ВЫСШАЯ ШКОЛА ЭКОНОМИКИ

p(B|A)yesgroup

Diffusion Models: Effective Language Modeling

Chimbulatov Egor

Center of Deep Learning and Bayesian Methods, HSE University

Why use language diffusion?

Current dominant method is **autoregression**: generating text consecutively, token by token.

The next token is ___

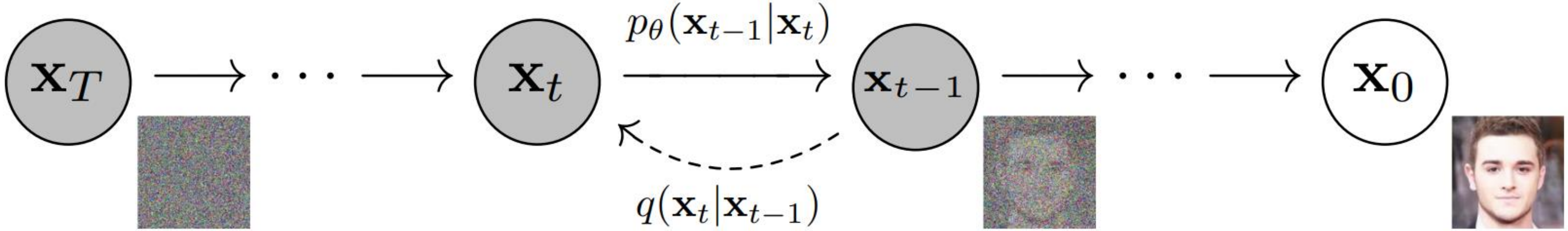
Most LLMs are built with this approach. However, it has several drawbacks

Why use language diffusion?

- Inability to correct previous mistakes
- The model does not think “ahead”
- Linear generation complexity

Diffusion Models

Given some forward noising process, learn NN to approximate the reverse process optimizing **ELBO**



Diffusion Models

- Forward process

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

- Reverse process (parameterized by the NN)

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

- Negative ELBO

$$\mathbb{E}_q\Bigg[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Bigg]$$

How to model text with diffusion?

Text is obviously discrete, while diffusion models are designed to inject noise in data. How do we inject noise in texts? There are **two approaches**

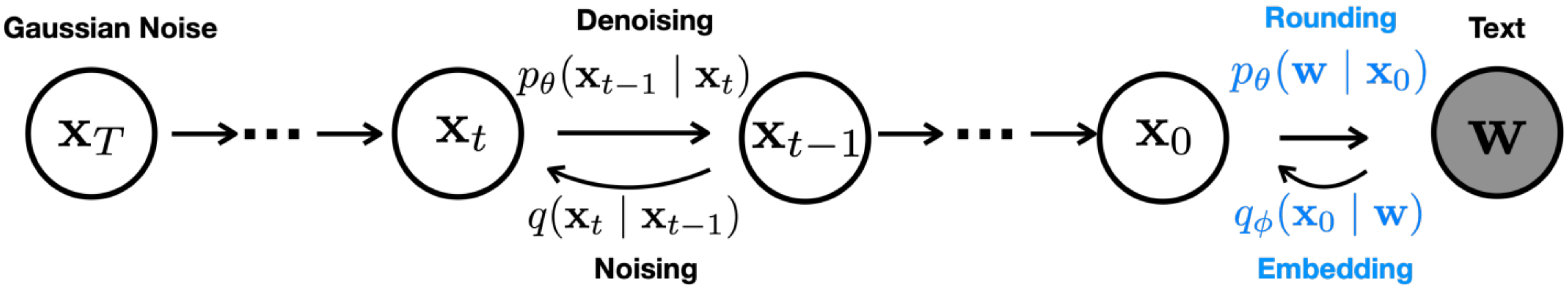
- **Continuous**: map text into continuous latent space
- **Discrete**: noising by changing tokens

Continuous Diffusion

Diffusion-LM

Straightforward approach was proposed in Diffusion-LM:

- Map tokens to **embeddings**
- Diffusion on embeddings
- **End-to-end** training



Diffusion-LM

This method had some serious drawbacks:

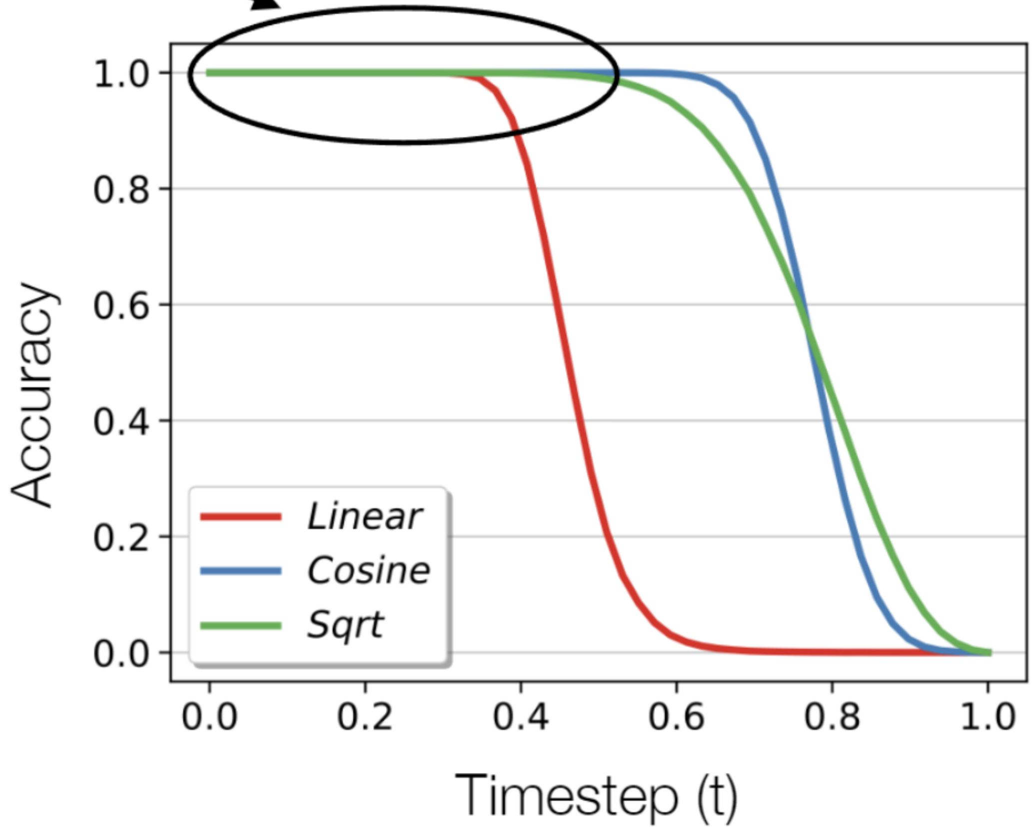
- Converges **slowly**
- Requires **a lot of steps** to produce higher quality samples
- Prone to **mode collapse**

