



Diffusion Language Models

汇报人：陈子康

Outline

- 基础简介
- 扩散语言模型的种类
- 扩散语言模型范式体系
- 扩散语言推理与优化
- 优点与挑战

Outline

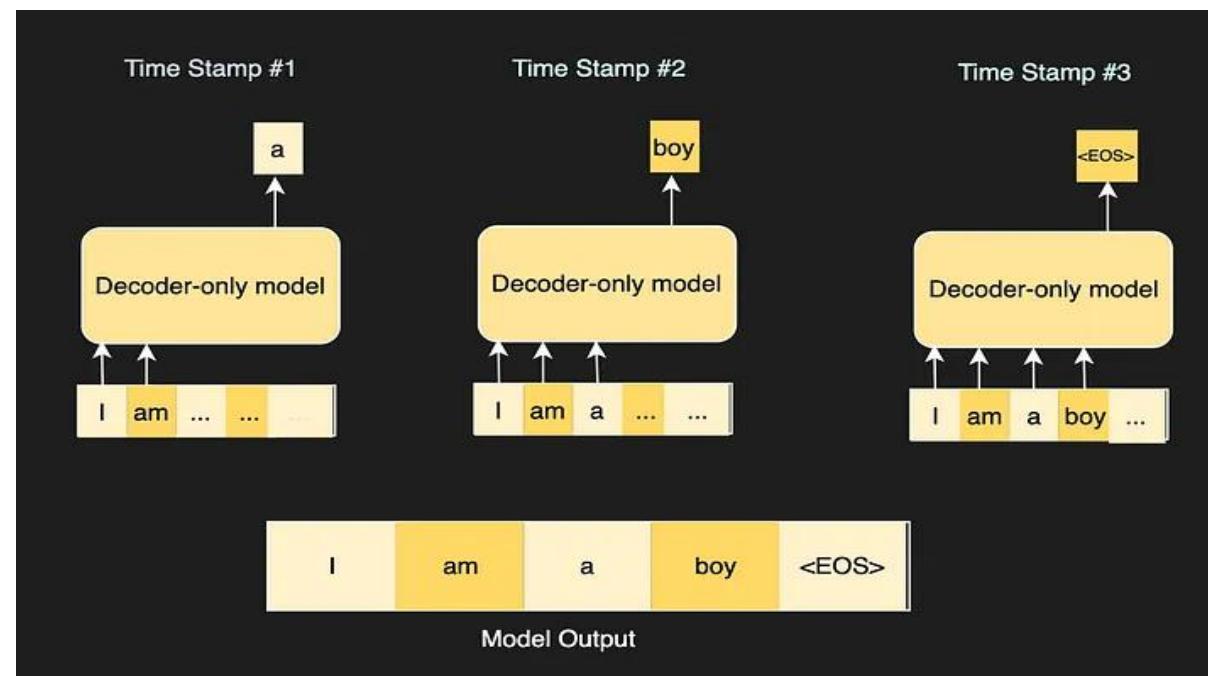
- 基础简介
- 扩散语言模型的种类
- 扩散语言模型训练策略
- 扩散语言推理与优化
- 特性优点与挑战

基础简介

自回归语言模型 (Autoregressive Language Models, AR): 以GPT系列为代表，AR模型通过预测下一个词元来生成文本。其概率计算方式是基于之前已生成的所有词元，采用单向的注意力机制 (Causal Attention)。其训练目标是最大化序列的对数似然:

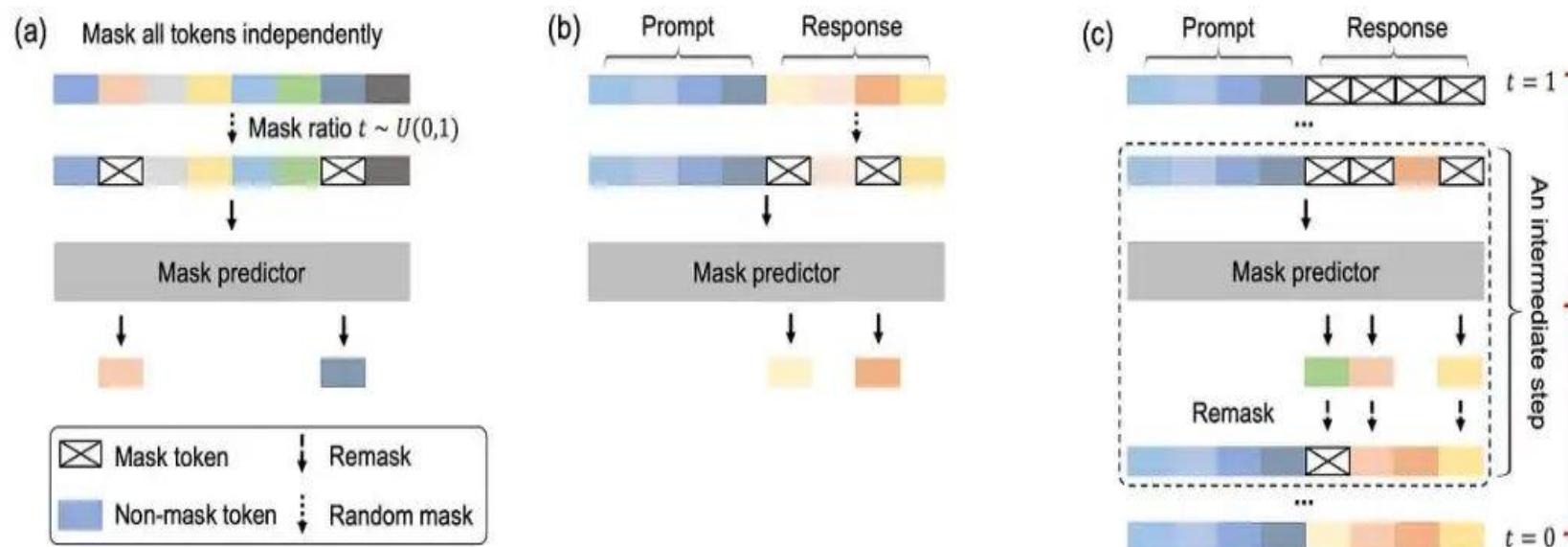
$$\mathcal{L}_{AR} = \mathbb{E}_{X \sim \mathcal{D}} \left[- \sum_{i=1}^n \log P_\theta(x_i | x_1, \dots, x_{i-1}) \right]$$

这种逐字生成的方式虽然保证了文本的连贯性，但也限制了并行计算能力，导致推理速度较慢。



基础简介

扩散语言模型 (DLM) 借鉴了图像生成中的扩散模型 (如DDPM)，通过逐步添加噪声破坏数据，再通过逆向去噪生成文本。与AR模型不同，DLM可并行更新整个序列。

