中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

Large Language Diffusion Models: Foundations and Practice

Presented by Wei Huang

October 27, 2025

Auto-Regressive V.S. Diffusion 示例

■ Auto-Regressive 示例

Auto-Regressive V.S. Diffusion 示例

- **Diffusion 示例**

1. Auto-Regressive 和 Diffusion 的核心区别：

■ 1.1 Auto-Regressive 和 Diffusion 的核心区别：

1. 生成顺序：Auto-Regressive是逐字生成，每生成一个字都依赖前面生成的内容，**有严格生成顺序**；而Diffusion 的生成范式里，**没有固定的生成顺序。**

■ 1.2 Diffusion 的优势：

2. 生成效率：Auto-Regressive 是逐字、串行式的，一次只能生成一个词；Diffusion 可以并行地生成多个词，生成效率更高。

3. 注意力差异：Auto-Regressive是单向注意力；Diffusion 是双向注意力。在需要聚合上下文信息的场景， Diffusion 会更有优势。

■ 1.3 Diffusion 的劣势：

4. 生成效率输出限制：Auto-Regressive 可以**按需生成任意长的输出文本**；Diffusion必须**预先确定输出的长度。**

2. 文本任务中Diffusion建模的合理性

2.1 文本任务本质是建模联合概率：

1. **目标：**学习真实分布 $p_{data}(x)$ ，其中 $x = (x_1, x_2, \ldots, x_L)$ 为离散的 token 序列

2. **训练：**优化参数化模型 $p_\theta(x)$ 去逼近 $p_{data}(x)$

3. **推理：**在 $p_\theta(x)$ 的高概率区域抽样，生成连贯文本。

2.2 训练目标：逼近真实的数据分布 p_{data}

$$\max_\theta \mathbb{E}_{p_{\text{data}}}[\log p_\theta(x)] \iff \min_\theta \mathrm{KL}(p_{\text{data}} \| p_\theta)$$

最大似然/最小 KL是所有现代生成模型 （AR、VAE、Diffusion..）的共同理论基础。

2. 文本任务中Diffusion建模的合理性

2.3 直接拟合联合分布的困难：维度爆炸

- 举个例子：词表 |V| ～ 30000，序列长 L ～ 10^3
- 若显式存储 $p_\theta(x)$：状态空间大小 |V|^L ，存储开销过大，且监督信号过于稀疏。
- 解决思路：**概率链式分解（语言模型）**或**引入隐变量（VAE，Diffusion）**，把高维问题化为若干低维子问题。

2. 文本任务中Diffusion建模的合理性

2.4 主流概率分布拆分方式

方式	核心公式	优点	局限	代表模型
链式分解	$p(x) = \prod_i p(x_i \mid x_{<i})$	训练简单，似然 $\log P$ 精确可算	推断串行，单向依赖	语言模型 (LM)
隐变量积分	$p_\theta(x) = \int p(z)\, p_\theta(x \mid z)\, dz$	采样可并行	需设计 $p(z)$ 与解耦机制	VAE, Diffusion, Flow-based, Energy-based

2.5 为什么 Diffusion 也适用于文本任务?

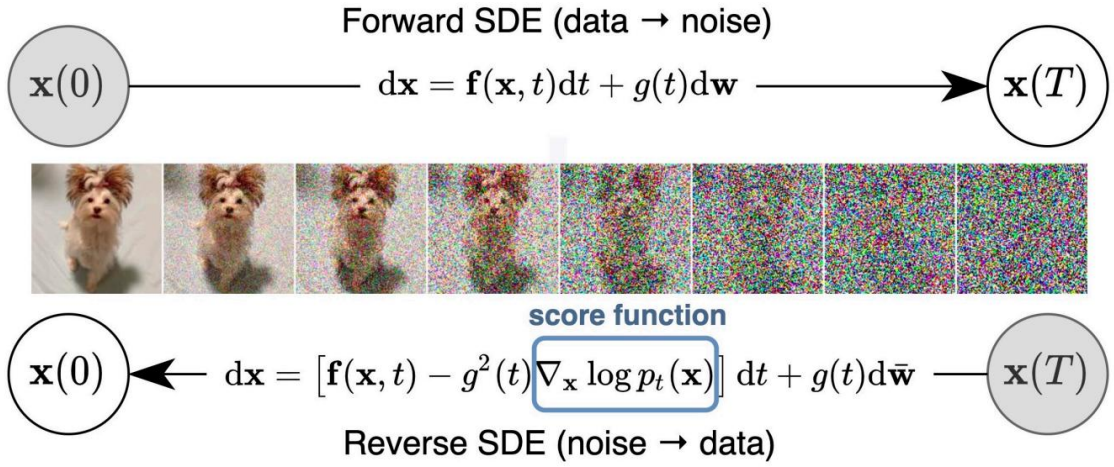
原因：因为 Diffusion 和 Auto–Regressive的共同目标都是建模联合概率。只是切入点不同而已。

