
Wide-In, Narrow-Out: Revokable Decoding for Efficient and Effective DLLMs

Feng Hong^{1,*} Geng Yu^{1,*} Yushi Ye¹ Haicheng Huang¹
Huangjie Zheng² Ya Zhang³ Yanfeng Wang³ Jiangchao Yao¹
¹Cooperative Medianet Innovation Center, Shanghai Jiao Tong University
²Apple ³School of Artificial Intelligence, Shanghai Jiao Tong University
{ feng.hong, Sunarker } @sjtu.edu.cn

Abstract

扩散大语言模型 (DLLMs) 作为自回归模型的一个引人注目的替代方案出现, 旨在实现快速的并行生成。然而, 现有的 DLLMs 受到严重的质量与速度权衡的困扰, 其中更快的并行解码导致显著的性能下降。我们将此归因于标准解码的不可逆性。为了解决这个问题, 我们引入了 Wide-In, Narrow-Out (WINO), 这是一种无需训练的解码算法, 可以实现可撤销解码。WINO 采用一种并行草稿验证机制, 积极地起草多个标记, 同时利用模型的双向上下文验证并重新掩盖可疑标记以进行改进。在像 LLaDA 和 MMDA 这样的开源 DLLMs 中经过验证, WINO 被证明可以决定性地改善质量与速度之间的权衡。例如, 在 GSM8K 数学基准测试中, 它将推理加速了 6 倍, 同时准确性提高了 2.58 个百分点; 在 Flickr30K 描述生成中, 它实现了 10 倍的加速同时提高了性能。More comprehensive experiments are conducted to demonstrate the superiority and provide an in-depth understanding of WINO. ¹

1 介绍

自回归 (AR) 大语言模型 (Radford et al., 2018, 2019), 如 GPT 系列 (OpenAI, 2022), have shown impressive performance in a ranging of language tasks。然而, 它们基础的逐个标记生成机制引入了固有的限制, 包括严重的推理延迟、易于错误传播 (Stechly et al., 2023; Valmeekam et al., 2023), 以及在保持全局一致性方面的挑战 (Mei et al., 2025)。作为回应, 扩散大语言模型 (DLLMs) 成为一种引人注目的非自回归替代方案, 旨在克服这些瓶颈。通过同时生成标记 (Li et al., 2022), DLLMs 理论上实现了大规模推理加速, 同时其原生的双向注意力提供了更好的一致性。DLLMs 的巨大潜力被专有的闭源系统展示出来 (例如水星编码器 (Inception Labs, 2025) 和双子扩散 (Google DeepMind, 2025)), 这些系统展示了惊人的速度, 超过每秒 1,000 个标记, 作为强有力的概念验证。

尽管有这个承诺, 开源 DLLM 的性能仍然令人失望。One critical bottleneck is that 它们陷入了一种严重的质量-速度权衡困境。Specifically, 为了实现高质量的输出, 这些模型通常被迫缓慢解码, 一次只生成一个标记, 这否定了它们主要的架构优势。试图通过并行生成多个标记来加速推理不可避免地导致输出质量显著下降 (Nie et al., 2025)。这种明显的权衡在很大程度上阻碍了 the open-source DLLM 作为其 AR 对手的可行高性能替代品。

我们将这种权衡与 DLLMs (Ou et al., 2025; Sahoo et al., 2024) 的一个基本缺陷联系起来: 它的不可逆性 of the standard decoding process。Specifically, the standard generation in diffusion steps 通常以一系列 [] 标记开始, 然后以贪婪的方式进行填充。一旦一个标记被解码, 该决定就是最终的, 无法修改, 即使在后续步骤中提供了更多的 informative 上下文也是如此。然而, 这对并行解码是 challenging, 其中初始标记是在信息非常有限的情况下生成的,

* Equal contribution

¹代码: <https://github.com/Feng-Hong/WINO-DLLM>

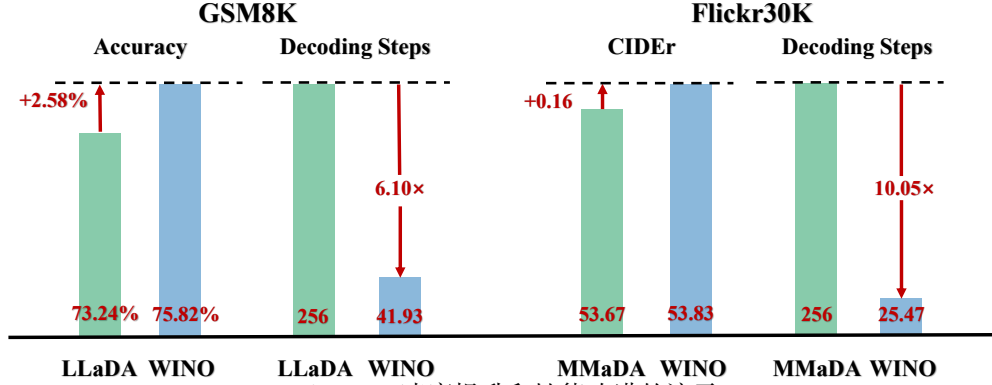


Figure 1: 速度提升和性能改进的演示。

这很容易导致早期错误被永久嵌入 accumulated，并在整个输出中传播。Therefore, such a rigid process essentially 阻止了 DLLMs 利用其双向注意力 (Seo et al., 2017) 的最大优势来修正早期错误 when the context progressively becomes rich。

为了解决这个问题，我们引入了 Wide-In, Narrow-Out (WINO)，这是一种新颖的解码算法，可以实现 DLLMs 的可撤销解码。WINO 采用了一种新颖的草稿与验证程序，能够并行运行。在每一步中，一个草稿模块根据宽松的阈值（“宽进”）积极地提出多个新标记。同时，一个验证模块利用新丰富的全局上下文重新评估所有先前生成的标记。任何未通过更严格验证检查的标记都会在未来的步骤中被重新遮掩以便再次细化（“窄出”）。This mechanism brings two merits: 1) it breaks the irreversibility of the conventional decoding in DLLMs, allowing the early error to be corrected for better performance; 2) it permits more aggressive token generation in each diffusion step for faster speedup with quality guarantee. Besides, our WINO is training-free and play-and-plug, which enables the general DLLMs 被证明是高效且有效的。

