ETH zürich

Beyond Autoregression: Discrete Diffusion for Complex Reasoning and Planning

Diego Arapovic
Supervisor: Andreas Plesner
Seminar in Deep Neural Networks
06. May 2025, Zürich

Beyond Autoregression:

Input sequence: “97, 38, 3, 17, 14”

Target sequence: “97 - 38 = 59, 59 – 17 = 42, 42/3 = 14”

Claim: Modelling choice matters!

Modelling Choices for Sequences

Task: model a sequence $\mathbf{x} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ from the data distribution $q(\mathbf{x})$

Autoregressive Modelling

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}_1) \prod_{n=2}^{N} p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_n \mid \boldsymbol{x}_{1:n-1}\right)$$

Discrete Diffusion Modelling

Forward Process

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$$

Backward Process

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{\mathbf{x}_{1:T}} p(\mathbf{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$$

Modelling Choices for Sequences

Task: model a sequence $\mathbf{x} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ from the data distribution $q(\mathbf{x})$

Autoregressive Modelling

$$p_\theta\big(\boldsymbol{x}_n \mid \text{“}97\ -\ 38\ =\ 59,\ 59\ -\ \text{”}\big)$$

Discrete Diffusion Modelling

Forward Process

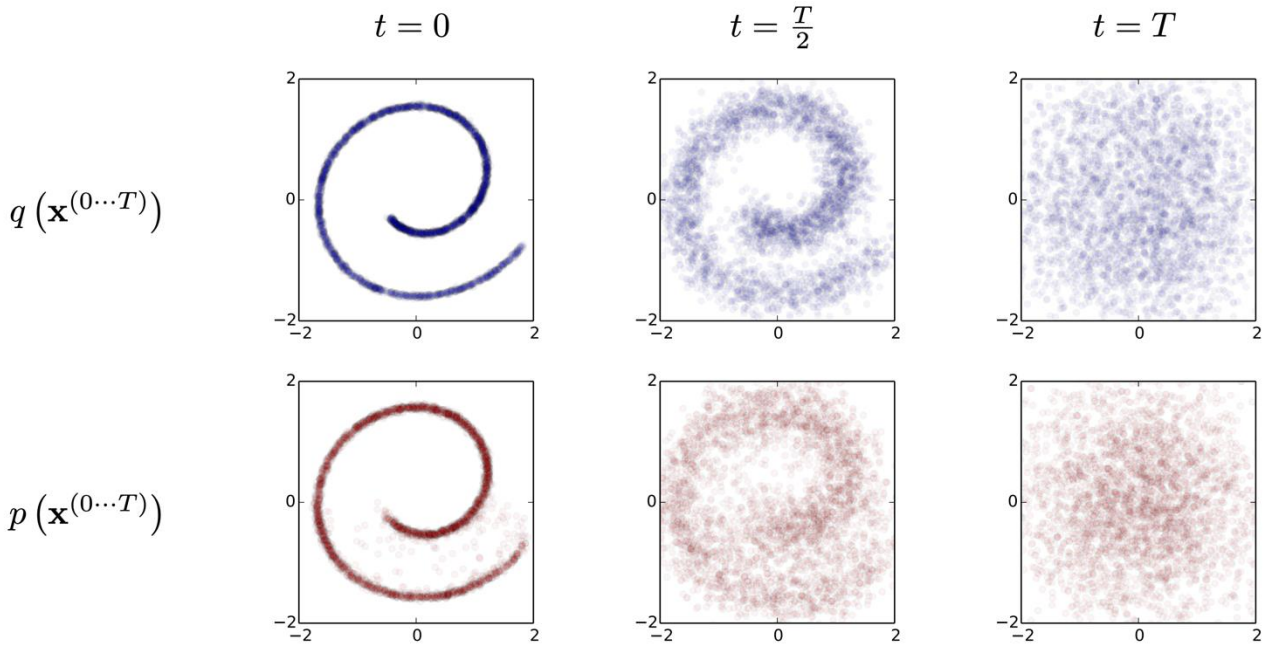
$$x_0: \quad \text{“}9\ 7 - 38 = 59, 5\ 9 - 17 = 42,\ 4\ 2/3 = 1\ 4''$$
$$x_t: \quad \text{“}m7 - 38 = 59, m9\, m17\, m\, 42mm2/3 = 1m''$$

Backward Process

$$x_T: \quad \text{“}mmmmmmmmmmmmmmmmmmmmmmmmmmmmmm''$$
$$x_t: \quad \text{“}m7 - 38 = 59, m9\, m17\, m\, 42mm2/3 = 1m'''$$
$$x_0: \quad \text{“}9\ 7 - 38 = 59, 5\ 9 - 17 = 42,\ 4\ 2/3 = 1\ 4''$$

Very Short History of Diffusion Modelling

Sohl-Dickstein et al. 2015



Ho et al. 2020

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at https://github.com/hojonathanho/diffusion.

1 Introduction

Deep generative models of all kinds have recently exhibited high quality samples in a wide variety of data modalities. Generative adversarial networks (GANs), autoregressive models, flows, and variational autoencoders (VAEs) have synthesized striking image and audio samples [14, 27, 3, 58, 38, 25, 10, 32, 44, 57, 26, 33, 45], and there have been remarkable advances in energy-based modeling and score matching that have produced images comparable to those of GANs [11, 55].