However, this work explored some **important findings**

Diffusion-LM

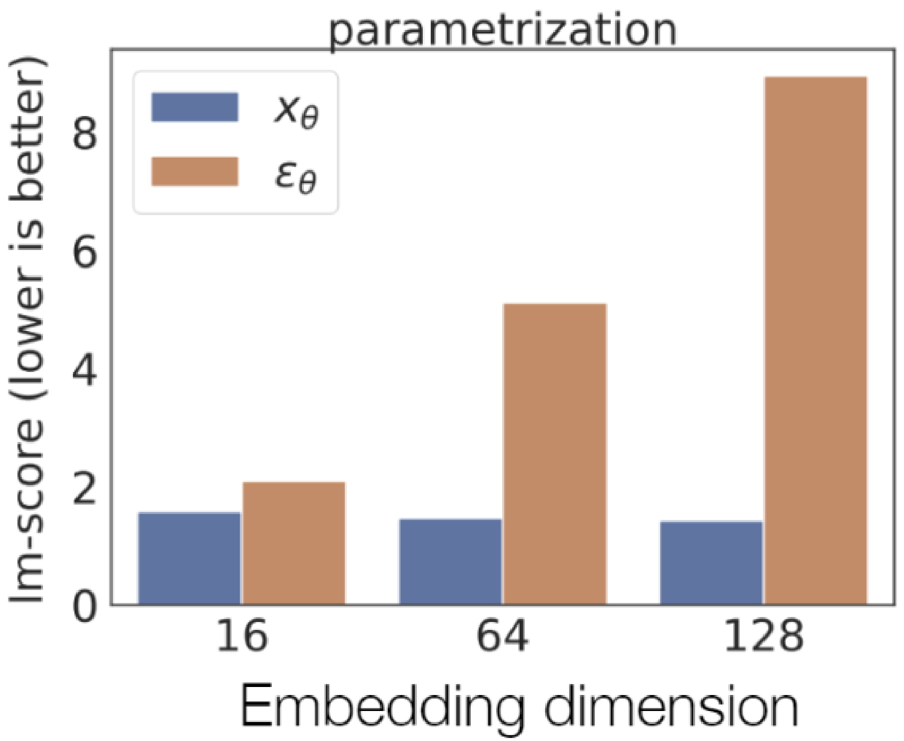
Continuous language diffusion requires more aggressive noise schedules since embeddings space is more **“clustered”**

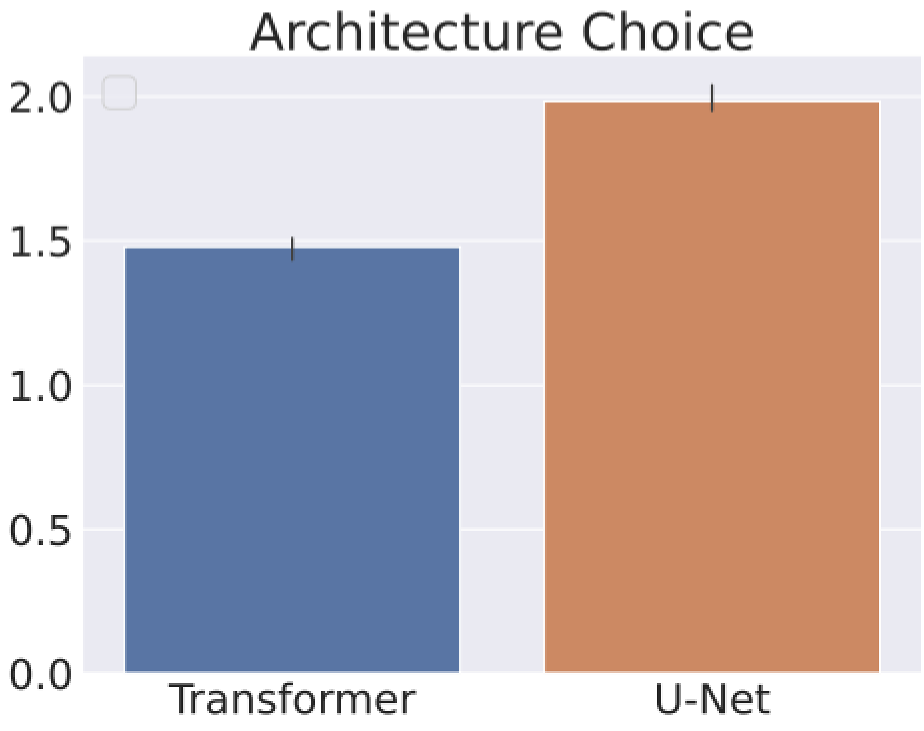
These steps are useless



Diffusion-LM

- **Object-prediction** is better than noise-prediction
- Transformers outperform U-net