我们的广泛实验表明，当应用于现有的开源模型如 LLaDA (Nie et al., 2025) 和 MMaDA (Yang et al., 2025) 时，WINO 能够大幅加快速度，同时也始终提高模型在语言和视觉-语言任务上的准确性。例如，如图 1 所示，在 GSM8K (Cobbe et al., 2021) 数学推理基准测试上，WINO 将推理加速了 $6\times$ ，同时将准确性提高了 2.58%；在 Flickr30K (Young et al., 2014) 图像描述基准测试上，解码速度提高了 $10\times$ ，且性能更高。通过使解码可撤销，WINO 在这一领域释放了 DLLMs 的潜力。

2 相关工作

基于扩散的语言模型。扩散模型 (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song et al., 2021)，最初在图像生成中被普及 (Nichol et al., 2022; Rombach et al., 2022; Saharia et al., 2022)，最近作为文本生成的自回归语言模型 (ARLMs) 的替代方案而受到关注。从连续域到离散域的这种扩展首先由 Sohl-Dickstein et al. (2015) 研究。随后，D3PM (Austin et al., 2021a) 提供了一个通用框架，它将扩散前向过程建模为由特定的状态转移矩阵在离散时间步骤上定义的离散状态马尔可夫链。Campbell et al. (2022) 后来将 D3PM 扩展到连续时间设置，利用了连续时间马尔可夫链 (CTMC) 的理论。最近，源自 D3PM 中的吸收状态扩散的遮蔽扩散模型 (MDMs) 的研究在小规模模型（例如，MDLM (Sahoo et al., 2024) 和 RADD (Ou et al., 2025)）和大规模应用（例如，LLaDA (Nie et al., 2025) 和 Dream (Ye et al., 2025)）中显示出可喜的结果。扩展这一研究方向，MMaDA (Yang et al., 2025) 引入了一种新型的多模态大规模扩散模型类，其特点是共享的概率公式和一种与模态无关的结构。

DLLM 加速技术。The existing acceleration study for DLLMs falls into two directions: KV cache and sampling compression. The former targets to build the KV cache for DLLMs due to its bidirectional full attention mechanism, unlike the causal attention of ARLMs. Typical works like Block Diffusion (Arriola et al., 2025), Fast-dLLM-cache (Wu et al., 2025) and dLLM-cache (Liu et al., 2025) respectively explore different caching mechanisms, which shows promising performance for speedup. Note that this direction is out of the scope of our work here. The latter direction focuses on optimizing the sampling process itself. For the classic low-confidence remasking strategy, several works have introduced novel sampling strategies to dynamically adjust the number of tokens predicted in parallel, thereby improving inference efficiency. Fast-dLLM-parallel (Wu et al., 2025)

adopts a straightforward approach by selecting tokens with confidence scores exceeding a predefined threshold. Meanwhile, Ben-Hamu et al. (2025) propose an entropy-bounded (EB) sampler, a drop-in replacement for conventional samplers that leverages an entropy-based unmasking procedure to dynamically decode multiple tokens per step while maintaining a predefined error tolerance. Although our WINO brings the acceleration promise due to sampling compression, different from these works, we explore to address the inherent limitation of standard decoding in DLLMs.

对于给定的提示 X ，设计了一个 DLLM 来生成一个具有预定义响应长度 L 的响应 $Y = [y_1, y_2, \dots, y_L]$ 。响应序列初始化为所有特殊的掩码标记 $Y^{(0)} = [\text{[掩码]}, \text{[掩码]}, \dots, \text{[MASK]}]$ 。解码过程通过总共 K 个去噪步骤迭代地改进响应序列 $Y^{(k)}$ 。接下来，我们详细说明 $K = L$ 的情况（即每步解码一个标记），因为现有模型通常在此设置下实现最佳性能 (Nie et al., 2025)。

在步骤 k 中，解码的目标是将序列 $Y^{(k-1)}$ 优化为 $Y^{(k)}$ 。给定词汇表 V 和参数为 θ 的模型，模型对响应序列的概率分布进行估计，表示为 $p_\theta(\hat{Y}|X, Y^{(k-1)})$ 。作为一个常见例子，在高置信度贪婪解码中，通过基于 $Y^{(k-1)}$ 对最有信心的 [掩码] 标记进行解码，从而获得 $Y^{(k)}$ ，即，

$$\begin{aligned} l^{(k)} &= \arg \max_{l \in \{l | y_l^{(k-1)} = \text{[掩码]}\}} \left(\max_{v \in V} p_\theta(\hat{y}_l = v | X, Y^{(k-1)}) \right), \\ y_l^{(k)} &= \begin{cases} \arg \max_{v \in V} p_\theta(\hat{y}_l = v | X, Y^{(k-1)}), & \text{if } l = l^{(k)}, \\ y_l^{(k-1)}, & \text{otherwise,} \end{cases} \quad \forall l \in \{1, 2, \dots, L\}. \end{aligned} \quad (1)$$

在完成所有 K 解码步骤后，最终生成的响应是 $Y = Y^{(K)}$ 。现有的 DLLMs，如 LLaDA (Nie et al., 2025) 和 MMaDA (Yang et al., 2025)，也可以通过每步生成多个 token（例如，2 个或 4 个）的简单并行采样来加速解码过程。然而，经验结果显示，这种策略通常会导致性能显著下降，尽管计算速度有所提升，但在实际效果上受到了限制。