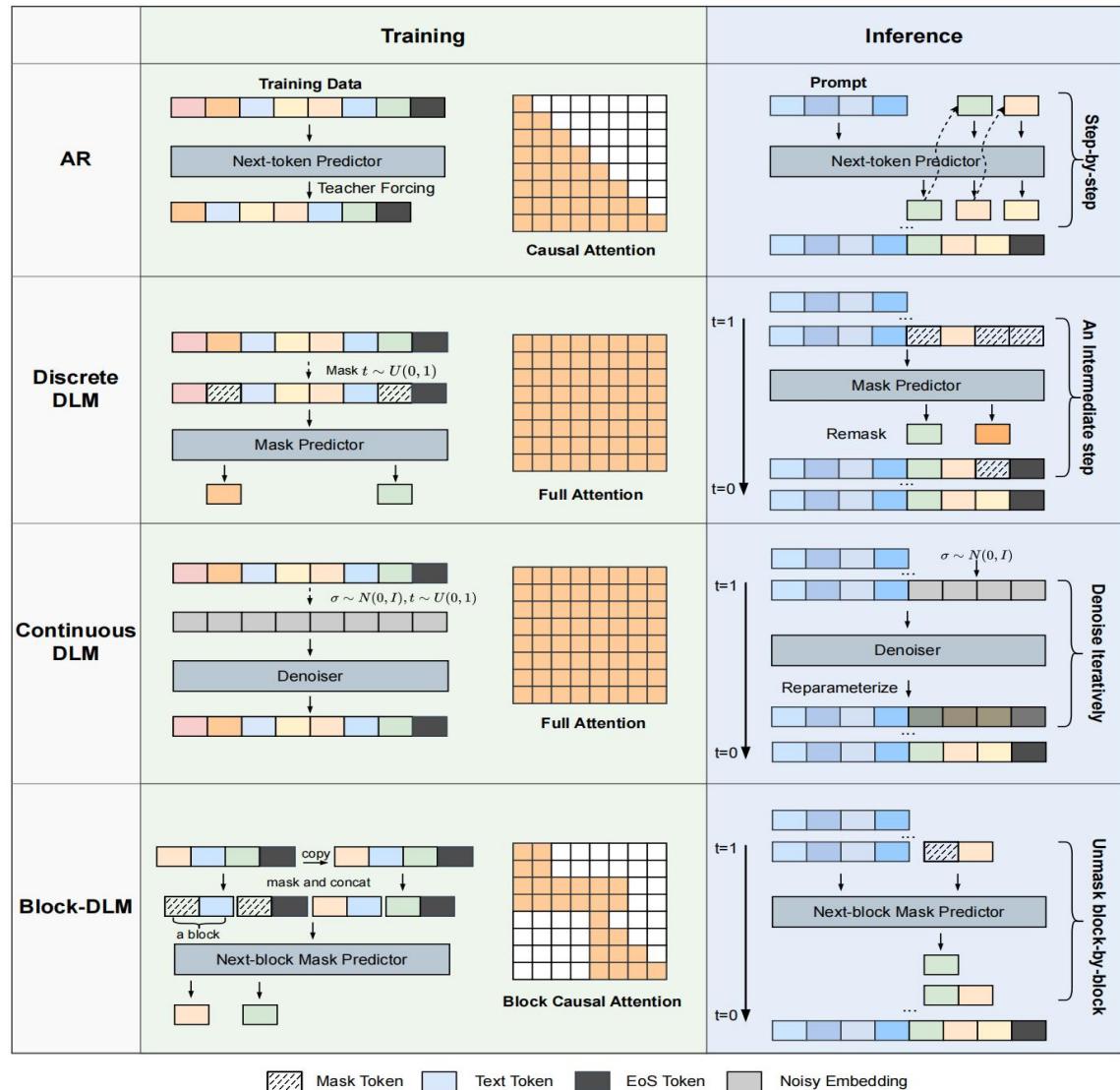


Outline

- 基础简介
- 扩散语言模型的种类
- 扩散语言模型训练策略
- 扩散语言推理与优化
- 特性优点与挑战

扩散语言模型的种类

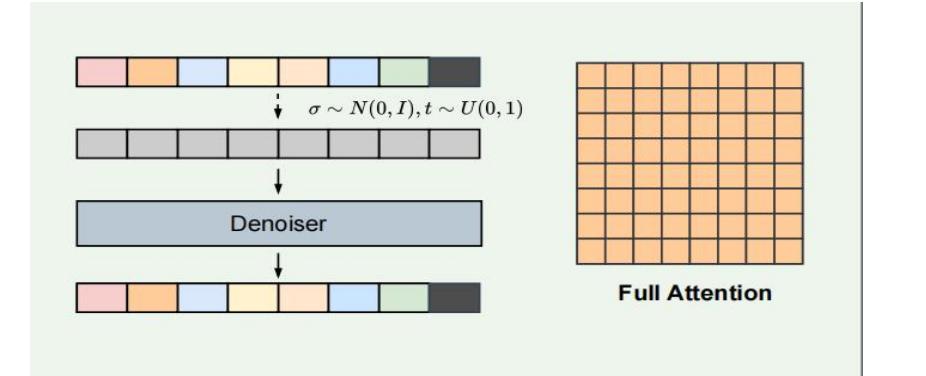
DLMs的核心思想源于非平衡热力学，它学习去逆转一个逐渐将数据变为噪声的“前向过程”。根据扩散过程操作的空间不同，DLMs主要分为连续空间和离散空间两大类。



扩散语言模型的种类

1. 连续空间扩散语言模型 (Continuous DLM)

连续空间DLM首先将离散的文本词元映射到连续的嵌入向量空间，然后在这个连续空间中执行扩散和去噪过程。



前向过程：也称为加噪过程。它通过一个预定义的马尔可夫链，在T个时间步内，逐步将原始数据词嵌入向量 x_0 转化为纯高斯噪声。在许多实现中，任意时刻t的加噪样本 x_t 可以直接通过 x_0 计算得出：

$$x_t = \alpha_t x_0 + b_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

ϵ ：从标准正态分布（高斯噪声）中采样的噪声。 α_t 和 b_t ：与时间步 t 相关的确定性函数，它们共同定义了噪声调度，即在每一步加入多少噪声以及保留多少原始信号。

反向过程：也称为去噪过程。模型学习如何逆转上述加噪过程。它从一个纯噪声 x_t 开始，通过迭代地去除噪声，最终恢复出接近原始数据 x_0 的清晰样本。模型的训练目标通常是最小化其预测值与真实值之间的差距，例如，预测每一步加入的噪声 ϵ 或原始数据 x_0 。一个简化的训练目标函数如下：

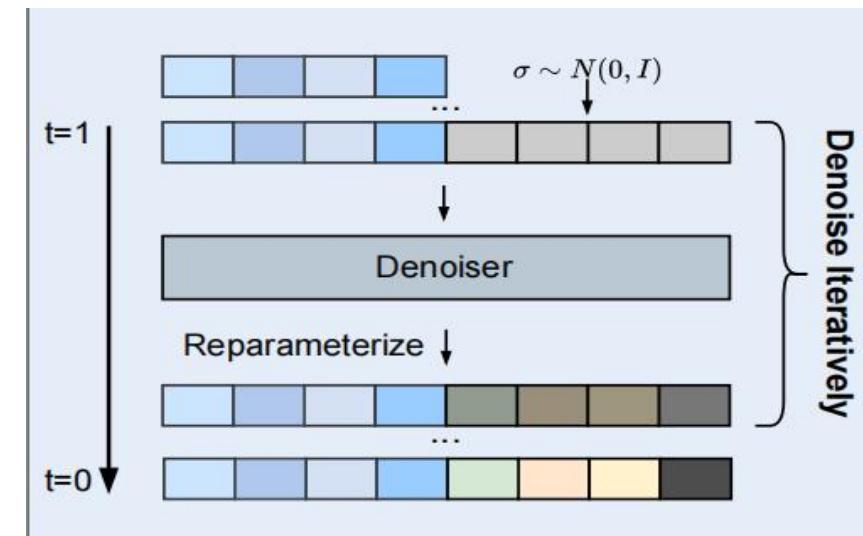
$$\mathcal{L}_{simple} = \mathbb{E}_{t, x_0, z} [||f_\theta(x_t, t) - z||^2]$$