Latent Continuous Diffusion

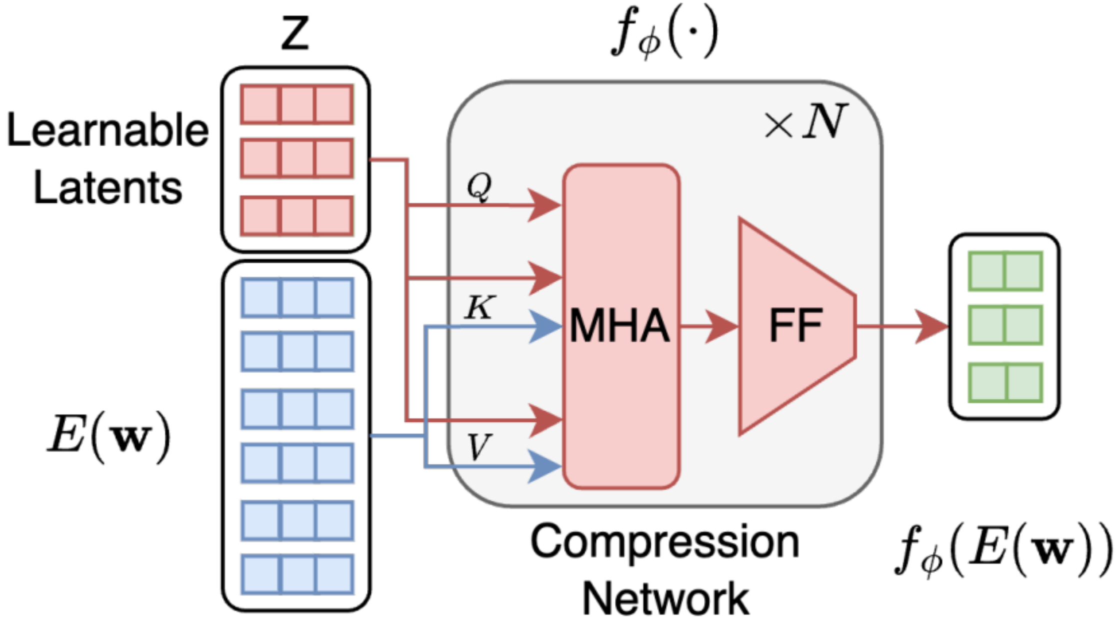
The next intuitive step in developing continuous diffusion was to construct appropriate **latent space,** because we know diffusion performs great in latent spaces

Most works on continuous language diffusion explore this direction

LD4TG

- **Compresses** pretrained LM outputs
- Decodes **autoregressively**

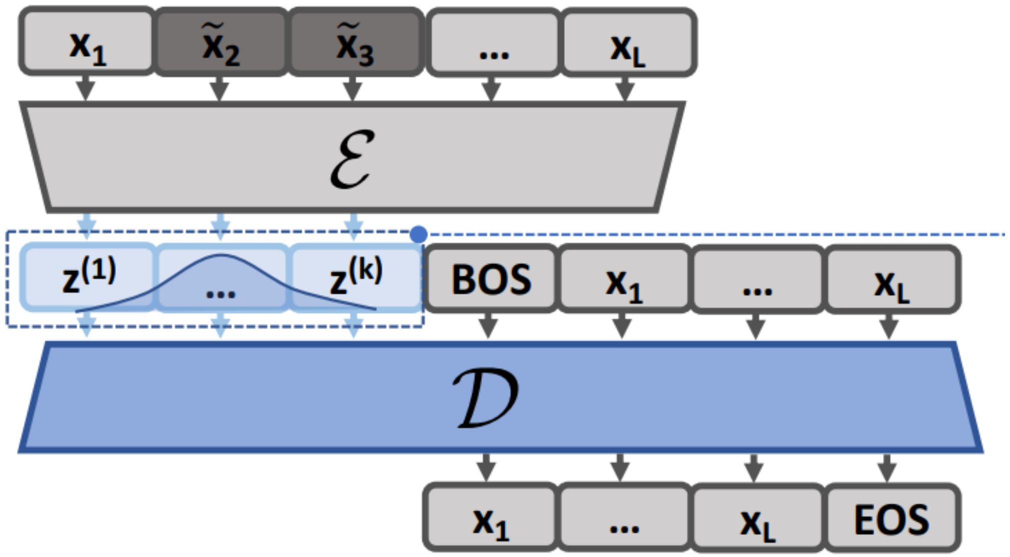
Achieved faster sampling, still stands as a strong baseline



PLANNER

- Map text to latent space with **encoder**
- Decode **autoregressively**
- **KL-regularize** latents