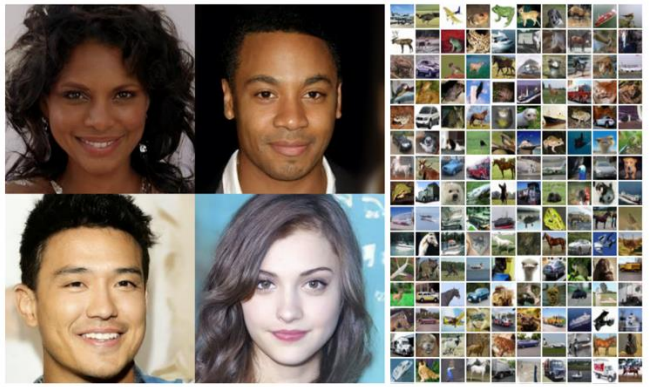
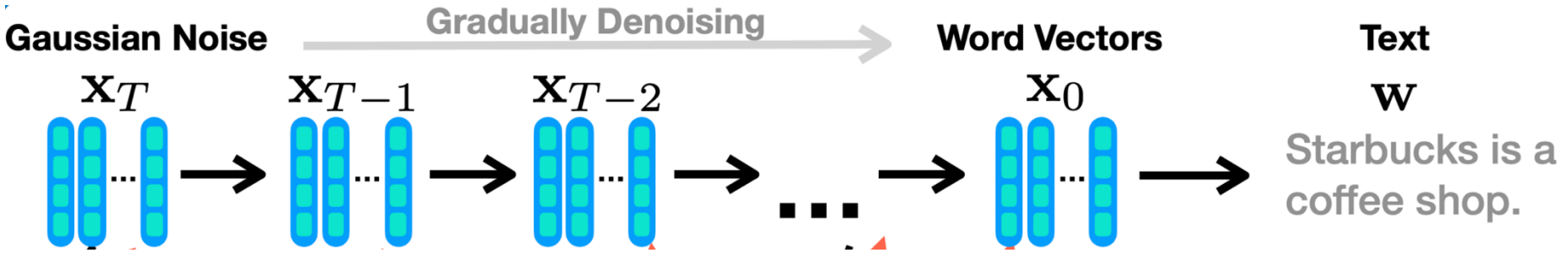


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

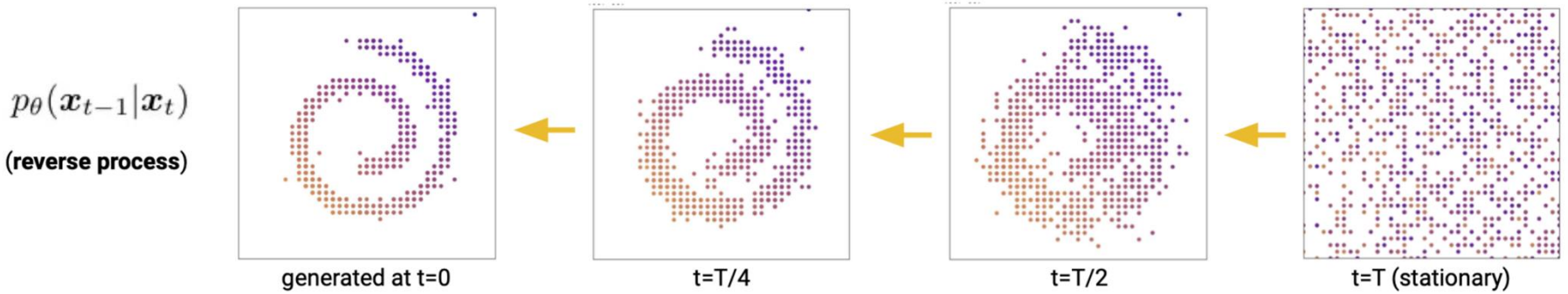
34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Momentum in Diffusion Modelling for Discrete Problems

2022: Diffusion LM, Li et al.



2021:
Hoogeboom et al.
Austin et al.

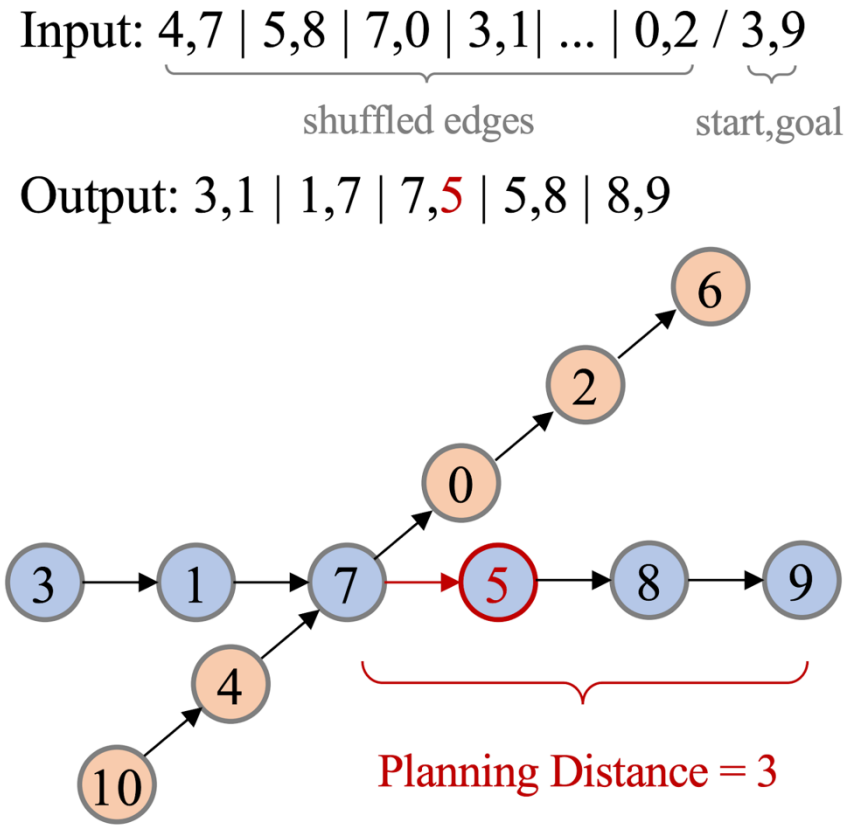


	7		3	1			5	
	1	5		8				
9			5				8	1
5		3	6			2		
2			1		9			4
		7			8	9		6
9	6				5			7
				6				
	2			9				