模型需要在所有时间步 t 和所有数据样本 x_0 上，尽可能准确地预测出目标 z 。可以是噪声 ϵ 、原始数据 x_0 或是其他相关量

扩散语言模型的种类

1. 连续空间扩散语言模型 (Continuous DLM)

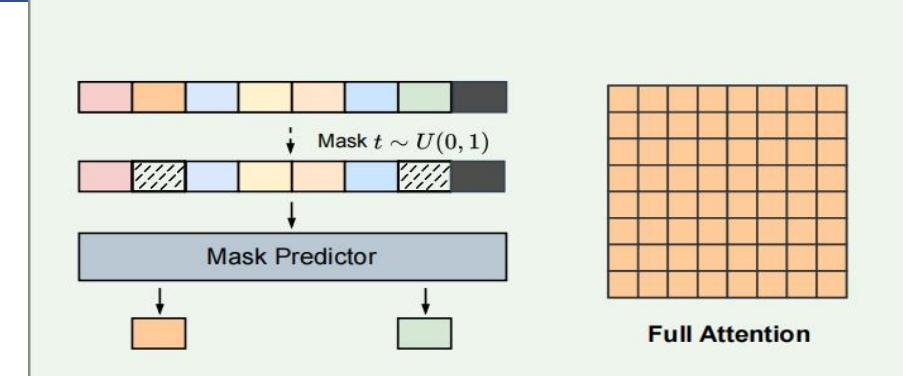
生成文本时，模型从一个随机噪声向量开始，迭代T步，每一步都进行一次去噪预测，直到最终得到一个清晰的嵌入向量，再通过最近邻搜索等方式将其映射回离散的词元。



扩散语言模型的种类

2. 离散空间扩散语言模型 (Discrete DLM)

离散空间DLM直接在词元表上定义扩散过程，避免了连续嵌入空间和最终的四舍五入步骤。



前向过程：此过程通过一个转移矩阵来逐步遮盖原始词元序列 x_0 。一个常见的做法是将词元以一定概率替换为特殊的`[MASK]`词元。在 t 时刻，序列状态 x_t 的概率由以下类别分布给出：

$$q(x_t|x_0) = \text{Cat}(x_t; p = x_0 \bar{Q}_t)$$

x_0 ：原始的one-hot编码的词元序列。 \bar{Q}_t ：从时间步1到 t 的累积转移矩阵。从原始序列 x_0 出发，经过 t 步掩盖后，得到序列 x_t 的概率。

反向过程：模型学习预测在给定掩码后序列 x_t 的情况下，原始词元 x_0 的概率分布。其训练目标被简化为一个仅在被掩码位置计算的交叉熵损失函数：

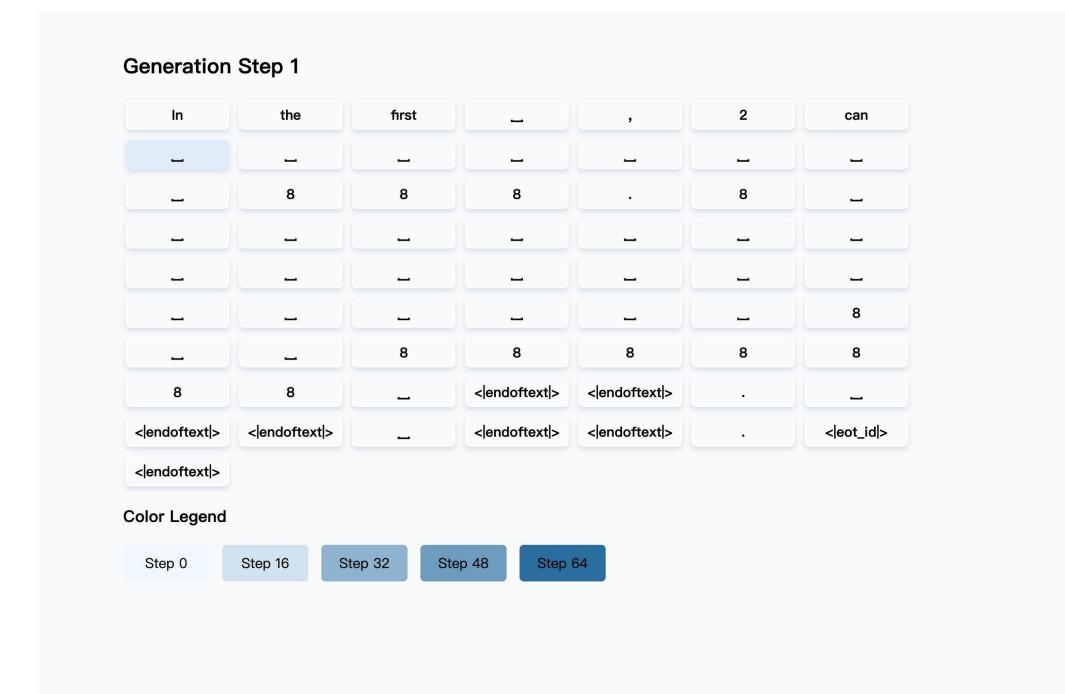
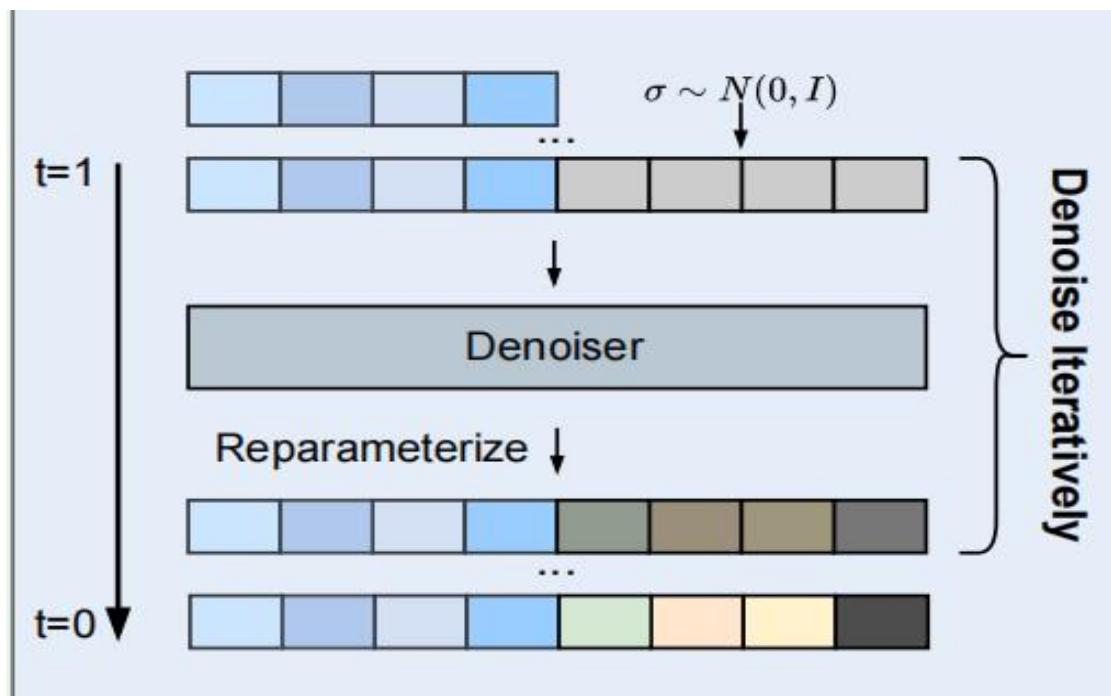
$$\mathcal{L}(\theta) = \mathbb{E}_{t,x_0,x_t} \left[-\frac{1}{|M|} \sum_{i \in M} \log p_\theta(x_{0,i}|x_t)_i \right]$$

