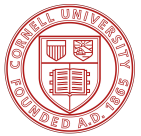
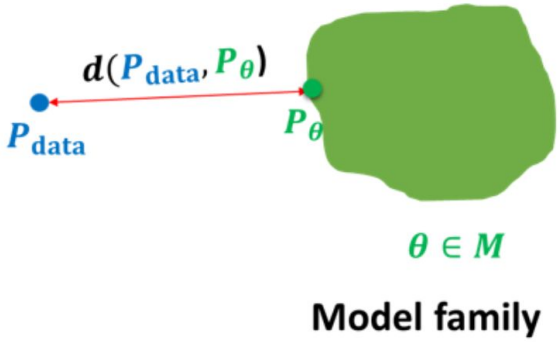
Discrete Diffusion Models



The Task of Generative Modeling

Suppose we are given a training set of examples, e.g., images of dogs

$x_i \sim P_{\text{data}}$
$i = 1, 2, \ldots, n$

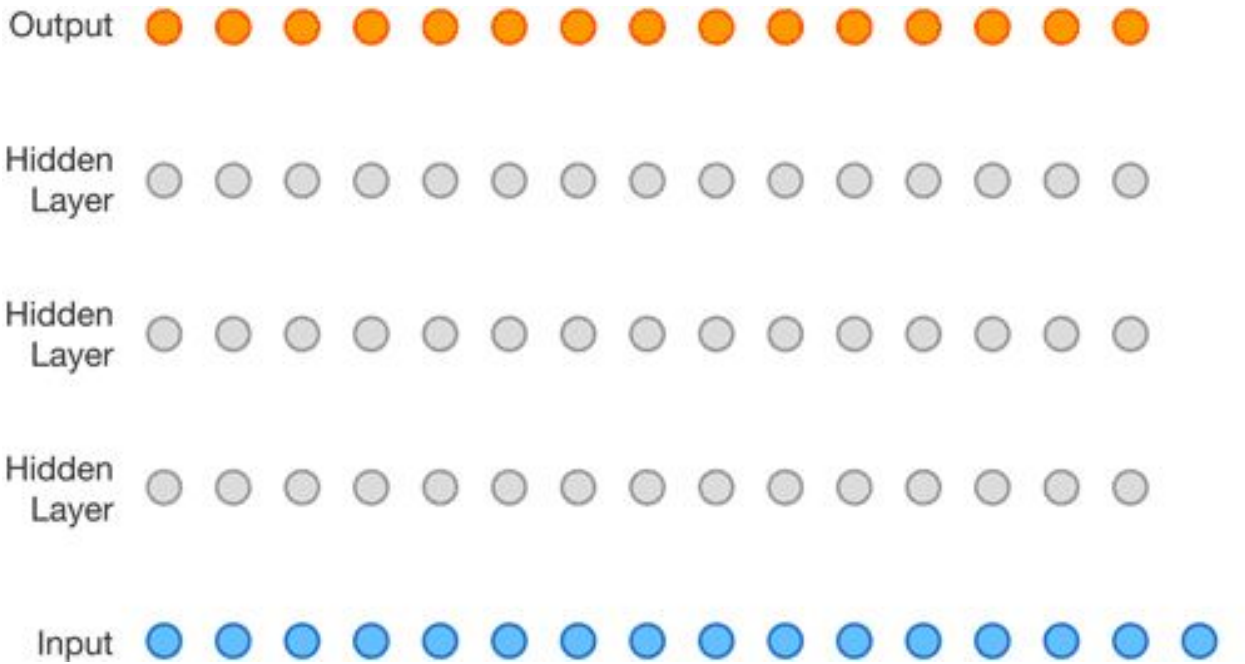


Our Goal: define a probability distribution $p(x)$ over images x.

1. **Modeling**: Define a family of distributions p_θ parameterized by θ.
2. **Learning**: Search for model parameters θ based on training data $\mathcal{D}$.
3. **Inference**: Generate data x by sampling from $p_\theta(x)$.

Last Lecture: Autoregressive Models (Overview)

Autoregressive models generate data sequentially by predicting the next element from previously generated ones.



Last Lecture: Autoregressive Models (Theory)

- Without loss of generality, we can use chain rule for factorization

$$p(x_1, \cdots, x_{784}) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \cdots p(x_n \mid x_1, \cdots, x_{n-1})$$

- We parameterize the conditionals using **logistic regression**:

$$p(x_1, \cdots, x_{784}) = p(x_1; \theta)p_{\text{logit}}(x_2 \mid x_1; \boldsymbol{\theta})p_{\text{logit}}(x_3 \mid x_1, x_2; \boldsymbol{\theta}) \cdots p_{\text{logit}}(x_n \mid x_1, \cdots, x_{n-1}; \boldsymbol{\theta})$$

Given training data $\mathcal{D} = \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(m)}\}$, **maximum likelihood learning** solves:

$$\max_{\theta} \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \sum_{i=1}^{n} \log p_{\text{neural}}(x_i | \mathbf{x}_{<i}; \theta)$$

Last Lecture: Autoregressive Models (Pros & Cons)

Autoregression

There

✅ Exact Likelihood Evaluation & Training

✅ Exact Sampling

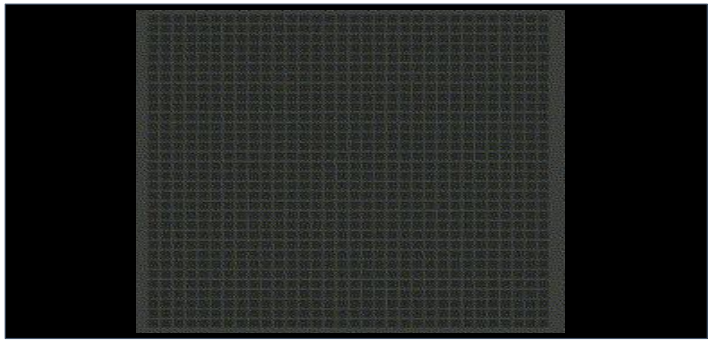
✅ Generates Sequences of Arbitrary Length

❌ Sequential (Slow) Generation

❌ No Latent Representations

❌ Compounding Errors At Generation

Today's Generative Modeling Landscape

	Autoregression	Diffusion
Images		
Text		]? trunk }) oĸa(()ns ..... (((Abstractreported–but',// ..... ((.(.(ns(handlerewis(___ originates.getDayROSS.).....cameractic¯ norms prosper taxing])); \Models. **This talk**

Diffusion vs. Autoregressive Language Models

Autoregression

There

❌ Sequential Generation

❌ Previous Tokens Can't Be Edited

❌ Fixed Number of Generation Steps

❌ Causal Context

Diffusion

Hirsch
account

✅ Parallel (fast) Generation

✅ Can Fix Previous Mistakes

✅ Generate in More/Fewer Steps

✅ Bidirectional Context

Diffusion vs. Autoregressive Language Models

Autoregression

There

Diffusion

Hirsch
account

Today's hardware in parallel, but LLM inference algorithms are sequential.
Diffusion is the right language modeling approach for today's hardware.

This Talk: Simple Masked Diffusion Language Models

$$x \sim p_\theta(x)$$

Many years later, as he faced the firing squad, Colonel Aureliano Buendía was to remember that distant afternoon when his father took him to discover ice. At that time Macondo was a village of twenty adobe houses, built on the bank of a river of clear water that ran along a bed of polished stones, which were white and enormous, like prehistoric eggs. The world was so recent that many things lacked names, and in order to indicate them it was necessary to point.

Credit for slides: Sasha Rush

Sampling from a Masked Language Model

____ _____ ______ __ __ _____ ___ ______ ______ _______ _________
_______ ___ __ ________ ____ _______ _________ ____ ___ ______
____ ___ __ ________ ____ __ ____ ____ _______ ___ _ _______ __
______ _____ _______ _____ __ ___ ____ __ _ _____ __ _____ _____
____ ___ _____ _ ___ __ ________ _______ _____ ____ _____ ___
_________ ____ __________ _____ ___ _____ ___ __ ______ ____ ____
______ ______ ______ ___ __ _____ __ ________ ____ __ ___
_________ __ ______

---- ----- ------ -- -- ----- --- ------ *squad,* ------- ---------
------- --- *to remember* ---- ------- --------- *when* --- ------
---- --- *to* -------- ---- -- ---- ---- ------- --- - ------- --
twenty adobe ------- ----- -- --- ---- -- - ----- *of* ----- *water*
---- --- *along* - --- -- *polished* ------- *which* ---- ----- ---
--------- ---- *prehistoric* ----- --- ----- --- -- ------ ---- *many*
------ ------ ------ --- *in* ----- -- -------- ---- -- ---
--------- -- ------

*Many years later, as he faced --- ------ squad, ------- ---------
Buendía was to remember that distant --------- when his ------
took --- to discover ---- At that ---- ------- was a village of
twenty adobe houses, built -- the bank of - river of clear water
that ran along - --- of polished stones, which were ----- and
--------- like prehistoric eggs. --- ----- --- so recent ---- many
things lacked names, and in ----- -- -------- them it was
--------- to ------*

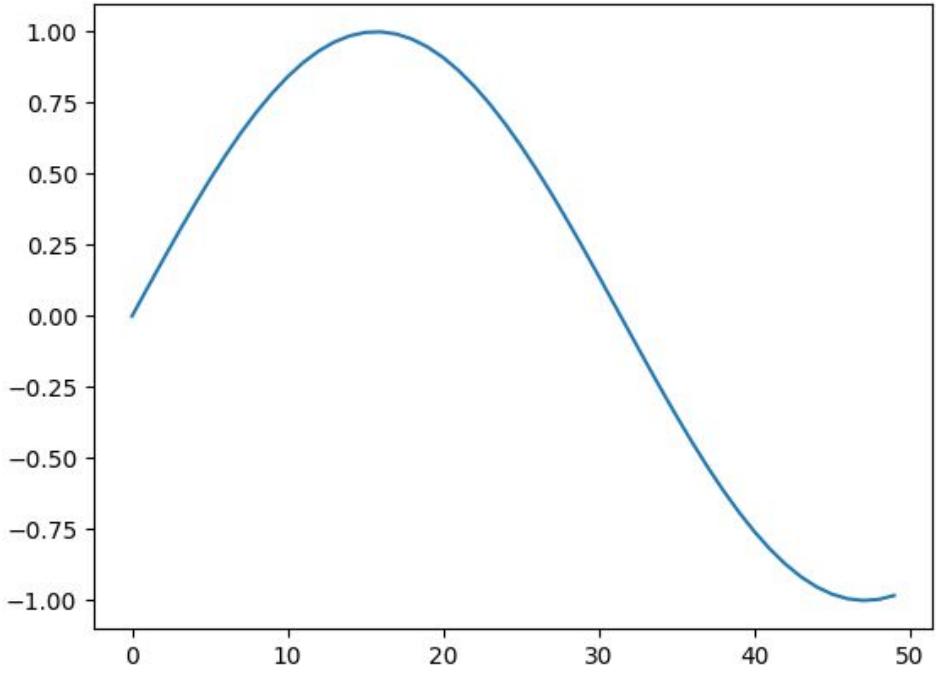
Many years later, as he faced the firing squad, Colonel Aureliano Buendía was to remember that distant --------- when his father took him to discover ice. At that time Macondo was a village of twenty adobe houses, built on the bank of - river of clear water that ran along a bed of polished stones, which were white and enormous, like prehistoric eggs. The world --- so recent that many things lacked names, and in order -- indicate them it was --------- to point.

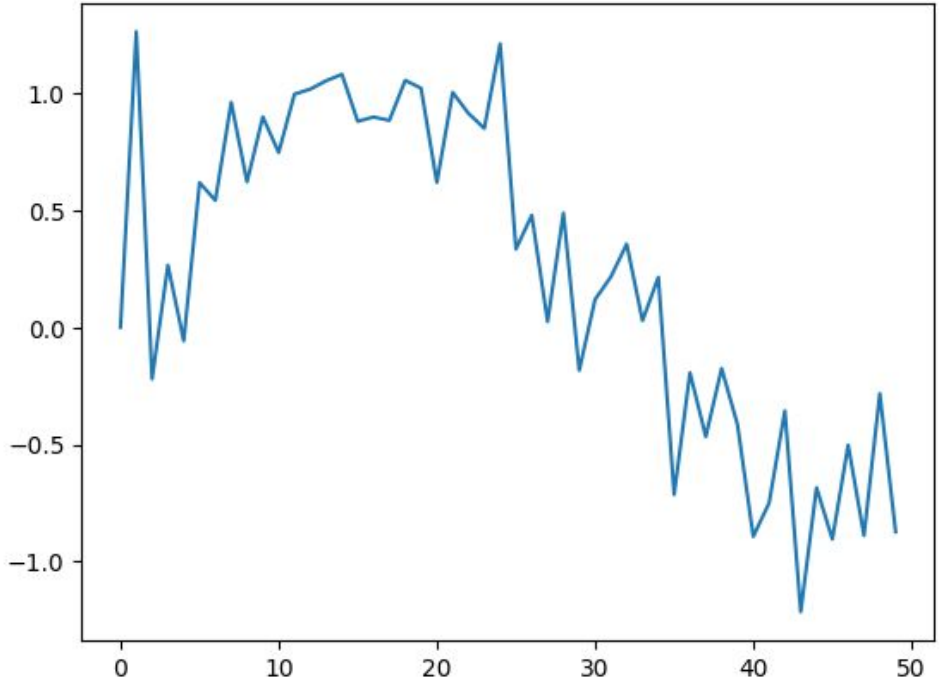
$$x \sim p_\theta(x)$$

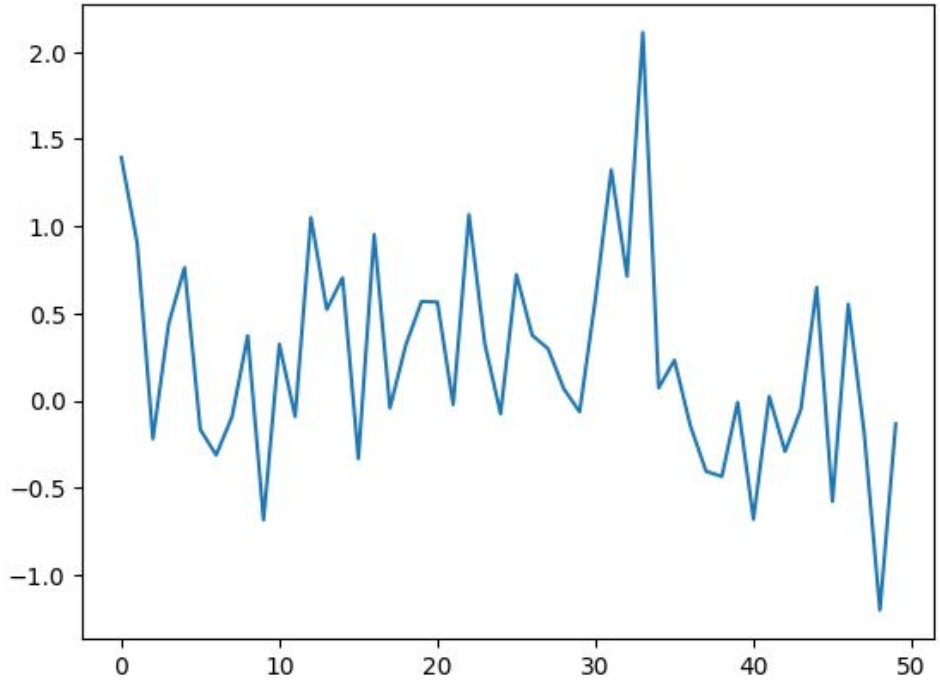
Many years later, as he faced the firing squad, Colonel Aureliano Buendía was to remember that distant afternoon when his father took him to discover ice. At that time Macondo was a village of twenty adobe houses, built on the bank of a river of clear water that ran along a bed of polished stones, which were white and enormous, like prehistoric eggs. The world was so recent that many things lacked names, and in order to indicate them it was necessary to point.

