

Discrete Diffusion Language Models

From Transition Matrix to Frontier Models

D3PM · SEDD · MDLM



Transition → Rate & Score → Reverse Design

00 Questions : 각 논문이 답하는 질문 하나씩을 따라간다

01

D3PM

How do tokens move?

Transition matrix Q

02

SEDD

How fast do they move back?

Rate matrix + ratio R

03

MDLM

How do we simplify reversal?

Reverse kernel + weighted CE

마지막: 지금의 모델 성능은 어디까지 왔고, 연구 질문은 무엇으로 이동했는가?

00 Running Example : A / B / MASK absorbing process로 모든 계산을 연결한다

R (rate matrix)

	A	B	M
A	-1	0	0
B	0	-1	0
M	1	1	0

Q = exp((ln 2)R)

	A	B	M
A	0.5	0	0
B	0	0.5	0
M	0.5	0.5	1

$$\mathbf{p}_0 = (0.7, 0.3, 0)^T$$

Columns: source · Rows: destination · Q: $\beta = 0.5$

The same process in three views

D3PM $Q = \exp((\ln 2)R)$

SEDD $d\mathbf{p}\tau/d\tau = R \mathbf{p}\tau$

MDLM $t = 1 - \exp(-\tau)$

$$t = 0.5 \leftrightarrow \tau = \ln 2 \quad / \quad t = 0.75 \leftrightarrow \tau = \ln 4$$

01 Transition Matrix : Q_{ij} = 상태 j에서 i로 갈 확률 (column convention)

Q

	A	B	M
A	.5	0	0
B	0	.5	0
M	.5	.5	1

Nonnegative

$Q_{ij} \geq 0$: 음의 전이 확률은 없다.

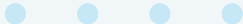
Mass conservation

각 column의 합은 1; 확률질량이 보존된다.

MASK $\rightarrow$ MASK = 1 : absorbing state는 forward에서 벗어나지 않는다.

01 Corruption Kernel : Q에 따라 파괴되는 정보의 형태가 달라진다

UNIFORM



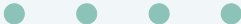
어떤 토큰으로든 이동

Vocab 전체에 질량 분산

$$Q = (1 - \beta)I + \frac{\beta}{K}\mathbf{1}\mathbf{1}^\top = \begin{bmatrix} .8 & .1 & .1 \\ .1 & .8 & .1 \\ .1 & .1 & .8 \end{bmatrix}$$

$\beta = 0.3, K = 3$ · 상태 (A, B, C)

DISCRETE GAUSSIAN



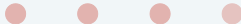
인접한 상태를 우선

순서·거리 구조가 존재할 때

$$Q_{ij} \propto e^{-\frac{4(i-j)^2}{(K-1)^2\beta}} \Rightarrow \begin{bmatrix} .8 & .15 & .05 \\ .15 & .7 & .15 \\ .05 & .15 & .8 \end{bmatrix}$$

순서 있는 상태 (1, 2, 3) · 예시 수치

ABSORBING



특정 MASK로 이동

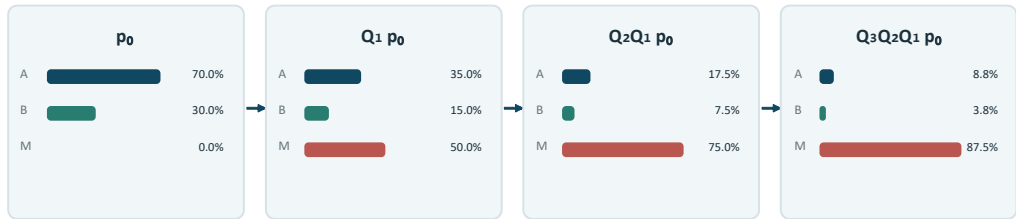
정보 삭제(erasure) 모델링

$$Q = (1 - \beta)I + \beta e_M\mathbf{1}^\top = \begin{bmatrix} .5 & 0 & 0 \\ 0 & .5 & 0 \\ .5 & .5 & 1 \end{bmatrix}$$

$\beta = 0.5$ · 상태 (A, B, M)

D3PM의 메시지: “noise”는 숫자 덧셈이 아니라 정보 전이를 정의하는 행렬이다.