半自回归扩散解码。这个策略被像 LLaDA (Nie et al., 2025) 和 MMaDA (Yang et al., 2025) 的 DLLM 广泛采用，它涉及将响应序列分成多个块并从左到右依次解码。在每个块中，应用上面描述的典型扩散解码策略。

虽然在结构上适合并行计算，但 DLLMs 面临一个关键瓶颈，妨碍了有效的多标记解码。在生成的早期阶段，稀疏的上下文常常导致模型在尝试并行解码时产生低质量或矛盾的标记。这个问题由于标准解码过程的 irreversibility 性质而被严重放大。由于这些有缺陷的初步预测被永久锁定，它们不可避免地在此渐进扩散步骤中传播并降低最终生成质量，迫使在并行解码的速度和串行解码的质量之间作出不理想的权衡。

To improve the trade-off, one critical point is to abandon the assumption of irreversibility in the standard decoding of DLLMs, and build a process of revokable decoding for progressive refinement。这个原则允许模型不断修正它自身初步的并行输出。随着在生成过程中更多上下文的出现，模型可以纠正它的初步预测。这样的机制最终解决了核心冲突。它结合了并行生成的效率与考虑上下文修正的精准性，释放了 DLLMs 的真正潜力。

2.1 通过并行起草和验证进行迭代优化

Motivated by the above analysis and the design intuition, we propose a parallel Draft-and-Verify framework to enable revokable decoding for more efficient and higher-quality generation in DLLMs.

具体来说，我们的框架在每个解码步骤中并行执行两个模块：1) 起草：积极地将多个 [掩码] 标记解锁为有意义的候选标记；2) 验证：评估所有当前解锁的标记，并重新掩盖那些被认为质量低下的标记以进行进一步完善。我们采用最常见和通用的半自回归解码范式来呈现我们的方法。当块长度等于生成长度时，它相当于完全扩散解码。

2.1.1 起草

我们将整个序列表示为 $Y = [Y_{\text{left}}, Y_{\text{cur}}, Y_{\text{right}}]$ ，其中 Y_{left} 包含提示 X 和之前解码的块， $Y_{\text{cur}} = [y_{\text{cur},1}, \dots, y_{\text{cur},L_b}]$ 代表当前正在解码的块， Y_{right} 表示剩余待解码的块。这里， L_b 是块长度。在 k -th 解码步骤中，我们不是解码固定数量的标记，而是基于信任阈值 τ_1 执行积极

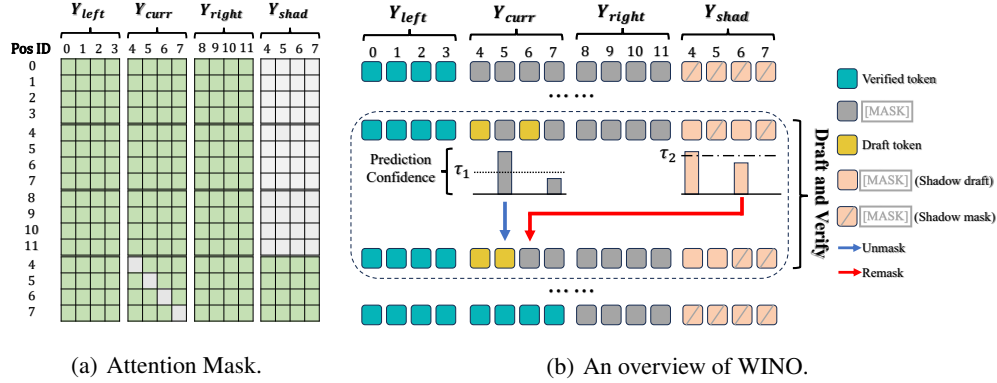


Figure 2: (a) 我们设计的注意力掩码的示意图。绿色方块表示 1，灰色方块表示 0，“Pos ID”是位置 ID 的缩写。(b) WINO 的概述。验证的 token 指代提示 X 或之前解码块中的 token。草稿 token 表示当前块中未被掩码的、直到当前解码步骤的 token。 $\boxed{\text{掩码}}$ （影子草稿）指代影子块中位置 ID 对应于草稿 token 的 token，而 $\boxed{\text{掩码}}$ （影子掩码）指代影子块中剩余的 token。

的多标记并行解码：

$$y_{cur,l}^{(k)} = \arg \max_{v \in V} p_{\theta}(\hat{y}_{cur,l} = v | Y), \text{ if } \max_{v \in V} p_{\theta}(\hat{y}_{cur,l} = v | Y) > \tau_1 \text{ and } y_{cur,l}^{(k-1)} = \boxed{\text{掩码}}. \quad (2)$$

Here, a relatively low confidence threshold τ_1 is adopted to allow more possible tokens to be decoded at each step, which will achieve the acceleration if only a few tokens among them are revoked during the verification module detailed in the next section. This will be demonstrated in the experiments.

2.1.2 验证

验证模块的设计原则是利用每个解码步骤中不断丰富的语义上下文（相对于早期步骤）来评估之前未遮蔽标记的质量。通过重新遮蔽质量较低的标记，解码过程可以变得可撤销和适应性强 for the proper early error correction。为了实现对解码标记的有效质量验证，我们设计了一个辅助的影子块，完全由 $\boxed{\text{掩码}}$ ， $Y_{shad} = [\boxed{\text{掩码}}] \times L_b$ 组成。此块被附加到序列 Y 上，形成扩展序列 $\tilde{Y} = [Y_{left}, Y_{curr}, Y_{right}, Y_{shad}]$ 。我们仔细设计了与 Y_{shad} 相关的位置信息 ID 和注意力掩码，以确保其输出可以有效验证在 Y_{curr} 对应位置上解码的标记的质量。

位置 ID。尽管 Y_{shad} 被附加到序列的右端，我们赋予它与 Y_{curr} 相同的位置 ID。因此， Y_{shad} 的输出对应于与 Y_{curr} 相同的位置，从而实现逐位验证。