Diffusion Background

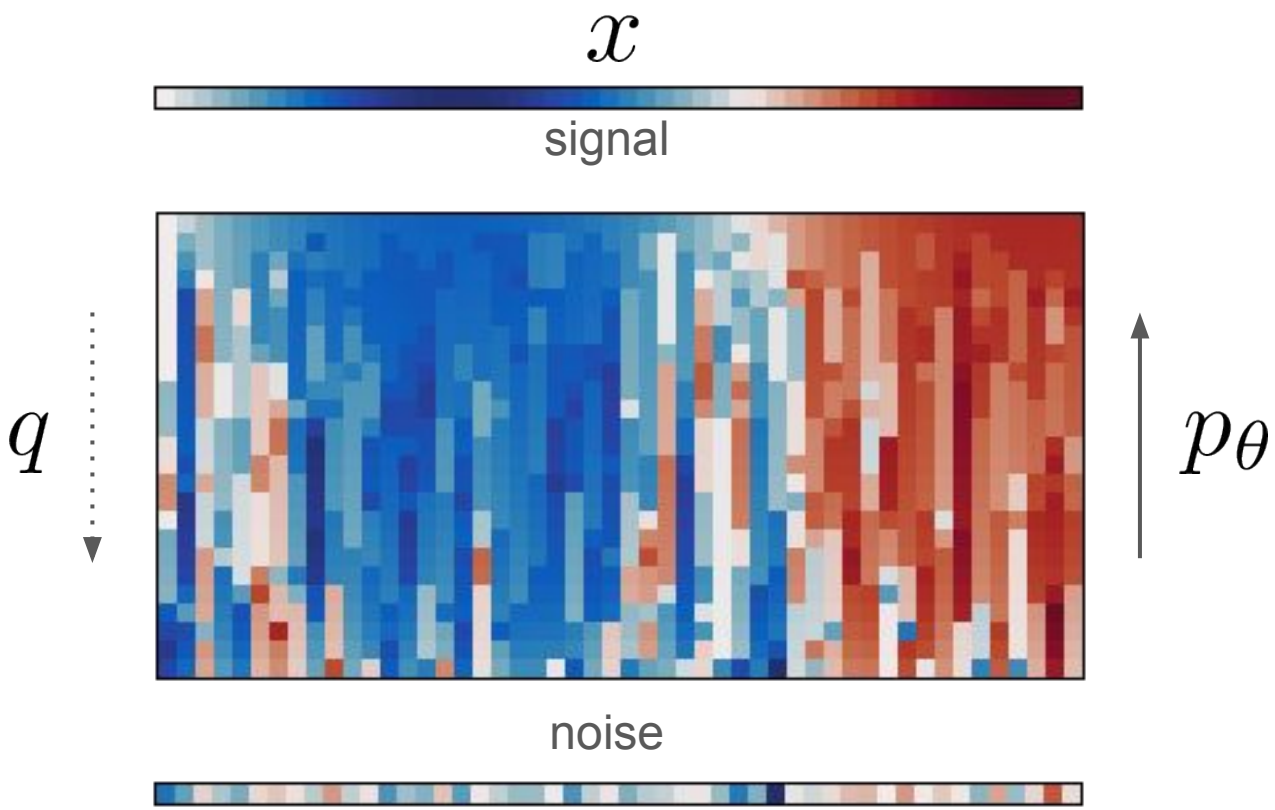
$$x \sim p_\theta(x)$$



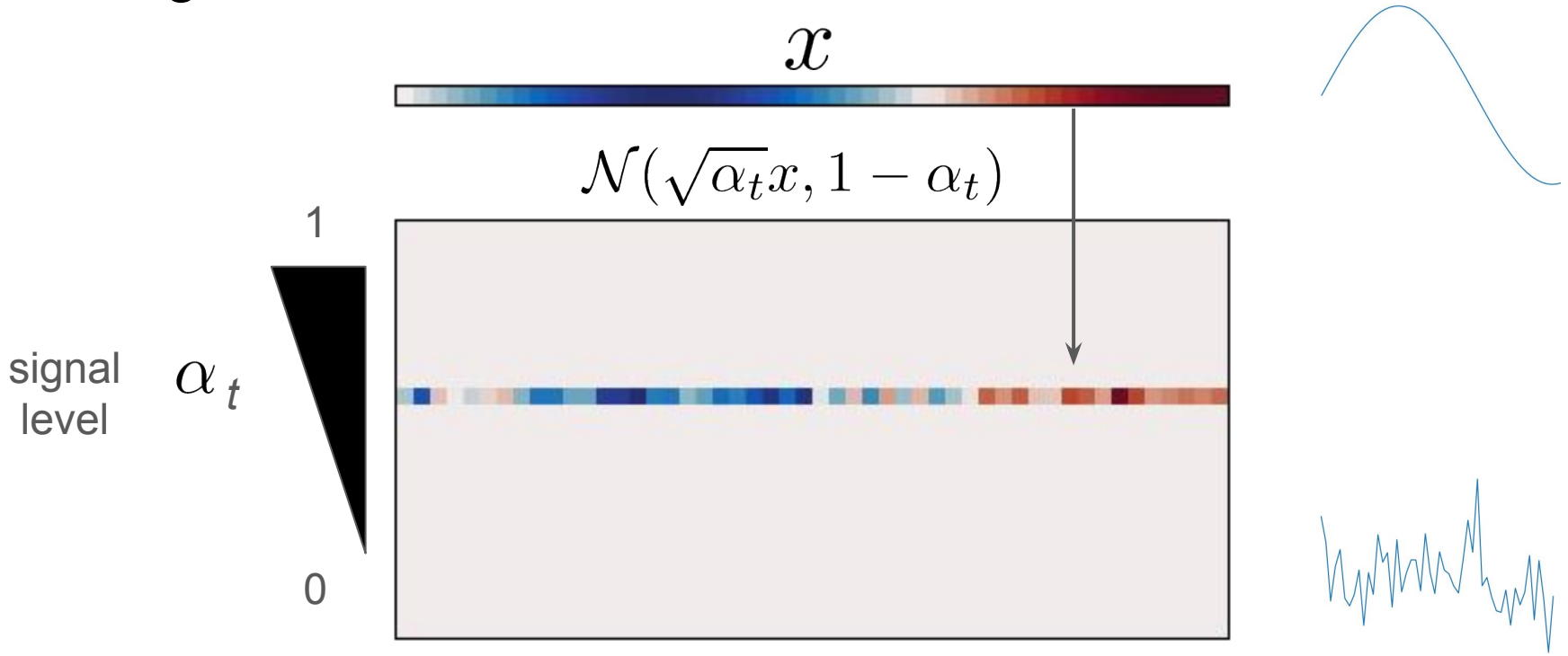

1.0
0.5
0.0
−0.5
−1.0
0
10
20
30
40
50


2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0
10
20
30
40
50

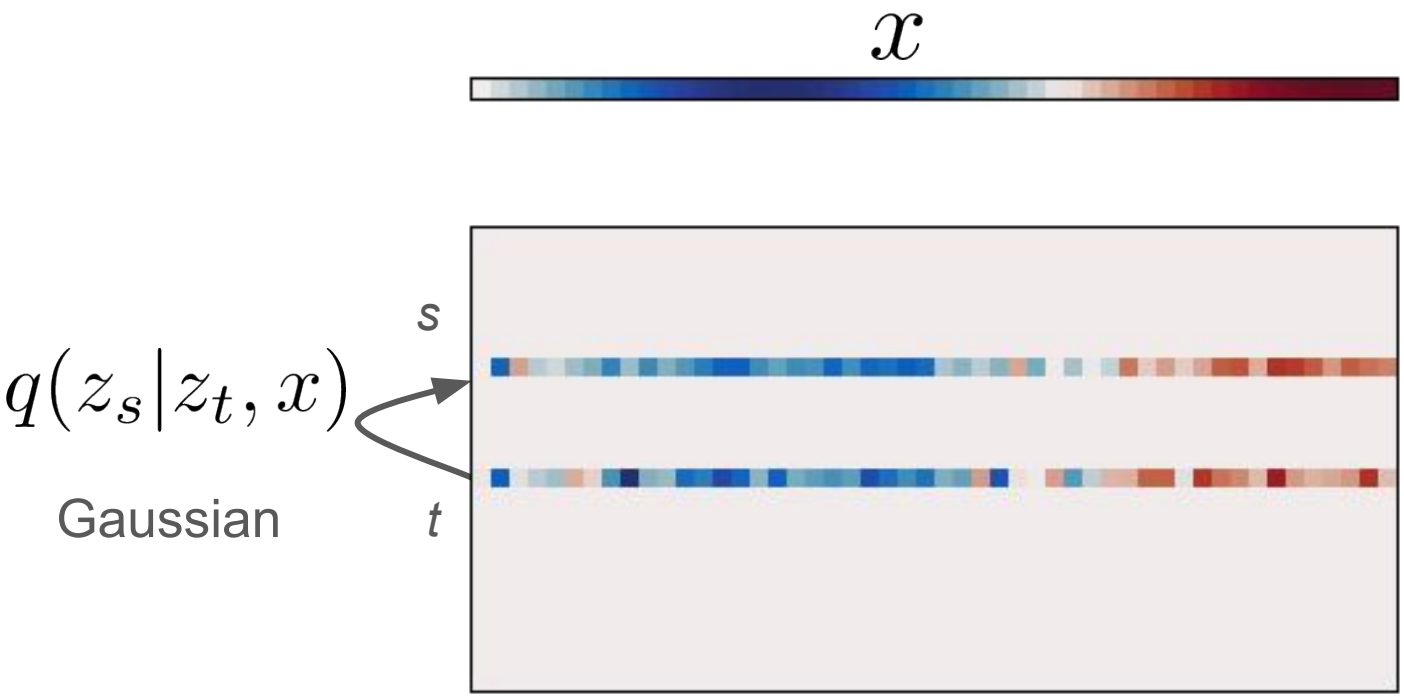
Gaussian Diffusion



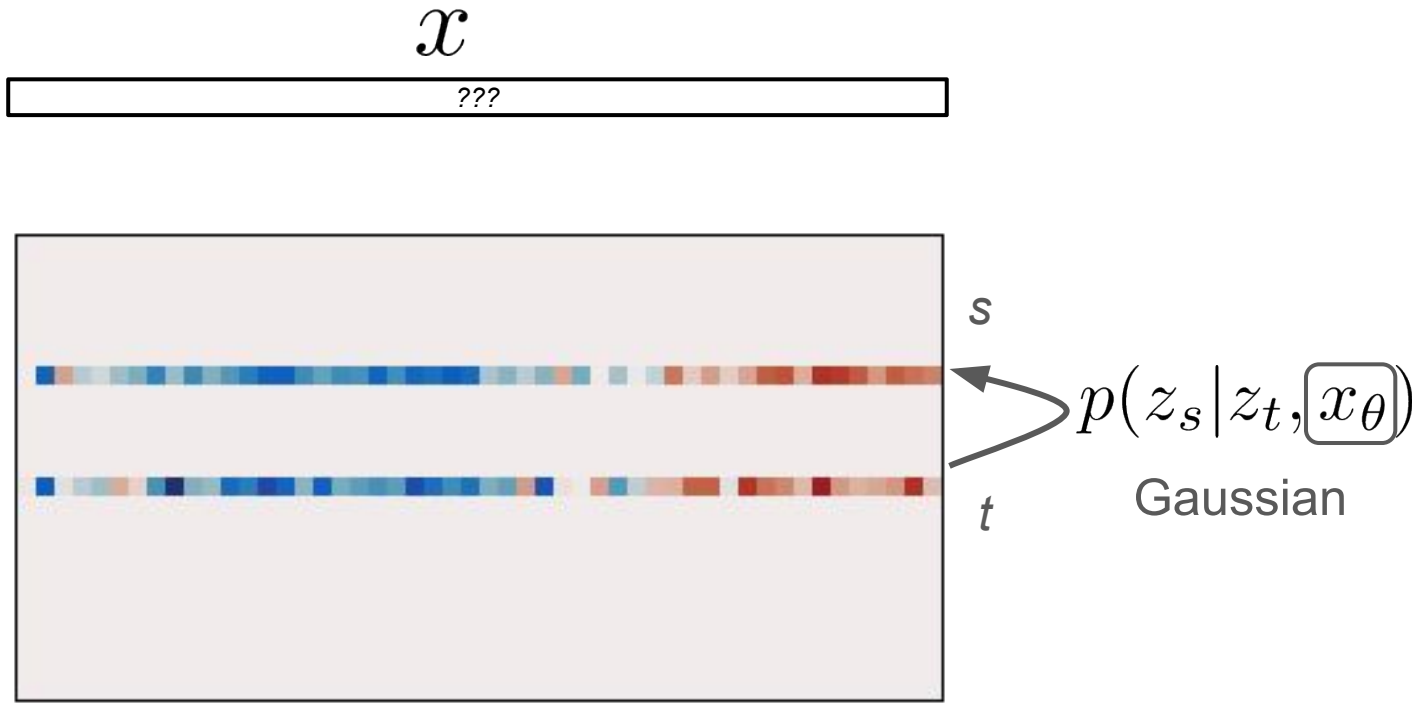
Noising Process: Gaussian Noise



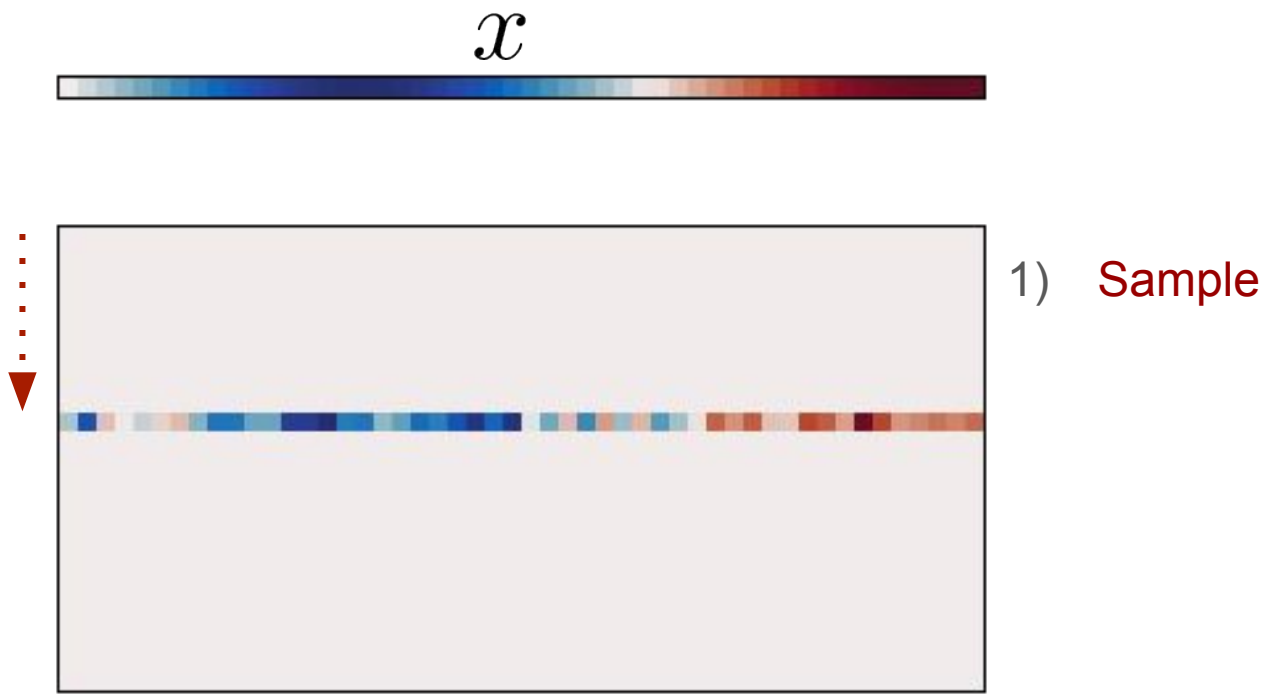
Gaussian Noise Implies Gaussian Posterior