01 Forward : Q를 곱하면 임의 시점의 x_t 를 직접 샘플링할 수 있다



$$q(x_t | x_0) = \text{Cat}(\bar{Q}_t e_{(x_0)}) \quad \cdot \quad \bar{Q}_t = Q_t Q_{t-1} \cdots Q_1$$

$\bar{Q}_t = Q_t \cdots Q_1$ 하나로 x_0 에서 x_t 를 바로 샘플링한다.

01 Two Steps : step당 생존 확률 0.5이면 두 step 뒤 A는 0.25만 남는다

Q

	A	B	M
A	.5	0	0
B	0	.5	0
M	.5	.5	1

Q²

	A	B	M
A	.25	0	0
B	0	.25	0
M	.75	.75	1

$x_0=A \rightarrow x_1: A \ 0.50 / M \ 0.50 \rightarrow x_2: A \ 0.25 / M \ 0.75$

01 Posterior : Bayes rule을 전이 행렬의 원소로 쓰기

BAYES

$$q(x_{t-1} = i \mid x_t = j, x_0 = k) = \frac{q(x_t = j \mid x_{t-1} = i) q(x_{t-1} = i \mid x_0 = k)}{q(x_t = j \mid x_0 = k)} = \frac{\overbrace{Q_t[j, i]}^{i \rightarrow j} \overbrace{\bar{Q}_{t-1}[i, k]}^{k \rightarrow i}}{\underbrace{\bar{Q}_t[j, k]}_{k \rightarrow j}}$$

Markov 성질로 $q(x_t \mid x_{t-1}, x_0) = q(x_t \mid x_{t-1})$ · 열 = 출발 상태 (이 발표의 표기)

VECTOR

$$q(x_{t-1} \mid x_t = j, x_0 = k) = \frac{(Q_t^\top e_j) \odot (\bar{Q}_{t-1} e_k)}{e_j^\top \bar{Q}_t e_k}, \quad \bar{Q}_t = Q_t Q_{t-1} \cdots Q_1$$

$Q_t^\top e_j$: 각 i 에서 j 로 갈 확률 (행 j) · $\bar{Q}_{t-1} e_k$: $x_0 = k$ 에서 각 i 에 있을 확률 (열 k)

ABSORBING

$$x_t = M: q(x_{t-1} = x_0 \mid \cdot) = \frac{\beta_t (1 - \bar{\beta}_{t-1})}{\bar{\beta}_t}, \quad q(x_{t-1} = M \mid \cdot) = \frac{\bar{\beta}_{t-1}}{\bar{\beta}_t}; \quad x_t \neq M: x_{t-1} = x_t$$

$\bar{\beta}_t = 1 - \prod_{i \leq t} (1 - \beta_i)$: t step 안에 MASK가 되었을 누적 확률

분자는 $x_0 \rightarrow x_{t-1} \rightarrow x_t$ 경로의 확률이고, 분모는 그 경로들의 합이다.

01 Posterior Example : $x_0 = A$, $x_2 = \text{MASK}$ 일 때 x_1 의 분포

$$Q = \begin{bmatrix} .5 & 0 & 0 \\ 0 & .5 & 0 \\ .5 & .5 & 1 \end{bmatrix}, \quad \bar{Q}_2 = Q^2 = \begin{bmatrix} .25 & 0 & 0 \\ 0 & .25 & 0 \\ .75 & .75 & 1 \end{bmatrix} \quad (\text{rows, cols} = A, B, M; \text{ cols} = \text{source})$$

$x_2 = M$, $x_0 = A$

$$\underbrace{Q^\top e_M}_{\text{row } M \text{ of } Q} = \begin{bmatrix} .5 \\ .5 \\ 1 \end{bmatrix} \odot \underbrace{\bar{Q}_1 e_A}_{\text{col } A \text{ of } Q} = \begin{bmatrix} .5 \\ 0 \\ .5 \end{bmatrix} = \begin{bmatrix} .25 \\ 0 \\ .5 \end{bmatrix} \xrightarrow{\div \bar{Q}_2[M,A] = .75} \begin{bmatrix} 1/3 \\ 0 \\ 2/3 \end{bmatrix}$$

$$x_2 = A : \begin{bmatrix} .5 \\ 0 \\ 0 \end{bmatrix} \odot \begin{bmatrix} .5 \\ 0 \\ .5 \end{bmatrix} = \begin{bmatrix} .25 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{\div .25} (1, 0, 0) \quad \text{still visible} \Rightarrow x_1 = A$$

$x_2 = \text{MASK}$ 를 보면, x_1 은 1/3 확률로 아직 A였고 2/3 확률로 이미 MASK였다.