注意力掩码。如 Fig. 2(a) 所示，我们在将 Y_{shad} 纳入序列 $\tilde{Y}$ 后精心设计了注意力掩码。具体来说， Y_{left} 、 Y_{curr} 和 Y_{right} 中的标记可以自由地相互关注，但不允许关注 Y_{shad} 。相反， Y_{shad} 中的每个标记都可以关注除 Y_{curr} 中对应位置之外的所有标记。

通过上述位置标识和注意力掩码的设计，我们达到了以下特性：

- 对于当前块 Y_{curr} 中的任何标记，附加 Y_{shad} 不会影响模型的输出。形式上讲，
$$p_{\theta}(\hat{y}_{cur,l} | Y) = p_{\theta}(\hat{y}_{cur,l} | \tilde{Y}).$$
- 对于 Y_{shad} 中的任何标记，以下性质成立。例如，考虑 Fig. 2(b) 中的标记 $y_{shad,3}$ ，其被分配了位置 ID 6。
 - 它与 $y_{cur,3}$ 共享相同的位置 ID，并被允许关注 Y_{left} 和 Y_{right} ；
 - 通过明确禁止关注 $y_{cur,3}$ ，有效避免了验证过程中的信息泄漏；
 - 对于 Y_{curr} 中的所有其他位置，每个位置恰好由一个解码的标记（来自 Y_{curr} ）和一个 $\boxed{\text{掩码}}$ 在 Y_{shad} 中出现在这些位置上。前者在解码过程中提供了逐渐丰富的上下文语义，而后者则用于规范 Y_{curr} 中解码标记的置信度，反映了解码过程中的不确定性以及可能需要的进一步优化。

Table 1: 语言领域多样化推理和生成基准上的性能和推理速度比较。

Benchmark	Method	Accuracy	Steps	Step Reduction	TPS	TPS Speedup
GSM8K Math Reasoning	LLaDA	73.24	256	1.00 ×	17.76	1.00 ×
	WINO	75.82 (+2.58)	41.93 (-214.07)	6.10 ×	100.53 (+82.77)	5.66 ×
MATH Math Reasoning	LLaDA	32.00	256	1.00 ×	17.62	1.00 ×
	WINO	34.20 (+2.20)	74.44 (-181.56)	3.44 ×	55.86 (+38.24)	3.17 ×
HumanEval Code Generation	LLaDA	37.80	256	1.00 ×	14.52	1.00 ×
	WINO	42.07 (+4.27)	93.32 (-162.68)	2.74 ×	37.19 (+22.67)	2.56 ×
MBPP Code Generation	LLaDA	36.40	256	1.00 ×	18.52	1.00 ×
	WINO	36.40 (+0.00)	96.57 (-159.43)	2.65 ×	45.39 (+26.87)	2.45 ×
Countdown Logical Reasoning	LLaDA	24.21	256	1.00 ×	17.22	1.00 ×
	WINO	33.20 (+8.99)	105.88 (-150.12)	2.41 ×	38.97 (+21.75)	2.26 ×
Sudoku Logical Reasoning	LLaDA	14.23	256	1.00 ×	11.61	1.00 ×
	WINO	15.20 (+0.97)	131.96 (-124.04)	1.94 ×	21.11 (+9.50)	1.82 ×
ARC-E Commonsense Reasoning	LLaDA	59.13	256	1.00 ×	17.26	1.00 ×
	WINO	81.19 (+22.06)	40.19 (-215.81)	6.37 ×	101.61 (+84.35)	5.89 ×
ARC-C Commonsense Reasoning	LLaDA	51.87	256	1.00 ×	17.10	1.00 ×
	WINO	73.89 (+22.02)	47.41 (-208.59)	5.40 ×	85.42 (+68.32)	5.00 ×

通过上述特别设计的位置 ID 和注意力掩码，验证模块可以正式表示为：

$$y_{\text{cur},l}^{(k)} = [\text{MASK}], \text{ if } p_{\theta}(\hat{y}_{\text{shard},l} = y_{\text{cur},l}^{(k-1)} | \tilde{Y}) < \tau_2 \text{ and } y_{\text{cur},l}^{(k-1)} \neq [\text{掩码}], \quad (3)$$

，其中 τ_2 是用于验证的置信度阈值。

总之，在解码步骤 k 中，我们的框架使得起草和验证过程都可以在一次前向传递中完成：

$$y_{\text{cur},l}^{(k)} = \begin{cases} \arg \max_{v \in V} p_{\theta}(\hat{y}_{\text{cur},l} = v | \tilde{Y}), & \text{if } \max_{v \in V} p_{\theta}(\hat{y}_{\text{cur},l} = v | \tilde{Y}) > \tau_1 \text{ and } y_{\text{cur},l}^{(k-1)} = [\text{MASK}], \\ [\text{掩码}], & \text{if } p_{\theta}(\hat{y}_{\text{shard},l} = y_{\text{cur},l}^{(k-1)} | \tilde{Y}) < \tau_2 \text{ and } y_{\text{cur},l}^{(k-1)} \neq [\text{遮挡}], \\ y_{\text{cur},l}^{(k-1)}, & \text{otherwise.} \end{cases} \quad (4)$$

我们使用 Eq. (4) 中的步骤迭代地优化整个 Y_{cur} ，直到 Y_{cur} 中的所有标记不再是 $[\text{掩码}]$ 。我们设定起草阈值 τ_1 和验证阈值 τ_2 使得 $\tau_1 < \tau_2$ 。较低的 τ_1 通过允许更多标记并行生成加速解码过程，而较高的 τ_2 通过执行更严格的接受标准确保最终输出的质量。我们将这种设计哲学称为“宽进，窄出” and term our method as WINO in short。