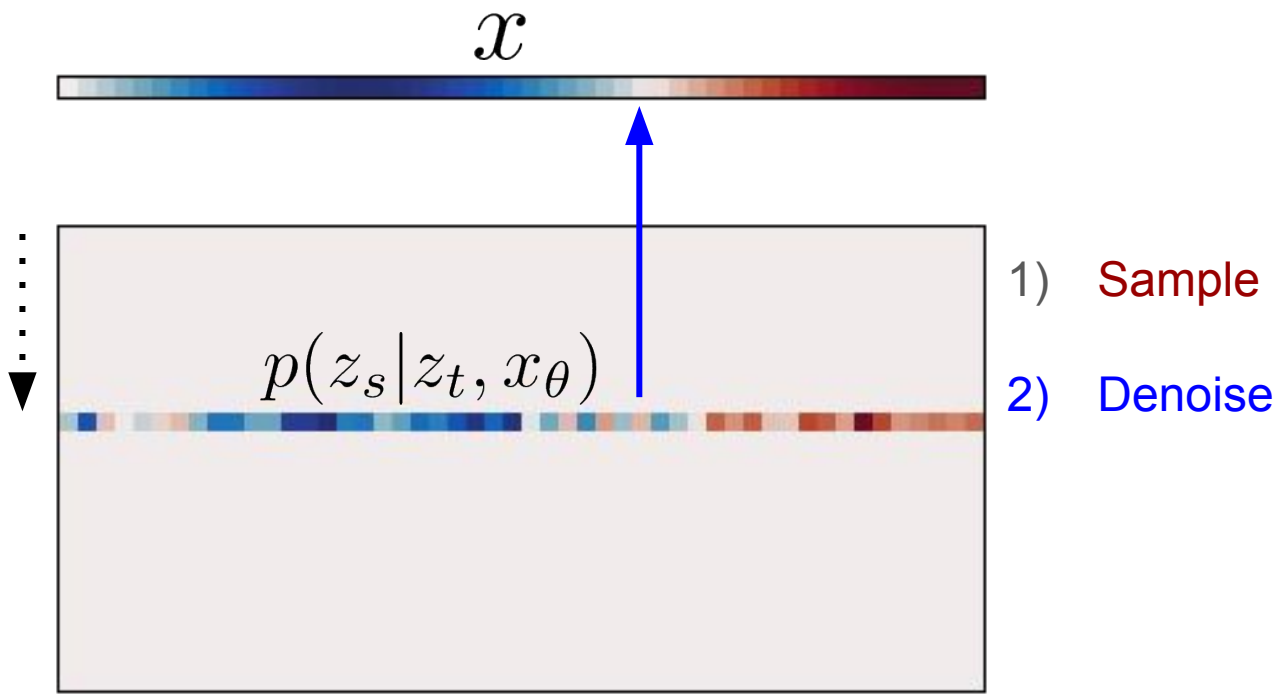
Reverse Process: Denoising



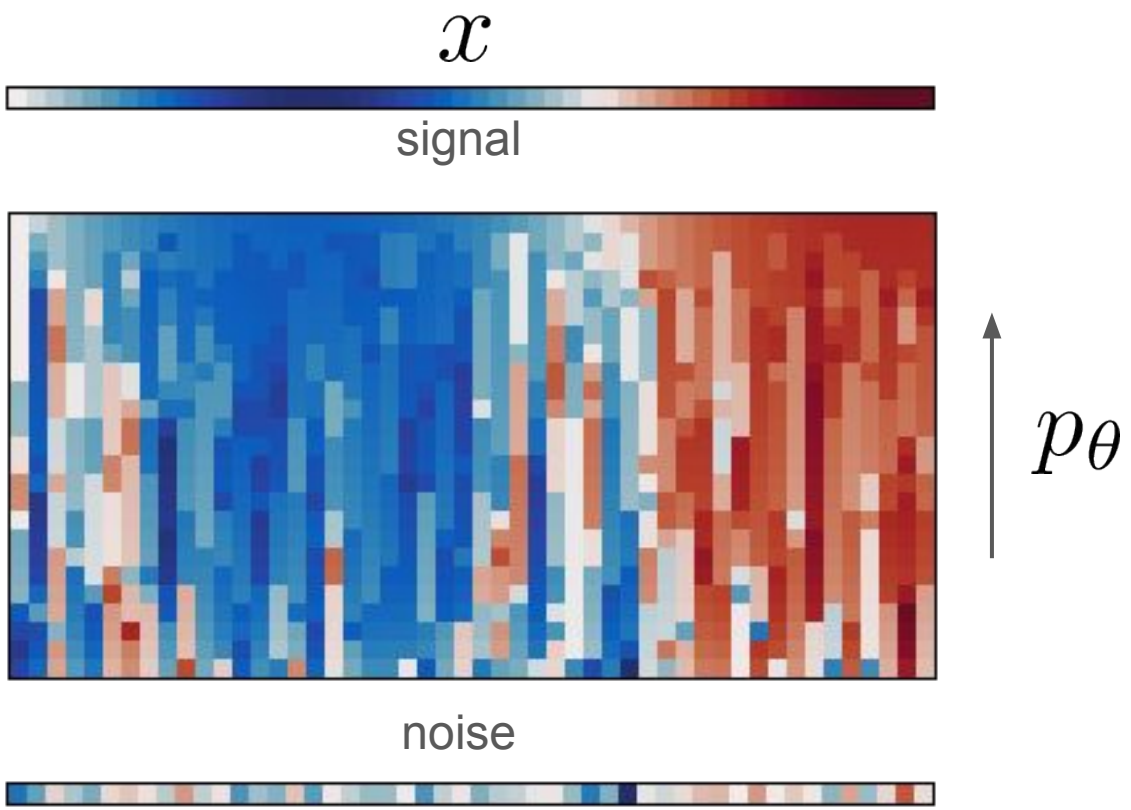
Learning to Denoise



Learning to Denoise



Generation: Iterative Denoising

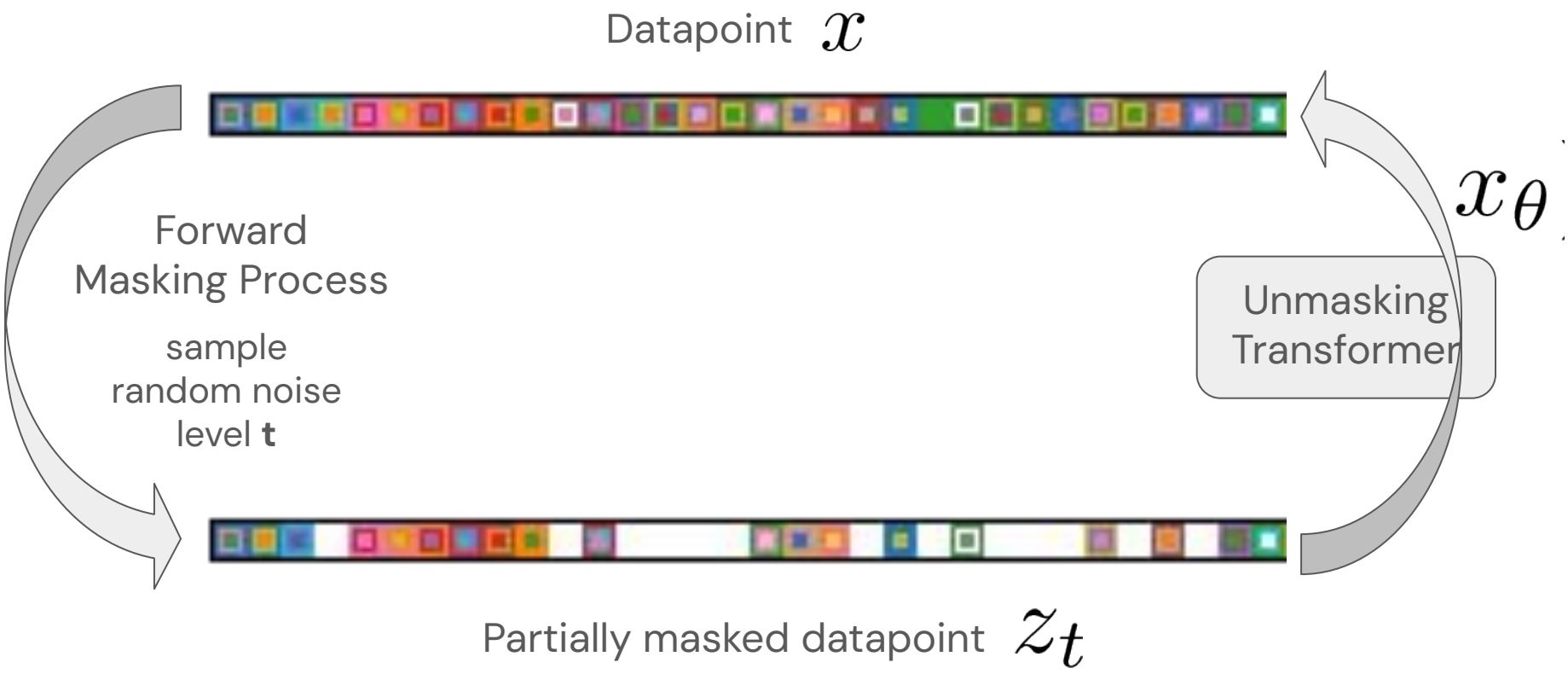


Simple Discrete Masking Diffusion

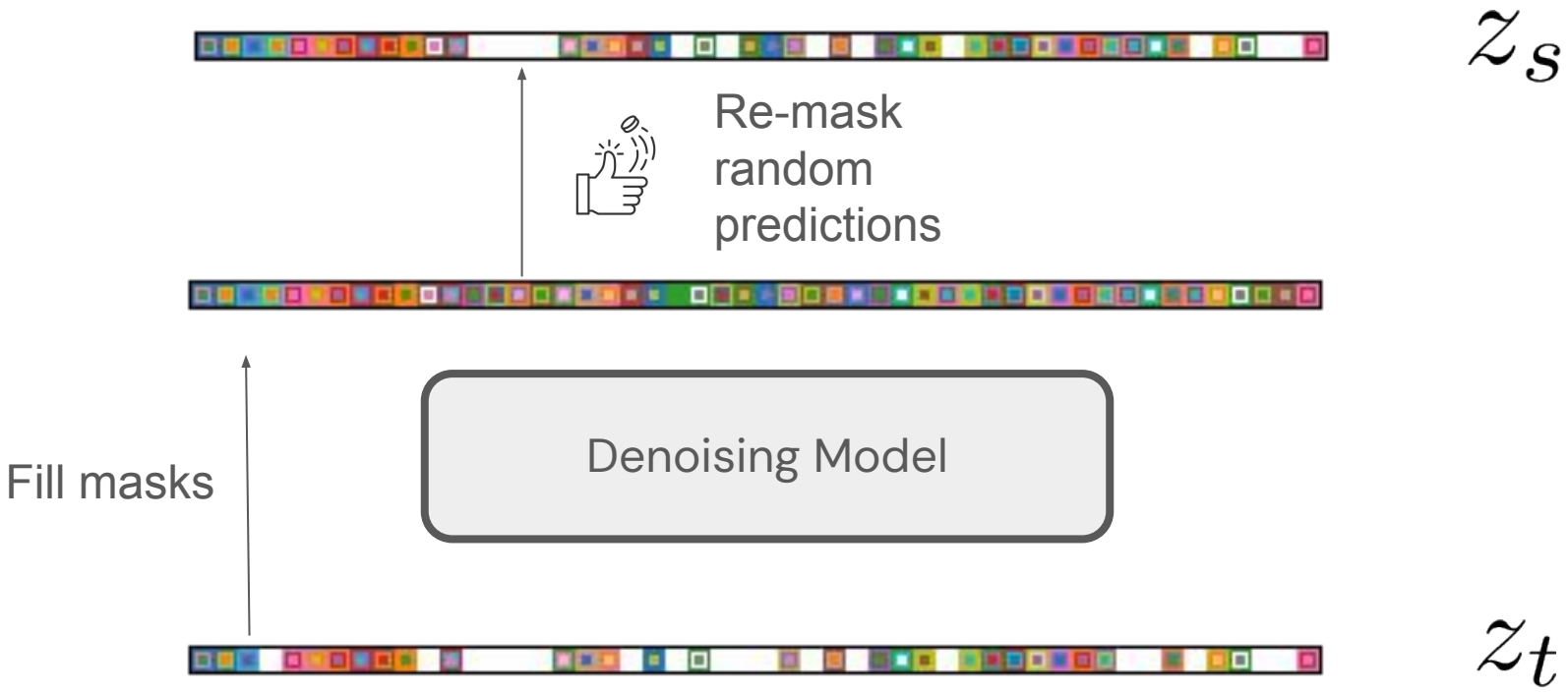
Many years later, as he faced the firing squad, Colonel Aureliano Buendía was to remember that distant afternoon when his father took him to discover ice. At that time Macondo was a village of twenty adobe houses, built on the bank of a river of clear water that ran along a bed of polished stones, which were white and enormous, like prehistoric eggs. The world was so recent that many things lacked names, and in order to indicate them it was necessary to point.