01 Reverse Model : $p_{\theta}(x_{t-1} | x_t)$ 를 x_0 예측으로 바꾸기

DIRECT

$$p_{\theta}(x_{t-1} | x_t) = \text{Cat}(x_{t-1}; \text{softmax}(f_{\theta}(x_t, t)))$$

가능하지만, forward가 허용하지 않는 전이에도 확률을 줄 수 있다

x_0 -PARAM

$$p_{\theta}(x_{t-1} | x_t) \propto \sum_{\tilde{x}_0} q(x_{t-1}, x_t | \tilde{x}_0) \tilde{p}_{\theta}(\tilde{x}_0 | x_t) = \sum_{\tilde{x}_0} q(x_t | x_{t-1}) q(x_{t-1} | \tilde{x}_0) \tilde{p}_{\theta}(\tilde{x}_0 | x_t)$$

Posterior의 x_0 자리에 모델이 예측한 분포 $\tilde{p}_{\theta}(\tilde{x}_0 | x_t)$ 를 넣는다 (D3PM 식 4)

VECTOR

$$p_{\theta}(x_{t-1} | x_t = j) \propto (Q_t^{\top} e_j) \odot (\bar{Q}_{t-1} \tilde{p}_{\theta}(\cdot | x_t)) \quad (\text{posterior: } e_k \rightarrow \tilde{p}_{\theta})$$

EXAMPLE

$$\tilde{p}_{\theta} = (.7, .3, 0) : \begin{bmatrix} .5 \\ .5 \\ 1 \end{bmatrix} \odot \begin{bmatrix} .35 \\ .15 \\ .5 \end{bmatrix} = \begin{bmatrix} .175 \\ .075 \\ .5 \end{bmatrix} \xrightarrow{\div .75} (0.233, 0.100, 0.667)$$

모델은 x_0 만 예측하고, 나머지는 정확한 posterior가 처리한다.

$$\text{KL} (q(x_{t-1} | x_t, x_0) \parallel p_{\theta}(x_{t-1} | x_t))$$

Known target q

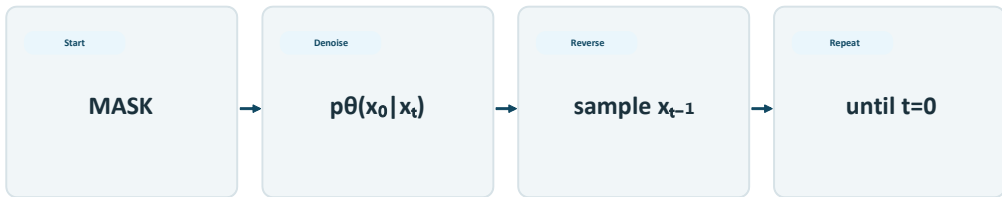
Forward Q와 데이터 x_0 로 Bayes posterior 계산

Learned p_{θ}

Neural denoiser의 clean-token 예측으로 reverse 구성

D3PM에서는 일반적인 Q도 처리한다. MASK-only 단순화는 이후 MDLM에서 핵심이 된다.

01 Inference : x_0 를 모르므로 p_θ 의 예측으로 역전이를 반복한다



핵심: Forward가 결정한 q 와 데이터에서 학습한 p_θ 를 결합해 역방향을 수행한다.

$$Q(\Delta\tau) = \exp(\Delta\tau R) \approx I + \Delta\tau R$$

Discrete view

한 번에 이동할 확률 $Q(\Delta\tau)$

Continuous view

순간적으로 이동하는 rate R

SEDD의 새 기여는 rate 자체가 아니라, 역방향 rate에 필요한 discrete score 학습이다.