3 实验

3.1 实验设置

数据集和基线。我们进行实验以评估 WINO 在不同类型的任务和领域中的表现。具体来说，对于语言领域，我们将 WINO 与 LLaDA 的标准解码在八个任务上进行比较：GSM8K (Cobbe et al., 2021)、MATH (Hendrycks et al., 2021)、HumanEval (Chen et al., 2021)、MBPP (Austin et al., 2021b)、Countdown (Zhao et al., 2025)、Sudoku (Zhao et al., 2025)、ARC-E (Clark et al., 2018) 和 ARC-C (Clark et al., 2018)，涵盖四类文本生成任务——数学推理、代码生成、逻辑推理和常识推理。对于视觉-语言领域，我们评估 WINO 在 MMaDA (Yang et al., 2025) 标准解码上的表现，涉及六个多模态理解任务：Flickr30k (Young et al., 2014)、AI2D (Kembhavi et al., 2016)、Math-Vision (Wang et al., 2024)、MathVista (Lu et al., 2024)、MMM (Yue et al., 2024) 和 ScienceQA (Lu et al., 2022)，跨越四种类型的多模态任务——图像标注、图表理解、数学推理和多学科推理。为清晰起见，我们在 MMM 的验证集和 MathVista 的官方小测试集上进行测试。

评估细节。所有基准测试都以零样本方式进行评估。关于所考虑基准的性能指标，我们在 Flickr30K 基准中使用 CIDEr 度量 (Vedantam et al., 2015)，而对于其余所有基准则使用准确度。为了评估我们方法的推断效率，我们通过在一个基准的所有样本中取平均值来测量每个任务的基线和 WINO 的所需解码步骤以及每秒生成的标记数 (TPS)。TPS 表示模型每秒

Table 2: 在视觉-语言领域的多样化生成和推理基准上的性能和推断速度对比。我们使用 CIDEr 作为 Flickr30K 的性能指标，其他基准使用准确率作为性能指标。

Benchmark	Method	Metric	Steps	Step Reduction	TPS	TPS Speedup
Flickr30K Captioning	MMaDA	53.67	256	1.00 ×	6.41	1.00 ×
	WINO	53.83 (+0.16)	25.47 (-230.53)	10.05 ×	55.11 (+48.70)	8.60 ×
AI2D Chart Understanding	MMaDA	54.86	256	1.00 ×	6.31	1.00 ×
	WINO	57.19 (+2.33)	30.90 (-225.10)	8.30 ×	46.04 (+39.73)	7.30 ×
Math-Vision Math Reasoning	MMaDA	8.55	256	1.00 ×	6.22	1.00 ×
	WINO	9.57 (+1.02)	44.69 (-211.31)	5.73 ×	31.17 (+24.95)	5.01 ×
MathVista-mini Math Reasoning	MMaDA	31.10	256	1.00 ×	6.21	1.00 ×
	WINO	31.40 (+0.30)	33.45 (-222.55)	7.65 ×	41.96 (+35.75)	6.76 ×
MMM-Val Multi-discipline Reasoning	MMaDA	18.56	256	1.00 ×	6.02	1.00 ×
	WINO	24.00 (+5.44)	38.47 (-217.53)	6.65 ×	36.13 (+30.11)	6.00 ×
ScienceQA Multi-discipline Reasoning	MMaDA	30.89	256	1.00 ×	6.07	1.00 ×
	WINO	42.24 (+11.35)	28.12 (-227.88)	9.10 ×	49.45 (+43.38)	8.15 ×

Table 3: 实验结果关于不同生成长度，块长度 L_b 等于 128。基线每个解码步骤揭示 1 个标记。

Benchmark	Generation Length	Method	Accuracy	Steps	Step Reduction	TPS	TPS Speedup
GSM8K	256	LLaDA	73.24	256	1.00 ×	17.76	1.00 ×
		WINO	75.82	41.93	6.10 ×	100.53	5.66 ×
	512	LLaDA	74.60	512	1.00 ×	11.84	1.00 ×
		WINO	79.91	68.53	7.47 ×	82.64	6.98 ×
MMM-Val	256	MMaDA	18.56	256	1.00 ×	6.02	1.00 ×
		WINO	24.00	38.47	6.65 ×	36.13	6.00 ×
	512	MMaDA	18.44	512	1.00 ×	5.01	1.00 ×
		WINO	23.44	64.82	7.90 ×	35.01	6.99 ×

生成的平均标记数。我们定义步骤减少为基线的所需解码步骤与 WINO 的比例，TPS 加速为基线的 TPS 与 WINO 的比例。

实现细节。对于模型配置，我们采用开源的 LLaDA-8B-Instruct² 进行语言领域任务，MMaDA-8B-MixCoT³ 进行视觉语言领域任务。我们采用在 LLaDA (Nie et al., 2025) 中介绍的半自回归采样策略，其中输出序列被划分成多个块，并从左到右生成。在我们的评估中，除非特别说明，我们将生成长度设置为 256，基线模型的解码步骤设置为 256，块长度设置为 128。对于包含推理过程的输出序列，我们提取最终答案进行评估。对于 WINO 的超参数，我们将验证阈值 τ_2 设置为 0.9，草拟阈值 τ_1 在范围 $\{0.4, 0.5, 0.6, 0.7\}$ 内调整。

3.2 主要结果

文本生成的性能和加速。我们报告了 LLaDA-8B-Instruct 在有和没有 WINO 的情况下，在 Tab. 1 的语言域任务上的性能、解码步骤数和吞吐量 (TPS)。除 MBPP 任务外，WINO 在所有文本推理和生成基准测试上，使用比基线 LLaDA 少得多的解码步骤，显著提高了准确性，在 MBPP 任务上，WINO 达到了与 LLaDA 相同的性能。例如，WINO 在 GSM8K 上将准确性提高了 2.58%，并减少了 $6.10 \times$ 步骤，TPS 加速了 $5.66 \times$ 。这些实验结果展示了 WINO 在提升生成质量和推理效率方面的有效性。