3. Diffusion 在文本任务中的应用：LLaDA

3.1 Paper 信息

- 论文名：Large Language Diffusion Models
- 论文链接：https://arxiv.org/abs/2502.09992
- 单位：人大；蚂蚁集团
- 发布时间：2025.01.18

3.2 图像中 Diffusion 示例：加噪＋去噪



3. Diffusion 在文本任务中的应用：LLaDA

3.3 LLaDA 中的加噪和去噪：

- **加噪：**
  - 加噪形式：将文本的 token 直接替换成<MASK>
  - 加噪流程：为了模拟diffusion的加噪过程，需要有从完全干净到完全噪声的样本。这项工作中，并不是对一条样本连续加噪，而是对整个数据集中的每一条样本都只随机加噪一次，加噪比例 $t \in [0, 1]$。和 Bert 的MLM任务类比，MLM任务是对数据集的每一条样本添加固定的噪声比例；这里是对每条样本添加随机的噪声比例。
  - 加噪例子：今天天气真不错啊 →今天<MASK><MASK>真不错啊

3. Diffusion 在文本任务中的应用：LLaDA

3.3 LLaDA 中的加噪和去噪：

- **去噪：**
  - 去噪形式：将文本中的 <MASK> 根据上下文预测成具体的 token
  - 去噪流程：

    Step 1：<MASK> <MASK> <MASK> <MASK> <MASK> <MASK> <MASK> <MASK>

    Step 2：今天 <MASK> <MASK> <MASK> <MASK> <MASK> <MASK>

    Step 3：今天 ＜MASK> <MASK> <MASK> 不错<MASK>

    Step 4：今天 <MASK> <MASK>真不错啊

    Step 5：今天天气真不错啊

3. Diffusion 在文本任务中的应用：LLaDA

3.4 LLaDA 的训练流程（预训练+SFT）：

- **预训练：**
  - 训练集规模：2.3Ttokens
  - 训练任务：类似于Bert的MLM任务，根据上下文预测<MASK>处的token。每条样本随机 mask 掉比例为t的token，然后进行**单次预测**。
  - 损失函数（实际用 cross-entropy 实现）：

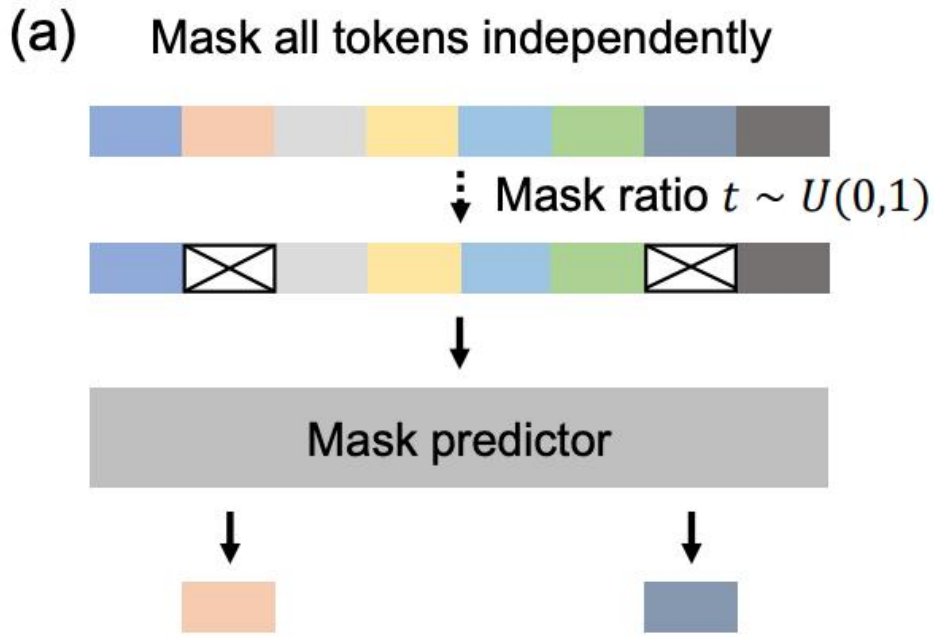
$$\mathcal{L}(\theta) \triangleq -\mathbb{E}_{t,x_0,x_t}\left[\frac{1}{t}\sum_{i=1}^{L}\mathbf{1}[x_t^i=\mathrm{M}]\log p_\theta(x_0^i|x_t)\right],$$