Establish Superiority of DDM vs AR

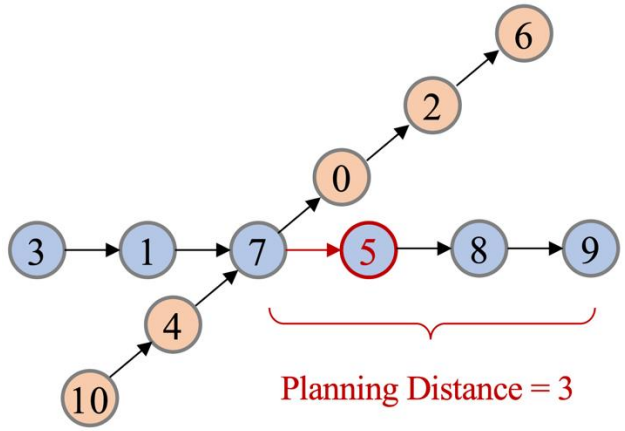
- Synthetic Graph Task
  - Input: shuffled edges of a graph, start and goal node
  - Goal: output edges from start to goal node

- Difficulty: Choosing the next node at node 7
  - Model has to look 3 nodes ahead
  - «Planning Distance»

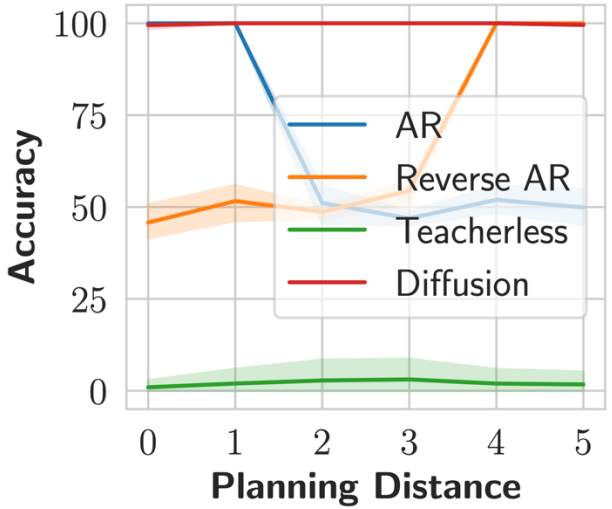


Establish Superiority of DDM vs AR

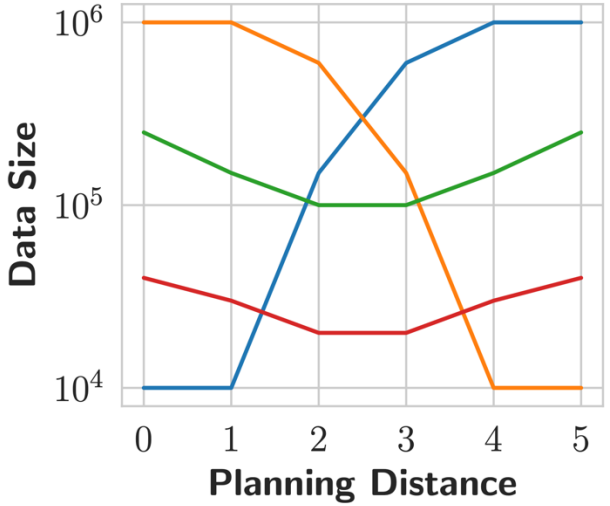
- Experimental Setup:
  - 10 nodes
  - 1 crossing at random location
  - symmetric around the crossing



50k Samples
Accuracy per PD



Data for
90% accuracy per PD



- Scaling the models:
  - 3-Layer transformer, 6B params
  - LLaMA 7B

🔑 Take-away:

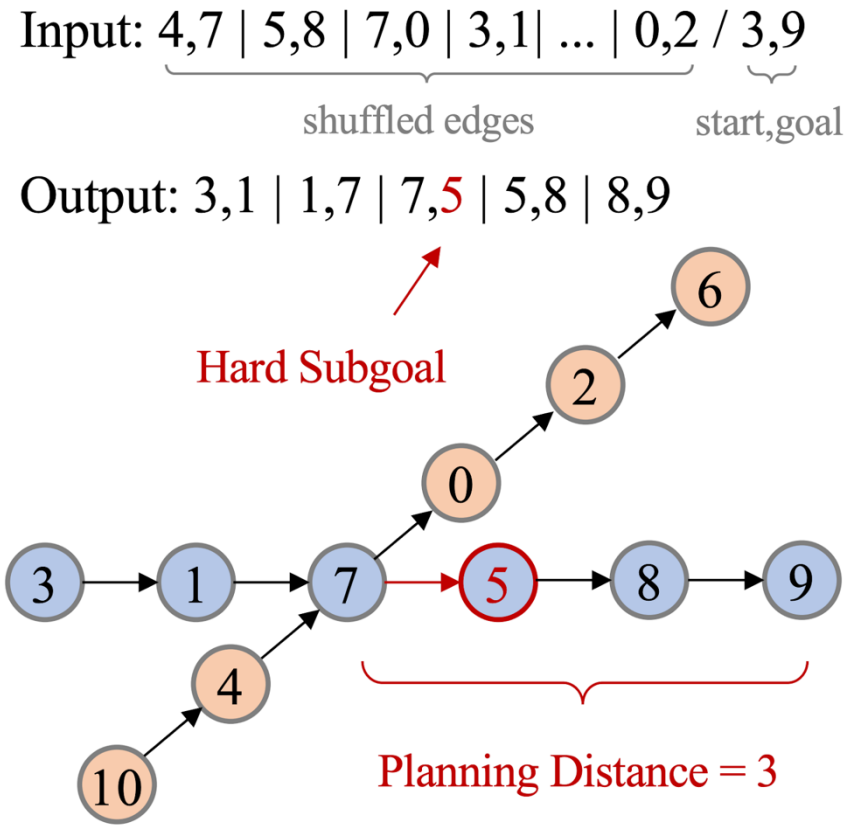
Tokens vary in difficulty to predict!

