

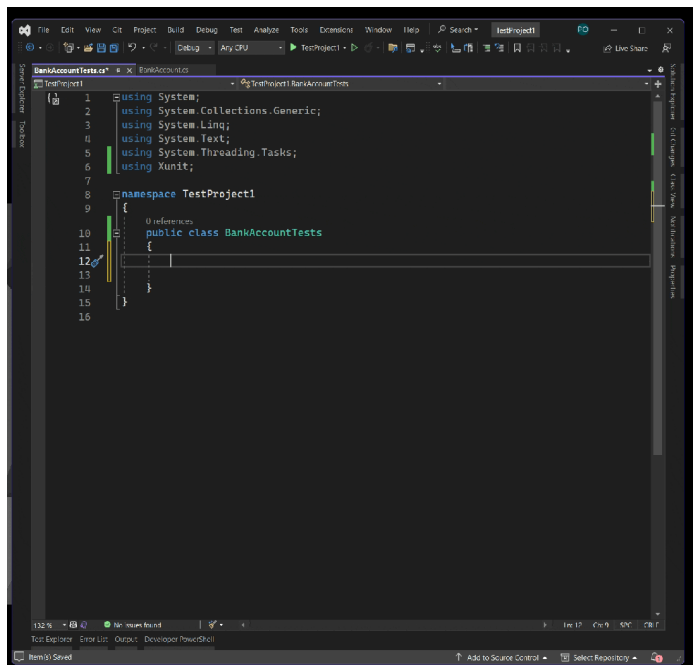


# 高效多模态生成方法与应用

邓志杰

上海交通大学

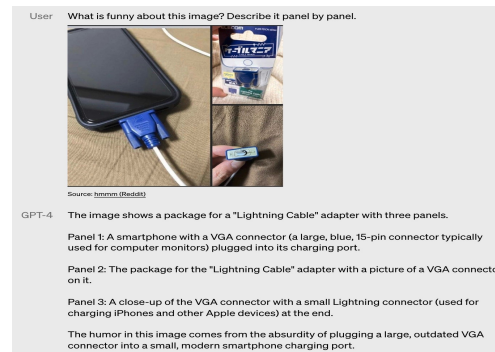
# 背景：语言生成已产生巨大实用价值



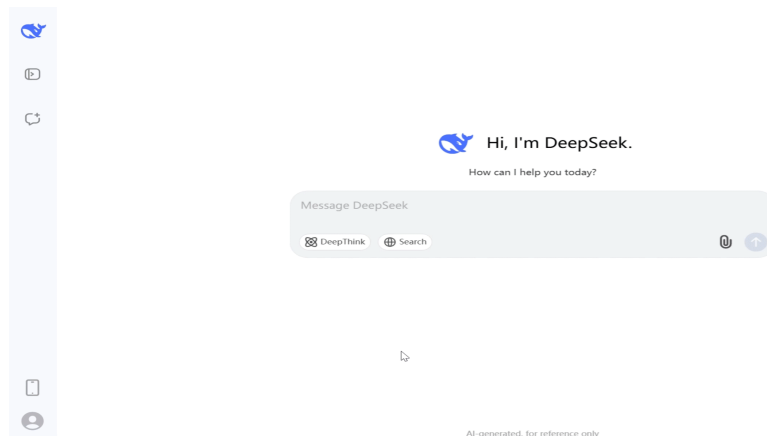
Coding assistant



Software development



Multimodal understanding



AI-generated, for reference only

Reasoning

## 背景：视觉生成构建起“世界模拟器”



*Sora by OpenAI*



*Vidu by ShengShu*

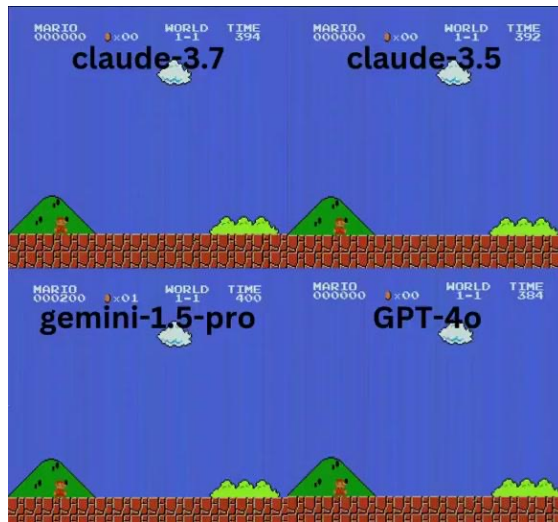


*W.A.L.T*

# 生成式大模型的最终形态: Agent (MLLM + memory + planning skills + tool use)



**Manus (a newest agent even better than OpenAI Deep Research)**



**Game Agent**



**Embodied Agent**

**Virtual vs. reality**

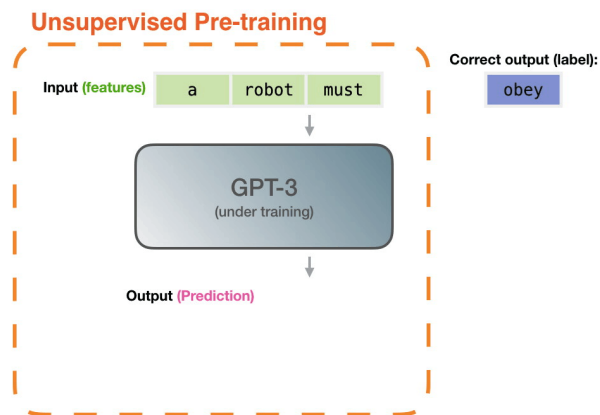




## 生成式大模型需要如何发展？

- Diverse use case -> **Cross modality** is needed
- Memory -> Long context -> **Efficiency** matters
- Planning + tool use -> **Reason** is important