多模态理解和推理的性能和加速。我们评估了将 WINO 整合到 MMaDA 中所带来的性能和效率提升，并在 Tab. 2 中总结了结果。与原版 MMaDA 相比，WINO 在所有基准上展示了显著且持续的推理效率提升。值得注意的是，与在文本领域任务的结果相比，加速效果更加显著。此外，WINO 在保持 Flickr30K、Math-Vision 和 MathVista 上相当结果的同时，极大提升了 AI2D、MMM 和 ScienceQA 上的任务性能。这些结果表明，WINO 在多模态领域中始终表现出性能提升和推理加速的效果。

速度提升与任务复杂度之间的关系。如 Tab. 1 和 Tab. 2 所示，我们观察到在所有基准测试中，速度提升的程度与任务性能之间存在一致的正相关关系。值得注意

²<https://huggingface.co/GSAI-ML/LLaDA-8B-Instruct>

³<https://huggingface.co/Gen-Verse/MMaDA-8B-MixCoT>

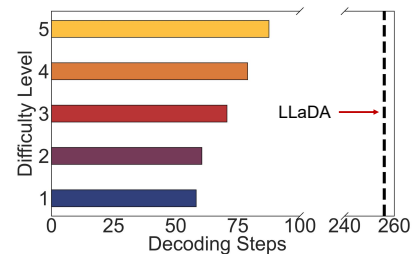


Figure 3: WINO 在具有不同难度级别的 MATH 基准测试子集上的解码

Table 4: 在块长度 L_b 等于生成长度的完全扩散设置中进行实验的结果。基线在每个解码步骤中解码 1 个标记。

Benchmark	Generation Length	Method	Accuracy	Steps	Step Reduction	TPS	TPS Speedup
GSM8K	256	LLaDA	34.34	256	1.00 ×	17.73	1.00 ×
		WINO	58.22 (+23.88)	38.77	6.60 ×	93.61	5.28 ×
	128	LLaDA	58.60	128	1.00 ×	23.23	1.00 ×
		WINO	62.32 (+3.72)	23.95	5.34 ×	114.29	4.92 ×
MMM-Val	256	MMaDA	17.22	256	1.00 ×	6.11	1.00 ×
		WINO	22.44 (+5.22)	24.94	10.26 ×	50.03	8.19 ×
	128	MMaDA	15.33	128	1.00 ×	6.70	1.00 ×
		WINO	23.11 (+7.78)	19.14	6.69 ×	39.94	5.96 ×

Table 5: 关于没有验证模块的 WINO 变体的实验结果。

Benchmark	Method	Accuracy	Steps	Step Reduction	TPS	TPS Speedup
GSM8K	LLaDA	73.24	256	1.00 ×	17.76	1.00 ×
	Only Draft ($\tau_1 = 0.6$)	70.28	34.79	7.36 ×	130.89	7.37 ×
	Only Draft ($\tau_1 = 0.9$)	72.33	81.39	3.15 ×	56.12	3.16 ×
	WINO	75.82	41.93	6.10 ×	100.53	5.66 ×
MMM-Val	MMaDA	18.56	256	1.00 ×	6.02	1.00 ×
	Only Draft ($\tau_1 = 0.6$)	19.89	35.63	7.18 ×	43.22	7.18 ×
	Only Draft ($\tau_1 = 0.9$)	18.56	79.74	3.21 ×	19.38	3.22 ×
	WINO	24.00	38.47	6.65 ×	36.13	6.00 ×

的是，当基准模型在任务得分上表现较高时（相较于更复杂的任务），步骤减少和 TPS 加速值显著更高。例如，WINO 在相对简单的字幕任务 Flickr30K 上实现了 $10.05 \times$ 的步骤减少，而在更具挑战性的数学推理基准测试 Math-Vision 上仅减少 $5.73 \times$ 步骤。这一现象暗示如果基础模型在目标任务上表现更好，则 WINO 的增益可能更明显。受限于固定的解码步骤和每步生成的固定字数，基准模型在推理效率上表现欠佳，特别是对于较简单的推理和生成任务。为了进一步研究这一现象，我们评估了 WINO 在按难度级别分类的 MATH 基准子集上的解码步骤。如图 3 所示，随着难度级别的降低，WINO 实现了逐渐更大的加速，突显了其根据任务复杂度自适应优化推理速度的能力。

3.3 消融研究和进一步分析。

关于不同的生成长度。在 Tab. 3 中，我们评估了 WINO 在不同生成长度上的性能表现，其中块长度 L_b 固定为 128，基线方法在每一步解码时都会解开 1 个 token（以达到最佳生成性能）。当生成长度设置为 512 时，WINO 在显著减少解码步骤的情况下仍然能实现可比或更好的任务性能，展示了 WINO 在不同生成长度上的有效性。

关于全扩散解码（而非半自回归解码）。在 Tab. 4 中，我们比较了基线和 WINO 应用全扩散解码的情况，这意味着块长度 L_b 设置为等于生成长度。相比于在 Tab. 1 和 Tab. 2 中的半自回归解码结果，WINO 在全扩散设置下显示出显著的准确率提升。值得注意的是，尽管 LLaDA 在 GSM8K 的全扩散解码中遭遇了显著的准确率下降，但 WINO 在使用更少解码步骤的情况下保持了合理的表现。这些结果表明，WINO 在全扩散解码场景中应用时可以显著提升模型性能和计算效率的潜力。

关于验证模块的消融实验。我们对一个不包含验证模块的 WINO 变体进行了一项消融研究，通过将验证阈值 τ_2 设置为零来实现。如图 5 所示，与 WINO 相比，这种变体在两个基准上表现出明显的性能下降。具体而言，当起草阈值 τ_1 较小时（对应于表中的 0.6），每个解码步骤会取消对更多候选标记的掩码，这自然会引入更高比例的不可靠标记，最终降低输出质量。相反，当 τ_1 较大时（对应于表中的 0.9），每个解码步骤被取消掩码的候选标记较少，这可以缓解错误传播，但以计算效率为代价。关键是，没有验证模块，生成过程缺乏纠

