Partition Generative Modeling

Deschenaux et al. 2025

Reviewed & Presented by Joon Hyeok Kim

Contents

D3PM : General but heavy framework!

Forward transition probabilities matrix

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

$$Q_t = \begin{bmatrix} q(x_t = 1 \mid x_{t-1} = 1) & q(x_t = 2 \mid x_{t-1} = 1) & \cdots & q(x_t = N \mid x_{t-1} = 1) \\ q(x_t = 1 \mid x_{t-1} = 2) & q(x_t = 2 \mid x_{t-1} = 2) & \cdots & q(x_t = N \mid x_{t-1} = 2) \\ \vdots & \vdots & \ddots & \vdots \\ q(x_t = 1 \mid x_{t-1} = N) & q(x_t = 2 \mid x_{t-1} = N) & \cdots & q(x_t = N \mid x_{t-1} = N) \end{bmatrix}$$

- Props.)
  - Massive $Q_t \in N^2$ where N is the size of the vocabulary

D3PM continues

Question) What Q should we choose?

Answer) Absorbing State Transition Matrix

- Def.)
  - For the absorbing state m,

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

- Prop.)
  - This is equivalent to the interpolation notation : $Q_t = (1 - \beta_t)\mathbf{I} + \beta_t \mathbf{1} e_m^\top$
    - where e_m is a vector with a 1 on the absorbing state and 0s elsewhere.
  - Use this for the **MASK** token!

- e.g.) When the absorbing token $m = 2$

$$(1-\beta_t)\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} + \beta_t \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \underbrace{1}_{\text{abs}} & 0 & \cdots & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1-\beta_t & \beta_t & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & \beta_t & 1-\beta_t & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \underbrace{\beta_t}_{\text{abs}} & 0 & \cdots & 1-\beta_t \end{bmatrix}$$

D3PM continues

Forward Process

One Step Marginal

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

- i.e.) Sample $\boldsymbol{x}_t$ from the categorical distribution $\boldsymbol{x}_{t-1}\boldsymbol{Q}_t$

One Shot Marginal

$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathrm{Cat}(\boldsymbol{x}_t; \boldsymbol{p} = \boldsymbol{x}_0\overline{\boldsymbol{Q}}_t)$: the t-step marginal

- where $\overline{\boldsymbol{Q}}_t = \boldsymbol{Q}_1\boldsymbol{Q}_2 \cdots \boldsymbol{Q}_t$

Posterior

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

D3PM continues

Reverse Process

$$p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) \propto \sum_{\tilde{\boldsymbol{x}}_0} \underbrace{q(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t \mid \tilde{\boldsymbol{x}}_0)}_{\text{term 0}} \quad \underbrace{\tilde{p}_\theta(\tilde{\boldsymbol{x}}_0 \mid \boldsymbol{x}_t)}_{\text{nn}_\theta(\boldsymbol{x}_t) \text{ predicts this!}}$$

$$\underbrace{q(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t \mid \tilde{\boldsymbol{x}}_0)}_{\text{term 0}} = \underbrace{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \tilde{\boldsymbol{x}}_0)}_{\text{term 1}} \cdot \underbrace{q(\boldsymbol{x}_{t-1} \mid \tilde{\boldsymbol{x}}_0)}_{\text{term 2}}$$

$$\underbrace{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \tilde{\boldsymbol{x}}_0)}_{\text{term 1}} = q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$$

$$\underbrace{q(\boldsymbol{x}_{t-1} \mid \tilde{\boldsymbol{x}}_0)}_{\text{term 2}} = \tilde{x}_0 \overline{Q}_{t-1} = \tilde{x}_0 Q_1 \cdots Q_{t-1} \quad \leftarrow \text{Expensive } \overline{Q}$$