M 是被掩码的位置集合。 $p_\theta(x_{0,i}|x_t)$ 是模型在给定掩码后序列 x_t 的情况下，预测第 i 个位置的原始词元为 $x_{0,i}$ 的概率。

扩散语言模型的种类

2. 离散空间扩散语言模型 (Discrete DLM)

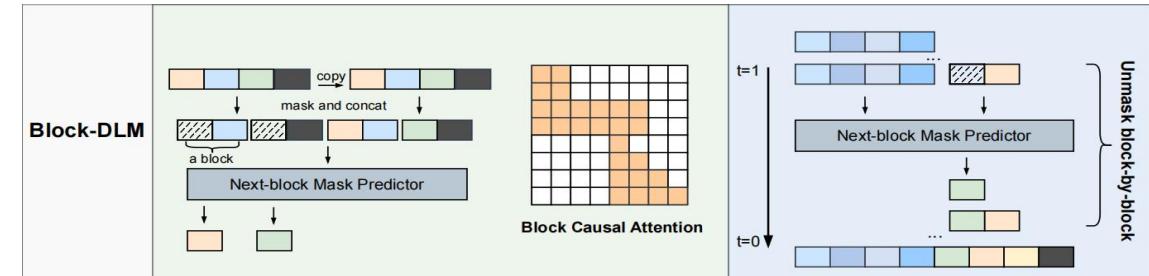
在推理时，模型从一个完全由 [MASK] 组成的序列开始，在每一步迭代中，模型预测整个序列，然后根据预测的置信度和噪声调度，将一部分置信度最高的 [MASK] 替换为预测的词元（这个过程称为“unmasking”），其余位置则重新设为 [MASK]（“remasking”），如此循环直到所有 [MASK] 都被替换。



扩散语言模型的种类

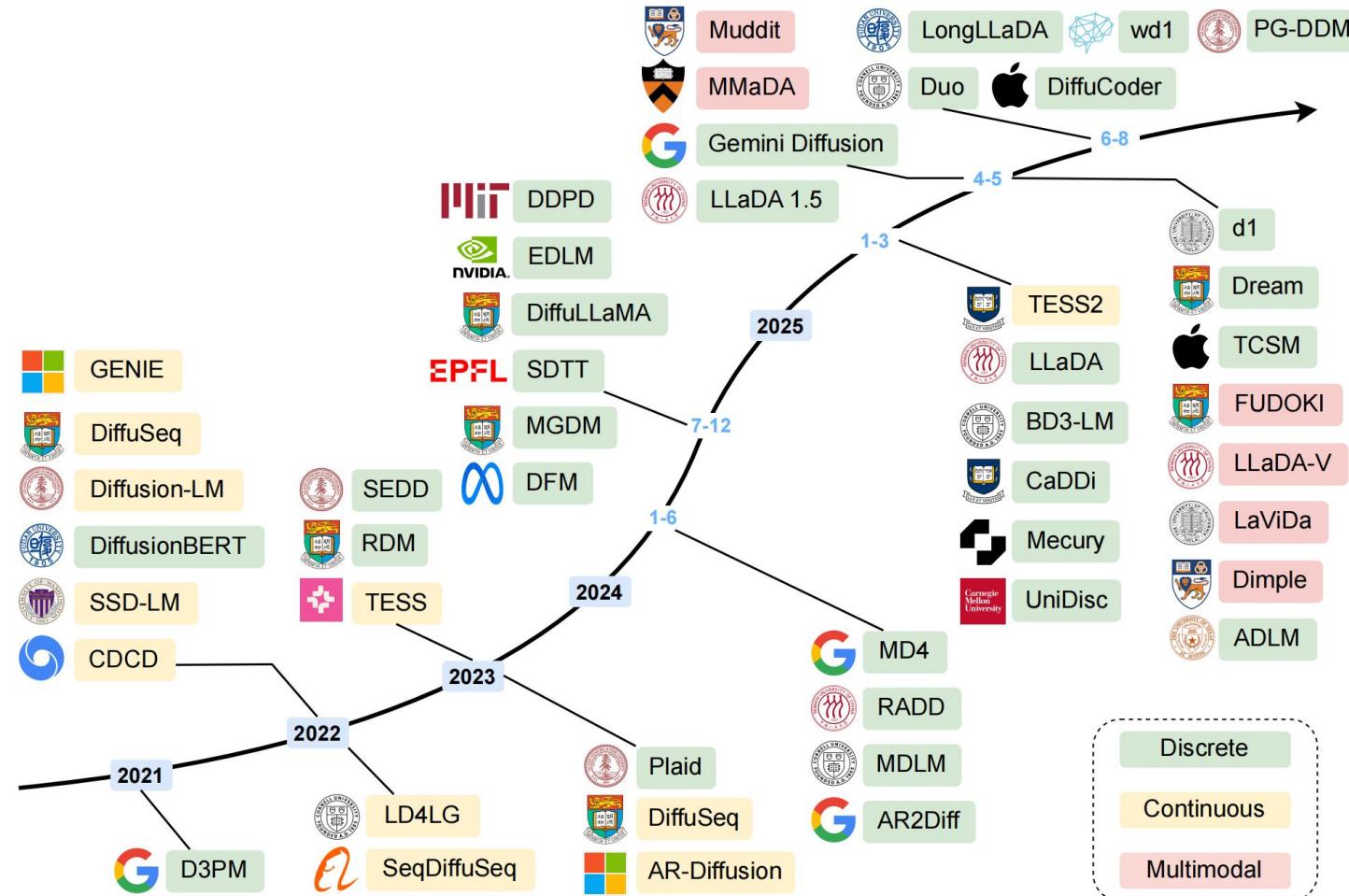
3. 块状扩散语言模型 (Block-DLM)