Variational Paragraph Embedder

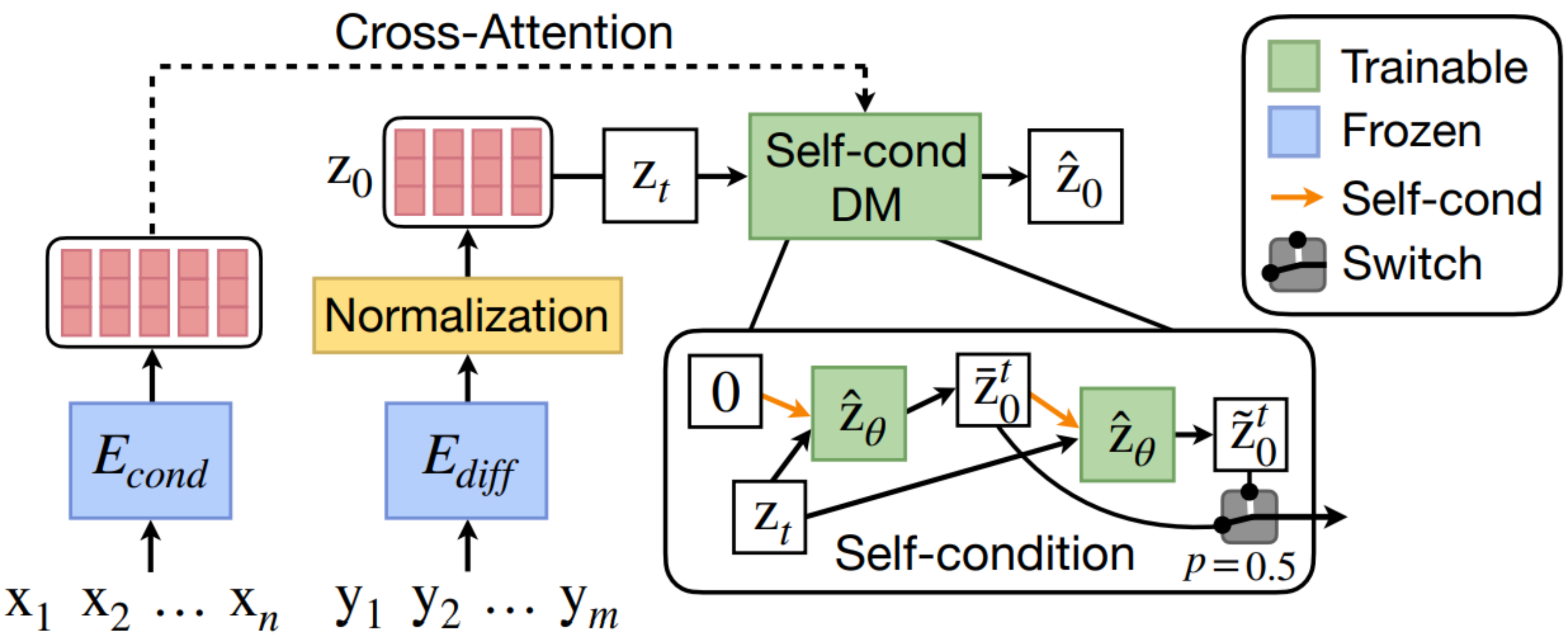


Step further…

Both **LD4TG** and **PLANNER** laid foundation for exploring latent diffusion. However, both methods decode text **autoregressively** from diffusion-generated latents, thus mitigating theoretical motivation for using diffusion