D3PM continues

Loss

$$L_\lambda = L_{\text{vb}} + \lambda \mathbb{E}_{q(\boldsymbol{x}_0)} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \big[-\log \tilde{p}_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_t) \big]$$

$$L_{\text{vb}} = \mathbb{E}_{q(\boldsymbol{x}_0)} \Bigg[\underbrace{D_{\text{KL}}[q(\boldsymbol{x}_T|\boldsymbol{x}_0)||p(\boldsymbol{x}_T)]}_{L_T} + \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \big[D_{\text{KL}}[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)] \big]}_{L_{t-1}} \underbrace{-\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} [\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)]}_{L_0} \Bigg]. \tag{1}$$

forward posterior

reverse denoiser

Drawback

- q and p_θ are N dimensional distributions, which are heavy.
- Parameterization is complicated (Consider all possible $\tilde{x}_0$s)

MDLM : Single Token Problem

Idea

Keep the **absorbing state transition** to use the MASK token!

Instead of considering N dimensional distribution Q_t, focus only on **two** tokens!

1. original token $\mathbf{x}$
2. mask token $\mathbf{m}$

Inheriting the **interpolation notation** in the forward process

- $q(\mathbf{z}_t \mid \mathbf{x}) = \mathrm{Cat}(\mathbf{z}_t\ ;\ \alpha_t \mathbf{x} + (1-\alpha_t)\mathbf{m})$

i.e.) Sample the intermediate state $\mathbf{z}_t$ from the distribution $\alpha_t \mathbf{x} + (1-\alpha_t)\mathbf{m}$

MDLM : Single Token Problem continues

Forward Process

Marginal

$$q(\mathbf{z}_t \mid \mathbf{x}) = \mathrm{Cat}(\mathbf{z}_t \; ; \; \alpha_t \mathbf{x} + (1 - \alpha_t)\mathbf{m})$$

- Then, $\mathbf{z}_t$ is sampled from $\alpha_t \mathbf{x} + (1 - \alpha_t)\mathbf{m}$.
  - i.e.) $\mathbf{z}_t = \begin{cases} \mathbf{x} & \text{with } p = \alpha_t \\ [\mathrm{MASK}] & \text{with } p = 1 - \alpha_t \end{cases}$

Posterior

$$q(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x}) = \begin{cases} \mathrm{Cat}(\mathbf{z}_s \; ; \; \mathbf{z}_t) & \mathbf{z}_t \neq \mathbf{m} \\ \mathrm{Cat}\left(\mathbf{z}_s \; ; \; \dfrac{(1 - \alpha_s)\mathbf{m} + (\alpha_s - \alpha_t)\mathbf{x}}{1 - \alpha_t}\right) & \mathbf{z}_t = \mathbf{m} \end{cases}$$

- where $(s < t)$

MDLM : Single Token Problem continues

Reverse Process : SUBS Parameterization- Model)

$$p_\theta(\mathbf{z}_s \mid \mathbf{z}_t) = q(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x} = \mathbf{x}_\theta(\mathbf{z}_t, t)) = \begin{cases} \mathrm{Cat}(\mathbf{z}_s \; ; \; \mathbf{z}_t) \\ \mathrm{Cat}\left(\mathbf{z}_s \; ; \; \dfrac{(1-\alpha_s)\mathbf{m} + (\alpha_s - \alpha_t)\mathbf{x}_\theta(\mathbf{z}_t, t)}{1-\alpha_t}\right) \end{cases}$$

- Desc.)
  - Similar to the forward posterior.
    - Only difference is that $\mathbf{x}$ is parameterized as $\mathbf{x}_\theta(\mathbf{z}_t, t)$
  - Cases
    - If $\mathbf{z}_t$ is masked (initial state),
      - we draw $\mathbf{z}_s$ from the distribution $\dfrac{(1-\alpha_s)\mathbf{m} + (\alpha_s - \alpha_t)\mathbf{x}_\theta(\mathbf{z}_t, t)}{1-\alpha_t}$
    - Else (i.e. already unmasked),
      - $\mathbf{z}_s = \mathbf{z}_t$ definitively.