02 Rate $\neq$ Probability : 열 합은 0, off-diagonal은 비음수

R

	A	B	M
A	-1	0	0
B	0	-1	0
M	1	1	0

$$R(M,A) = 1$$

A가 MASK가 되는 순간 전이율

$$P(A \rightarrow M \text{ in } \Delta\tau) \approx \Delta\tau$$

작은 $\Delta\tau$ 에서만 확률로 환산

R columns sum to 0

각 상태에서 나가는 rate와 diagonal 균형

Rate는 단위시간당 전이율이며, $\Delta\tau$ 를 곱해야 확률이 된다.

02 Forward ODE : Q를 반복하는 대신 분포의 순간 변화율을 쓴다

$$d p_{\tau} / d \tau = R p_{\tau} \quad \rightarrow \quad p_{\tau} = \exp(\tau R) p_0$$

$p_{\tau}(A)$

$$0.7 e^{(-\tau)}$$

$p_{\tau}(B)$

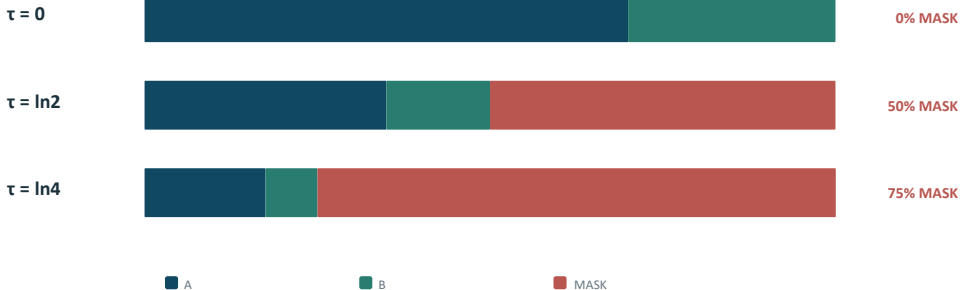
$$0.3 e^{(-\tau)}$$

$p_{\tau}(M)$

$$1 - e^{(-\tau)}$$

원본 토큰 A:B의 70:30 비율은 유지되고, MASK 질량만 점점 커진다.

02 Mass Flow : 같은 forward process를 연속시간 R로 읽는다



A와 B의 질량은 70 : 30 비율을 유지한 채 MASK로 이동한다.

02 Reverse Rate : 역방향 rate에는 데이터 빈도를 반영한 확률이 필요하다

$$\bar{R}_\tau(y, x) = R_\tau(x, y) \times p_\tau(y) / p_\tau(x)$$

MASK $\rightarrow$ A

$$1 \times .175 / .750 = .233$$

MASK $\rightarrow$ B

$$1 \times .075 / .750 = .100$$

문제: 실제 언어모델에서는 $p_\tau(x)$ 가 너무 복잡해서 확률을 직접 계산할 수 없다.

$$s\theta(x, \tau)_y \approx p\tau(y) / p\tau(x)$$

If $x_0 = A$

Forward ratio $q\tau(A|A) / q\tau(M|A) = 1/3$

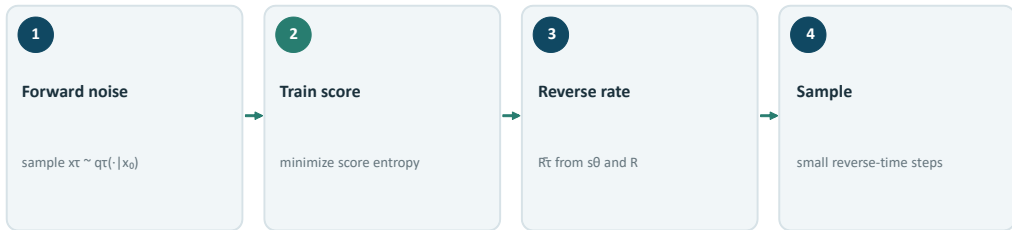
If $x_0 = B$

Forward ratio $q\tau(A|B) / q\tau(M|B) = 0$

$$E[\text{target} \mid x\tau=M] = 0.7 \times 1/3 + 0.3 \times 0 = 0.233$$

SEDD는 $p\tau$ 자체가 아니라 확률비 $p\tau(y) / p\tau(x)$ 를 학습한다.