其中，E代表期望，t代表mask的比例，x_0代表干净样本，x_t代表加噪后的样本。注意，损失函数表明，每条样本权重并非一致。Mask 掉的比例越高，代表难度越高，样本对应的权重就越低，避免过度惩罚。

3. Diffusion 在文本任务中的应用：LLaDA

3.4 LLaDA 的训练流程（预训练+SFT）：

- **预训练：**
  - 示意图：



3. Diffusion 在文本任务中的应用：LLaDA

3.4 LLaDA 的训练流程（预训练+SFT）：

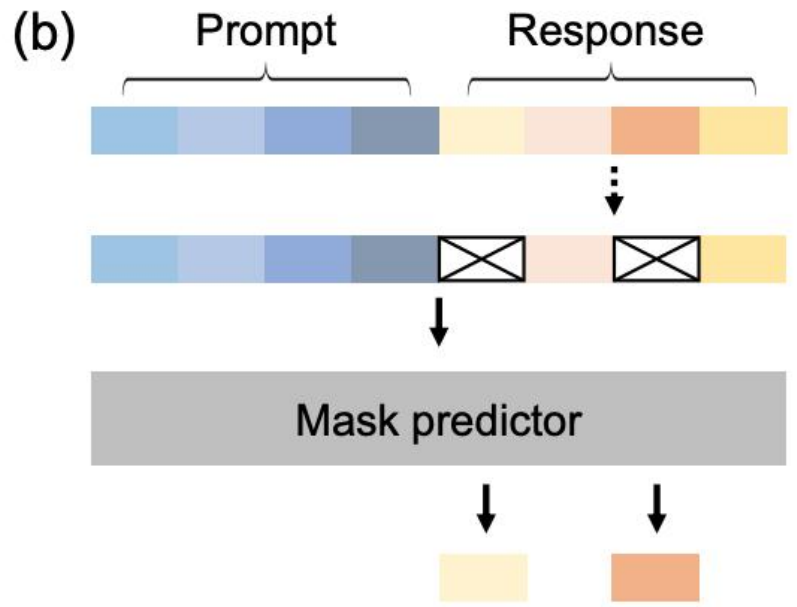
- **SFT：**
  - 训练集规模：4.5 million pairs，（p_0, x_0）。其中，p_0为prompt，x_0为response。
  - 训练任务：与预训练非常类似。区别只在于，SFT 中不 mask p_0，只 mask x_0
  - 损失函数：

$$-\mathbb{E}_{t,p_0,r_0,r_t}\left[\frac{1}{t}\sum_{i=1}^{L'}\mathbf{1}[r_t^i=\mathrm{M}]\log p_\theta(r_0^i|p_0,r_t)\right],$$

3. Diffusion 在文本任务中的应用：LLaDA

3.4 LLaDA 的训练流程（预训练+SFT）：

- **SFT：**
  - 示意图：



3. Diffusion 在文本任务中的应用：LLaDA

3.5 LLaDA 的推理流程：

- **推理流程介绍：**
  - 训练任务LLaDA 的推理流程就是将输入的噪声 <MASK>逐渐还原成有语义的token。我们在上文中已经给了一个简化的去噪例子。
  - 区别在于，LLaDA 的实现中并不是每个时间步只去噪一部分，实际上会有两步：1.把所有的噪声都去掉，2.Remask 回一部分。叠加之后，相当于每步只去噪了一部分。