Many years later, -- he faced --- firing squad, ------- Aureliano Buendía was to remember that ------- afternoon when his father took him to discover ice. At that time Macondo was a village of twenty adobe houses, built on the ---- of a river of clear water that ran ----- a bed of polished stones, which were ----- --- enormous, like prehistoric eggs. The ----- was so recent that many things lacked names, and in order to indicate them -- was necessary to point.

Model Parameterization & Training: Unmasking Transformer



One Step of Generation



Multiple Steps of Generation

Many years later, as he faced squad,
Buendía was to remember that distant when his
took to discover At that was a village of
twenty adobe houses, built the bank of river of clear water
that ran along of polished stones, which were and
like prehistoric eggs. so recent many
things lacked names, and in them it was
to

Multiple Steps of Generation

Many years later, as he faced the firing squad, Colonel Aureliano Buendía was to remember that distant when his father took him to discover ice. At that time Macondo was a village of twenty adobe houses, built on the bank of river of clear water that ran along a bed of polished stones, which were white and enormous, like prehistoric eggs. The world so recent that many things lacked names, and in order indicate them it was to point.

Multiple Steps of Generation

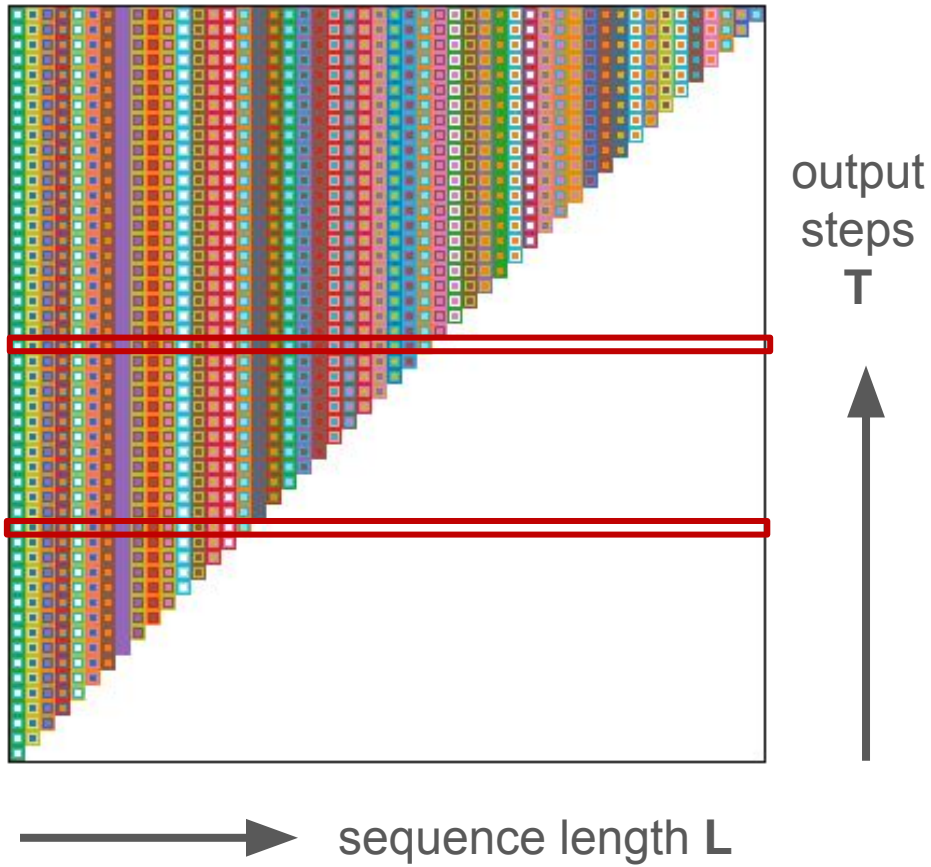
Many years later, as he faced the firing squad, Colonel Aureliano Buendía was to remember that distant afternoon when his father took him to discover ice. At that time Macondo was a village of twenty adobe houses, built on the bank of a river of clear water that ran along a bed of polished stones, which were white and enormous, like prehistoric eggs. The world was so recent that many things lacked names, and in order to indicate them it was necessary to point.