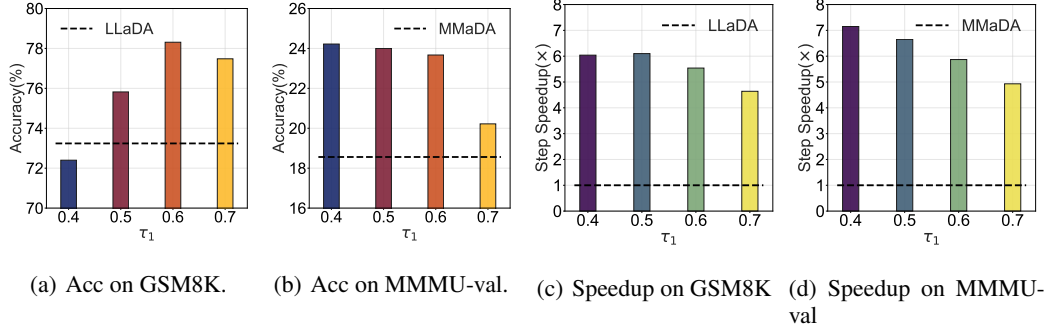


Figure 5: 在验证阈值 τ_2 固定为 0.9 的情况下，对起草阈值 τ_1 进行的实验。

正错误预测的机制。因此，即使在大 τ_1 情况下，模型也可能无法取得可比的性能，这凸显了验证模块在维持生成质量中的必要性。

与简单并行采样的比较。现有 DLLMs 的解码过程可以通过每一步生成多个标记来加速，即简单并行采样。然而，如 Tab. 6 所示，直接增加 DLLMs 每一步生成的固定标记数量会导致显著的性能下降。例如，GSM8K 的准确率从 73.24% (1 个标记/步) 下降到 64.67% (4 个标记/步)。相比之下，WINO 的起草和验证过程允许生成过程中的灵活解码，这在实现更快推理速度的同时显著提高了任务性能。

GPU 内存使用。为了方便高效地验证非掩码标记的质量，WINO 引入了一个辅助阴影块，其大小等于整个半自回归解码过程的指定块长度 L_b 。因此，WINO 产生了额外的 GPU 内存成本来存储辅助阴影块。我们评估了 WINO 的 GPU 内存成本，并在 Fig. 4 中展示了结果。相比所有基线，WINO 的额外内存成本保持在较小的水平。例如，在 GSM8K 上，WINO 相比标准 LLaMA 解码，仅增加了 2.4% 的 GPU 内存使用 (从 16.18 GB 增加到 16.57 GB)。在其他数据集上，WINO 引入的额外内存开销同样可以忽略不计。

阈值调节的效果。在 Fig. 5 中，我们呈现了对不同起草阈值 τ_1 的 WINO 的评估结果。我们的实验表明，WINO 在不同基准和 τ_1 值上始终优于基线，无论是在任务性能还是推理效率方面。当 τ_1 值降低时，在每个解码步骤中会取消遮盖更多的候选标记，从而通过减少所需的解码步骤来加速推理。然而，这样做的代价是可能引入更多不可靠的预测，可能会增加验证模块纠错的负担。根据经验，我们发现将起草阈值 τ_1 设置在 0.5 到 0.7 的范围内可以实现最佳平衡，既能保持竞争力的任务性能，又能保证高效的生成。

案例研究: 解码动态。为了对基线和 WINO 的解码过程进行细粒度的检查，我们展示了两个生成示例——一个来自 GSM8K，一个来自 MMMU-val。如图所示，基线可能在早期解码阶段生成错误的标记。由于基线生成的标记在后续解码步骤中保持不变，错误的上下文信息在整个生成过程中传播，最终导致生成结果质量低下。相比之下，WINO 通过迭代的起草和验证机制实现了对生成标记的动态优化，这减轻了错误的累积并促进了高质量解码输出。

Table 6: 使用简单并行采样的基线实验结果。

Benchmark	Method	Steps	Accuracy	Step Reduction
GSM8K	LLaDA	256	73.24	1.00 ×
		128	71.11	2.00 ×
	WINO	64	64.67	4.00 ×
		41.93	75.82	6.10 ×
MathVista-mini	MMaDA	256	31.10	1.00 ×
		128	30.02	2.00 ×
	WINO	64	29.80	4.00 ×
		33.45	31.40	7.65 ×

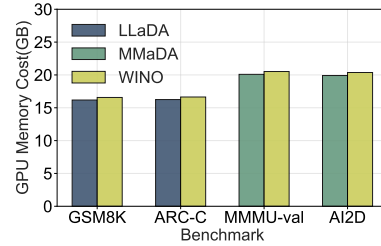


Figure 4: GPU 内存使用情况。

Case: GSM8K, Generation Length 256, Block Length 128, 2 blocks in total

Question: In a family, there are 2 brothers and 3 sisters. All sisters are the same age, which is 16. One of the brothers is 12 years old, which is half the age of the older brother. What is the total age of all these siblings?

GT: 84.

LLaDA:

Step: 128 (第一块解码完成). 1st block:

<reasoning>

1. We know that all three sisters are 16 years old.
2. One of the brothers is 12 years old, which is half the age of the older brother.
3. Let's denote the age of the older brother as x .
4. According to the problem, 12 is half the age of the older brother, so we can write the equation: $12 = \frac{x}{2}$.
5. Solving for x , we multiply both sides by 2: $x = 24$.
6. Therefore, the

Step: 162 (早期过度自信错误). 2nd block:

older brother is 24 years old.