3. Diffusion 在文本任务中的应用：LLaDA

3.5 LLaDA 的推理流程：

- **推理流程介绍：**
  - 还是用上面的例子，假设去噪的时间步为4，每次实际去噪的比例就是：1/4* 100%=25%：

Step 1：＜MASK> <MASK> <MASK> <MASK> <MASK> <MASK> <MASK><MASK>

Step 2（去噪）：今天树叶飞鸟游鱼

Step 2'（Remask）：今天 <MASK> <MASK> <MASK> <MASK> <MASK> <MASK>

Step 3（去噪）：今天黑神话不错吗

Step 3'（Remask）：今天 <MASK> <MASK> <MASK> 不错 <MASK>

Step 4（去噪）：今天海洋真不错啊

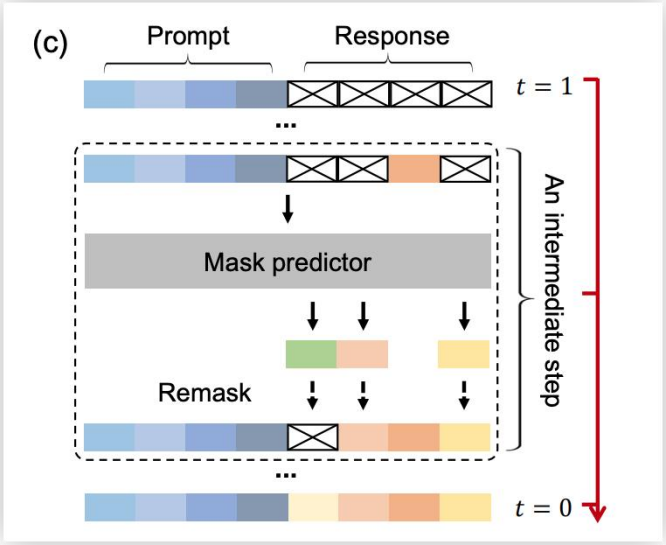
Step 4'：今天<MASK><MASK>真不错啊

Step 5（去噪）：今天天气真不错啊

3. Diffusion 在文本任务中的应用：LLaDA

3.5 LLaDA 的推理流程：

- **Remask策略：**
  - 随机 Remask
  - **保留 Confidence 最高的，剩下的 Remask（效果最佳）**
  - 模仿 Auto-Regressive 模型，每次保留左边，Remask 右边
- **流程图：**



3. Diffusion 在文本任务中的应用：LLaDA

■ 3.6 LLaDA 的定量性能评估：

Table 1: **Benchmark Results of Pre-trained LLMs.** * indicates that models are evaluated under the same protocol, detailed in Appendix B.6. Results indicated by † and ¶ are sourced from Yang et al. [25, 26] and Bi et al. [32] respectively. The numbers in parentheses represent the number of shots used for in-context learning. "-" indicates unknown data.

	LLaDA 8B*	LLaMA3 8B*	LLaMA2 7B*	Qwen2 7B†	Qwen2.5 7B†	Mistral 7B†	Deepseek 7B¶
Model	Diffusion	AR	AR	AR	AR	AR	AR
Training tokens	2.3T	15T	2T	7T	18T	-	2T
General Tasks							
MMLU	**65.9** (5)	65.4 (5)	45.9 (5)	70.3 (5)	74.2 (5)	64.2 (5)	48.2 (5)
BBH	49.7 (3)	**62.1** (3)	39.4 (3)	62.3 (3)	70.4 (3)	56.1 (3)	39.5 (3)
ARC-C	45.9 (0)	**53.1** (0)	46.3 (0)	60.6 (25)	63.7 (25)	60.0 (25)	48.1 (0)
Hellaswag	70.5 (0)	**79.1** (0)	76.0 (0)	80.7 (10)	80.2 (10)	83.3 (10)	75.4 (0)
TruthfulQA	**46.1** (0)	44.0 (0)	39.0 (0)	54.2 (0)	56.4 (0)	42.2 (0)	-
WinoGrande	74.8 (5)	**77.3** (5)	72.5 (5)	77.0 (5)	75.9 (5)	78.4 (5)	70.5 (0)
PIQA	73.6 (0)	**80.6** (0)	79.1 (0)	-	-	-	79.2 (0)
Mathematics & Science							
GSM8K	**70.3** (4)	48.7 (4)	13.1 (4)	80.2 (4)	85.4 (4)	36.2 (4)	17.4 (8)
Math	**31.4** (4)	16.0 (4)	4.3 (4)	43.5 (4)	49.8 (4)	10.2 (4)	6.0 (4)
GPQA	25.2 (5)	**25.9** (5)	25.7 (5)	30.8 (5)	36.4 (5)	24.7 (5)	-
Code							
HumanEval	**35.4** (0)	34.8 (0)	12.8 (0)	51.2 (0)	57.9 (0)	29.3 (0)	26.2 (0)
HumanEval-FIM	**73.8** (2)	73.3 (2)	26.9 (2)	-	-	-	-
MBPP	40.0 (4)	**48.8** (4)	23.2 (4)	64.2 (0)	74.9 (0)	51.1 (0)	39.0 (3)
Chinese							
CMMLU	**69.9** (5)	50.7 (5)	32.5 (5)	83.9 (5)	-	-	47.2 (5)
C-Eval	**70.5** (5)	51.7 (5)	34.0 (5)	83.2 (5)	-	-	45.0 (5)