这类模型试图结合自回归模型和扩散模型的优点，在并行性与强大的时序依赖建模能力之间取得平衡。一个典型的策略是采用块状半自回归生成方式。



模型以自回归的方式生成一个个词元块 (block)，而在每个块内部，词元则是通过扩散过程并行生成的。

扩散语言模型的种类



扩散语言模型的种类

Model	Parameters	Diffusion type	Noise schedule	Task	Training data
D3PM [24]	70M	Discrete	Mutual information	Language	65B tokens
Diffusion-LM [22]	100M & 300M	Continuous	Square-root	Language	-
Diffuseq [50]	91M	Continuous	Square-root	Language	565K sentence pairs
SSD-LM [74]	400M	Continuous	Cosine	Language	123B tokens
DiffusionBERT [25]	110M	Discrete	Spindle	Language	16B tokens
CDCD [51]	1.3B	Continuous	-	Language	315B tokens
LD4LG [63]	188M	Continuous	Cosine	Language	5.2M sentence pairs
SeqDiffuSeq [140]	65M & 110M	Continuous	Adaptive	Language	45B tokens
TESS [58]	125M & 355M	Continuous	Linear	Language	-
MDLM [63]	110M	Discrete	Log-linear	Language	622B tokens
DFM [69]	1.7B	Discrete	Linear & Cubic	Language & Code	2.5T tokens
TESS-2 [59]	7B	Continuous	Log-linear	Language	360B tokens
LLaDA [28]	1B & 8B	Discrete	Linear	Language & Code	2.3T tokens
Mecury [32]	-	Discrete	Log-linear	Code	Trillions tokens
LLaDA-1.5 [85]	8B	Discrete	Linear	Language	2.3T tokens
MMADA [31]	8B	Discrete	Log-linear	Language	900B image-text tokens
Dream [26]	7B	Discrete	Log-linear	Language & Code	580B tokens
LLaDA-V [29]	8.4B	Discrete	Linear	Multimodal	3M image-text samples
LaViDa [96]	8.4B	Discrete	Convex	Multimodal	1.6M image-text samples
Dimple [30]	7B	Discrete	Log-linear	Multimodal	0.8B tokens
LongLLaDA [73]	8B	Discrete	Log-linear	Language & Code	2.3T tokens
DiffuCoder [84]	7B	Discrete	Log-linear	Code	130B tokens

Outline

- 基础简介
- 扩散语言模型的种类
- **扩散语言模型训练策略**
- 扩散语言推理与优化
- 特性优点与挑战

扩散语言模型训练策略

与自回归语言模型近似DLMs的训练过程可以分为两个主要阶段：预训练（Pre-training）和后训练（Post-training）
预训练学习把被逐步「加噪」的文本恢复成干净文本；离散 DLM 在词元空间做掩码/还原，连续 DLM 在向量空间做噪声/去噪。

1. 预训练常见三条路径

- (1) 从 AR LLM 迁移初始化：把已训练好的自回归模型（如 LLaMA、Qwen 等）权重拿来做 DLM 的起点，显著节省算力与时间；如 DiffuLLaMA / DiffuGPT / Dream-7B 都走这条路线。
- (2) 从图像扩散骨干迁移（多模态）：如 D-DiT、Mudit 直接用预训练的 MM-DiT/SD3 系列作视觉分支/骨干，再联合训练文本分支。
- (3) 从零训练：如 LLaDA- 8B，直接用离散扩散目标端到端预训练，验证了从头规模化的可行性。

监督微调目的是像 AR 模型那样在指令/对话/任务数据上做有监督学习，只是要尊重扩散/掩码的范式。

离散 DLM（如 LLaDA/Dream）：保留 prompt（不掩码），在响应段按比率掩码，只对被掩码的响应 token 计交叉熵；这样模型学会「看条件、补答案」。

连续 DLM（如 TESS2）：只对响应段嵌入加噪并去噪，目标同样是条件生成。

扩散语言模型训练策略

2. 后训练 (Post-training) - 激发推理能力

为了提升 DLM 在数学推理等复杂任务上的能力，研究者们正在探索将强化学习 (RL) 等后训练技术应用于 DLM。扩散语言模型在训练方法上存在一个关键问题：其训练目标（通常是随机遮盖部分文本并让模型预测）与模型实际生成文本时的推理路径（一步步解码的过程）不匹配。这种不匹配限制了模型的优化效果。

TraceRL：

核心思想是让模型的训练过程与其自然的推理轨迹对齐。简单来说，它不再仅仅根据最终生成的答案好坏给予奖励或惩罚，而是关注并优化整个生成答案的“思考路径”或“轨迹”。

生成与记录轨迹：在训练时，首先让被训练的语言模型针对一个任务（如数学题）生成一个完整的解答。这个生成过程不是一次性完成的，而是通过多个解码步骤来完成的，每一步都会解码出一些新的词元 (tokens)。这些步骤的集合就构成了轨迹。

奖励整个轨迹：在得到最终答案后，会有一个外部的奖励机制（例如，判断数学题答案是否正确）来给出一个总分。TraceRL将这个分数作为对整个生成轨迹的奖励或惩罚，然后用这个信号来更新模型的参数。这样一来，模型学习到的就不仅仅是“什么答案是好的”，更是“通过什么样的步骤可以得到好答案”。

Outline

- 基础简介
- 扩散语言模型的种类
- 扩散语言模型训练策略
- 扩散语言推理与优化
- 特性优点与挑战

扩散语言推理与优化

1. KV Cache

由于扩散语言模型双向注意力机制的特性，其无法利用自回归语言模型的KV Cache做法。
然而诸如Fast-DLLM 等方法想到利用近似 KV-Cache这一想法

Fast-DLLM通过实验结果表明，虽然掩码扩散语言模型 (MDLLM) 在前向计算时，理论上需要让每个token对输入序列上所有位置的 (Key / Value) 进行attn，从而导致其K和V每次forward都需要集体更新，但相邻去噪步中同一位置的 K、V 向量余弦相似度是比较高的（他们仅测了LLaDA和Dream这两个开源模型）。

据此，即便严格意义上的 KV-Cache 并不适用于 MDLLM，仍可借助这一高相似性，在局部时间窗口（如同一 block 内的连续denoising步）内实施一种近似KV-Cache机制，从而减少重复计算。

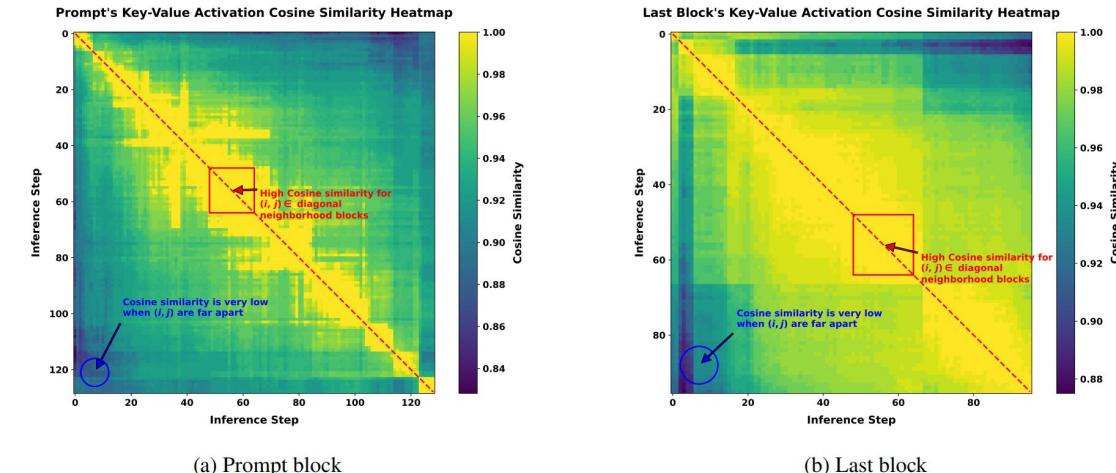


Figure 3 | Heatmaps of Key-Value Activation Cosine Similarity Across Inference Steps in LLaDA-Instruct. (a) Cosine similarity heatmap for the prompt block, averaged over all prompt tokens. (b) Cosine similarity heatmap for the last block, averaged over all tokens in the last block (used to represent suffix tokens, as the last block always belongs to the suffix before its own decoding). In both (a) and (b), high similarity is observed near the diagonal ($i \approx j$), indicating that Key-Value activations at adjacent inference steps within a block are highly similar. The red boxed regions highlight this effect, supporting the use of an approximate block-wise KV Cache: cached activations from previous steps can be safely reused during block decoding with minimal loss in accuracy. The DualCache strategy, which additionally caches suffix tokens, further demonstrates negligible differences in activations during block decoding, enabling greater acceleration with competitive accuracy.