Example: Autoregressive Unmasking

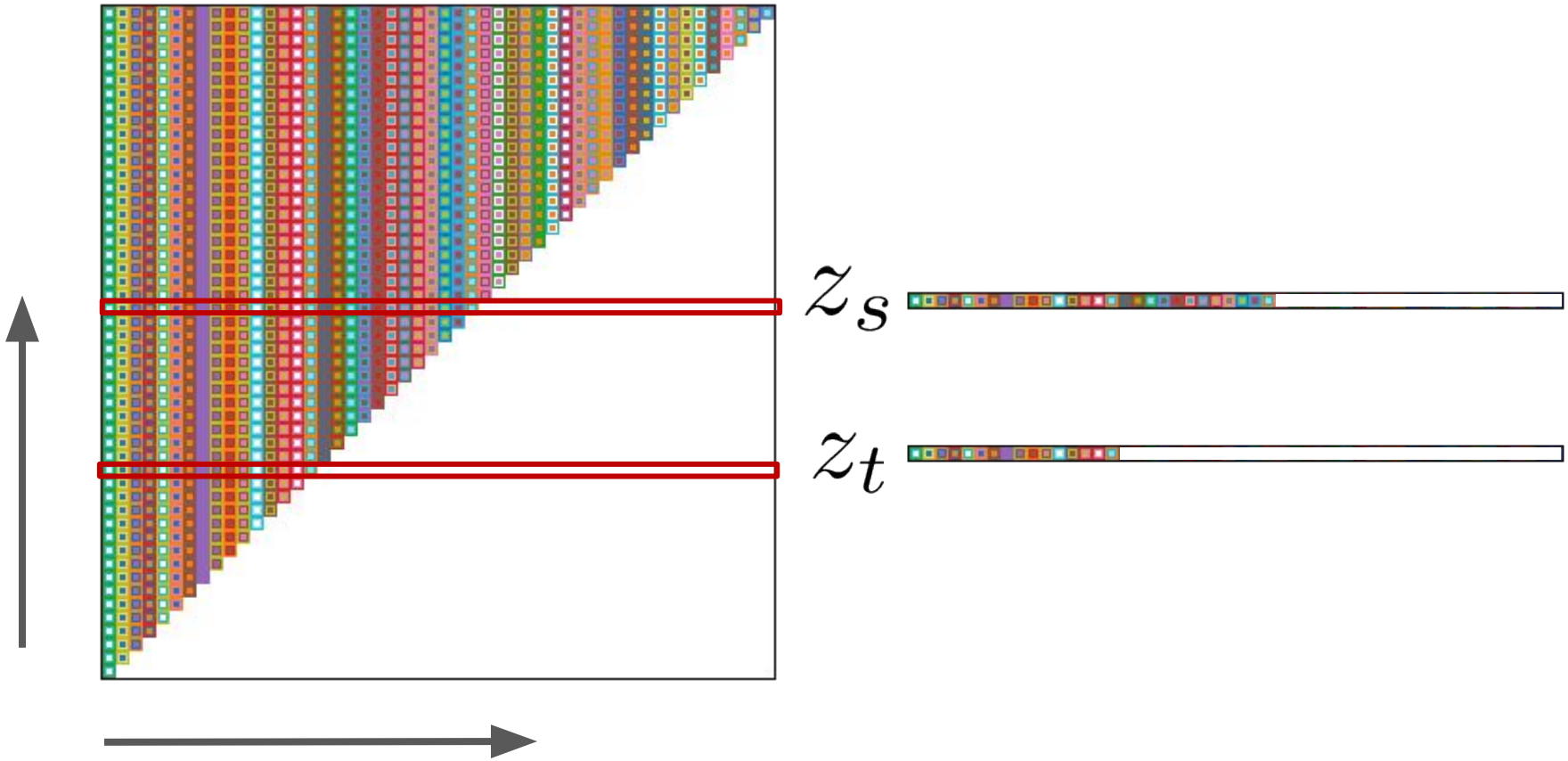
There

z_s

z_t



Example: Autoregressive Unmasking

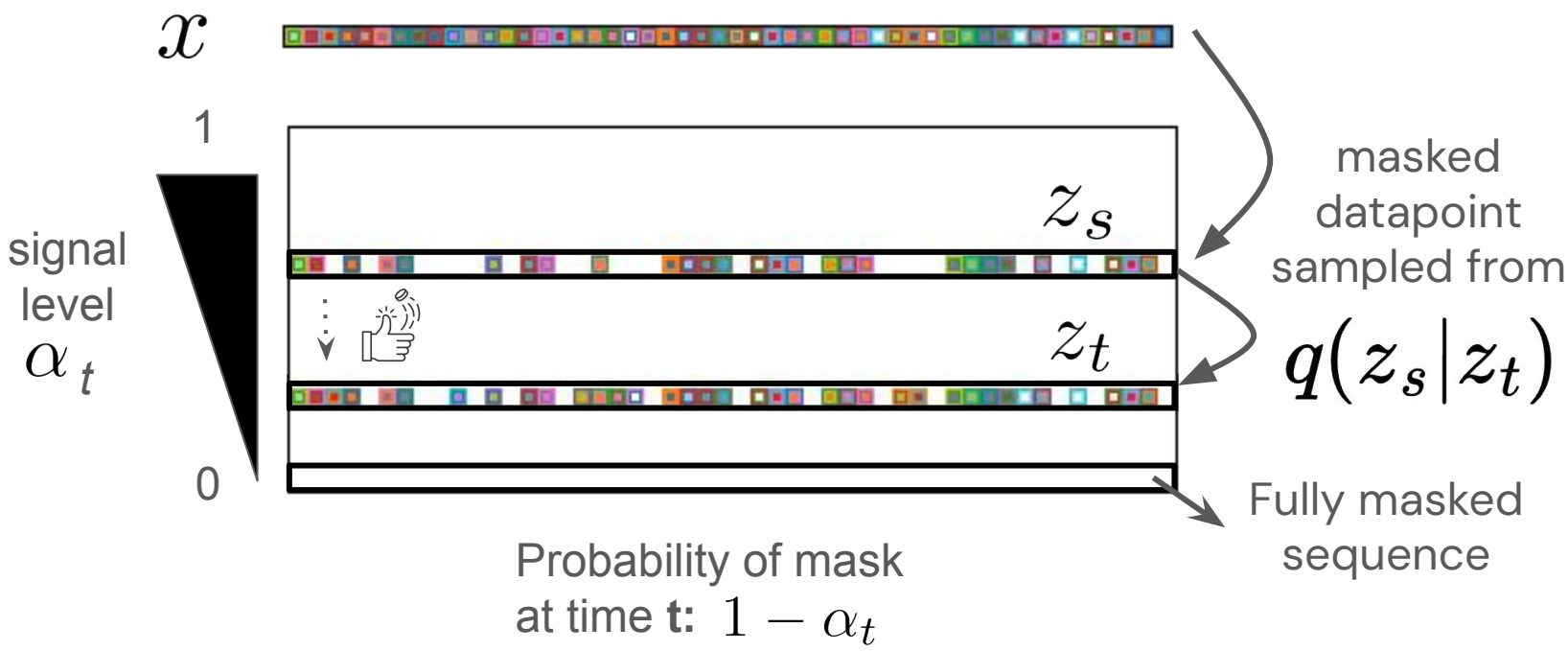


Modeling: Defining a Generative Model **p**

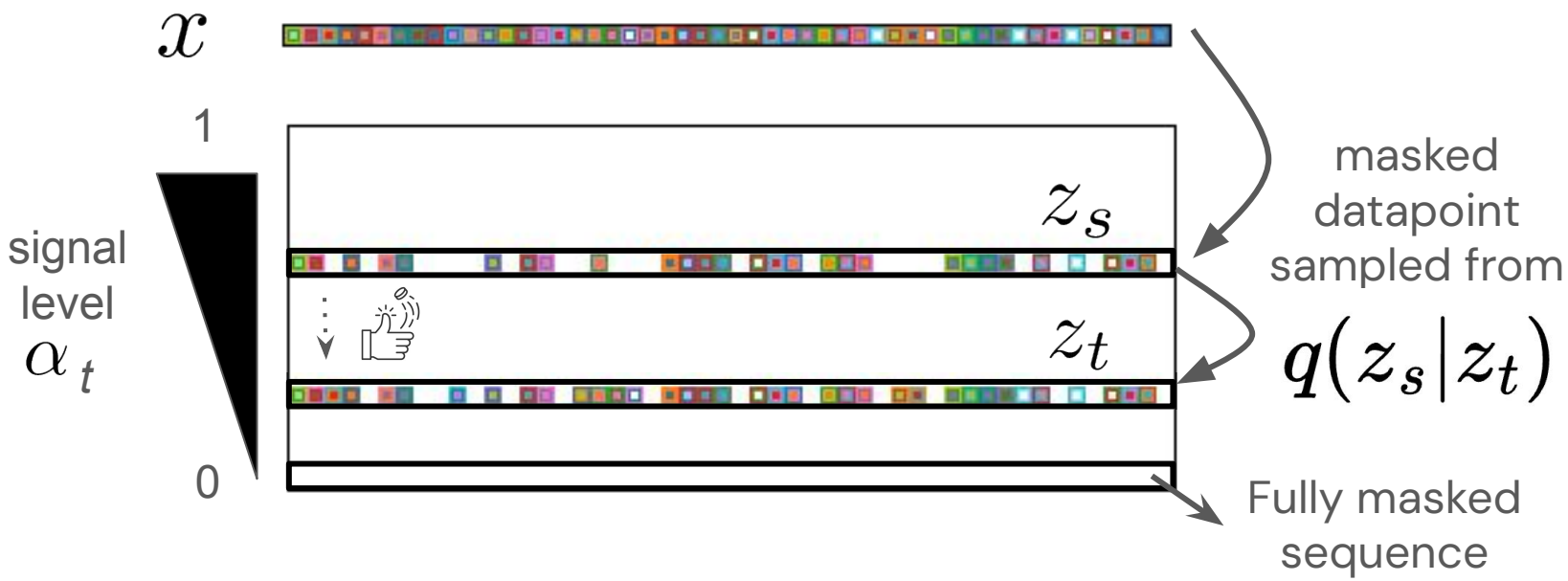
Hirsch account

$$x, z_0, \ldots, z_T \sim p(x|z_0) \prod_{t=1}^{T} p(z_{t-1}|z_t)$$

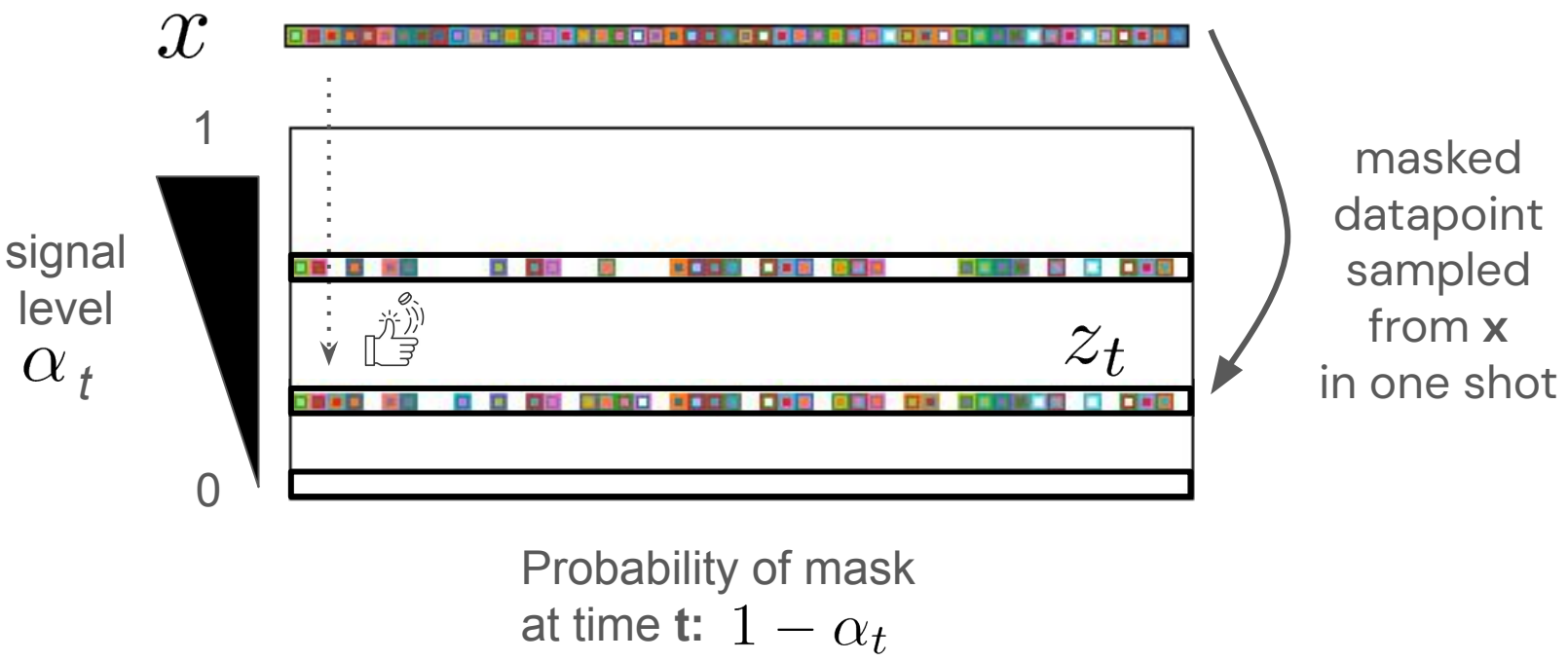
How Do We Train **p**? To Reverse Noise Process **q**



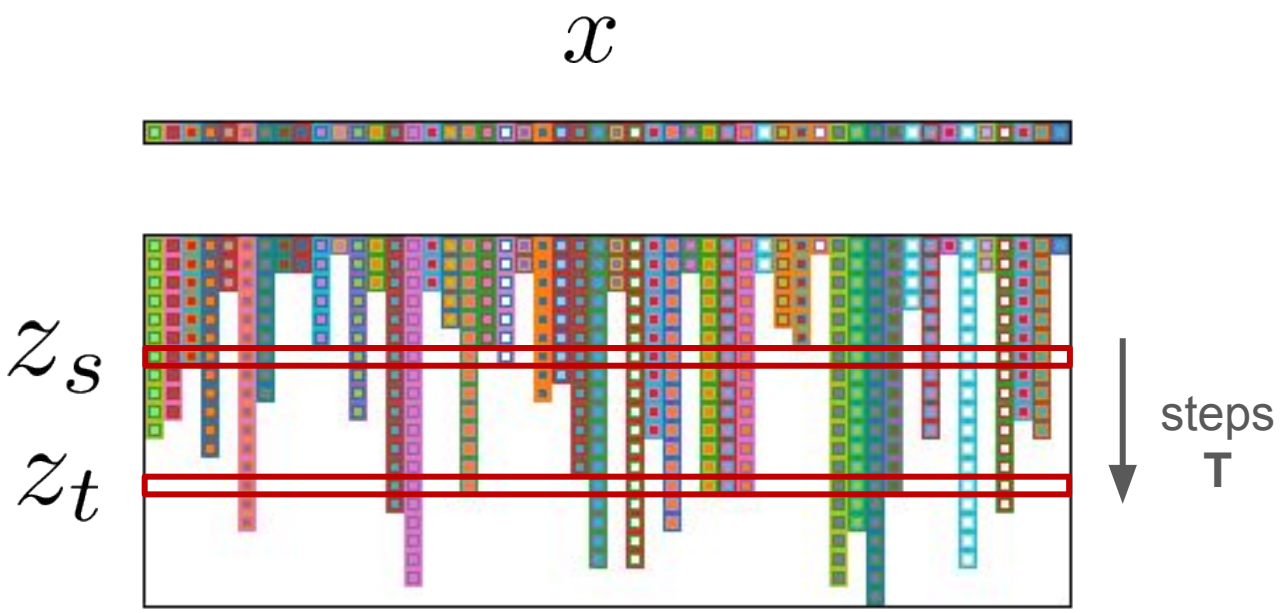
How Do We Train **p**? To Reverse Noise Process **q**



How Do We Train **p**? To Reverse Noise Process **q**



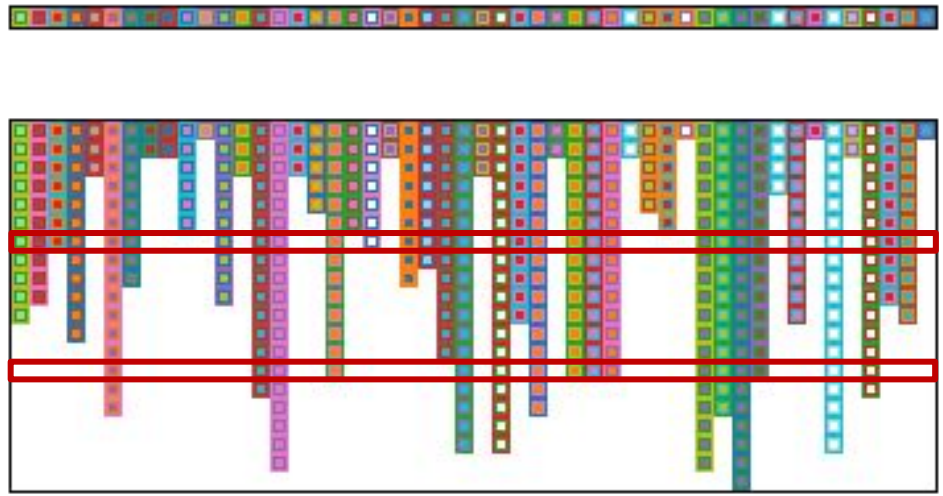
How Do We Train **p**? To Reverse Noise Process **q**



The noise process defines a path from clean **x** to masked **z**'s.

How Do We Train **p**? To Reverse Noise Process **q**

x



$q(z_s|z_t, x)$

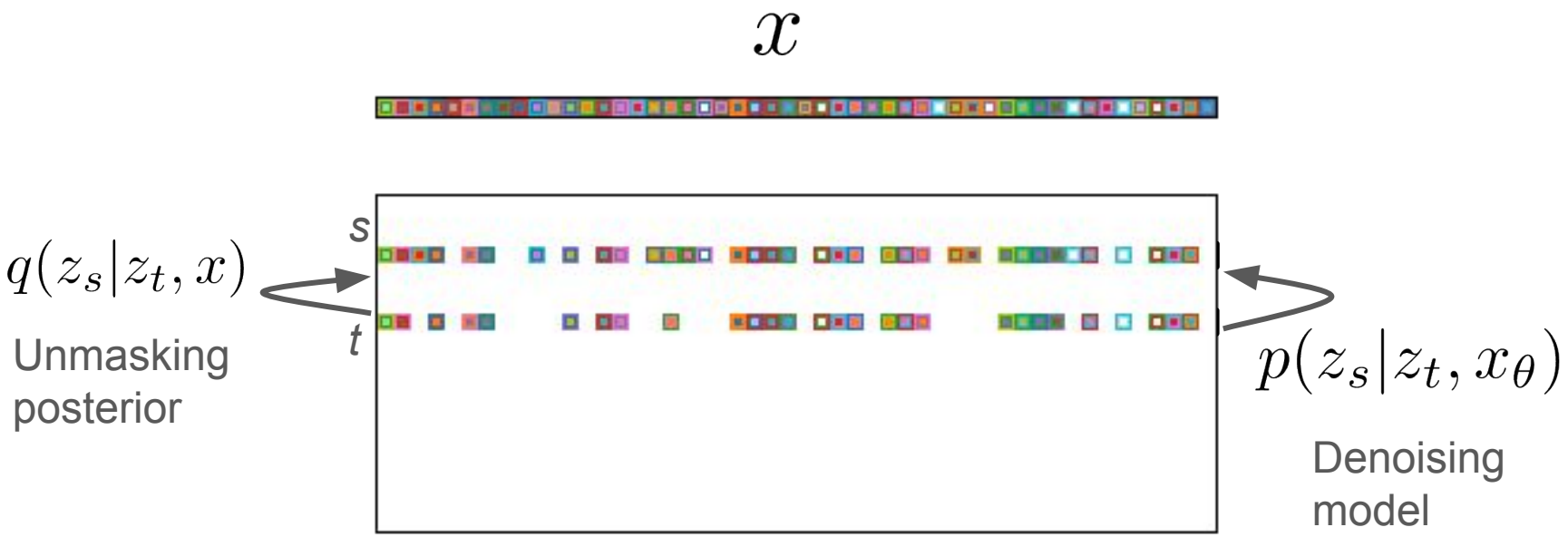
Optimal unmasking process

$p(z_s|z_t, x_\theta)$

Learned denoising model

The noise process defines a path from clean **x** to masked **z**'s.

How Do We Train **p**? To Reverse Noise Process **q**



Learning **p**: A Variational Inference Perspective

The classical approach to learning a latent variable **p** is to maximize the ELBO:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|(\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}|\mathbf{x})]$$

The (negative) ELBO has (in general) the following form:

$$\mathcal{L}_{\text{NELBO}} = \mathbb{E}_q \Big[\mathcal{L}_{\text{reconstr}} + \underbrace{\mathbb{E}_t \text{KL}(q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})||p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t))}_{\mathcal{L}_{\text{diffusion}}} + \mathcal{L}_{\text{prior}} \Big]$$

Learning **p**: Simplifying the ELBO

The (negative) ELBO for masked diffusion has (in general) the following form:

$$\mathcal{L}_{\text{NELBO}} = \mathbb{E}_q \Bigg[\mathcal{L}_{\text{reconstr}} + \underbrace{\mathbb{E}_t \text{KL}(q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x}) || p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t))}_{\mathcal{L}_{\text{diffusion}}} + \mathcal{L}_{\text{prior}} \Bigg]$$

Naively, an unsimplified discrete diffusion loss equals the following:

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E} \left[\frac{\alpha_s - \alpha_t}{1 - \alpha_t} \log \frac{\alpha_t \mathbf{x}_\theta(\mathbf{z}_t)^\top \mathbf{m} + (1 - \alpha_t)}{(1 - \alpha_t)\mathbf{x}_\theta(\mathbf{z}_t)^\top \mathbf{x}} + \frac{1 - \alpha_s}{1 - \alpha_t} \log \frac{(1 - \alpha_s)(\alpha_t \mathbf{x}_\theta(\mathbf{z}_t)^\top \mathbf{m} + (1 - \alpha_t))}{(1 - \alpha_t)(\alpha_s \mathbf{x}_\theta(\mathbf{z}_t)^\top \mathbf{m} + (1 - \alpha_s))} \right] \mathbf{z}^\top \mathbf{m}$$

Learning **p**: Simplifying the ELBO

The (negative) ELBO for masked diffusion has (in general) the following form:

$$\mathcal{L}_{\text{NELBO}} = \mathbb{E}_q \Bigg[\mathcal{L}_{\text{reconstr}} + \underbrace{\mathbb{E}_t \text{KL}(q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x}) || p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t))}_{\mathcal{L}_{\text{diffusion}}} + \mathcal{L}_{\text{prior}} \Bigg]$$

Our work uses properties of asking to obtain the following simplified form:

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{t, \mathbf{z}_t} \frac{\alpha_{t-1} - \alpha_t}{1 - \alpha_t} \log p_\theta(\mathbf{x}|\mathbf{z}_t)$$

Learning **p**: Simplifying the ELBO

The (negative) ELBO for masked diffusion has (in general) the following form:

$$\mathcal{L}_{\text{NELBO}} = \mathbb{E}_q \Big[\mathcal{L}_{\text{reconstr}} + \underbrace{\mathbb{E}_t \text{KL}(q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x}) || p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t))}_{\mathcal{L}_{\text{diffusion}}} + \mathcal{L}_{\text{prior}} \Big]$$

If we take the number of time steps **T** to infinity, we get the following bound:

$$\mathcal{L}_{\text{NELBO}} = -\mathbb{E}_q \int_{\alpha=0}^{\alpha=1} \frac{\log p_\theta(\mathbf{x}|\mathbf{z}_\alpha)}{1-\alpha} \, d\alpha$$

Learning **p**: Simplifying the ELBO

The (negative) ELBO for masked diffusion has (in general) the following form:

$$\mathcal{L}_{\text{NELBO}} = \mathbb{E}_q \Big[\mathcal{L}_{\text{reconstr}} + \underbrace{\mathbb{E}_t \text{KL}(q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x}) || p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t))}_{\mathcal{L}_{\text{diffusion}}} + \mathcal{L}_{\text{prior}} \Big]$$