其中作者声称，LLaDA **的主要** baseline **是** LLaMA2 7B **和** LLaMA3 8B**。从结果看，**LLaDA **优于**LLaMA 2 7B**，和** LLaMA3 8B **处于可比的水平。**

3. Diffusion 在文本任务中的应用：LLaDA

■ 3.7 LLaDA 的定性 Case Study

Table 3: **Visualization of the Sampling Process and a Generated Multi-round Dialogue.** In the response of LLaDA, darker colors indicate tokens predicted in the later stages of sampling, while lighter colors correspond to earlier predictions.

Sampling Process	
User	Lily can run 12 kilometers per hour for 4 hours. After that, she runs 6 kilometers per hour. How many kilometers can she run in 8 hours?
LLaDA	In 4 hours , Lily runs 12 * 4 = 48 kilometers . After 4 hours she runs 6 * 4 = 24 kilometers . In total she runs 48 + 24 = 72 kilometers in 8 hours . Result 72

浅色代表先生成，深色代表后生成。从生成顺序上看，LLaDA 最开始生成的，是人名和其他名词；到后期才会生成需要推理得到的数字结果。总体上，生成顺序是先生成简单的、更容易确定的token，再生成困难的token。

3. Diffusion 在文本任务中的应用：LLaDA

3.8 LLaDA 的意义：

- 证明了扩散模型在 LLMs上的可行性，为未来的研究开辟了新的方向
- 扩散模型的并行范式，具有大幅缩短推理时间的潜质

3.9 LLaDA 的限制：

- 输出定长。需要预先更定输出的长度，不够灵活。
- 训练开销大。文章中报告预训练的时间是 0.13 milion H800 hours，相当于300卡 H800连续训练15天，每天24小时。
- 无配套框架。目前尚未有针对性优化的推理框架，推理开销很高。

3. Diffusion 在文本任务中的应用：LLaDA

3.8 LLaDA 的意义：

- 证明了扩散模型在 LLMs上的可行性，为未来的研究开辟了新的方向
- 扩散模型的并行范式，具有大幅缩短推理时间的潜质

3.9 LLaDA 的限制：

- 输出定长。需要预先更定输出的长度，不够灵活。
- 训练开销大。文章中报告预训练的时间是 0.13 milion H800 hours，相当于300卡 H800连续训练15天，每天24小时。
- 无配套框架。目前尚未有针对性优化的推理框架，推理开销很高。

4.从文本任务走向多模态：LLaDA-V, LaViDa

■ 4.1 LLaDA–V (LLaDA–V: Large Language Diffusion Models with Visual Instruction Tuning)

文章概览：

这篇文章和市面上训练MLLM的文章（如LLaVA）的最大区别，可能就是把LLM 部分替换成了LLaDA。做法非常直接的一篇文章。

- **架构：**采用经典的「视觉编码器+MLP投影器+语言模型」架构。视觉编码器（SigLIP 2）提取图像特征，MLP投影器将其映射到 LLaDA的嵌入空间。LLaDA 语言塔则负责处理融合后的多模态输入并生成回复。特别地，LLaDA–V采用了双向注意力机制，允许模型在预测时全面理解对话上下文。
- **训练阶段：**
  - Stage 1（对齐编码器）：对齐视觉编码器和语言模型的embedding。冻结视觉编码器和语言模型，只训练MLP
  - Stage 2（指令微调）：Visual Instruction Tuning
  - Stage 3（更难任务的指令微调）：Multimodal Reasoning Enhancement