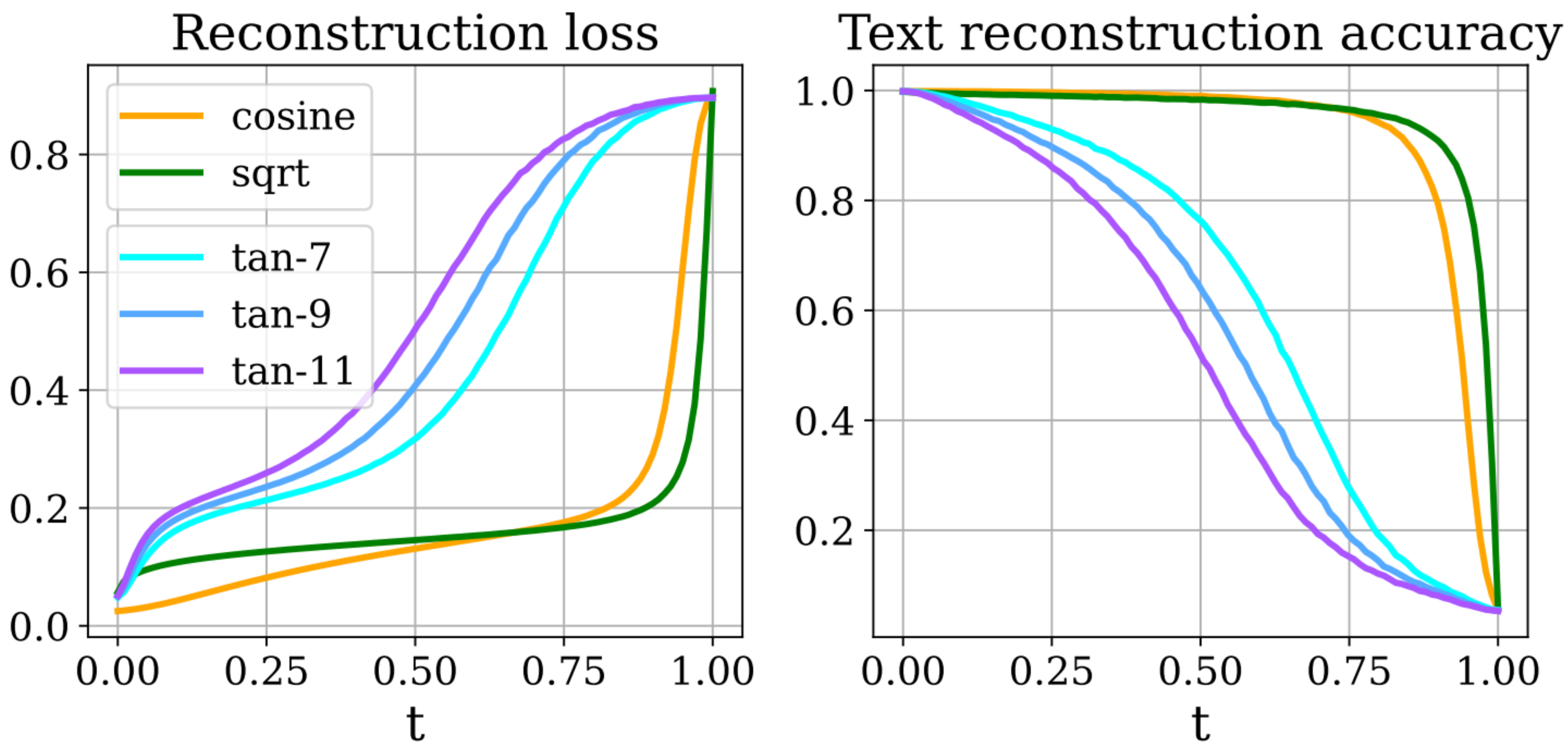
Text Encoding Diffusion Model

- Diffusion in **pretrained LM** space
- Decode **non-autoregressively**



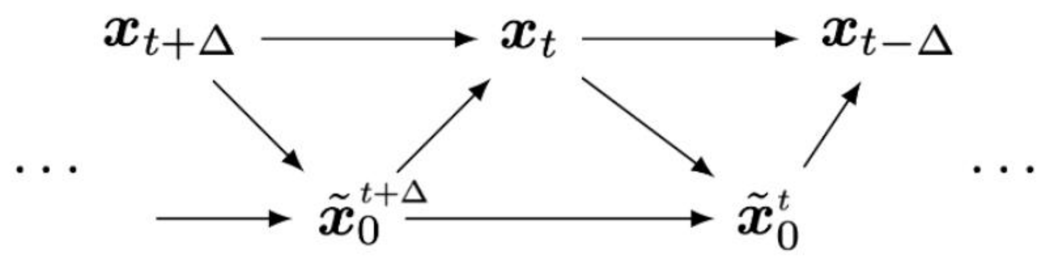
Text Encoding Diffusion Model

- Even more **aggressive schedule**



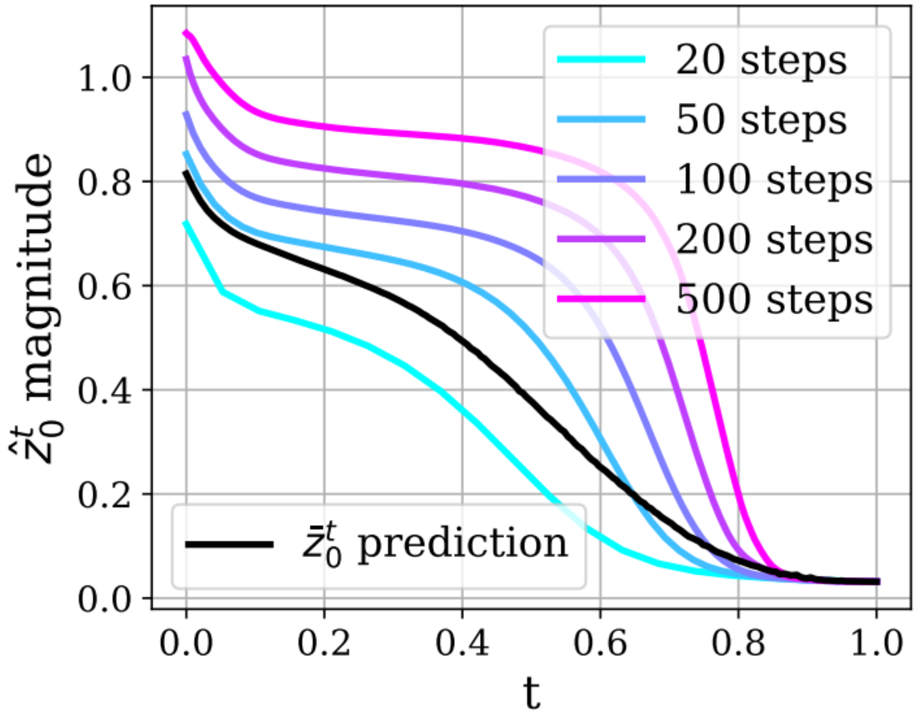
Text Encoding Diffusion Model