→ «Subgoal Imbalance»

Establish Superiority of DDM vs AR

- Subgoal: Task of predicting a certain token n.
- Subgoal Imbalance
- Proposition:

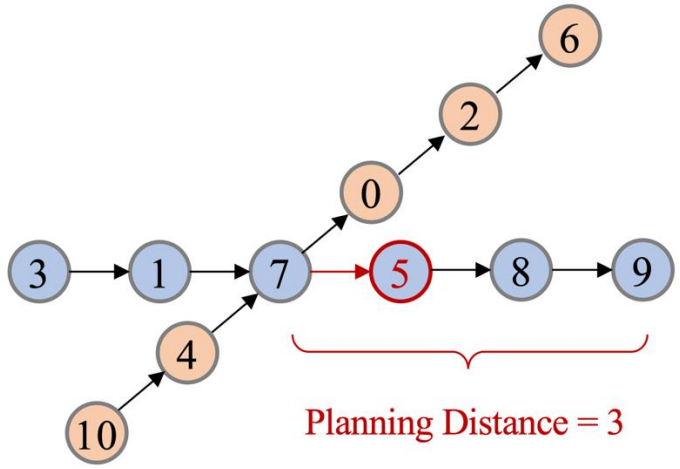
  → Difficulty of learning a certain subgoal depends on the underlaying model.



Token-level loss as common ground

- Full input:

$$\underbrace{4,7|5,8|7,0|...|0,2}_{\text{shuffled edges}} / \underbrace{3,9}_{\text{start,goal}} \text{[SEP]} \underbrace{3,1|1,7|7,5|5,8|8,9}_{\text{target output}}$$



$$\mathcal{L}_{\text{AR}} = \sum_{n=1}^{N} -\log p_{\text{AR}}\left(\boldsymbol{x}_n \mid \boldsymbol{x}_{1:n-1}\right)$$

$$\underbrace{4,7|5,8|7,0|...|0,2/3,9\text{[SEP]}3,1|1,7|7,}_{x_{1:n-1}} x_n \underbrace{|5,8|8,9}_{\text{ignored}}$$

$$\mathcal{L}_{\text{DM}} = \sum_{n=1}^{N} -\log p_{\text{DM}}\left(\boldsymbol{x}_n \mid \boldsymbol{x}_{\neq n}\right)$$

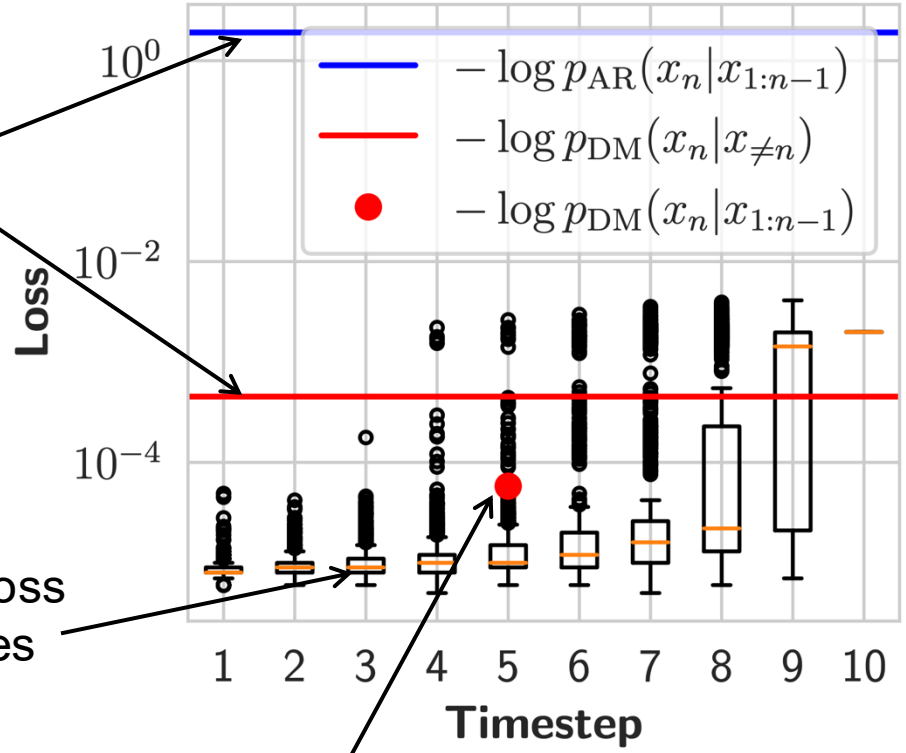
$$4,7|5,8|7,0|...|0,2/3,9\text{[SEP]}[\text{M}],1|1[\text{M}][\text{M}]|7,5[\text{M}]x_n,8|[\text{M}],9$$

Checking for Subgoal Imbalance

$$\mathcal{L}_{\text{DM}} = \sum_{n=1}^{N} -\log p_{\text{DM}}(\boldsymbol{x}_n \mid \boldsymbol{x}_{\neq n})$$

$$= \sum_{n=1}^{N} \sum_{t=1}^{T} w(t) \cdot \underbrace{u(\mathbf{x}_0, \mathbf{x}_t, n; \boldsymbol{\theta})}_{\substack{\text{cross entropy loss} \\ \text{on token } n \text{ at noise } t}}$$

Negative Log-Likelihood
for fixed PD = 3



Cross entropy loss
Over 1K samples

NLL with only
AR style context

Take-aways:

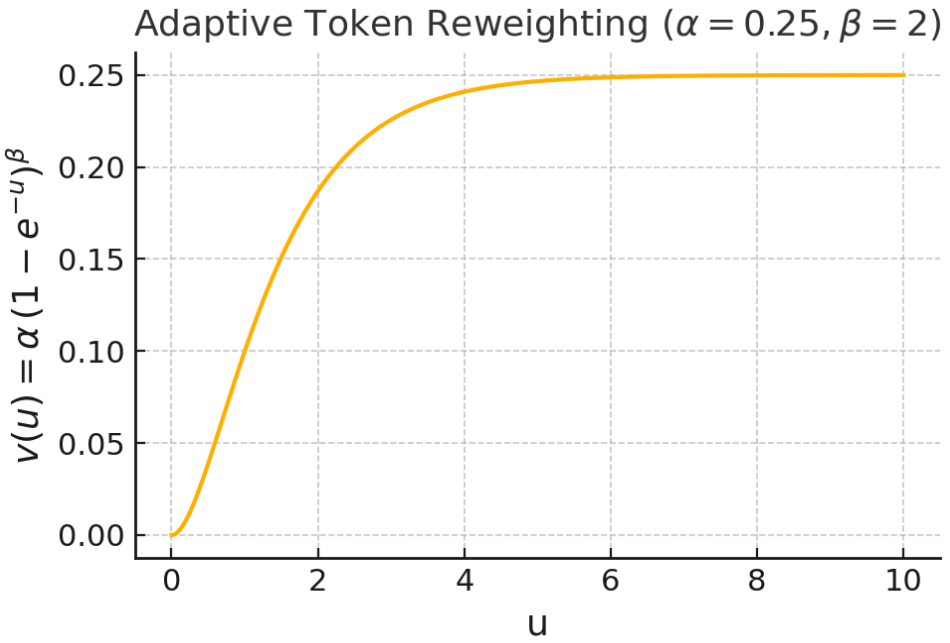
→ Modelling choice impacts difficulty
→ Many views help the model learn (Sudoku intuition)
→ Especially interesting for hard subgoals