4.从文本任务走向多模态：LLaDA-V, LaViDa

模型结构示意图：

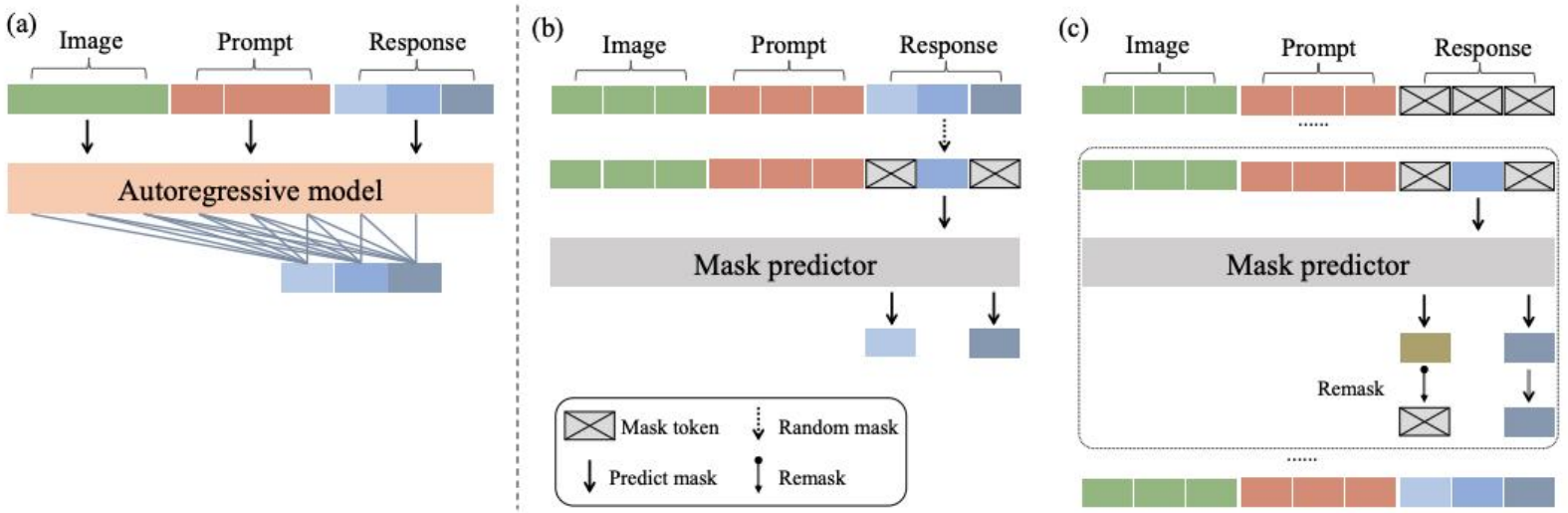
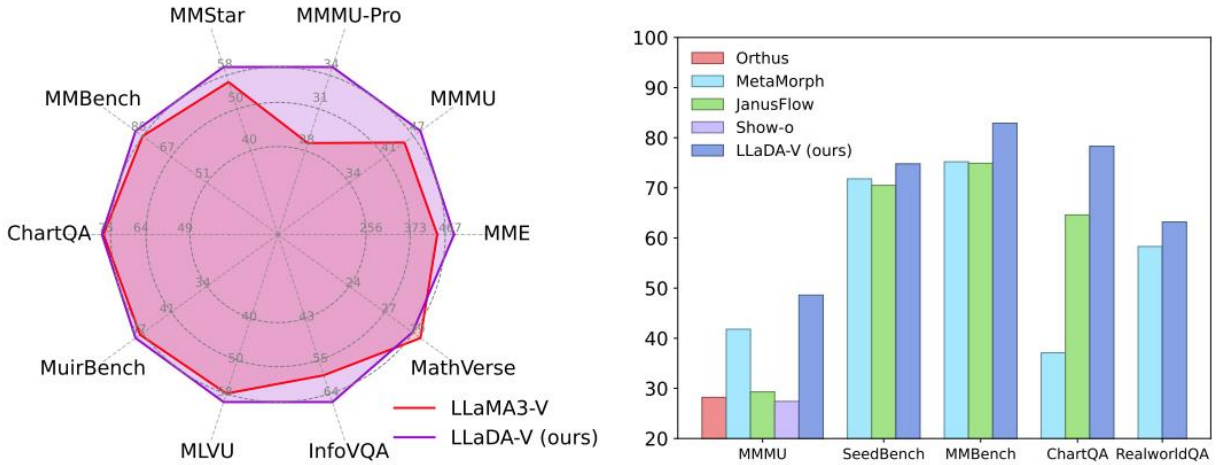


Figure 2: **Overview of Autoregressive Approaches and LLaDA-V.** Image representations are generated by an encoder and an MLP projector (not explicitly shown). (a) Autoregressive Training: Given image features and the input prompt, autoregressive models are trained to predict the response through next-token prediction. (b) LLaDA-V's Training: Image features and the input prompt remain unmasked, while only the response is randomly masked. (c) LLaDA-V's Inference: As time step t decreases from 1 to 0, generation begins with a fully masked response and iteratively predicts tokens.