扩散语言推理与优化

由于他们是直接在LLaDA和Dream上apply的，所以为了保证训练和inference之间的一致性，在每次forward时，Fast-DLLM依然是需要将prompt和所有的blocks都送入model的。这也使得他们进一步propose了dual caching，对应的普通的KV-caching可以看作是prefix caching，因为只存储当前block之前的K和V，而dual caching则是由于fast-dllm每次都需要将prompt和所有的blocks都送入model，所以对于当前处理的block之后的blocks，也做了caching。

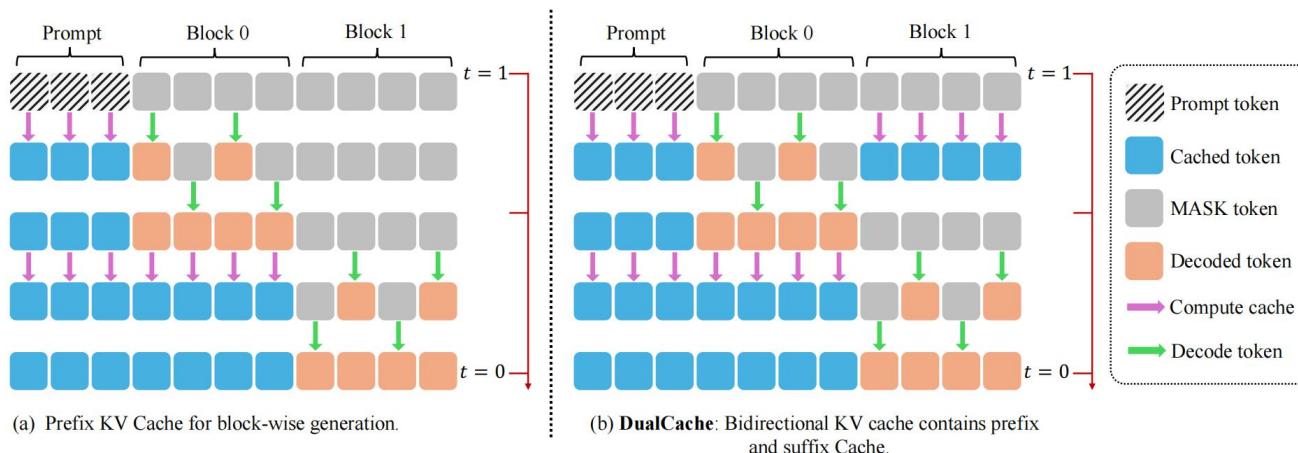
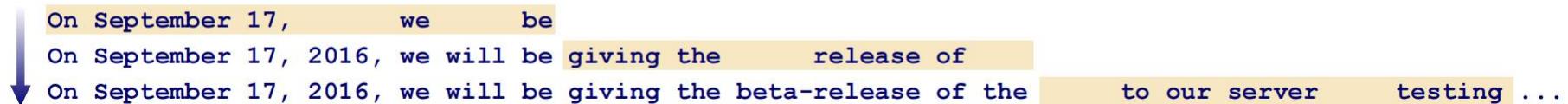


Figure 2 | **Illustration of our Key-Value Cache for Block-Wise Decoding.** (a) During prefix-only caching, the KV cache is computed once for the prompt and reused across multiple decoding steps within each block. The cache is updated after completing a block to maintain consistency, with negligible overhead. (b) DualCache extends this approach by caching both prefix and masked suffix tokens, further accelerating decoding. The high similarity of KV activations across steps allows effective reuse with minimal approximation error.

扩散语言推理与优化

Block Diffusion (Ours): ✓ High quality ✓ Arbitrary-length ✓ KV caching ✓ Parallelizable

1. KV Cache

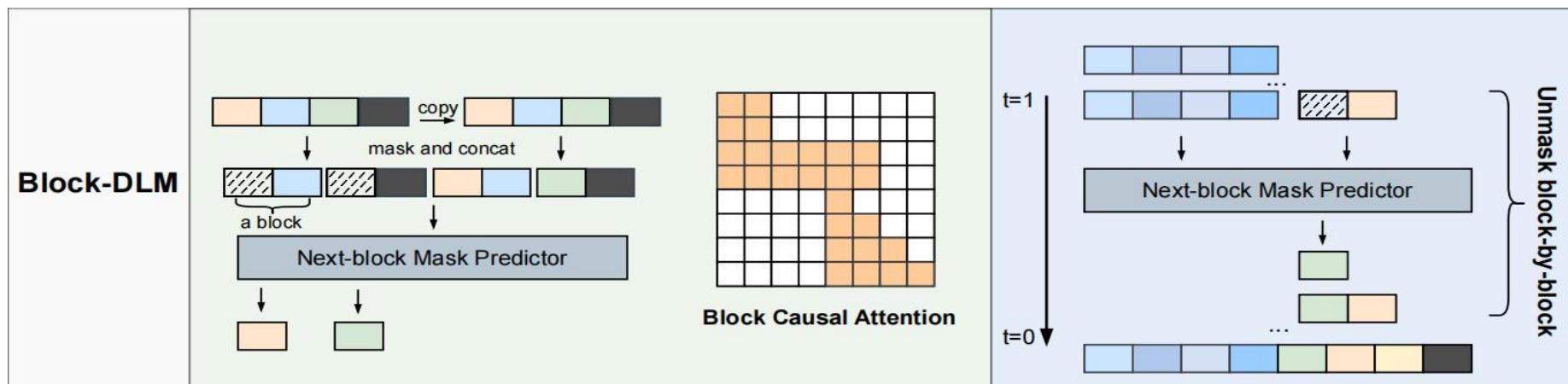


BD3LM: 基于Block-DLM下的一致性更高的 KV-Cache

在训练阶段引入 半因果掩码 (semi-causal mask) , 使 MDLLM 天然支持 Semi-KV 缓存；但模型行为也随之向自回归 (AR) 范式靠拢。

相较 Fast-DLLM，本方法：使用KV-caching采样和不使用KV-caching采样对于BD3LM而言两者是统一的，不存在近似；推断时仅需处理 当前 block 及其所有前置 block，不必再将完整输入序列送入模型，计算量更低。

主要劣势是在换推理时的block size时必须重新训练或至少微调现有MDLLM，而 Fast-DLLM 可直接作用于已训练完毕的模型，无需额外调整。



扩散语言推理与优化

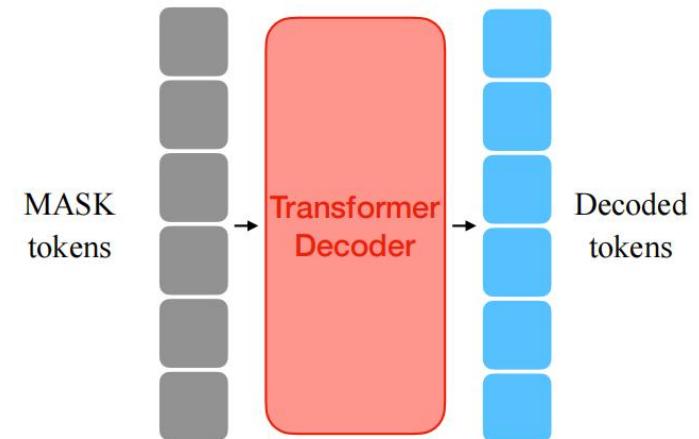
2. 并行解码

并行解码旨在利用DLM的并行生成能力，使其不是按照规定的step解码一定数目

置信度感知解码 (Confidence-aware Decoding): 如Fast-dLLM，该方法只揭示那些预测概率超过特定阈值的词元，从而在不牺牲质量的情况下实现高达27.6倍的加速。