Multi-Granularity Diffusion Modelling (MGDM)

$$\mathcal{L}_{\mathrm{DM}} = \sum_{n=1}^{N} \sum_{t=1}^{T} \underbrace{w(t)}_{\text{weighs } \mathbf{x}_t} \; u(\mathbf{x}_0, \mathbf{x}_t, n; \boldsymbol{\theta})$$

$$\mathcal{L}_{\mathrm{MGDM}} = \sum_{n=1}^{N} \sum_{t=1}^{T} w(t) \; \underbrace{v(\mathbf{x}_t, n)}_{\text{token difficulty}} \; u(\mathbf{x}_0, \mathbf{x}_t, n; \boldsymbol{\theta})$$

$$v\left(\mathbf{x}_{t,n}\right) = \alpha \left(1 - \exp\left(-u(\cdot)\right)\right)^{\beta}$$



Experiments - Countdown

"97, 38, 3, 17, 14"

"97 - 38 = 59, 59 – 17 = 42, 42/3 = 14"

Training from scratch

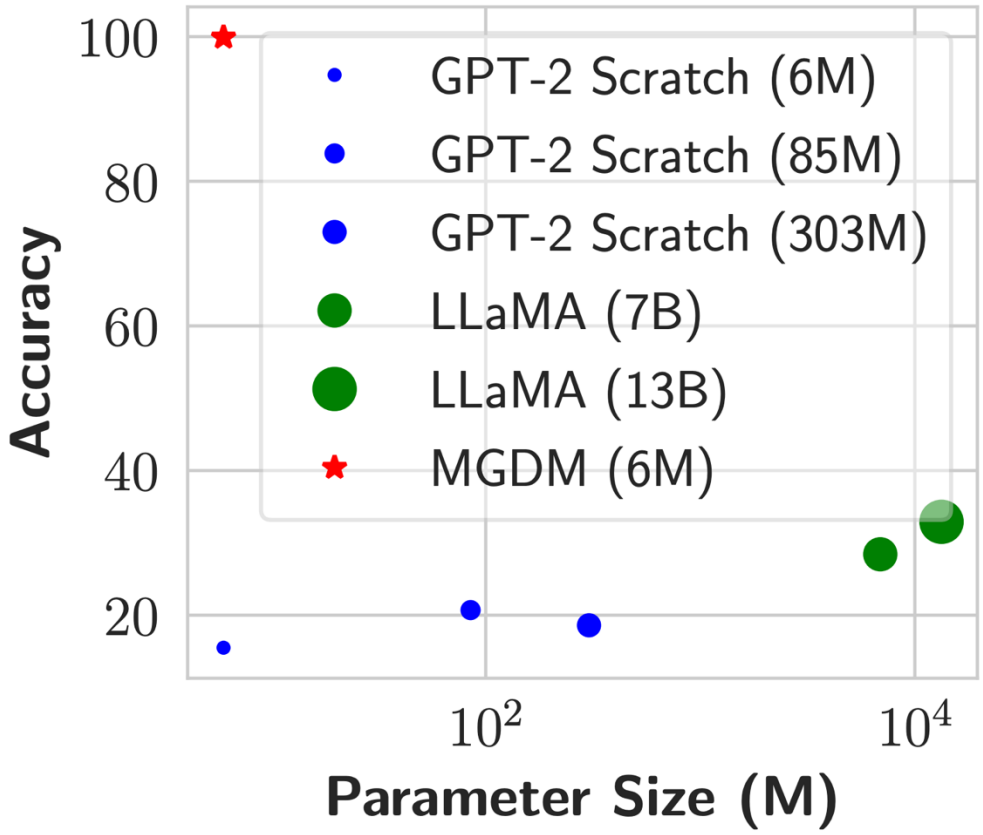
	Params	CD 3	CD 4	CD 5
Autoregressive				
GPT-2 Scratch	6M	94.1	31.9	4.3
	85M	95.9	45.8	5.1
	303M	96.4	41.3	4.5
Stream-of-Search	250M	-	54.2	-
LLaMA	7B	95.7	41.1	6.7
	13B	96.5	51.1	7.4
Diffusion				
VDM	85M	99.1	73.4	16.3
D3PM	85M	99.4	83.1	27.6
RDM	85M	99.5	87.0	45.8
MGDM (Ours)	6M	98.1	52.0	27.0
	85M	99.5	**91.5**	**46.6**
	303M	**99.9**	88.3	39.0

In-Context Learning

	Acc.	# Token
Prompting		
GPT-4 IO	7.3	x28
GPT-4 CoT	4.0	x61
GPT-4 CoT-SC	9.0	x241
GPT-4 ToT	74.0	x186
Supervised training		
GPT-2 Scratch	18.8	**x1**
MGDM	**76.0**	**x1**