4.从文本任务走向多模态：LLaDA-V, LaViDa

核心结论：

LLaDA–V展现出更强的数据可扩展性，特别是在多学科知识（如MIMMU）基准上。尽管 LLaDA–8B **在纯文本任务上略逊于** LLaMA3–8B，**但** LLaDA–V **在11个 多模态任务中超越了**LLaMA3–V。**这表明扩散架构在多模态任务上面具备一定的优势。**



(a) LLaDA-V vs. LLaMA3-V (b) LLaDA-V vs. non-autoregressive MLLMs

Figure 1: **Benchmark Results.** (a) LLaDA-V demonstrates superior performance on more benchmarks compared to LLaMA3-V when trained on the same dataset, particularly excelling in multidisciplinary knowledge and mathematical reasoning tasks. (b) LLaDA-V achieves state-of-the-art performance in multimodal understanding among both hybrid autoregressive-diffusion (such as MetaMorph [31] and Show-o [28]) and purely diffusion-based models.

4.从文本任务走向多模态：LLaDA-V, LaViDa

■ 4.2 LaViDA （LaViDa: A Large Diffusion Language Model for Multimodal Understanding）

文章概览：

这篇文章的模型架构和上一篇LLaDA-V**非常类似。但是除去模型架构外，本文有许多有趣的想法。创新点可以总结成：**1. 训练数据增强 2. 设计 KV cache 加速 3. 设计解码 Schedule

- **架构：**采用模型架构和LLaDA-V几乎一样。采用经典的「视觉编码器+MLP投影器＋语言模型」架构。视觉编码器（SigLIP 2）提取图像特征，MLP投影器将其映射到 LLaDA的嵌入空间。LLaDA 语言塔则负责处理融合后的多模态输入并生成回复。

4.从文本任务走向多模态：LLaDA-V, LaViDa

- **架构示意图：**

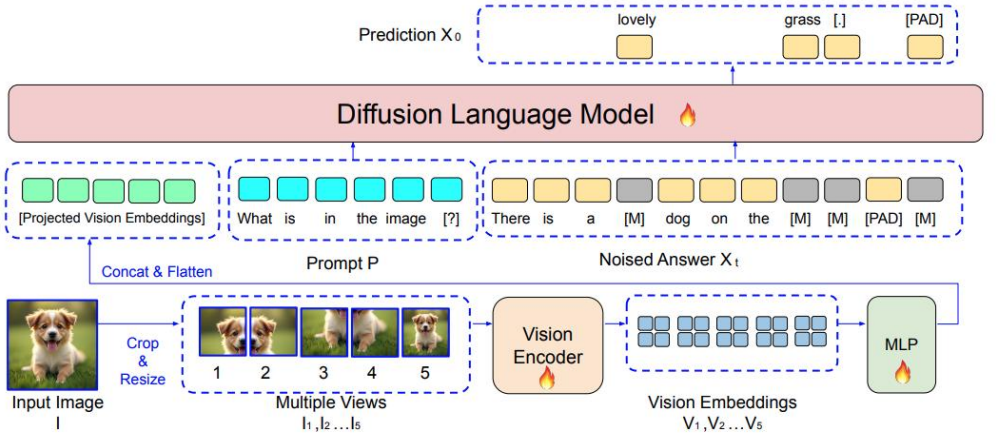


Figure 2: **Overall design of LaViDa.** LaViDa's architecture consists of a vision encoder, a diffusion language model, and an MLP vision projector. The bottom half of the figure illustrates the image encoding process, while the top half depicts the diffusion language modeling process. These two pipelines are described in detail in Sec. 3.1.

- **训练阶段（同样和LLaDA-V类似）：**
  - Stage 1（对齐编码器）：对齐视觉编码器和语言模型的embedding。冻结视觉编码器和语言模型，只训练MLP
  - Stage 2（指令微调）：Visual Instruction Tuning
  - Stage 3（更难任务的指令微调）：Reasoning Distillation