02 Pipeline : Score를 예측하면 reverse generator가 정해진다



SEDD가 해결한 것: 분포의 값을 모르는 상황에서도 역방향 rate를 학습한다.

03 Absorbing : MASK만 쓰면 복원 후보가 두 종류로 줄어든다

General discrete corruption

$A \rightarrow A, B, C, \dots$

Absorbing MASK corruption

$A \rightarrow A \text{ or MASK}$

상태 공간은 커도 한 위치의 posterior support는 {MASK, original token}뿐이다.

03 Forward : Linear schedule의 t는 위치별 누적 mask 확률이다

t = 0.00

The	cat	sleeps	.
-----	-----	--------	---

t = 0.50

The	[M]	sleeps	[M]
-----	-----	--------	-----

t = 0.75

[M]	[M]	sleeps	[M]
-----	-----	--------	-----

t = 1.00

[M]	[M]	[M]	[M]
-----	-----	-----	-----

Single-position rule

$$P(x_t=M \mid x_0) = t$$

$$P(x_t=x_0 \mid x_0) = 1-t$$

주의: 전체 문장에서 MASK의 개수는 무작위 — 기대 개수만 $L \times t$.

$$q(x_s \mid x_t=M, x_0) = (s/t) \delta_M + ((t-s)/t) \delta_{x_0}$$

KEEP MASK

$$s / t = 2/3$$

REVEAL ORIGINAL

$$(t-s)/t = 1/3$$

예제: $t=0.75 \rightarrow s=0.50$. 이미 MASK인 것 중 1/3을 복원한다.

$$p_{\theta}(x_s | x_t=M) = (s/t) \delta M + ((t-s)/t) p_{\theta}(x_0 | x_t)$$

Rule 1 · No MASK prediction

$p_{\theta}(x_0 | x_t)$ 의 vocab 확률에서 MASK를 0으로 설정

Rule 2 · Carry over

이미 드러난 token은 다음 단계에서도 그대로 유지

완전한 reverse distribution이 clean-token logits + 결정된 mixing weight로 표현된다.

$$KL(q \parallel p_{\theta}) = - ((t-s)/t) \log p_{\theta}(x_0 \mid x_t)$$



$$L = E_t [(1/t) \sum_i 1[x_{t,i}=M] \cdot (-\log p_{\theta}(x_{0,i} \mid x_t))]$$

1/t는 단순한 masked-token 개수 보정이 아니라 연속시간 ELBO에서 나온 시간 가중치.

03 Reverse Step : SEDD와 같은 숫자여도, 여기서 0.233은 확률이다

$$t = .75 \rightarrow s = .50 \quad p_{\theta}(A \mid M) = .70, p_{\theta}(B \mid M) = .30$$

MASK

2 / 3

0.667

A

$(1/3) \times 0.7$

0.233

B

$(1/3) \times 0.3$

0.100

Reverse distribution: $P(M) + P(A) + P(B) = 1.000$

03 Inference : 학습은 한 번의 corruption, 생성은 denoiser의 반복 호출

t=1.00	[M]	[M]	[M]	[M]	[M]
t=0.67	The	[M]	[M]	[M]	.
t=0.33	The	cat	[M]	here	.
t=0.00	The	cat	sleeps	here	.

Step 수 감소 → 덜 계산하지만, 서로 의존하는 토큰을 너무 일찍 확정하면 품질이 떨어질 수 있다.

	D3PM	SEDD	MDLM
Key object	Q_t : transition	R_t : rate	p_θ : clean-token
Forward	$Q_t = \prod Q$	$dp/d\tau = R p$	$\text{Mask}(x_0, t)$
Learning	KL + aux CE	Score entropy	Weighted masked CE
Reverse	Bayes + p_θ	Ratio $\times$ rate	SUBS mixing
Take-away	Which transitions?	Which reverse rates?	How simplify reverse?