- Self-conditioning increases confidence, improving performance



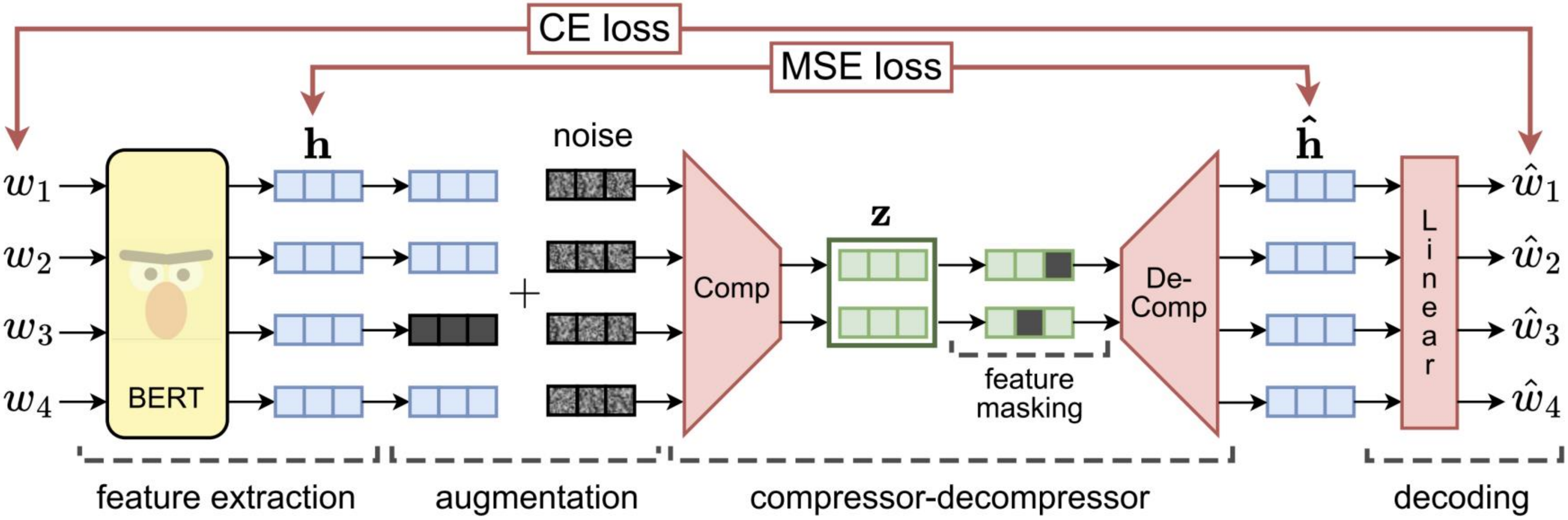
(b) Self-Conditioning on the previous x_0 estimate.

$$\tilde{x}_0^t = f_\theta(x_t, t, \tilde{x}_0^{t+\Delta})$$



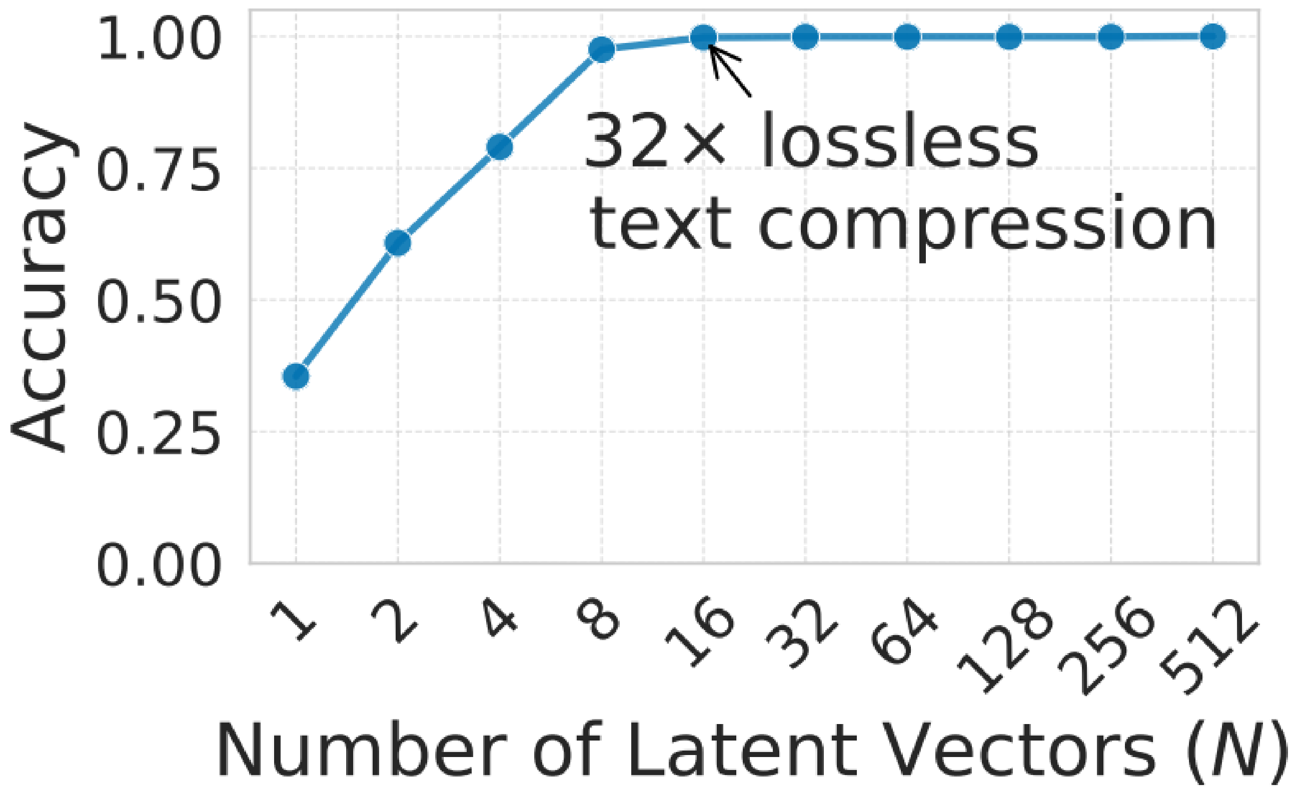
COSMOS

- Similar to LD4TG, compresses **pretrained LM** outputs
- Proposes procedure to **smoothen and compress** latent space