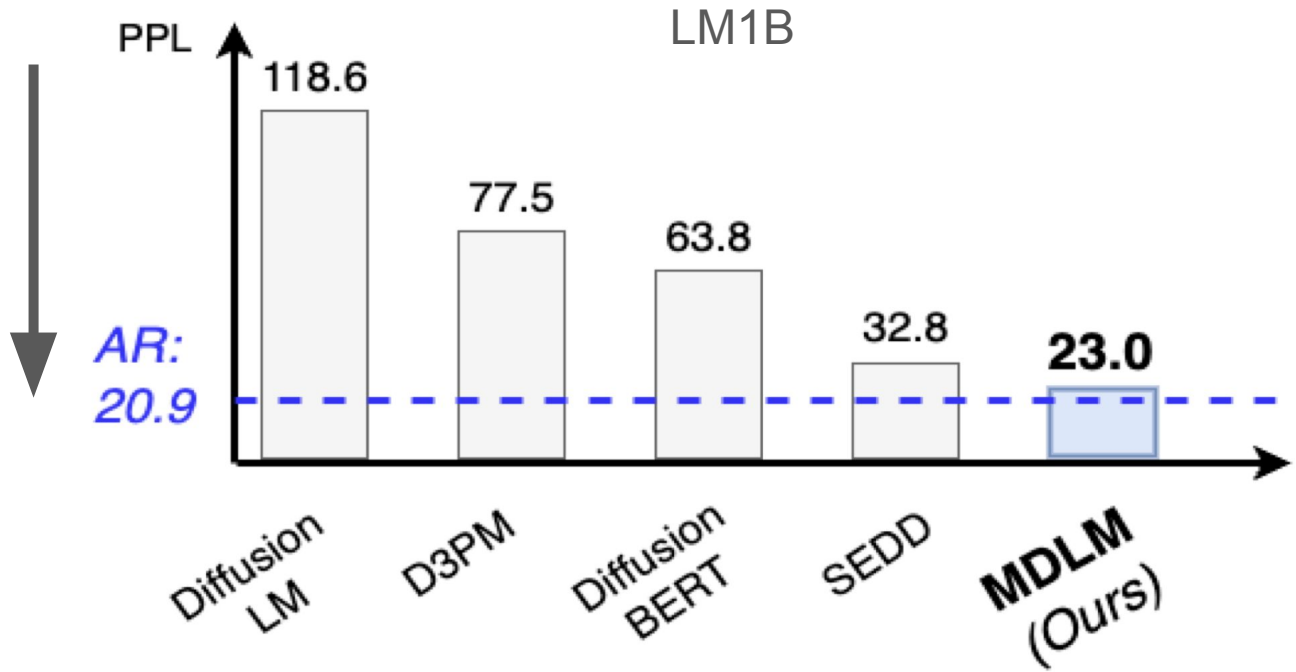
If we take the number of time steps **T** to infinity, we get the following bound:

$$\mathcal{L}_{\text{NELBO}} = -\mathbb{E}_q \int_{\alpha=0}^{\alpha=1} \frac{\log p_\theta(\mathbf{x}|\mathbf{z}_\alpha)}{1-\alpha} \, d\alpha$$

Observations

- The model is trained as a BERT model with a randomized masking rate
- However, the model admits principled sampling algorithms: generative BERT!
- This BERT defines probabilistic model *p* with an ELBO for evaluation

Experiments: Masked Diffusion Helps Close Gap To AR






Austin et al., "Structured Denoising Diffusion Models in Discrete State-Spaces." NeurIPS 2021
Li et al. "Diffusion-LM Improves Controllable Text Generation", NeurIPS 2022
Lou et al. "Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution", ICML 2024

Features of Discrete Diffusion

Parallel Generation

Autoregression

There

❌ Sequential

❌ Memory bound

❌ One token/step

Diffusion

Hirsch
account

✅ Parallel

✅ Compute bound

✅ Multiple tokens/step

Diffusion accelerates inference, producing multiple tokens/step.

Variable-Length Generation

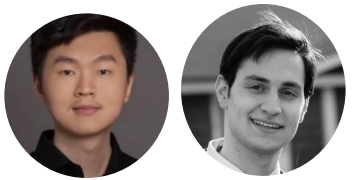
Autoregression	Block Diffusion	Diffusion
There are three possible categories of the typical rate of return for companies that share…	On	Hirsch will need to take account data that's to be released by tomorrow.
❌ Sequential	✅ Parallel	✅ Parallel
❌ Causal context only	✅ Semi-global context	✅ Global context
✅ Higher quality	✅ Higher quality	❌ Lower quality
✅ Arbitrary-length	✅ Arbitrary-length	❌ Fixed-length
✅ KV caching	✅ KV caching	❌ No KV caching

Arriola et al. "Block Diffusion: Interpolating Between Diffusion And Autoregressive Approaches", ICLR 2025 (Oral)

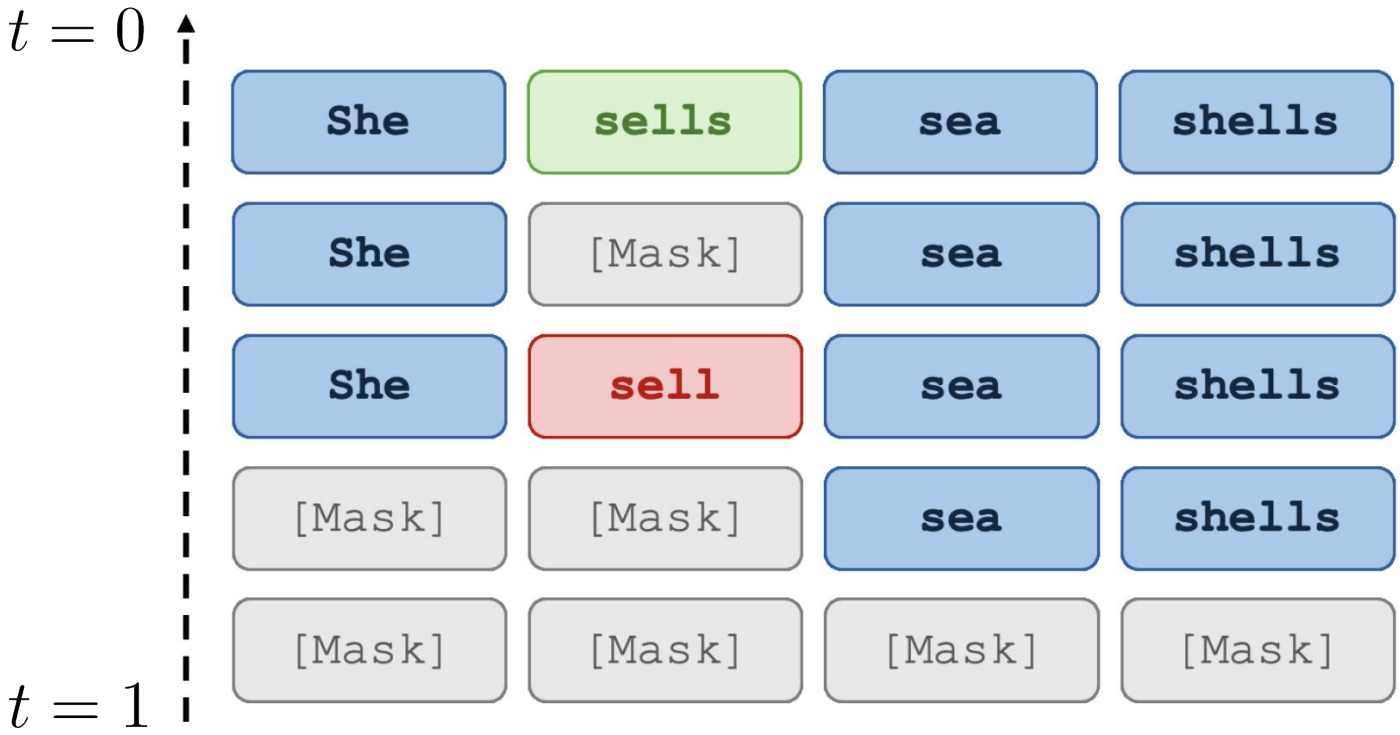
Iterative Refinement

✗ Autoregressive models: once a token is produced, it cannot be changed

✓ Diffusion models: can perform **iterative error correction**



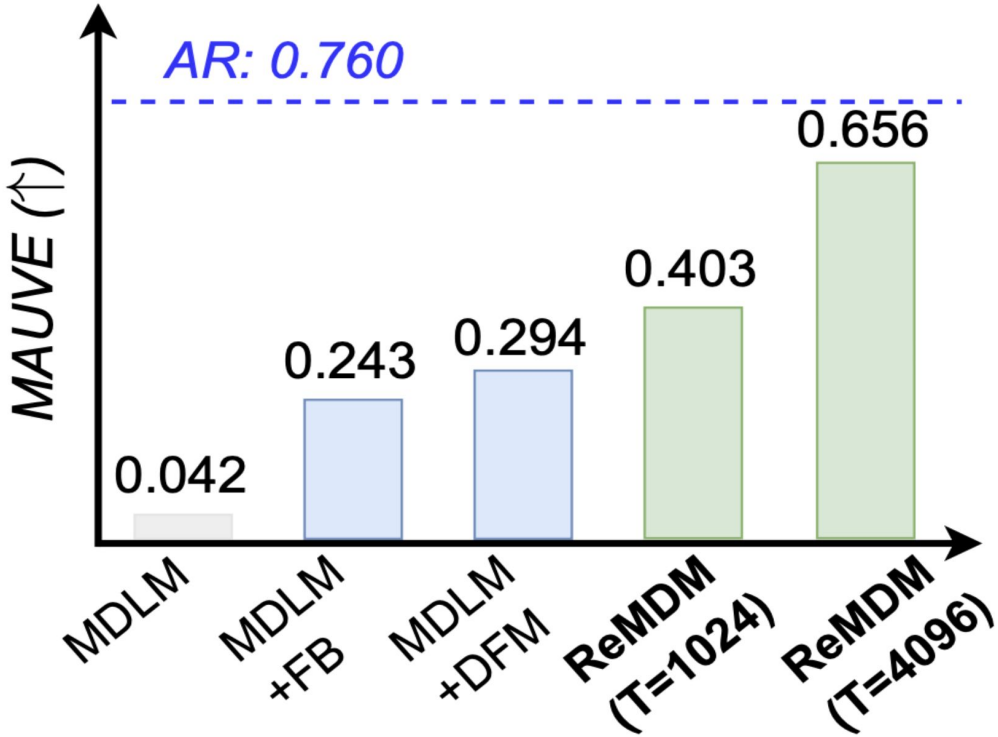
Iterative Refinement: ReMDM Sampler



Model produces "sell" because it's a valid completion, but later corrects itself

Wang, Schiff, et al. "Remasking Discrete Diffusion Models with Inference-Time Scaling", Neurips 2025


Inference-Time Scaling

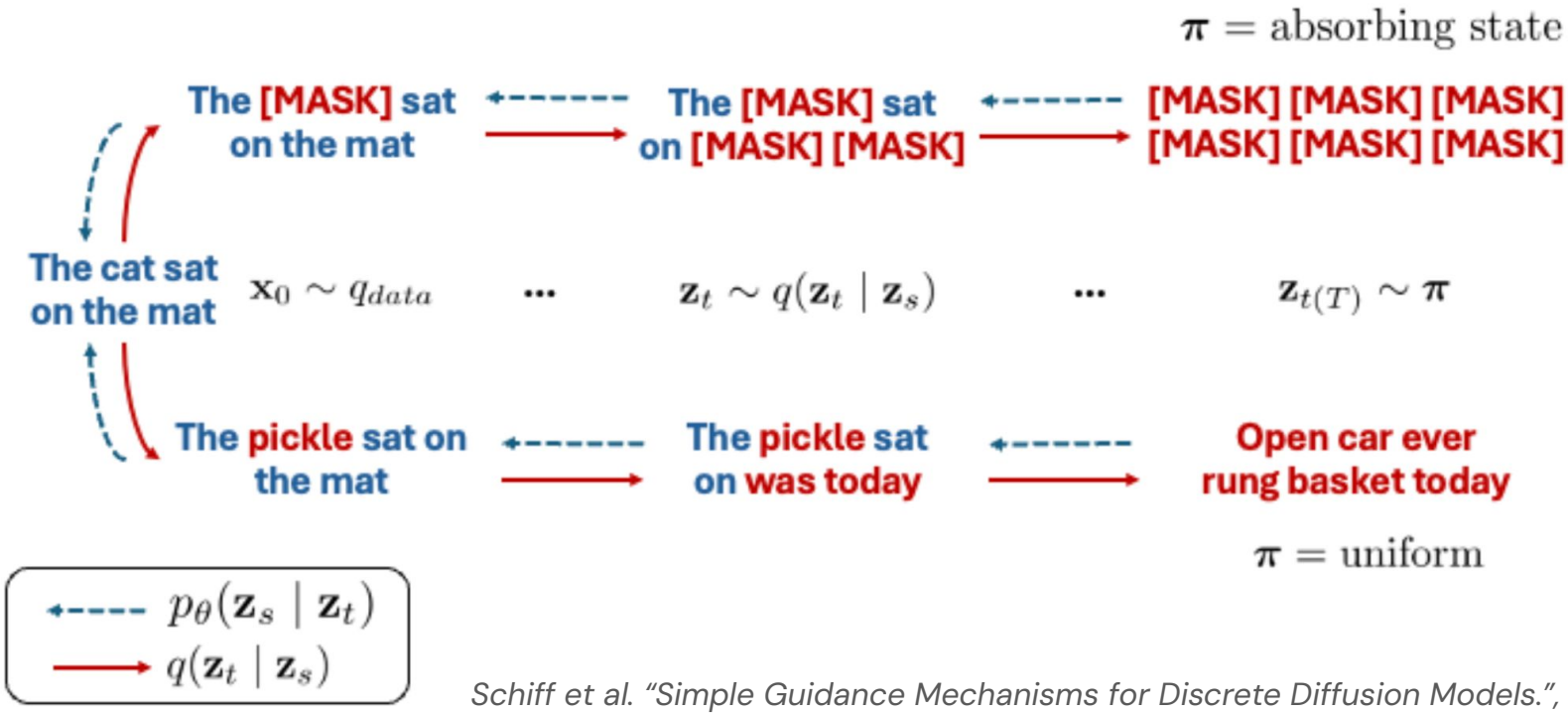


Diffusion modeling with inference-time scaling further closes the gap in sample quality to AR.

Wang, Schiff, et al. "Remasking Discrete Diffusion Models with Inference-Time Scaling", Neurips 2025

Uniform Noise Diffusion Language Models

An alternative to masked diffusion is to randomly swap words.