세 논문은 같은 absorbing 과정을 전이 행렬, rate, reverse 설계로 다룬다.

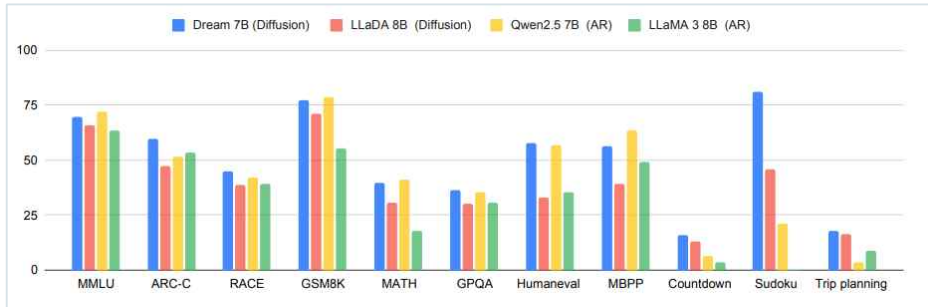
05 Perplexity : LM1B 테스트 PPL 상한 (↓), 규모와 학습량이 다른 항목은 구분

		Parameters	PPL (↓)
<i>Autoregressive</i>	Transformer-X Base [13]	0.46B	23.5
	OmniNet _T [61]	100M	21.5
<i>Diffusion</i>	BERT-Mouth [64] [†]	110M	≤142.89
	D3PM (absorb) [1]	70M	≤76.90
	Diffusion-LM [30] [†]	80M	≤118.62
	DiffusionBert [26]	110M	≤63.78
	SEDD [33] (33B tokens)	110M	≤32.79
<i>Autoregressive (Retrained)</i>	Transformer (33B tokens)	110M	22.32
	Transformer (327B tokens)		20.86
<i>Diffusion (Ours)</i>	MDLM (33B tokens)	110M	≤27.04
	MDLM (327B tokens)		≤ 23.00

출처: MDLM (NeurIPS 2024), LM1B test perplexity · ≤표시는 diffusion 모델의 likelihood bound(상한)

D3PM 70M / SEDD-MDLM 110M · 다른 학습량 · Diffusion 수치는 PPL upper bound

05 7-8B Models : Dream 7B · LLaDA 8B와 AR base model 비교

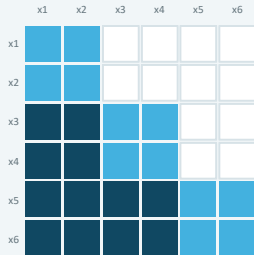


출처: Dream 7B (2025) · Diffusion: Dream 7B, LLaDA 8B / AR: Qwen2.5 7B, LLaMA 3 8B

일부 코드/계획 성능은 강력하지만 학습 토큰 수와 초기화 방법이 달라 인과적 비교는 어렵다.

05 Block Diffusion : 블록 사이는 AR, 블록 안은 diffusion

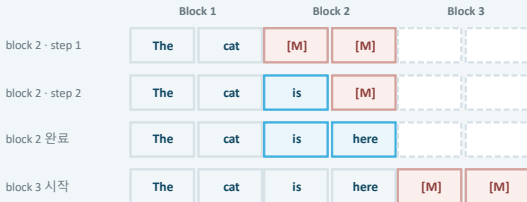
ATTENTION | $L = 6, L' = 2$



■ 블록 안: 양방향 ■ 앞 블록: 참조

행 = query, 열 = key · 뒤 블록은 보지 않는다

GENERATION



$L' = 1$

$1 < L' < L$

$L' = L$

AR

Block Diffusion

MDLM

블록 사이는 AR이라 KV cache와 가변 길이가 가능하고, 블록 안은 병렬로 복원한다.

100B

LLaDA2.0-flash · MoE

- | | | |
|----|-----------------------------|-----------------------|
| 01 | AR pretrained | knowledge inheritance |
| 02 | Diffusion adaptation | block → full → block |
| 03 | Post-training | SFT + DPO |

결론은 “Diffusion이 AR보다 항상 우수하다”가 아니라 frontier-scale 학습·배포의 가능성이다.