7.<|mdm_mask|>...<|mdm_mask|>

</reasoning>

<answer>

96

</answer><|eot_id|><|endoftext|>

Final (误差累积):

<reasoning>

1. We know that all three sisters are 16 years old.
2. One of the brothers is 12 years old, which is half the age of the older brother.
3. Let's denote the age of the older brother as x .
4. According to the problem, 12 is half the age of the older brother, so we can write the equation: $12 = \frac{x}{2}$.
5. Solving for x , we multiply both sides by 2: $x = 24$.
6. Therefore, the older brother is 24 years old.
7. The younger brother, being the only other brother, 也是 24 岁.
8. The total age of the two brothers is $24 + 24 = 48$.
9. The total age of the three sisters is $3 \times 16 = 48$.
10. The total age of all the siblings is $48 + 48 = 96$.

</reasoning>

<answer>

96

</answer><|eot_id|><|endoftext|>

WINO:

Step: 35 (第一个区块解码完成). 1st block:

<reasoning>

1. We know that all three sisters are 16 years old.
2. One of the brothers is 12 years old, which is half the age of the older brother.
3. Let's denote the age of the older brother as x .
4. According to the problem, 12 is half the age of the older brother, so we can write the equation: $12 = \frac{x}{2}$.
5. Solving for x , we multiply both sides by 2: $x = 24$.
6. Therefore, the

Step: 37 (绘图误差). 2nd block:

older brother is 24 years old.

7.<|mdm_mask|>...<|mdm_mask|> = 96<|mdm_mask|><|mdm_mask|><|mdm_mask|>

</reasoning><|mdm_mask|>

<answer>

9<|mdm_mask|>

</answer><|eot_id|><|endoftext|>

Step: 42 (改进: 仍然错误). 2nd block:

older brother is 24 years old.

7. The total age of<|mdm_mask|> sisters<|mdm_mask|> \((3<|mdm_mask|>\text{times } 16 = 4<|mdm_mask|> \dots \)

<|mdm_mask|> total age of all the<|mdm_mask|> is \((48 + <|mdm_mask|><|mdm_mask|> = 86 \)) years.

</reasoning>

<answer>

86

</answer><|eot_id|><|endoftext|>

Final (通过迭代优化所有正确):

<reasoning>

1. We know that all three sisters are 16 years old.
2. One of the brothers is 12 years old, which is half the age of the older brother.
3. Let's denote the age of the older brother as x .
4. According to the problem, 12 is half the age of the older brother, so we can write the equation: $12 = \frac{x}{2}$.
5. Solving for x , we multiply both sides by 2: $x = 24$.

```

6. Therefore, the older brother is 24 years old.
7. The total age of the sisters is  $3 \times 16 = 48$  years.
8. The total age of the brothers is the sum of the ages of the 12-year-old and the 24-year-old brother:  $12 + 24 = 36$  years.
9. The total age of all the siblings is  $48 + 36 = 84$  years.
</reasoning>
<answer>
84
</answer><|eot_id|><|endoftext|>

```

在这项工作中，我们推出了 Wide-In, Narrow-Out (WINO)，这是一种无需训练的解码算法，通过使生成过程可撤销来解决扩散大型语言模型 (DLLMs) 中关键的质量速度权衡。WINO 通过采用并行的草稿和验证机制，克服了不可撤销的标准解码的限制，使模型能够积极生成标记，并利用其完整的双向上下文迭代地纠正错误。我们在 LLaDA 和 MMaDA 等现有开源模型上的实验表明，WINO 在显著提高多种语言和视觉语言任务的准确性的同时，将推理速度加快到最多 $10 \times$ 。虽然我们承认未来在结构上的改进空间，WINO 通过重新思考解码过程本身，从根本上提升了 DLLMs 的实用性，确立了它们作为自回归系统真正高效和高质量替代品的地位。

References

- Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.
- Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17981–17993, 2021a.
- Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021b.
- Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *CoRR*, abs/2505.24857, 2025.
- Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018.

- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.
- Google DeepMind. Gemini diffusion. Google DeepMind Models, 2025. URL <https://deepmind.google/models/gemini-diffusion>.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks*, 2021.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- Inception Labs. Introducing mercury: The first commercial diffusion-based language model. Inception Labs Blog, 2025. URL <https://www.inceptionlabs.ai/introducing-mercury>.
- Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Min Joon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth A dozen images. *CoRR*, abs/1603.07396, 2016.
- Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.
- Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang, and Linfeng Zhang. dlm-cache: Accelerating diffusion large language models with adaptive caching. *CoRR*, abs/2506.06295, 2025.
- Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *ICLR*, 2024.
- Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, Chenlin Zhou, Jiayi Mao, Tianze Xia, Jiafeng Guo, and Shenghua Liu. A survey of context engineering for large language models, 2025. URL <https://arxiv.org/abs/2507.13334>.
- Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR, 2022.
- Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *CoRR*, abs/2502.09992, 2025.
- OpenAI. Chatgpt: Optimizing language models for dialogue. OpenAI Blog, November 2022. URL <https://openai.com/blog/chatgpt/>.
- Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.
- Subham S. Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.
- Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *ICLR*, 2017.
- Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *CoRR*, abs/1503.03585, 2015.
- Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. GPT-4 doesn’t know it’s wrong: An analysis of iterative prompting for reasoning problems. *CoRR*, abs/2310.12397, 2023.
- Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. Can large language models really improve by self-critiquing their own plans? *CoRR*, abs/2310.08118, 2023.
- Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015.
- Ke Wang, Juntong Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. In *NeurIPS*, 2024.
- Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. *CoRR*, abs/2505.22618, 2025.
- Ling Yang, Ye Tian, Bowen Li, Xinchun Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *CoRR*, abs/2505.15809, 2025.
- Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b, 2025. URL <https://hkunlp.github.io/blog/2025/dream>.
- Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Trans. Assoc. Comput. Linguistics*, 2:67–78, 2014.

Xiang Yue, Yuansheng Ni, Tianyu Zheng, Kai Zhang, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhui Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *CVPR*, pages 9556–9567. IEEE, 2024.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *CoRR*, abs/2504.12216, 2025.