Schiff et al. "Simple Guidance Mechanisms for Discrete Diffusion Models.", ICLR 2025

Improved Controllable & Guided Generation

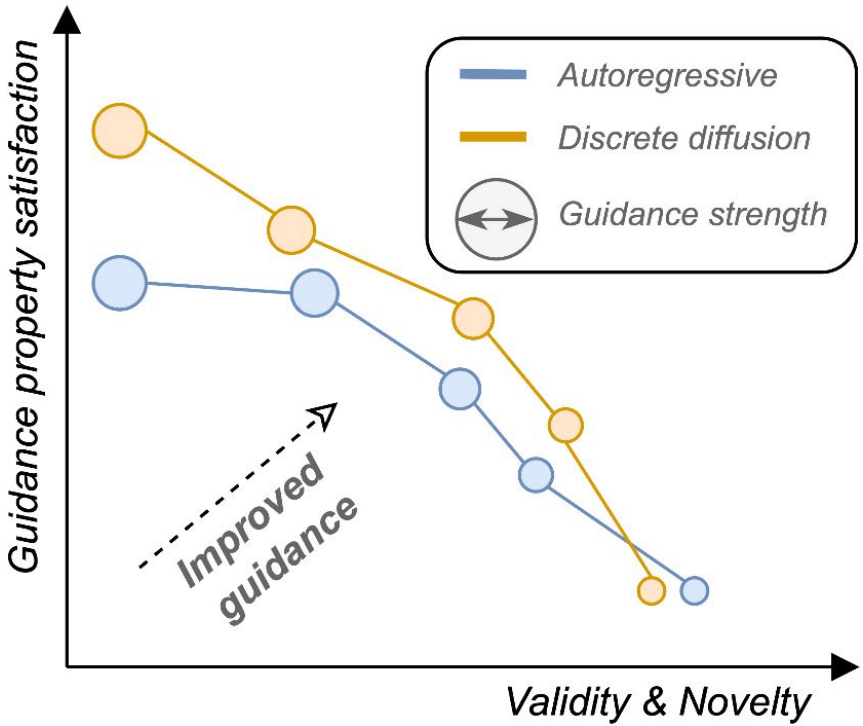
✗ Autoregressive models: make **“local”** predictions

✓ Diffusion models: make **“global”** (and **iterative)** refinements

An ink sketch style illustration of a small hedgehog holding a piece of watermelon with its tiny paws, taking little bites with its eyes closed in delight.

Figure credit: OpenAI

Improved Controllable & Guided Generation



Consider generating **x** to maximize property **y** (e.g,. binding affinity of drug).

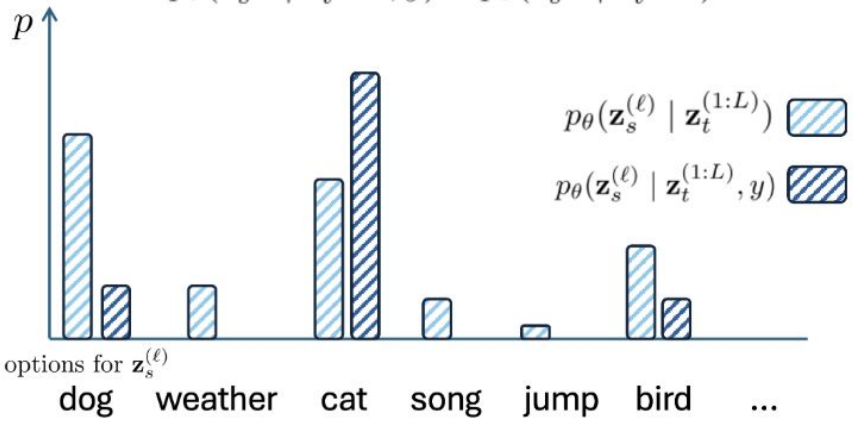
Diffusion yields better Pareto frontier between realistic **x** and high **y.**

Schiff et al. "Simple Guidance Mechanisms for Discrete Diffusion Models.", ICLR 2025

Suppose you need to update this word: The $\overbrace{\text{pickle}}^{\mathbf{z}_t^{(\ell)}}$ sat on the mat

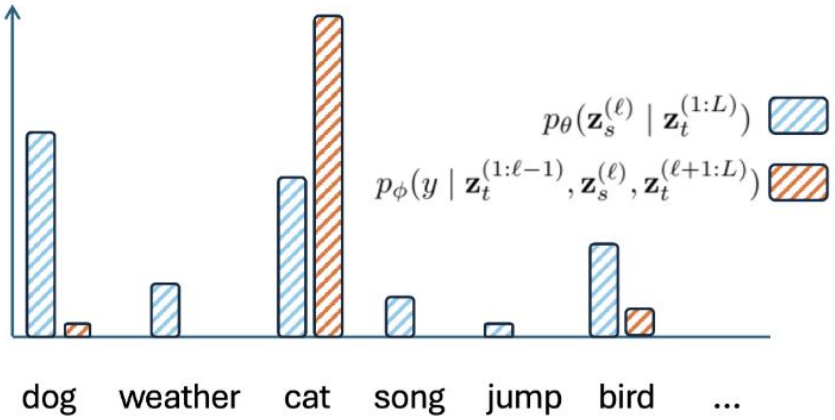
Discrete Classifier-Free Guidance

$$p_\theta^\gamma(\mathbf{z}_s^{(\ell)} \mid \mathbf{z}_t^{(1:L)}, y) \propto p_\theta(\mathbf{z}_s^{(\ell)} \mid \mathbf{z}_t^{(1:L)}, y)^\gamma \cdot p_\theta(\mathbf{z}_s^{(\ell)} \mid \mathbf{z}_t^{(1:L)})^{1-\gamma}$$



Discrete Classifier-Based Guidance

$$p_\theta^\gamma(\mathbf{z}_s^{(\ell)} \mid \mathbf{z}_t^{(1:L)}, y) \propto p_\phi(y \mid \mathbf{z}_t^{(1:\ell-1)}, \mathbf{z}_s^{(\ell)}, \mathbf{z}_t^{(\ell+1:L)})^\gamma \cdot p_\theta(\mathbf{z}_s^{(\ell)} \mid \mathbf{z}_t^{(1:L)})$$



Schiff et al. "Simple Guidance Mechanisms for Discrete Diffusion Models.", ICLR 2025

Classifier-Based and Classifier-Free Guidance

Suppose we have a classifier $p(y|\mathbf{z}_t)$ of property y from noisy $\mathbf{z}_t$.

$$\underbrace{\log p(\mathbf{z}_s|\mathbf{z}_t, y)}_{\text{conditional distribution}} = \underbrace{\log p(y|\mathbf{z}_t, \mathbf{z}_s)}_{\text{classifier}} + \underbrace{\log p(\mathbf{z}_s|\mathbf{z}_t)}_{\text{unconditional model}} + c$$

Classifier-Based and Classifier-Free Guidance

Suppose we have a classifier $p(y|\mathbf{z}_t)$ of property y from noisy $\mathbf{z}_t$.

$$\underbrace{\log p(\mathbf{z}_s|\mathbf{z}_t, y)}_{\text{conditional distribution}} = \underbrace{\log p(y|\mathbf{z}_t, \mathbf{z}_s)}_{\text{classifier}} + \underbrace{\log p(\mathbf{z}_s|\mathbf{z}_t)}_{\text{unconditional model}} + c$$

Discrete Classifier-Based Guidance (Schiff et al., 2024)

We can use this classifier to guide generation, and sample from

$$\underbrace{\log p^{(\gamma)}(\mathbf{z}_s|\mathbf{z}_t, y)}_{\text{new unnormalized distribution}} = \gamma \underbrace{\log p(y|\mathbf{z}_t, \mathbf{z}_s)}_{\text{guidance term}} + \underbrace{\log p(\mathbf{z}_s|\mathbf{z}_t)}_{\text{original diffusion model}}$$

where $\gamma > 0$ is a guidance strength parameter.

This extends to sequences if $p(\mathbf{z}_s|\mathbf{z}_t)$ is fully-factored across tokens.

Suppose we have a conditional model $p(\mathbf{z}_s|\mathbf{z}_t, y)$ and an unconditional $p(\mathbf{z}_s|\mathbf{z}_t)$. Recall the CBG factorization is:

$$\log p^{(\gamma)}(\mathbf{z}_s|\mathbf{z}_t, y) = \gamma \cdot \underbrace{\log p(y|\mathbf{z}_t, \mathbf{z}_s)}_{\text{apply Bayes rule}} + \log p(\mathbf{z}_s|\mathbf{z}_t) + c$$

By Bayes rule, $\log p(y|\mathbf{z}_t, \mathbf{z}_s) = \log p(\mathbf{z}_s|\mathbf{z}_t, y) - \log p(\mathbf{z}_s|\mathbf{z}_t) + c$.

Suppose we have a conditional model $p(\mathbf{z}_s|\mathbf{z}_t, y)$ and an unconditional $p(\mathbf{z}_s|\mathbf{z}_t)$. Recall the CBG factorization is:

$$\log p^{(\gamma)}(\mathbf{z}_s|\mathbf{z}_t, y) = \gamma \cdot \underbrace{\log p(y|\mathbf{z}_t, \mathbf{z}_s)}_{\text{apply Bayes rule}} + \log p(\mathbf{z}_s|\mathbf{z}_t) + c$$

By Bayes rule, $\log p(y|\mathbf{z}_t, \mathbf{z}_s) = \log p(\mathbf{z}_s|\mathbf{z}_t, y) - \log p(\mathbf{z}_s|\mathbf{z}_t) + c$.

Discrete Classifier-Free Guidance (Schiff et al., 2024)

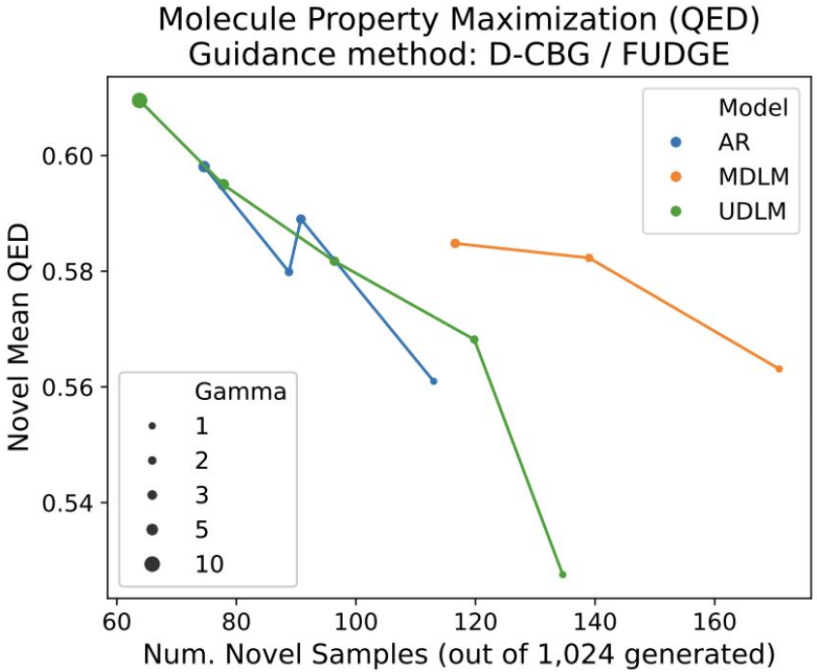
We can use this classifier to guide generation, and sample from

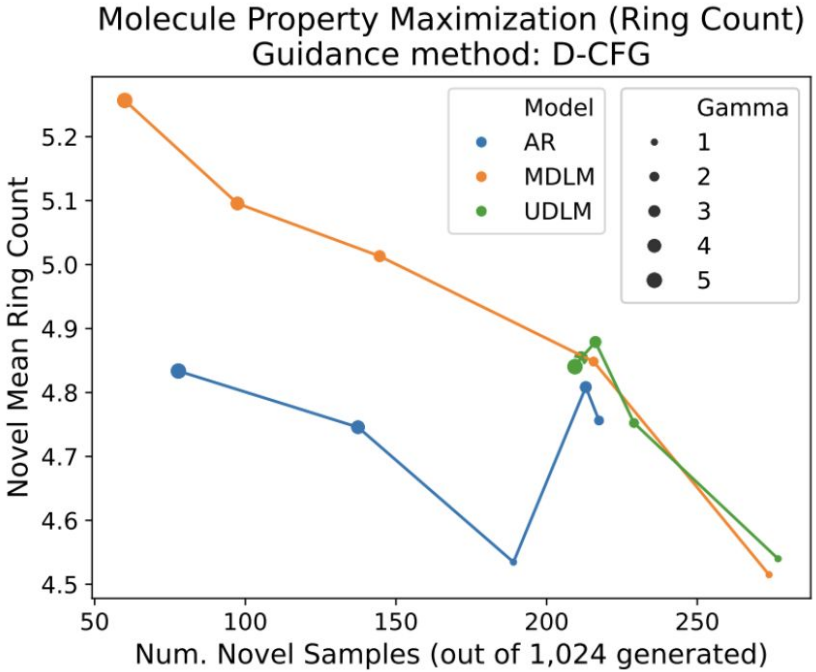
$$\underbrace{\log p^{(\gamma)}(\mathbf{z}_s|\mathbf{z}_t, y)}_{\text{new unnormalized distribution}} = \gamma \underbrace{\log p(\mathbf{z}_s|\mathbf{z}_t, y)}_{\text{conditional model}} + (1-\gamma) \underbrace{\log p(\mathbf{z}_s|\mathbf{z}_t)}_{\text{unconditional model}}$$

where $\gamma > 0$ is a guidance strength parameter.

In practice, we parameterize $p(\mathbf{z}_s|\mathbf{z}_t, y), p(\mathbf{z}_s|\mathbf{z}_t)$ with same model.

Improved Controllable & Guided Generation

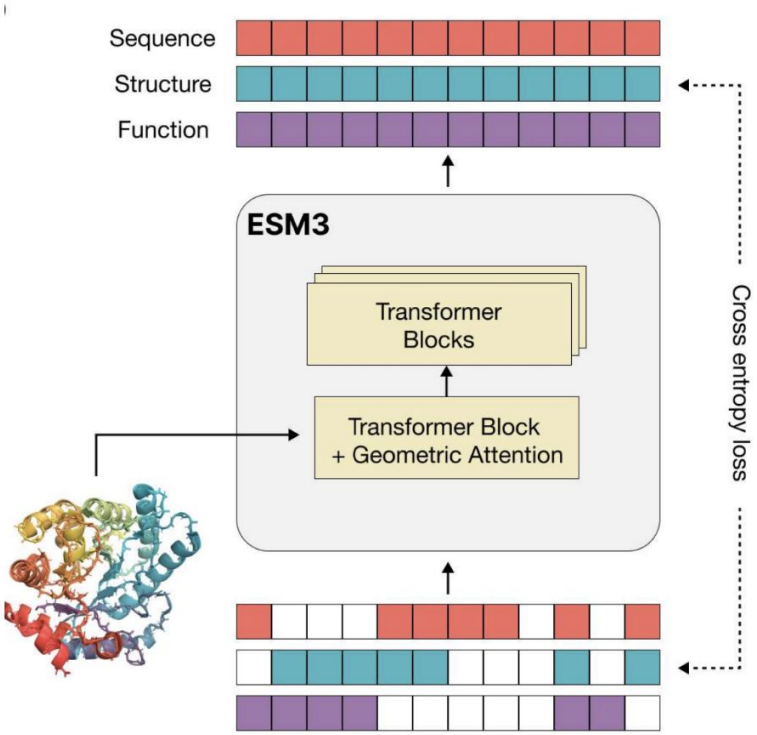




Schiff et al. "Simple Guidance Mechanisms for Discrete Diffusion Models.", ICLR 2025

Large Diffusion Language Models

Protein Language Models are Discrete Diffusion Models

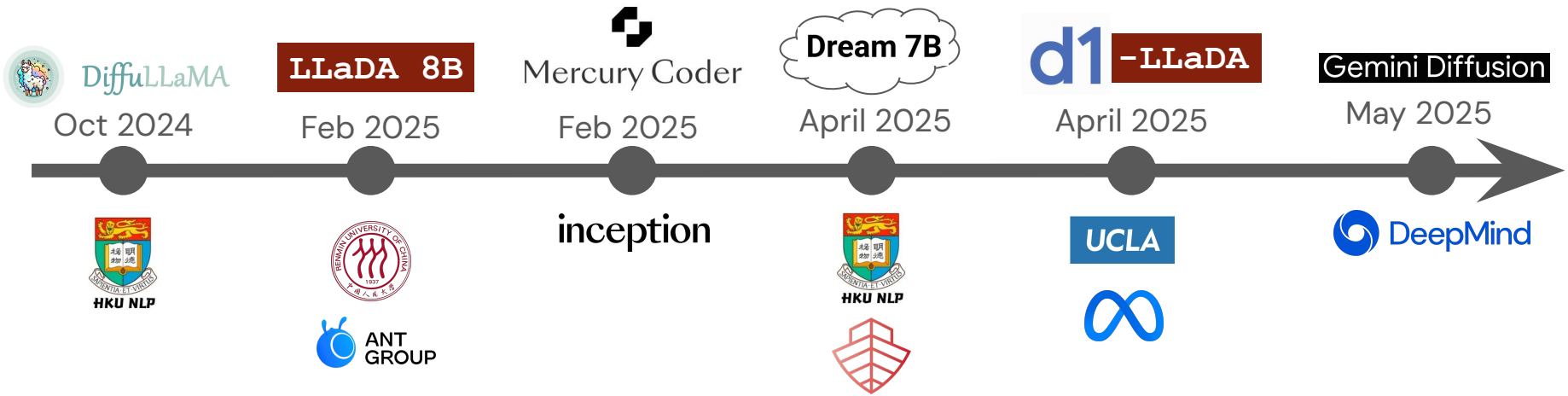


Recent ESM3 100B protein language models use masked diffusion for training and sampling.