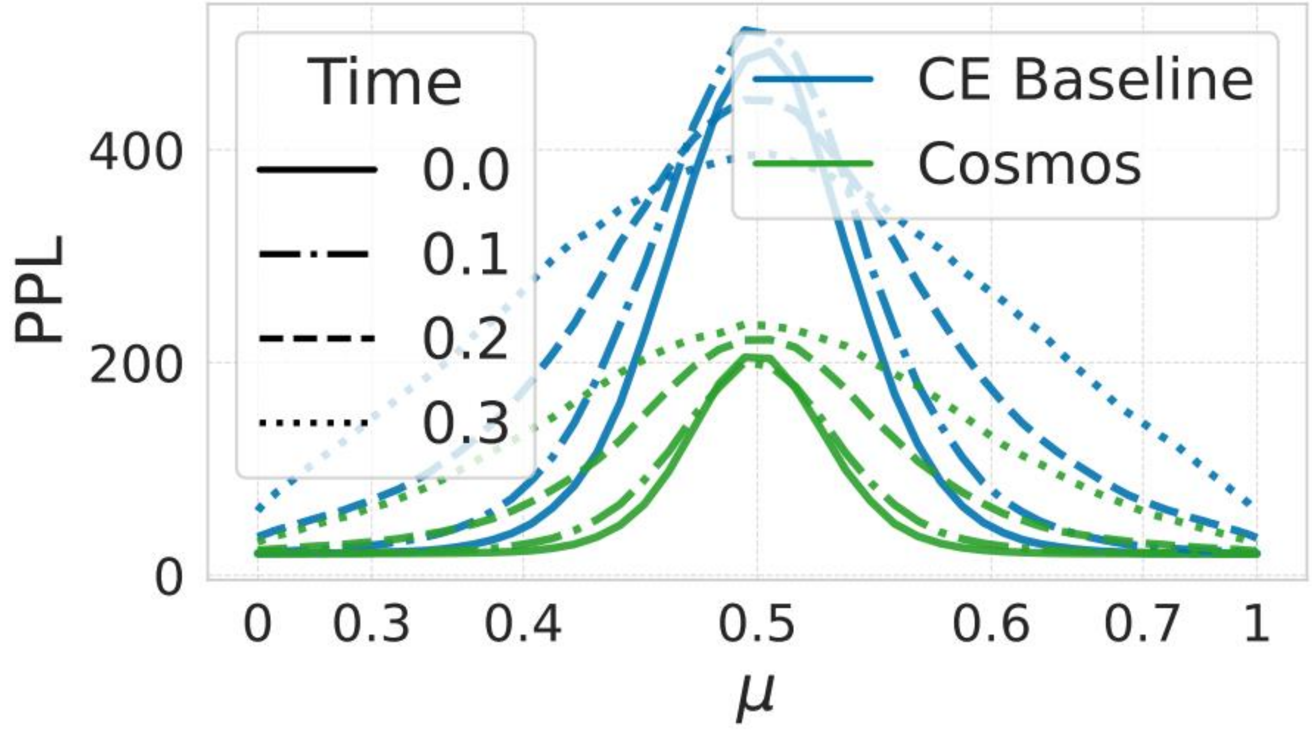
COSMOS

Text can be compressed in a quite “tight” latent space. However, this does not mean it will be easily modeled



COSMOS

- **MSE-regularizations** on encoder activations
- **Perturbations** of encoder activation (masking and noising)
- Latent-space feature masking



Continuous Diffusion Summary

Advantages:

- Corrects its mistakes
- Pretty fast
- Common methods from CV-Diffusion are applicable

Disadvantages:

- Often relies on pre-trained models
- Often requires training autoencoder separately
- Causal language generation is not straightforward

Discrete Diffusion

Discrete Denoising Diffusion Probabilistic Models

The idea is pretty simple. Let's define stochastic matrix that defines probabilities of token transitions:

$$[\boldsymbol{Q}_t]_{ij} = q(x_t = j | x_{t-1} = i)$$

Then the forward process is:

$$q(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}) = \mathrm{Cat}(\boldsymbol{x}_t; \boldsymbol{p} = \boldsymbol{x}_{t-1} \boldsymbol{Q}_t)$$

Discrete Denoising Diffusion Probabilistic Models

The marginal and posterior distributions:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathrm{Cat}\left(\boldsymbol{x}_t; \boldsymbol{p} = \boldsymbol{x}_0\overline{\boldsymbol{Q}}_t\right), \quad \text{with} \quad \overline{\boldsymbol{Q}}_t = \boldsymbol{Q}_1\boldsymbol{Q}_2\ldots\boldsymbol{Q}_t$$

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} = \mathrm{Cat}\left(\boldsymbol{x}_{t-1}; \boldsymbol{p} = \frac{\boldsymbol{x}_t\boldsymbol{Q}_t^\top \odot \boldsymbol{x}_0\overline{\boldsymbol{Q}}_{t-1}}{\boldsymbol{x}_0\overline{\boldsymbol{Q}}_t\boldsymbol{x}_t^\top}\right)$$

Optimizing NELBO (which is CE in disguise)

$$L_\lambda = L_{\mathrm{vb}} + \lambda\, \mathbb{E}_{q(\boldsymbol{x}_0)}\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[-\log \widetilde{p}_\theta(\boldsymbol{x}_0|\boldsymbol{x}_t)]$$

Discrete Denoising Diffusion Probabilistic Models

Uniform transition matrix

$$[\boldsymbol{Q}_t]_{ij} = \begin{cases} 1 - \frac{K-1}{K}\beta_t & \text{if} \quad i = j \\ \frac{1}{K}\beta_t & \text{if} \quad i \neq j \end{cases}$$

T = 0	The great brown fox hopped over the lazy dog.
T = 10	The vast black fox hopping over the lazy cat.
T = 20	Their vast tripped this jumping upon walked organizations.
T = 25	Bunk scamper tripped this Sanchez walked organizations.