85M

Experiments - Sudoku



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

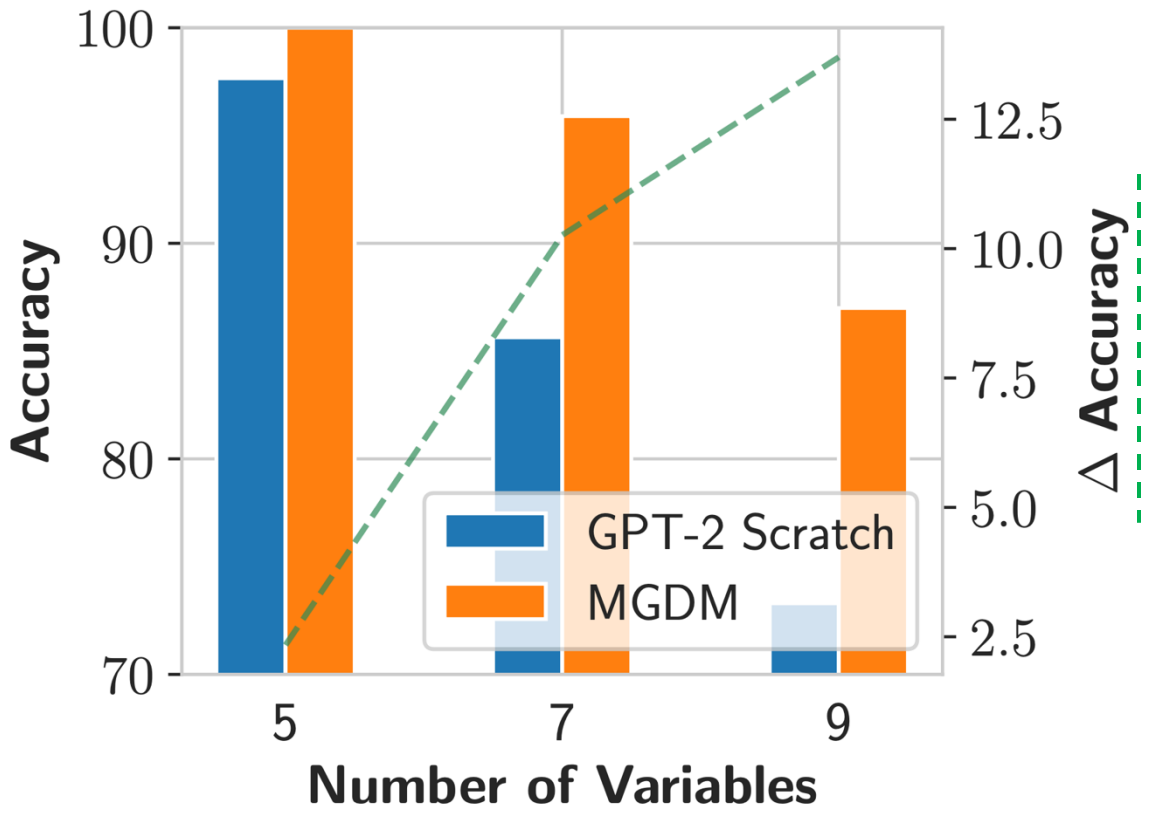
Limitations:

- Only easy Sudokus
- Experiment scope very small

Experiments - Boolean Satisfiability Problem (SAT)

$$(x_1 \vee \neg x_2) \ \wedge \ (\neg x_1 \vee x_2 \vee x_3) \ \wedge \ \neg x_1$$

$$x_1 = 0, \ x_2 = 0, \ x_3 = 1$$



Ablation Studies

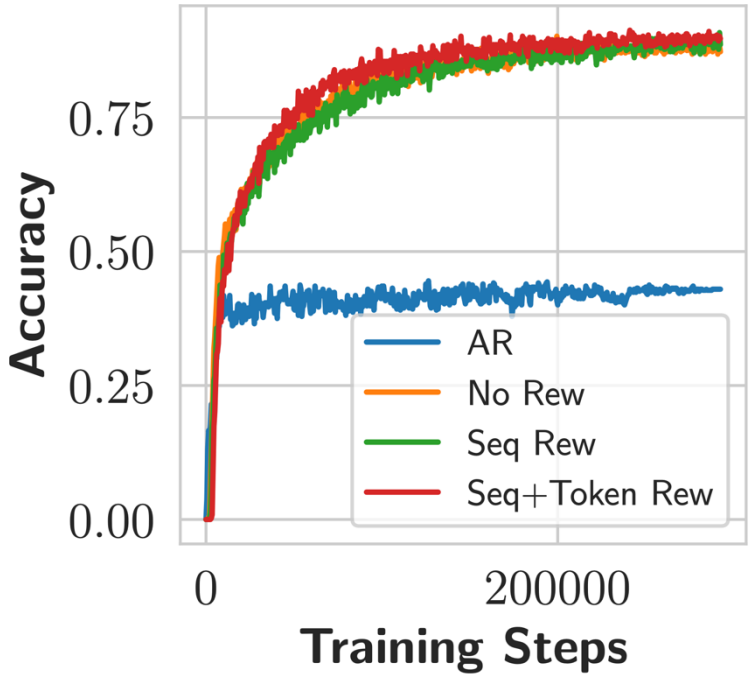
"97, 38, 3, 17, 14"

"97 - 38 = 59, 59 – 17 = 42, 42/3 = 14"

$$\mathcal{L}_{\text{MGDM}} = \sum_{n=1}^{N} \sum_{t=1}^{T} \underbrace{w(t)}_{\text{seq.-reweighting}} \underbrace{v(\mathbf{x}_t, n)}_{\text{token-reweighting}} u(\mathbf{x}_0, \mathbf{x}_t, n; \boldsymbol{\theta})$$

$$v(\mathbf{x}_{t,n}) = \alpha \left(1 - \exp(-u(\cdot))\right)^{\beta}$$

	Random	TopK
No reweighting	82.1	87.3
Original sequence-reweighting	83.1	88.5
+ token-reweighting (α=0.25, β=1)	84.9	**90.4**
+ token-reweighting (α=1, β=1)	82.4	89.3
+ token-reweighting (α=0.25, β=2)	82.4	87.9
Linear sequence-reweighting	79.6	87.0
+ token-reweighting (α=0.25, β=1)	83.2	88.0
+ token-reweighting (α=1, β=1)	86.7	90.4
+ token-reweighting (α=0.25, β=2)	85.6	**91.5**