MDLM : Single Token Problem continues

Loss

Start with ELBO

$$L_{\text{vb}} = \mathbb{E}_{q(\boldsymbol{x}_0)}\Bigg[\underbrace{D_{\text{KL}}[q(\boldsymbol{x}_T|\boldsymbol{x}_0)||p(\boldsymbol{x}_T)]}_{L_T} + \sum_{t=2}^{T}\underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\text{KL}}[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)]\right]}_{L_{t-1}}$$
$$\underbrace{-\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)]}_{L_0}\Bigg]. \quad (1)$$

forward posterior

reverse denoiser

- Put $L_{T-1} = \mathcal{L}_{\text{diffusion}}$

MDLM : Single Token Problem continues

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_q \left(\sum_{i=1}^{T} D_{\text{KL}} \left[q(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}) \parallel p_\theta(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) \right] \right)$$

$$= \sum_{i=1}^{T} \mathbb{E}_q \left(D_{\text{KL}} \left[\underbrace{q(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x})}_{\text{Forward Posterior}} \parallel \underbrace{p_\theta(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x}_\theta(\mathbf{z}_t, t))}_{\text{SUBS para'd Reverse}} \right] \right)$$

where $s =$

$\because \mathbf{z}_t \in \{\mathbf{x},$

$\vdots$

Rao-Blackwellized Likelihood Bounds backs up Simplification

$\vdots$

$$= \sum_{i=1}^{T} \mathbb{E}_q \left[\frac{\alpha_t - \alpha_s}{1 - \alpha_t} \underbrace{\log \langle \mathbf{x}_\theta(\mathbf{z}_t, t),\ \mathbf{x} \rangle}_{\text{Cross Entropy Loss!}} \right]$$

cf.) $\begin{cases} \mathbf{z}_t = \\ \mathbf{z}_t = \end{cases}$

MDLM : Single Token Problem continues

Continuous Time Loss

Making the number of steps $T \to \infty$, we may get

$$\mathcal{L}_{\text{NELBO}}^{\infty} = \mathbb{E}_q \int_{t=0}^{t=1} \frac{\alpha_t'}{1-\alpha_t} \underbrace{\log \langle \mathbf{x}_\theta(\mathbf{z}_t, t),\ \mathbf{x} \rangle}_{\text{Cross Entropy Loss!}} \mathrm{d}t$$

- where $\alpha_t' = \lim_{T \to \infty} T(\alpha_t - \alpha_s) \quad (\because T \to \infty \Rightarrow s \to t)$

MDLM dealing with length L sequence of tokens

Forward noising process is applied independently across a sequence.

- $q(\mathbf{z}_t^{1:L} \mid \mathbf{x}^{1:L}) = \prod_{\ell=1}^{L} q(\mathbf{z}_t^{\ell} \mid \mathbf{x}^{\ell})$

Denoising process factorizes independently across tokens, conditioned on a sequence of latents $\mathbf{z}_t^{1:L}$

- $p_\theta(\mathbf{z}_s^{1:L} \mid \mathbf{z}_t^{1:L}) = \prod_{\ell=1}^{L} p_\theta(\mathbf{z}_s^{\ell} \mid \mathbf{z}_t^{1:L})$

Optimization)

- $\mathcal{L}_{\text{NELBO}}^{\infty} = \mathbb{E}_q \int_{t=0}^{t=1} \frac{\alpha_t'}{1-\alpha_t} \sum_{\ell=1}^{L} \log \left\langle \mathbf{x}_\theta^{\ell}(\mathbf{z}_t^{1:L}, t), \mathbf{x}^{\ell} \right\rangle \mathrm{d}t$

PGM also solves Mask Generative Modeling problem

$$\mathcal{L}_{\text{MGM}} := \mathbb{E}_{\mathbf{x}\sim\mathcal{D},t\sim\mathcal{U}[0,1]}\left[w(t)\text{CE}(\mathbf{x}_\theta(\mathbf{z}_t,t),\mathbf{x})\right]$$

Accoring to them MDLM's NELBO loss is just a MGM loss with special weight

$$\mathcal{L}_{\text{NELBO}}^{\infty} = \mathbb{E}_q \int_{t=0}^{t=1} \frac{\alpha_t'}{1-\alpha_t} \sum_{\ell=1}^{L} \underbrace{\log\langle \mathbf{x}_\theta(\mathbf{z}_t,t),\ \mathbf{x}\rangle}_{\text{Cross Entropy Loss!}} \mathrm{d}t$$

- where $w(t) = \dfrac{\alpha_t'}{1-\alpha_t}$

PGM's Idea : Based on the MDLM framework...