Discrete Denoising Diffusion Probabilistic Models

Absorbing (masking) transition matrix

$$[\boldsymbol{Q}_t]_{ij} = \begin{cases} 1 & \text{if} \quad i = j = m \\ 1 - \beta_t & \text{if} \quad i = j \neq m \\ \beta_t & \text{if} \quad j = m, i \neq m \end{cases}$$

T = 0	The great brown fox hopped over the lazy dog.
T = 10	The great [MASK] fox hopped over [MASK] lazy dog.
T = 20	The [MASK][MASK] [MASK] ship over [MASK] lazy the.
T = 25	[MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK]

Masked Discrete Language Models

Basically, continues to explore **D3PM Absorbing** state, but with few changes:

- Improved architecture
- **Tighter** lower bound
- Faster **Semi-autoregressive** sampler

Masked Discrete Language Models

MDLM parametrization (SUBS) introduces two key properties:

- Zero Masking Probability
- Carry-Over Unmasking

With this changes, we can simplify the loss:

$$\mathcal{L}_{\text{diffusion}} = \sum_{i=1}^{T} \mathbb{E}_q [\mathrm{D}_{\mathrm{KL}}(q(\mathbf{z}_{s(i)} | \mathbf{z}_{t(i)}, \mathbf{x}) \| p_\theta(\mathbf{z}_{s(i)} | \mathbf{z}_{t(i)}))] = \sum_{i=1}^{T} \mathbb{E}_q \left[\frac{\alpha_{t(i)} - \alpha_{s(i)}}{1 - \alpha_{t(i)}} \log \langle \mathbf{x}_\theta(\mathbf{z}_{t(i)}), \mathbf{x} \rangle \right]$$

Masked Discrete Language Models

Most of MDLM success can be attributed to **improved architecture**:

- DiT woth RoPE instead of T5
- Better tokenization (larger vocabulary)

	PPL ($\leq$)
MDLM (47)	$\mathbf{27.04}\pm.01$
w/o continuous time (43)	$27.19\pm.07$
& w/o carry-over (41)	$28.56\pm.15$
& w/o zero masking (39)	$28.51\pm.15$

Generalized Interpolating Discrete Diffusion

Solves the problem of inability to **remask** tokens during generation:

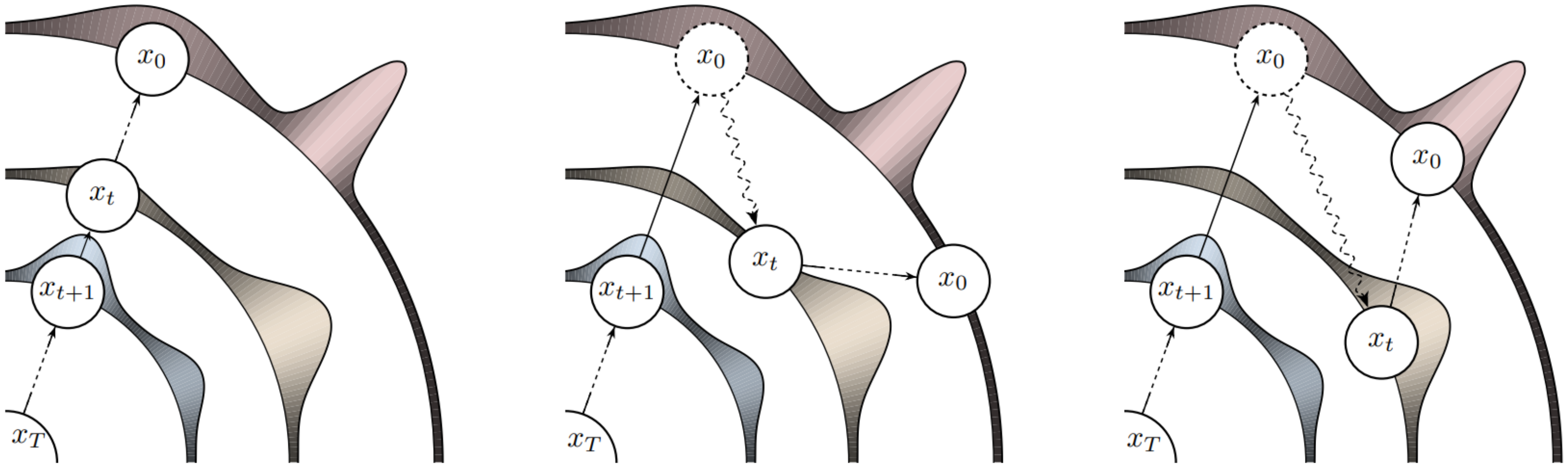
- Key feature is introducing mix of Absorbing and Uniform states

$$q_t(z_t|x) = \frac{1}{C_t}((1-t)\mathbf{x} + t\mathbf{m} + c_t\mathbf{u})$$

Guided Star-Shaped Masked Diffusion

Solves the problem of inability to **remask** tokens during generation:

- **Non-markovian** generation
- **Guided** noising with classificator



(a) MDLM (b) Star (c) G-Star

Large Diffusion Language models

Most Large Diffusion Models are Masked Diffusion

- Gemini Diffusion
- Seed Diffusion
- LLaDa

Large Diffusion Language models

- Easy to implement
- Scale good

Future of Diffusion Models in Language Modeling

- Current dominating paradigm is Masked Diffusion
- DLM do great in coding
- Continuous Diffusion will be scaled… soon (we are working on it)

Join our Text Diffusion Reading Group!