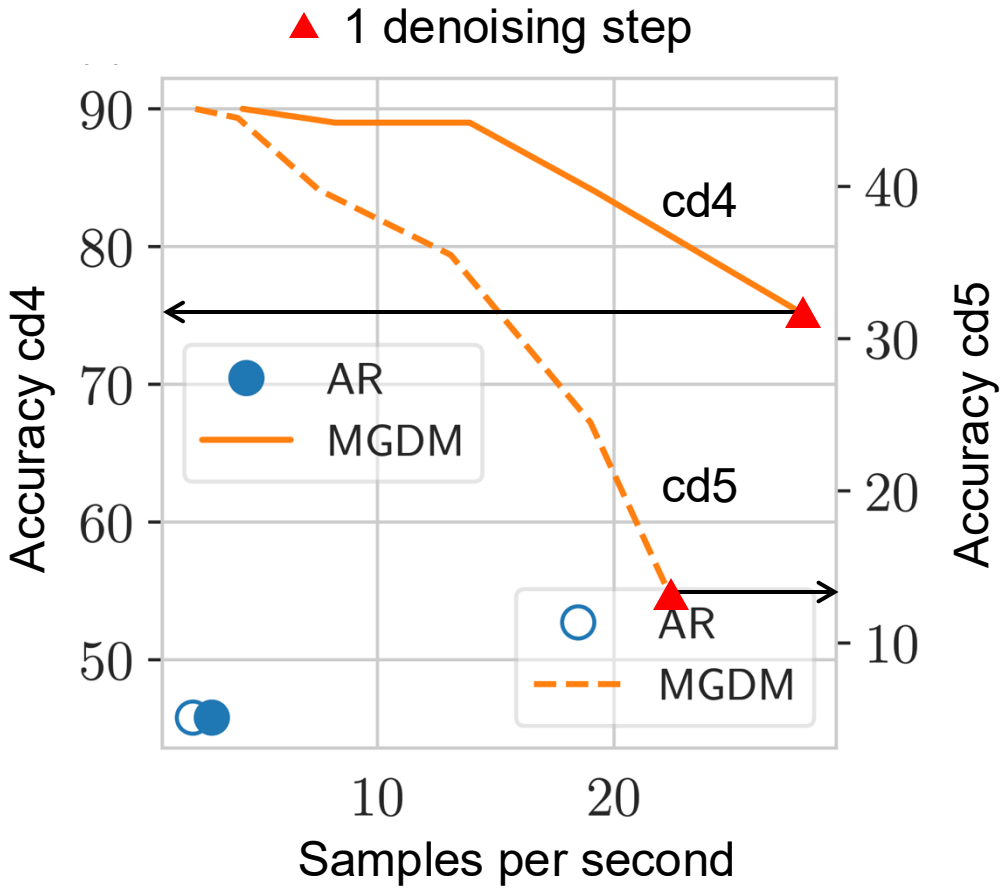


Model Speed at Inference

"97, 38, 3, 17, 14"

"97 - 38 = 59, 59 – 17 = 42, 42/3 = 14"

- Models
  - AR: GPT-2 Scratch 85M
  - DM: MGDM 85M
- Metric:
  - Samples processed per second
- Diffusion Model can adapt #denoising steps



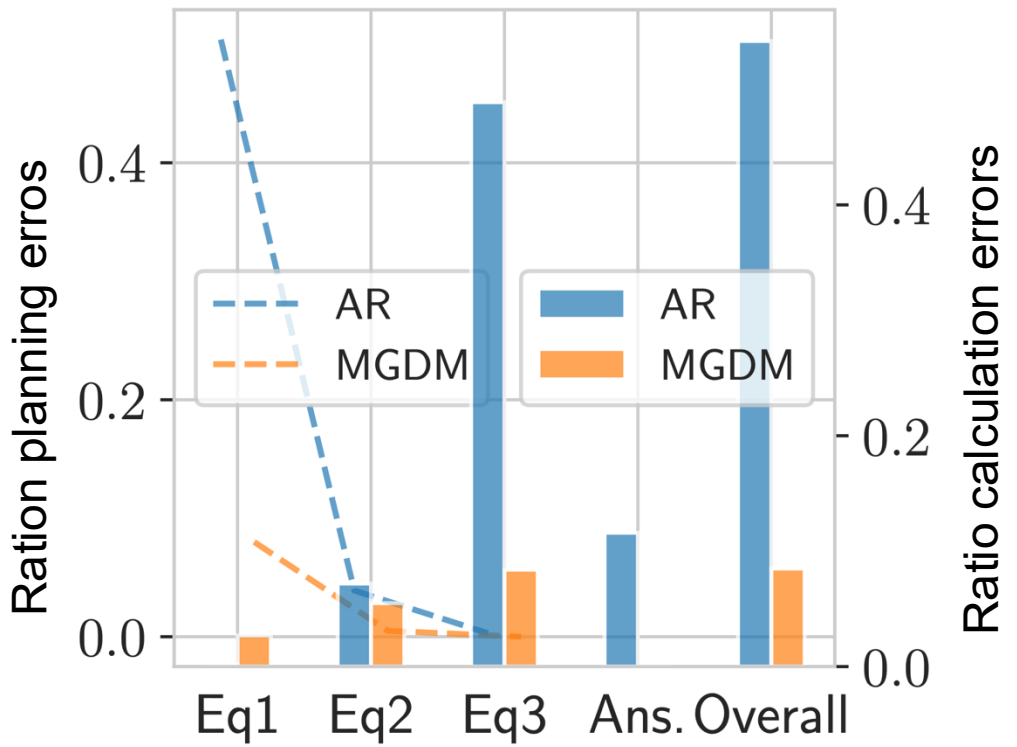
Error Analysis

"97, 38, 3, 17, 14"

"97 - 38 = 59, 59 – 17 = 42, 42/3 = 14"

- Models
  - AR: GPT-2 Scratch 85M
  - DM: MGDM 85M

- Metrics:
  - Ratio of calculation errors: $a \ \mathrm{OP} \ b \neq c$
  - Ratio of planning errors:
    → Equation at which model fails the task