- Treat **masked** and **unmasked** tokens as **group 0** and **group 1**
- Let each group learn each other in the opposite time schedule

Objective

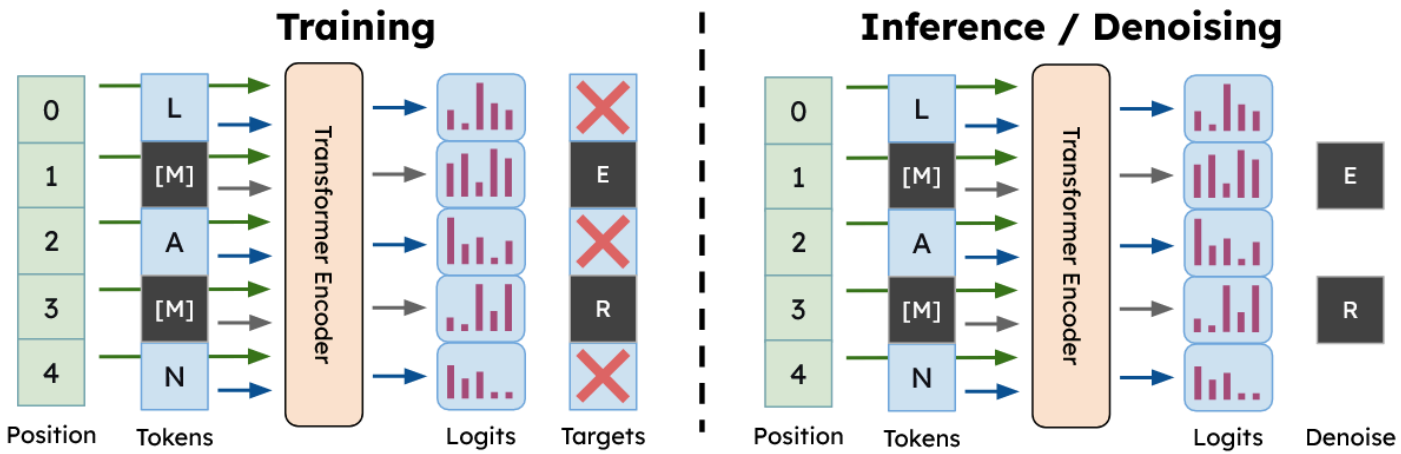
$$\mathcal{L}_{\text{PGM}} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D},t\sim\mathcal{U}[0,1]}\left[w^{\text{PGM}}(\mathbf{g},t)\ \text{CE}(\mathbf{x}_\theta(\mathbf{x};\mathbf{g};t),\mathbf{x})\right]$$

- where $w^{\text{PGM}}(\mathbf{g},t)_i = \begin{cases} w(t) & \text{if } \mathbf{g}_i = 0 \\ w(1-t) & \text{if } \mathbf{g}_i = 1 \end{cases}$

