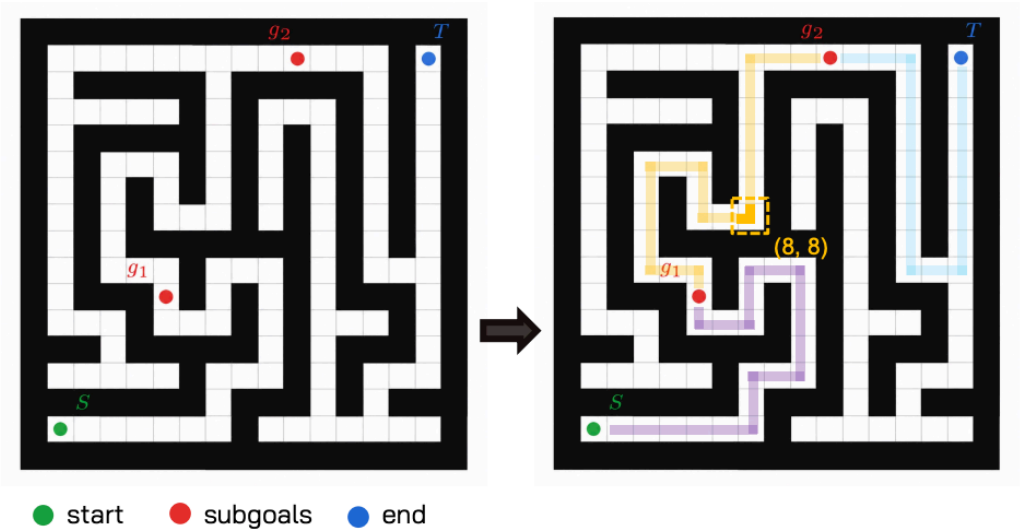
```
Step 000 | Sequence:
```

```
Step 001 | Sequence: 1
Step 003 | Sequence: (11
Step 006 | Sequence: 1()111
Step 007 | Sequence: 12()111
Step 009 | Sequence: 123()2111
Step 012 | Sequence: 123(443)2111
Step 016 | Sequence: 12=3=(443=)2=111
Step 017 | Sequence: =12=3=(443=)2=111
```

```
Step 018 | Sequence: =12=3=(443=)2=N111
Step 019 | Sequence: =12=3=(443=O)2=N111
Step 020 | Sequence: =12=3=(4c43=O)2=N111
Step 025 | Sequence: O=12=3=(4ccc43=O)O2=N11C1
Step 030 | Sequence: O=12=3=C(c4ccccc43=O)O2=N11CC1
Step 036 | Sequence: O=C1C2=C3=C(Cc4ccccc43=O)OC2=N1C1CC1
Step 037 | Sequence: O=C1C2=CC3=C(Cc4ccccc43=O)OC2=N1C1CC1
Step 043 | Sequence: O=C1C2=CC3=C(CCc4ccccc4C3=O)OC2=CN1C1CCCCC1
```

```
Step 044 | Sequence: O=C1C2=CC3=C(CCc4ccccc4C3=O)OC2=CN1C1CCCCC1
```

Goal-conditioned Planning via Insertion



Initial Anchors: $S \rightarrow g_1 \rightarrow g_2 \rightarrow T$

Choose gap: $S \rightarrow \square \rightarrow g_1 \rightarrow \square \rightarrow g_2 \rightarrow \square \rightarrow T$

$S \rightarrow \square \rightarrow g_1 \rightarrow \square \rightarrow g_2 \rightarrow \square \rightarrow T$

Insert: $S \rightarrow g_1 \rightarrow (8, 8) \rightarrow g_2 \rightarrow T$

	Braided			Imperfect			Perfect		
Method	Easy	Med	Hard	Easy	Med	Hard	Easy	Med	Hard
Monotonic (left-to-right) baselines									
FO-ARM	96.2	94.2	93.0	92.5	88.2	76.9	96.2	58.3	27.2
Non-monotonic baselines									
MDM	51.9	59.0	60.7	56.1	51.7	50.0	12.2	0.2	0.0
AO-ARM	54.2	61.8	65.4	60.3	55.2	51.3	11.7	1.1	0.0
LO-ARM	67.7	61.3	56.4	79.8	60.1	51.6	13.8	32.0	25.2
FlexMDM	86.9	86.7	89.5	78.5	83.1	84.2	0.1	0.0	0.0
IP (Ours)	**100.0**	**100.0**	**99.9**	**100.0**	**98.4**	**95.3**	**99.4**	**98.1**	**97.9**

Takeaways: Diffusion vs. Autoregression

- Both properties we believe to separate diffusion from AR are not so fundamental:
  - MDM/random-order AR can parallelize sampling as predictability grows during generation
  - With loss-reweighting, MDM/random-order AR can give diffusion-level perceptual quality
- Beyond AR and diffusion:
  - From fixed-order and random-order to learning-order
  - Insertion is critical for non-monotonic sequence generation
  - Scalable learning algorithms exist for these more powerful ways to generate

Main Papers

Shi, Han, Wang, Doucet, & Titsias. Simplified and generalized masked diffusion for discrete data. *NeurIPS 2024*.

Shi, & Titsias. Demystifying Diffusion Objectives: Reweighted Losses are Better Variational Bounds. *Preprint 2025*.

Wang, Shi, Heess, Gretton, & Titsias. Learning-Order Autoregressive Models with Application to Molecular Graph Generation. *ICML 2025*.

Zhang, Wang, Gretton, Ying, van Dijk, Titsias, & Shi. Variational Learning for Insertion-based Generation. *ICML 2026*.

Joint work with Michalis Titsias, Kehang Han, Zhe Wang, Arnaud Doucet, Yangtian Zhang, Nicolas Heess, Arthur Gretton, Rex Ying, David van Dijk

Appendix

Variational Learning for Insertion-based Generation

- Define the variational distribution over permutations as a Plackett–Luce (PL) model

$$q_\theta(\sigma \mid y) = \prod_{i=1}^{L} q_\theta(\sigma_i \mid \sigma_{<i}, y) \qquad q_\theta(\sigma_i \mid \sigma_{<i}, y) \propto \exp(g_{\theta,\sigma_i})$$

- The ELBO further decomposes to

Exact marginalization

$$\sum_{i=1}^{L} \mathbb{E}_{q_\theta(\sigma_{<i}|y)} \left[\mathbb{E}_{q_\theta(\sigma_i|\sigma_{<i},y)} \left[\log \frac{p_\phi(y_{\sigma_i} \mid f(\sigma_{\le i}), y_{\sigma_{<i}}) p_\phi(f(\sigma_{\le i}) \mid y_{\sigma_{<i}})}{q_\theta(\sigma_i \mid \sigma_{<i}, y)} \right] \right]$$

Sampling: Parallel Gumbel Top-k
Gradients: REINFORCE Leave-One-Out