Beyond Autoregression - Contributions

- Show DDM outperform AR in reasoning and planning
  - Highlight ARs limitations

- Explain why DM are stronger on reasoning through the lense of Subgoal Imbalance

- Propose Multi-Granular Diffusion Modelling (MGDM)
  - Better loss and accuracy
  - Faster convergence
  - Theoretically motivated

- Paper has clear scope

Beyond Autoregression - Limitations

- Experiments
  - Plots not well introduced, axes not 0 aligned
  - Limited task diversity
  - Non-satisfactory complexity of experiments (Sudoku, Countdown
  - Metrics not properly introduced
  - Testes only on short outputs
  - Limited ablation studies
  - Teacherless and reverse AR not tested on «real» tasks
  - No generalization experiments, less/more complex instances

Beyond Autoregression - Limitations

- Missing quantification of training cost
- Theoretically motivated but no further analysis
  - Convergence guarantees
- Lacking interpretability compared to LLMs
- Connection to trendy LLMs missing
  - No mention how this extends to general language generation

MGDM is a promising proof-of-concept

but practicality, robustness, and applicability remain to be explored

Thank you for your attention!

Backup: Algorithms

Algorithm 1 Training MGDM

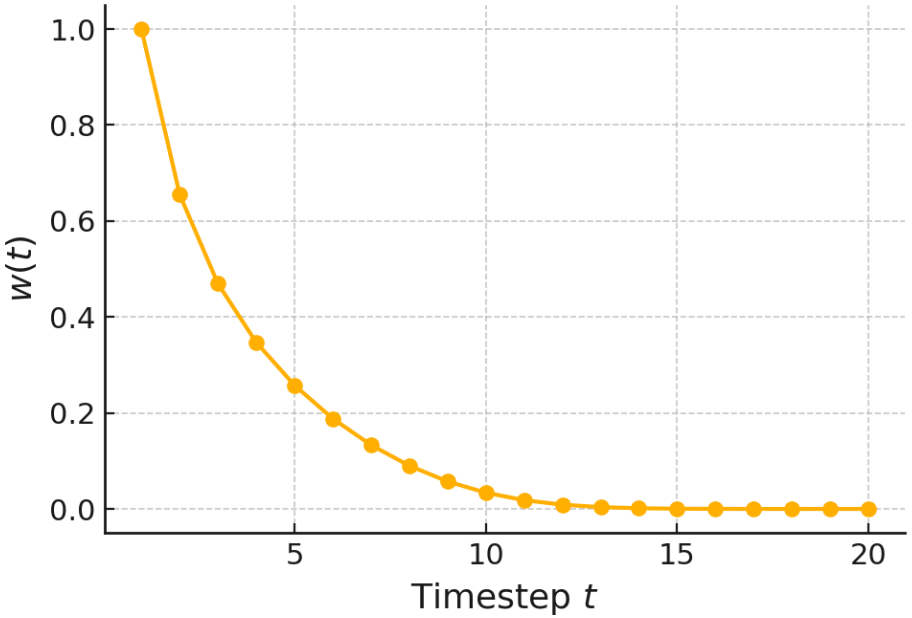
Input: neural network $f(\cdot;\boldsymbol{\theta})$, data distribution $p_{\text{data}}(\mathbf{x}_{0,1:N})$, a custom sequence reweighting term $w(t)$, token reweighting parameters α and γ, timesteps T.
Output: model parameters $\boldsymbol{\theta}$.
repeat
  Draw $\mathbf{x}_{0,1:N} \sim p_{\text{data}}(\mathbf{x}_{0,1:N})$;
  Draw $t \in \text{Uniform}(\{1,\dots,T\})$;
  Draw $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)$;
  for $n = 1, 2, \dots, N$ **do**
    Let $u(\mathbf{x}_0, \mathbf{x}_t, n; \boldsymbol{\theta}) \coloneqq \mathbf{1}_{\mathbf{x}_{t,n} \neq \mathbf{x}_{0,n}} \mathbf{x}_{0,n}^\top \log f(\mathbf{x}_t;\boldsymbol{\theta})_n$;
    Let $v(\mathbf{x}_{t,n}) = \alpha(1 - \exp u(\mathbf{x}_0, \mathbf{x}_t, n; \boldsymbol{\theta}))^\gamma$;
  end for
  $L_{\boldsymbol{\theta}} = -w(t) \sum_{n=1}^{N} v(\mathbf{x}_{t,n}) u(\mathbf{x}_0, \mathbf{x}_t, n; \boldsymbol{\theta})$;
  Minimize $L_{\boldsymbol{\theta}}$ with respect to $\boldsymbol{\theta}$;
until converged

Algorithm 2 Sampling from MGDM

Input: trained network $f(\cdot;\boldsymbol{\theta})$, mask token id $\boldsymbol{m}$, timesteps T, temperature τ.
Output: generated sample $\mathbf{x}_0$.
for $n = 1, 2, \dots, N$ **do**
  Initialize $\mathbf{x}_{T,n} = \boldsymbol{m}$;
end for
for $t = T, \dots, 1$ **do**
  Define indicator $\mathbf{e}_t = \text{TopK}\left(f(\mathbf{x}_t;\boldsymbol{\theta})\right)$ with indices in top-t/T values set to 1 and others 0;
  for $n = 1, 2, \dots, N$ **do**
    Draw $\widetilde{\mathbf{x}}_{0,n} \sim \text{Categorical}\left(f(\mathbf{x}_t;\boldsymbol{\theta})/\tau\right)$;
    $\mathbf{x}_{t-1,n} = \mathbf{e}_{t,n}\widetilde{\mathbf{x}}_{0,n} + (1 - \mathbf{e}_{t,n})\boldsymbol{m}$;
  end for
end for
Return $\mathbf{x}_{0,1:N}$.

Backup: Maths

$$w(t) = \frac{\alpha_{t-1} - \alpha_t}{1 - \alpha_t} \in (0, 1]$$



alpha_t: forward mask survival probability
→ defines the noise schedule

w(t): probability of a token to become masked at timestep t
→ Each token counted once in expectation over all noise steps