自适应并行解码 (Adaptive Parallel Decoding, APD): 通过一个轻量级的AR辅助模型动态调整并行度，在需要时牺牲吞吐量以换取更高的保真度。

推测性扩散 (Speculative Diffusion, SpecDiff): 使用一个并行的DLM作为“起草者”快速生成候选文本，然后由一个更大的AR模型进行验证和修正，实现高效生成。



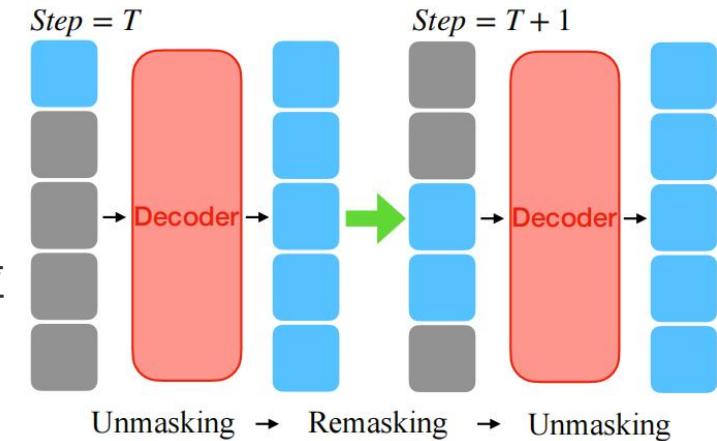
扩散语言推理与优化

3. 揭示/重掩码

离散扩散语言模型常用“mask-predict”范式：长度固定、初始全是 [MASK]。每个迭代步 t ，模型对所有位置给出分布，揭示（unmask）高置信度位置、保留或重掩（remask）低置信度位置，迭代到没有 [MASK] 为止。

置信度排序/阈值：按最大类概率从高到低揭示；或设全局阈值 p^* ，高于 p^* 的位置一次性定稿。（Fast-dLLM 用阈值法带来数量级加速。）

重掩已生成：为进一步精修，允许把已揭示但置信度下降的 token 再度重掩，换取质量一算力可调的“推理时缩放”（ReMDM）。



慢一快两相：先小步谨慎揭示稳定 token，再快速批量定稿（SlowFast）。

扩散语言推理与优化

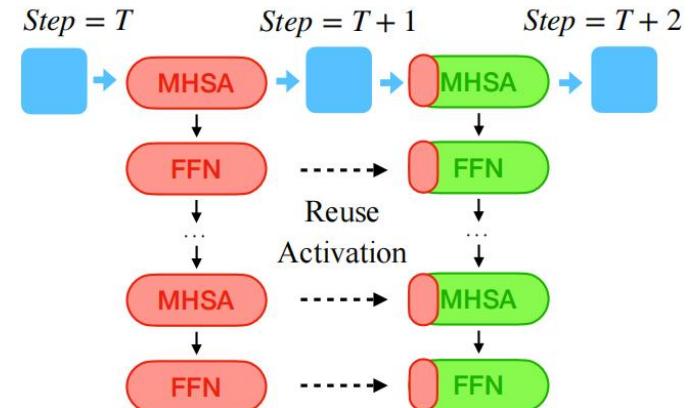
4. 特征缓存

扩散迭代相邻步之间中间表示（层激活）高度相似。特征缓存直接复用这些中间激活（不只 KV），在下一步做校验/小修，从而少算大量 FFN/MHSA 前向。与 AR 的 KV-Cache 不同，Feature Cache 覆盖更广，可缓存整层/子层输出，既可用于图像扩散，也能迁移到文本 DLM。

文本 DLM 中的两类冗余：

提示段 (prompt) 几乎静态：可以长间隔刷新；

回复段 (response) 稀疏变化：按位做轻量相似度验证（如 V-verify on values），只有“变了很多”的位置才重算局部激活。实践里带来最高 $\approx 9\times$ 的端到端加速。



Outline

- 基础简介
- 扩散语言模型的种类
- 扩散语言模型训练策略
- 扩散语言推理与优化
- 特性优点与挑战

特性优点与挑战

一、多模态与综合能力

1. Overall-GenEval: 多模态生成综合评估 (文本 + 图像)
2. MME (Multimodal Large Language Model Evaluation):
多模态大模型综合基准, 含 14 个子任务 (如 VQA、图像描述)。

二、推理与常识

1. CQA (Consumer Question Answering: 真实场景问答, 尤其医疗领域 (如波斯语医疗论坛数据))。
2. Hellaswag: 情境推理 (如 “切菜时刀滑落” 的后续事件), 对抗性干扰项设计。
3. PIQA (Physical Interaction: Question Answering) : 物理常识 (如 “如何用胶带修补杯子”), 需理解材料属性和因果关系。