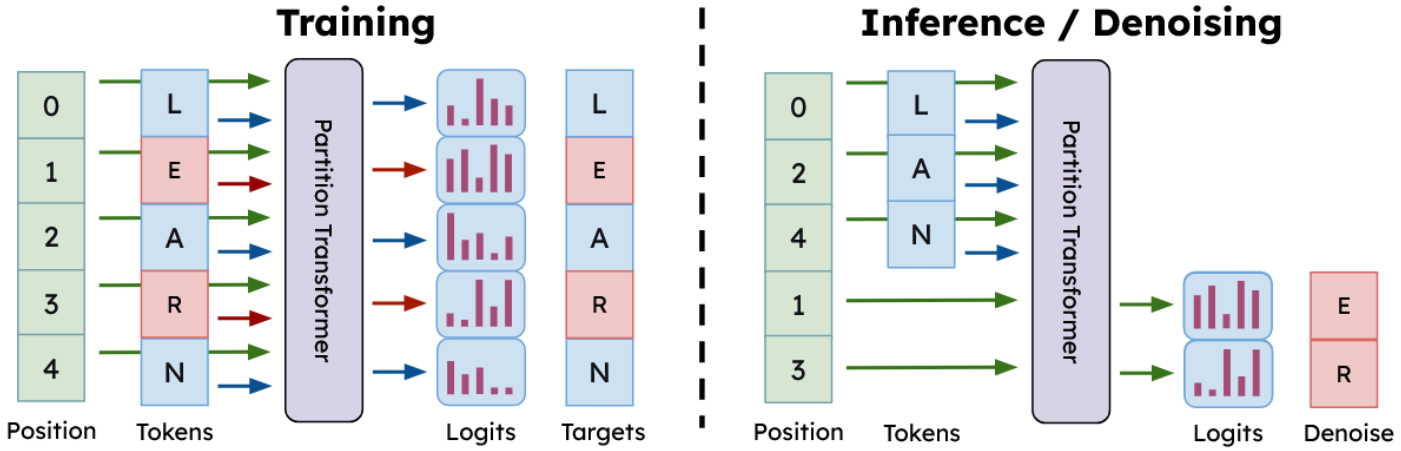
How?