小结：本文的模型架构和训练过程同样是常规的，但是本文还有另外三个创新点。

4.从文本任务走向多模态：LLaDA-V, LaViDa

核心创新点：

- **Complementary Masking.** 在正向加噪的过程中，每个样本除了有随机加噪的版本，还有另一个噪声取反的版本。举个例子：假设输入样本为：今天天气真不错啊。对应会有两个加噪的样本：样本1：今天<MASK><MASK>真不错啊。样本2：<MASK><MASK>天气<MASK><MASK><MASK><MASK>。作者声称，这种方式能显著增加样本的利用率，因为每个token都用到了。
- **KV-Cache.** Diffusion Language Model构建KV-Cache 有两个核心困难：1. 模型是双向的注意力，新生成的token会影响之前token序列的KV值；2.Token 生成顺序是不固定的。为了解决这两个困难，本文的想法是为作为前缀的图像 Token 和文本Prompt的Token计算单次 KV-Cache，后续不再更新。文章把这个范式称为Prefix-LM。文章证明，虽然冻结 Prefix的KV-Cache 会让性能略微下降，但是推理的时间开销会大幅降低。
- **Schedule Shift.** 本文的想法是，去噪阶段未必一定要均匀去噪。比如开始去噪的 token 多，后续的少；或者开始去噪的少，后续的多。去噪 Schedule 的形式化表达是：

$$t_i' = s_\alpha(t_i) = \frac{\alpha t_i}{1 + (\alpha - 1)t_i}$$

4.从文本任务走向多模态：LLaDA-V, LaViDa

核心实验结果：

- **Complementary Masking.**

(a) Effect of complementary masking.

	w/o Comp.M.	w/ Comp.M.
MME↑	260.00	**297.00**
MathVista↑	28.40	**33.40**
ScienceQA↑	48.74	**81.49**
MMMU↑	38.56	**41.78**
Runtime↓	**8.2 hr**	8.9 hr

- **KV-Cache.**

(a) Effect of KV Cache

Method	NFE	CIDEr ↑	Latency↓
Full-DLM	100%	**121.0**	7.65
Prefix-DLM	100%	117.3	1.93
Full-DLM	50%	118.6	4.13
Prefix-DLM	50%	114.8	**1.23**
Open-Lnxt-8B	–	111.8	1.71

- **Schedule Shift.**

(b) Effect of timestep shifting

	COCO Caption (CIDEr)↑			
NFE	25%	50%	75%	100%
cosine	87.7	102.2	110.8	117.3
linear	84.9	105.2	108.6	117.3
α=3	48.7	74.7	92.4	117.3
α=3^{-1}	**101.1**	**114.8**	**117.3**	**117.3**

4.从文本任务走向多模态：LLaDA-V, LaViDa

Take Away Messages：

- Complementary Masking 可以在少量增加训练开销的情况下，大幅提升模型的训练效果。这种方法的道理是增强训练样本的利用率。
- 使用 Prefix-LM的KV-Cache 可以在基本保持性能的情况下，大幅降低时间开销。
- 如果要使用 diffusion 的范式来加速推理（e.g. 每次去噪多个token），比较好的去噪策略是刚开始快速去噪，后续再缓慢精细去噪。

4. Rethinking LLaDA: Masked Model or Diffusion?

- **社区评价。有原教旨主义的** diffusion **研究人员，批评**LLaDA **更像是** masked model**，也就是大号的**Bert**，而非** Diffusion model**。**
- **理论研究。清华和** NVIDIA **在**ICLR 2025 **的文章也在理论上证明，**masked diffusion models **由于不含时的特性，实际上等价于** masked model**。文章名：**“MASKED DIFFUSION MODELS ARE SECRETLY TIMEAGNOSTIC MASKED MODELS AND EXPLOIT INACCURATE CATEGORICAL SAMPLING"
- **模型特性分析。**LLaDA**的核心特性：**

1. **加噪：使用**<MASK>**随机替换原始** token**，而非用词表里的词随机替换。**
2. **去噪：不需要显式编码时间。因为模型可以根据** <MASK>**的数量隐式推断时间点。而更原教旨的** diffusion **是含时的** SDE**，是否含时是显著的差异。**
3. Remask: LLaDA **最终使用的**Remask **策略是** Confidence-based**的策略，一种比较启发式的方法。**

4. Rethinking LLaDA: Masked Model or Diffusion?

- **演化路径。从** Discreate Diffusion Model –> Mased Diffusion Model –> Bert**的演化路径：**

Discrete Diffusion Model –> **采用** mask **转移核（而非** uniform **词表替换）** –> Masked Diffusion Model –> **证明其时间** t **在输入中是冗余的** –> **输入随机比例掩码改为固定比例掩码** –> BERT

上述演化路径表明，除去理论证明外，实践上 MDM **和** Bert **的核心区别就是随机比例掩码/固定比例掩码。**

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

智能算法安全重点实验室
Key Laboratory of AI Safety, CAS

Thank you for your attentions!