LLaDA2.1 · token-to-token

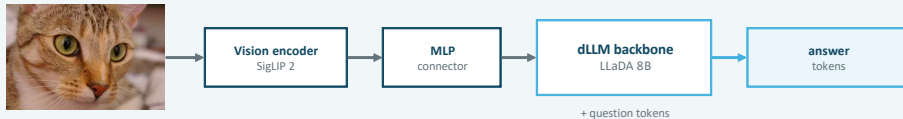
확정했던 토큰을 다시 수정하며 품질과 속도를 조절한다.
HumanEval+ : 논문 보고 892 TPS

LLaDA2.2 · insert / delete

생성 길이까지 수정 가능한 Levenshtein editing.
SWE-bench Verified : 모델 카드 49.28%

이제 연구 대상은 denoiser만이 아니라, 언제 확정하고 언제 다시 수정할지에 관한 decoding policy다.

ARCHITECTURE



TRAINING & GENERATION

학습



생성



조건 (항상 clean) · 답변만 mask하고 복원

이미지와 질문을 조건으로 두고, 답변 텍스트만 masked diffusion으로 생성한다.

05 Latency : Step 수가 줄어도 step당 계산량과 cache 재사용성은 다르다

$$\text{Latency} \approx \text{NFE} \times \text{cost per forward} + \text{overhead}$$

Autoregressive LLM

More calls, but highly effective causal KV cache

Full-attention dLLM

Fewer calls possible, but expensive full-seq denoising

동일 hardware · output length · batch · quality에서 latency / throughput / NFE를 함께 비교.

	Acc. (↑)	Throughput (↑)
<i>Prior work</i>		
SMDM-1B ^{*†}	58.5	-
LLaDa-8B	70.7	-
<i>Ours</i>		
AR-1.7B [†]	62.9	25.6
MDLM-1.7B [†]	58.8	24.6
Eso-LM-1.7B [†]	33.4	25.6
Duo-1.7B [†]	65.8	24.6

GSM8K (↑) · 동일 논문 내 비교 · Duo는 MDLM보다 정확도가 높지만 validation PPL은 불리

출처: Scaling Beyond Masked Diffusion (ICML 2026)

MASK가 항상 최선의 corruption kernel인지는 아직 열린 질문이다.

06 Few-Step : Posterior bridge를 활용한 few-step consistency

Teacher-like path

$x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow \dots \rightarrow x_8$

many network evaluations

Compressed path

$x_1 \xrightarrow{\hspace{10em}} x_8$

fewer evaluations; harder bridge

핵심 평가: 적은 NFE에서 품질을 유지하나? 실제 목표 분포와는 얼마나 일치하나?

06 Reuse : 입력 일부만 바뀌어도 주변 위치의 hidden state가 바뀐다

Step t

The

[M]

$$h(\text{The}) = f(\text{The}, [M])$$

Step s

The

cat

$$h(\text{The}) = f(\text{The}, \text{cat})$$

dLLM-Cache: up to 9.1× fewer FLOPs (LongBench-HotpotQA)

연구 질문: 어떤 layer·token·step에서 KV를 재사용해도 품질이 유지될까?

07 Takeaways : 세 가지 수학적 객체를 연구 문제에 연결한다

D3PM

Q

Design the corruption

SEDD

R

Learn reverse dynamics

MDLM

$p\theta$

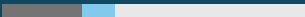
Simplify reverse sampling

Which would you change first: the kernel, the reverse rule, or inference?

Q 는 corruption을, R 은 reverse dynamics를, $p\theta$ 는 sampling을 정한다.



Thank You



MuliLAB Journal Club
인공지능응용학과 허세민

A1 Conventions : 열 = 출발 상태 (논문 원문과 전치 관계일 수 있음)

This seminar · column convention

$$p(\text{next}) = Q p(\text{current})$$
$$R(M, A) = A \rightarrow \text{MASK}$$

D3PM paper · row convention

$$p(\text{next})^T = p(\text{current})^T Q$$
$$Q(A, M) = A \rightarrow \text{MASK}$$

수학적 모델의 차이가 아니라 벡터/행렬의 표기 관례 차이이다.