Protein LMs can be used to accelerating folding

Hayes et al. "Simulating 500 million years of evolution with a language model", Science 2025

Large Diffusion Language Models



Diffusion LLMs Exhibit Favorable Scaling Laws Relative to AR

- Both uniform and masked diffusion LLMs exhibit similar scaling behavior to AR LLMs
- However, the optimal model sizes of dLLMs are larger

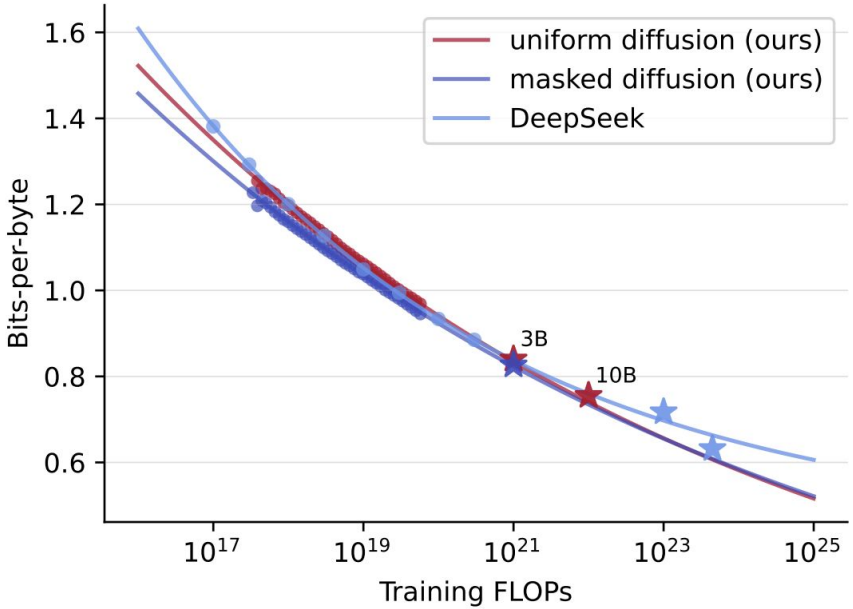
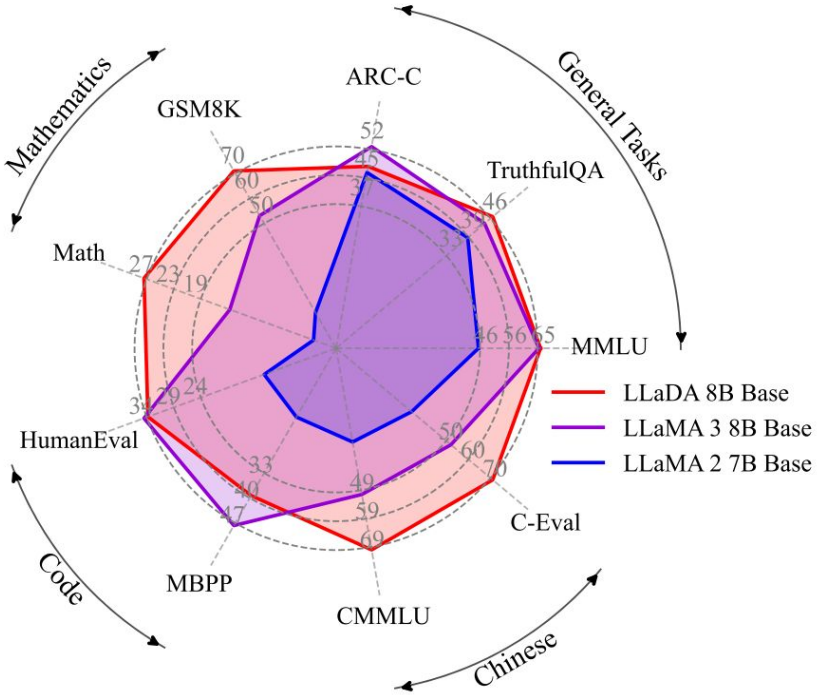
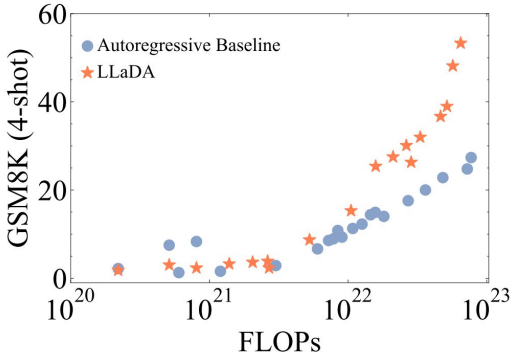


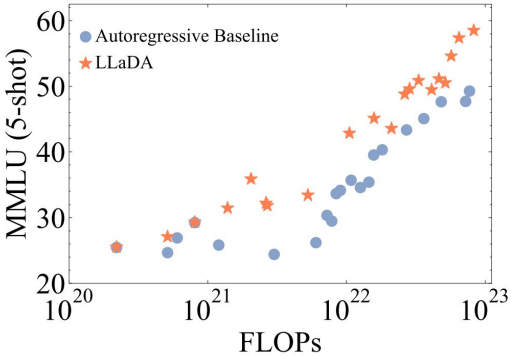
Figure 1: Our proposed scaling laws extrapolate well to 3B and 10B models trained on up to 50× larger compute budgets and suggest that DLMs can be competitive with ALMs at scale, even in compute-bound training settings.

von Rutte et al. "Scaling Behavior of Discrete Diffusion Language Models." arXiV preprint 2026

LLaDA: An 8B Open-Weights Diffusion LLM

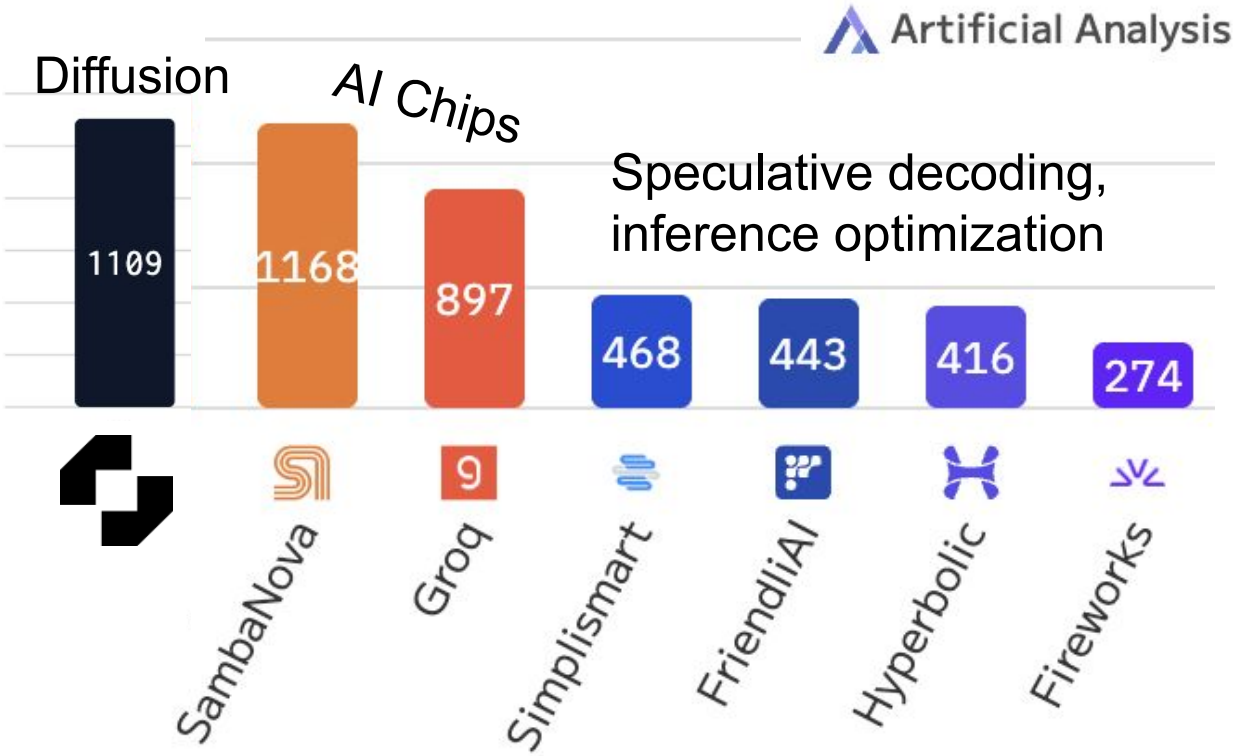






Nie et al. "Large Diffusion Language Models." arXiV preprint 2025.

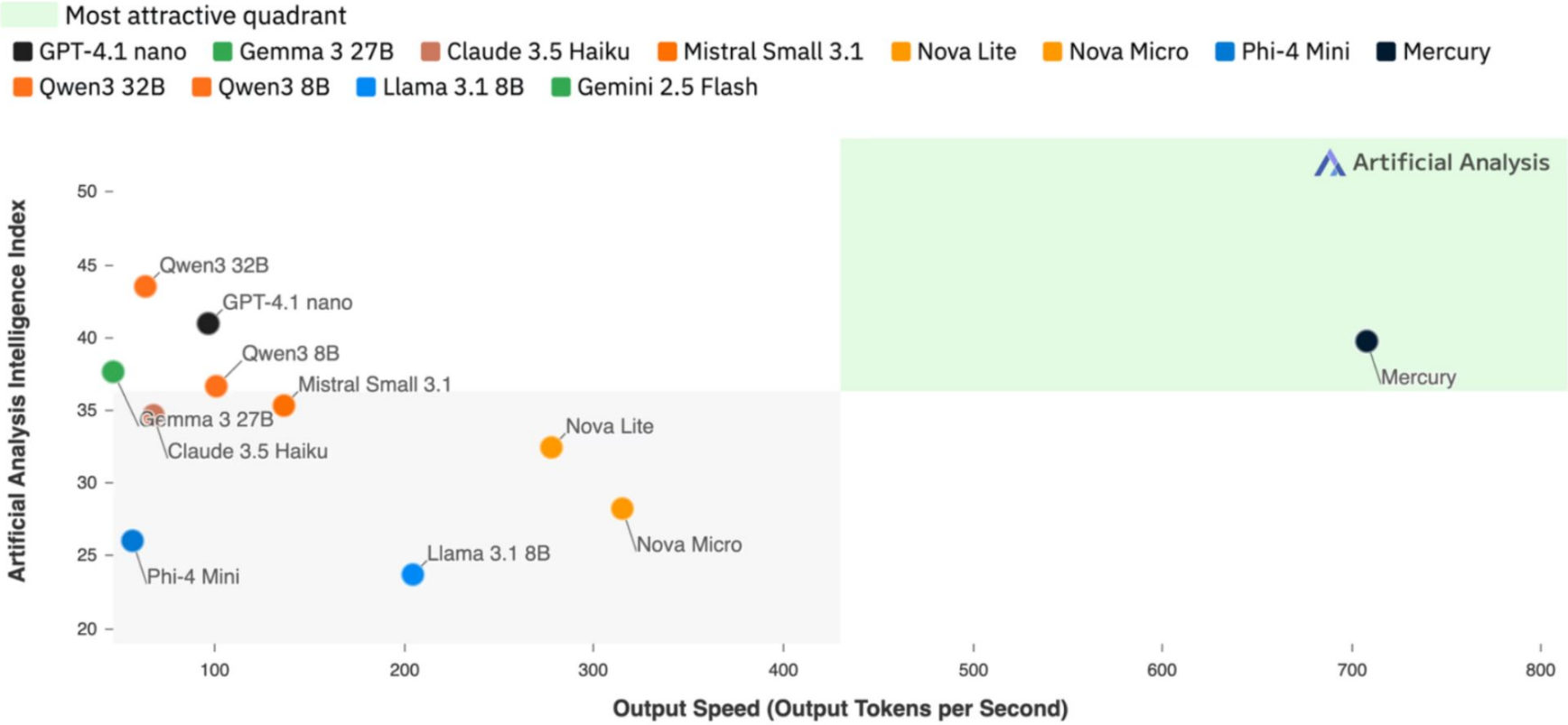

Mercury Coder
Iterations
0
AUTOREGRESSIVE LLM
LEFT-TO-RIGHT GENERATION
Iterations
0
INCEPTION DIFFUSION LLM
COARSE-TO-FINE GENERATION



Throughput of Llama 8b vs. Mercury Mini (higher quality) in tokens/second across inference providers

Intelligence vs. Output Speed: Small, Non-reasoning models

Artificial Analysis Intelligence Index; Output Speed: Output Tokens per Second



Conclusion

- Masked diffusion is emerging as an alternative to autoregressive models.
- Advantages of diffusion include:
  - **Fast parallel generation**: much greater tok/sec at inference time
  - **Built-in error correction**: improves quality via iterative refinement with bidirectional context
  - **Controllability**: easier to guide model because of error correction
  - **New form of inference-time scaling**: can denoise more or less at inference time
  - **Multimodality**: one paradigm for all modalities
- Large-scale diffusion language models are already here!

Simple Diffusion Language Models