三、专业领域与代码

1. HumanEvalPython 代码生成 (164 题), 严格单元测试评估 (Pass@k 指标)。
2. GSM8K小学数学多步应用题

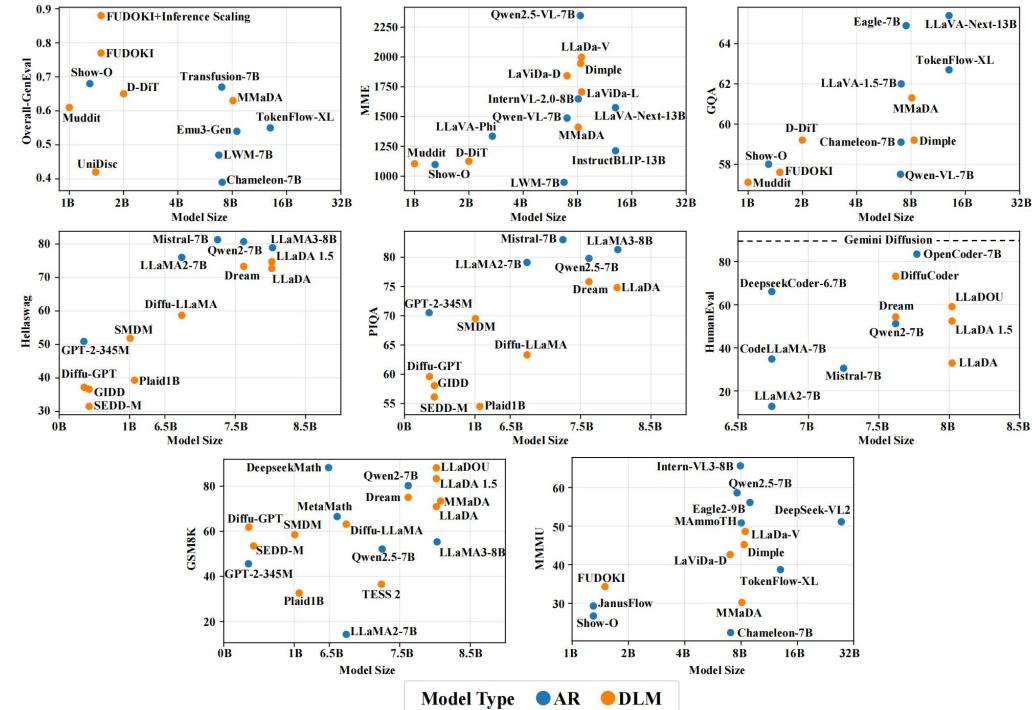


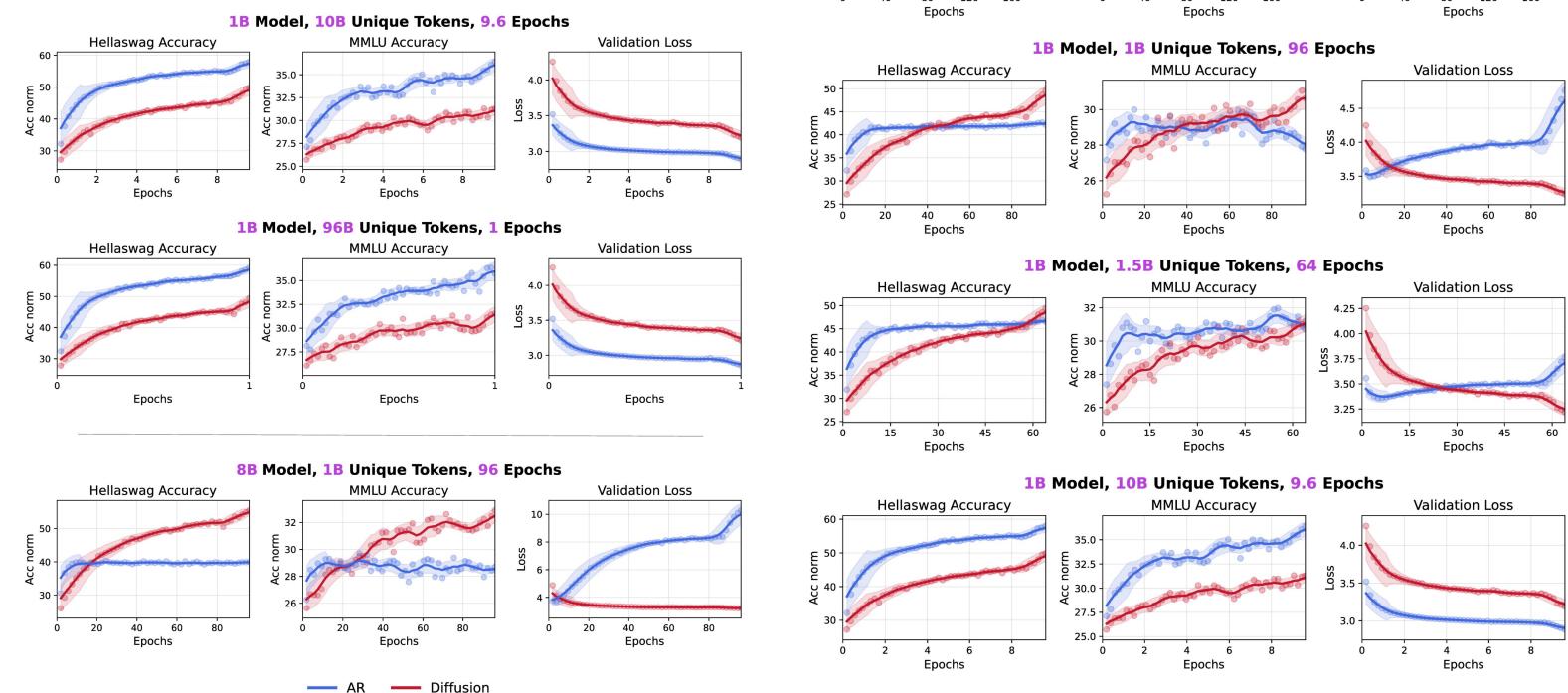
Fig. 6. Performance comparison on eight benchmarks: Overall-GenEval, MME, CQA, Hellaswag, PIQA, HumanEval, GSM8K, and MMMU. The horizontal axis in each subplot represents the model size, measured in the number of parameters. The vertical axis indicates the score under the corresponding benchmark, with higher scores reflecting better performance. Model types are distinguished by color: blue represents AR language models, while orange represents DLMs.

特性优点与挑战

DLM对于训练的数据要求特征：

1. 在数据是严重bound时，能通过重复有限的数据看到DLM超越ARM的那个点。
2. DLM比ARM有更高的数据上限。
3. 1b DLM模型仅用1b的无重复token训练时，可以达到HellaSwag 56%，MMLU 33%。数据从nemotron cc随机sample的，没有任何trick。

基本上一直重复训练DLM会一直有能力增长。



特性优点与挑战

DLM优点:

1. 并行生成 (吞吐高)

DLM 通过“迭代去噪”一次可并行生成多处位置，相比 AR 的逐 token 生成有天然并行性与更高吞吐。

2. 双向上下文 (表征更丰富)

训练/解码都用全局注意力，天然利用左右文，得到更细腻的上下文理解与可用于控制的表征

3. 迭代细化 (质量一算力可调)

“先定高置信区域、保留不确定区域继续打磨”的反复修正，可以逐步提升低置信度片段的连贯性与正确性（对应 Masked DLM 的揭示/重掩）。

4. 可控性强

可对“位置/结构”施加约束，用指导类技术（如无分类器指导）在解码时调风格、相关性等

5. 统一多模态的自然载体

在统一的去噪框架下同时建模文本与视觉等模态，便于“理解 + 生成”一体化

特性优点与挑战

DLM挑战:

1. 并行一质量权衡（“并行解码诅咒”）

一次性揭示过多位置会忽略位置间相关性，易造成不一致或错误累积；步数过少时现象更明显。假设我们要模型填补 “The list of poker hands that consist of two English words are: __, __.” 这两个空。正确的答案可能是 "high card", "two pairs", "full house" 等。注意，这两个词之间是高度相关的。但MDM在并行预测时，会为每个[MASK]位置生成一个概率分布，然后独立采样。这就可能导致它在一个位置预测出 "high"，在另一个位置预测出 "house"，组合起来就成了 "high house" —— 一个完全错误的答案。

2. 长序列与动态长度生成不友好

许多 DLM 以固定长度去噪训练，推理时也要对整段进行多步全局更新：若每步仍是全局注意力，复杂度可达 $O(N^3)$ ，且需要预设长度（即使提前生成了 [EOS]，剩余位置仍被更新浪费算力）。

3. 规模化验证不足

公开 DLM 最大多在8B量级，尚缺乏类似顶级 AR 模型那样的超大规模验证；不少工作依赖从 AR 继续训练或在有限数据上构建派生模型。

4. 训练效率与训推偏差

在常规掩码/时间步均匀采样下，平均仅~50% token 参与损失，关键答案 token 可能被漏掉；同时，训练时噪声/掩码分布与推理时的揭示节奏不一致，也会导致训推偏差。

思考

能否用到我们的场景？

1. DLM高速并行吞吐且可控生成质量适合一些对任务精确度不高，计算量小且要求时延高的场景上，比如做草稿模型。
- 2.DLM其生成长度带来的延时增加并不剧烈，延时主要依据扩散步骤T，可以考虑长文本，时延要求高的工作。