PAPER · ROW

$$q(x_{t-1} | x_t, x_0) = \text{Cat}\left(x_{t-1}; \frac{x_t Q_t^\top \odot x_0 \bar{Q}_{t-1}}{x_0 \bar{Q}_t x_t^\top}\right)$$

x_t, x_0 는 one-hot 행벡터, $Q_t[i, j] = q(x_t = j | x_{t-1} = i)$

THIS TALK · COLUMN

$$q(x_{t-1} | x_t = j, x_0 = k) = \frac{(Q_t^\top e_j) \odot (\bar{Q}_{t-1} e_k)}{e_j^\top \bar{Q}_t e_k}$$

$Q_t[i, j] = q(x_t = i | x_{t-1} = j), Q_{\text{col}} = Q_{\text{row}}^\top$

CHECK

$$\sum_i Q_t[j, i] \bar{Q}_{t-1}[i, k] = (Q_t \bar{Q}_{t-1})[j, k] = \bar{Q}_t[j, k] \Rightarrow \sum_i q(x_{t-1} = i | \cdot) = 1$$

두 식은 같은 posterior이며, 행렬을 전치한 표기 차이일 뿐이다.

$$P(\text{M at } s \mid \text{M at } t) = P(\text{M at } s) / P(\text{M at } t) = s / t$$

$$P(\text{reveal between } t \text{ and } s \mid \text{M at } t) = (t - s) / t$$

s 시점의 MASK는 t 시점에도 MASK이므로, 유지 확률은 s/t 다.

$$E_{x_0 | x_t=x} [q\tau(y|x_0) / q\tau(x|x_0)] = p\tau(y) / p\tau(x)$$

$s - r \log s$: r 의 조건부 평균에 최적해 $s = E[r]$ (양수 조건)

주의: 개별 샘플의 target이 곧 정확한 주변분포 비율이라는 뜻은 아니다.

$$t = 1 - e^{-\tau} \quad \Leftrightarrow \quad \tau = -\log(1-t)$$

SEDD at $\tau = \ln 4$

MASK $\rightarrow$ A : instantaneous rate 0.233 per unit τ

MDLM $t=.75 \rightarrow s=.50$

MASK $\rightarrow$ A : finite-step probability 0.233

0.233 = 0.233 하지만 의미와 시간 단위는 다르다.

Step $KL = (\Delta t/t) \times CE(\text{masked correct token})$

Σ over steps $\rightarrow \int_0^1 (1/t) CE(\text{masked correct token}) dt$

Estimate with $t \sim \text{Uniform}(0,1)$

MASK 비율 t 와 interval mask 확률 $\beta_{s \rightarrow t} = (t-s)/(1-s)$ 는 서로 다르다.

BERT-style MLM

Masked positions predict tokens
Often fixed corruption recipe
Generation possible through iterative schemes

MDLM

Random continuous masking levels
Explicit reverse kernel (SUBS)
Likelihood-bound weighted CE

차이는 “BERT는 생성 불가능”이 아니라 generative process의 명시적 확률모형이다.

A8 Metrics : PPL 상한, 정확도, TPS, FLOPs는 서로 다른 지표다

PPL ↓

정확한 생성 확률 자체가 아니라 upper bound일 수 있다.

Accuracy ↑

동일 benchmark, prompt, decoding budget인지 확인

TPS ↑

GPU, precision, batch, output length, quality 조건 확인

FLOPs ↓

컴퓨팅 절감과 wall-clock latency 개선은 구분

PPL 상한, 정확도, TPS, FLOPs 감소는 서로 직접 비교할 수 없다.

- | | | |
|---|--|-------------------------------|
| 1 | D3PM (NeurIPS 2021) | Transition matrix & posterior |
| 2 | SEDD (ICML 2024) | Reverse rates & score entropy |
| 3 | MDLM (NeurIPS 2024) | SUBS & weighted masked CE |
| 4 | Dream / LLaDA2.0 | Large-scale capability |
| 5 | Scaling / Consistent / dLLM-Cache | Research directions |

D3PM → SEDD → MDLM 순서로 읽고, frontier 모델과 연구 방향으로 넘어간다.