- Enable this using the architecture

MDLM vs PGM

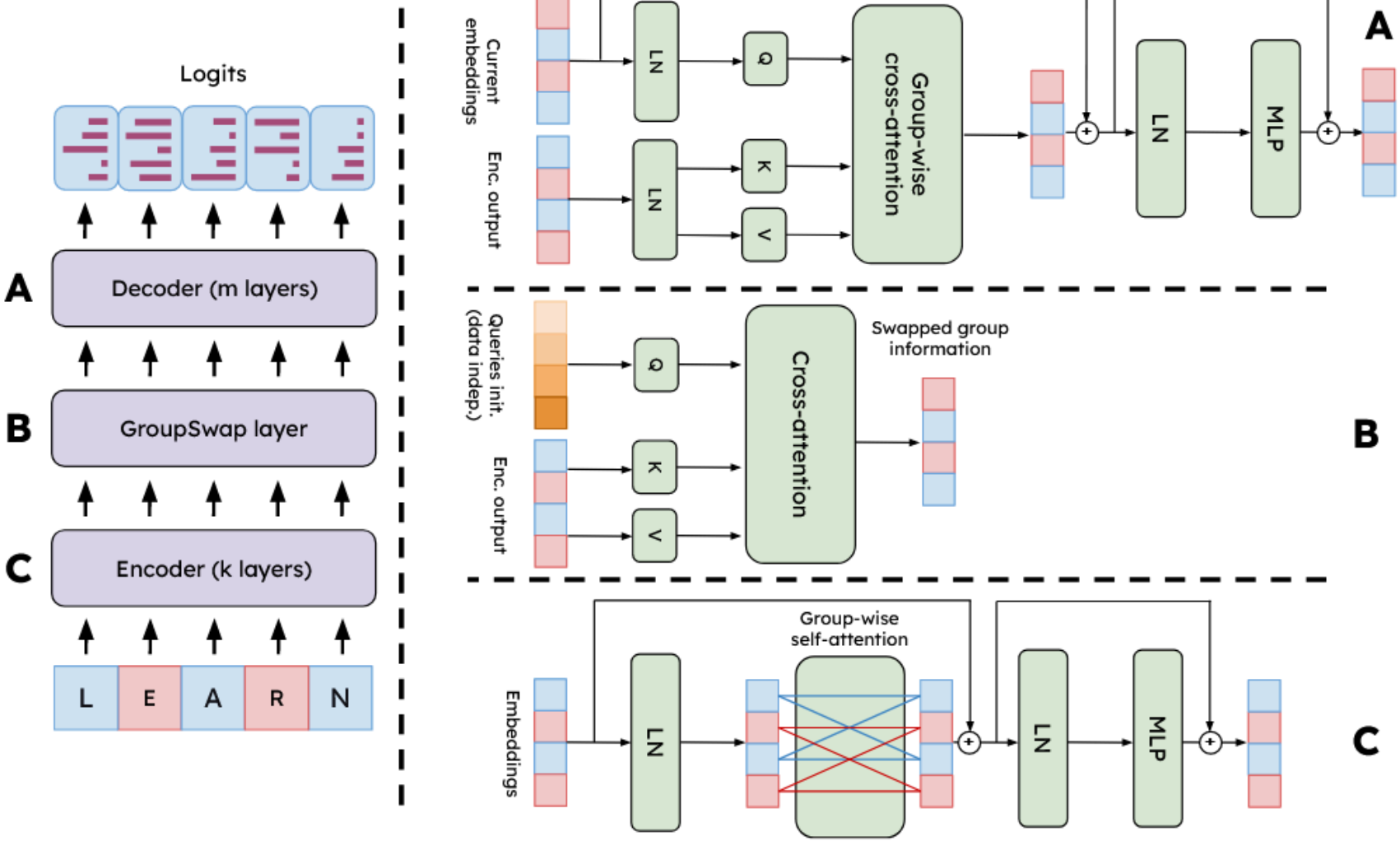


(a) Masked Generative Modeling (MGM)



(b) Partition Generative Modeling (PGM)

Partition Transformer



Encoder

- Desc.)
  - Partition-wise **self-attention** blocks
- Prop.)
  - Similar to standard bidirectional transformer blocks
  - Different on that separate groups do not attend each other

Decoder

- Desc.)
  - Cross-attention layer
    - Keys and Values are computed based on the output of the encoder
    - Queries are computed using either...
      - the output of the GroupSwap layer (Current Implementation)
      - the output of the previous decoder block (Future work?)
- Prop.)
  - No self-attention layer
    - Why?)
      - Compute predictions solely at the positions that we will decode.
      - Paper argues that this allows efficient generation.

GroupSwap Layer

- Goal)
  - Recall that encoder localized each group information.
  - This layer allows information exchange between groups.
    - Why?)
      - Prediction on group 1 should be based on group 0
      - and vice versa
- Implementation)
  - Cross Entropy
    - Queries
      - Data-Independent
      - Data-Dependent
    - Keys and Values
      - Inputted from the encoder
      - No speicified details provided.

Experiments

Language Modeling : PGM vs MDLM

- PGM achieved....
  - lower validation **perplexity**
  - 5 times higher sampling **throughput**
  - faster inferece without sacrificying **downstream performance**
    - i.e. tasks never trained
  - higher accuracy after **distillation**

Image Modeling : PGM vs MaskGIT (VQGAN on ImageNet)

- PGM (64 steps) achieved....
  - improved FID score
  - 3.9 times faster

Isolating the effect of Complementary Masking

- Setting)
  - Use only MDLM
  - Input complementary masked sequences
    - e.g.) "L[M]A[M]N", "[M]E[M]R[M]"
  - Compare it with the normal MDLM.
- Result

Model ↓	#Params	Val. PPL	Latency (sec) ↓	TP (tok/sec) ↑
LM1B (ctx len. 128)				
MDLM	170M	27.67	3.78	1'081.57
MDLM† (Compl. masking)	170M	**25.72**	3.78	1'081.57
PGM 6 / 6	171M	26.80	**2.12**	**1'930.93**
OpenWebText (ctx len. 1024)				
MDLM	170M	23.07	31.41	1'043.22
MDLM† (Compl. masking)	170M	22.98	31.41	1'043.22
PGM 8 / 8	203M	22.61	**5.86**	**5'585.57**
PGM 6 / 6 (dim. 1024)	268M	**21.43**	5.93	5'518.09