- 我们需要： **高效多模态**生成，同时可以**慢思考**

## 挑战1：语言、视觉生成范式存在分歧



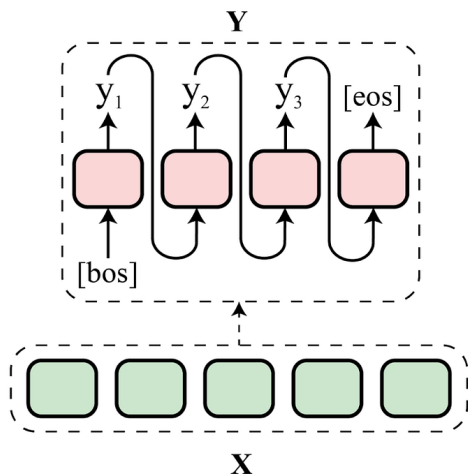
VS.



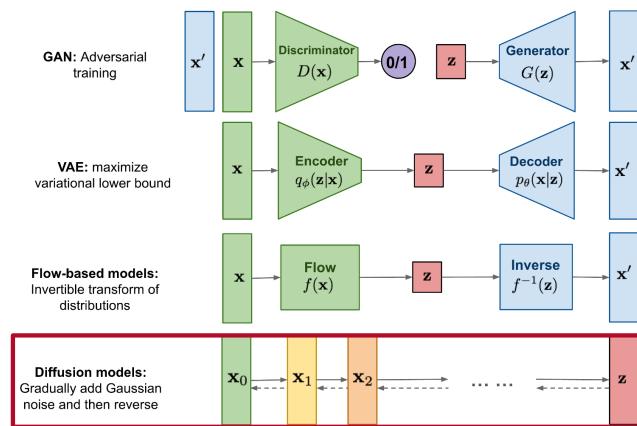
语言：自回归生成，刻画长程依赖

图像：扩散建模，准确预测连续细节

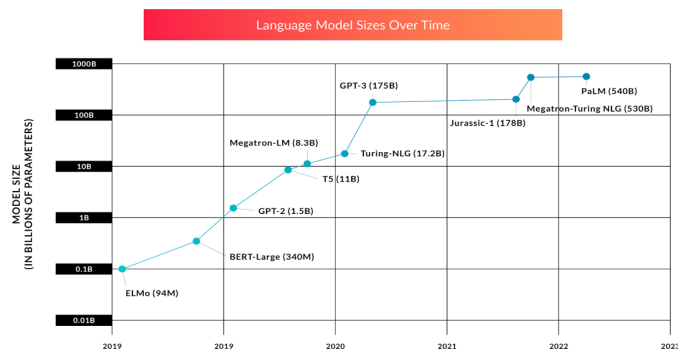
## 挑战2: 顺序推理效率低, 速度慢, 成本高



每个token需要一次模型前传



每个去噪步需要一次模型前传



模型本身的大尺寸 (更多的flops、内存占用)



## 挑战3：对推理能力的兼顾为模型效率提出了新的要求

Dynasor-CoT: training-free and plug-and-play LLM reasoning acceleration

[Twitter](#) | [Blog](#) | [Code](#) | [Paper](#)

How to use Dynasor-CoT?

Token Saving: 0    Speedup: 1.00x    Pending...

Baseline    Dynasor-CoT (Adaptive Compute)

Enter your Question

The point  $S(a, b)$  lies on the line with the equation  $53x + 2y = 12.5$  When  $5a = 45$ , what is the value of  $5b$ ?

Submit    Random

生成的思维链中包含大量冗余“自我怀疑”

OpenAI o3

What impact did recent breakthroughs in battery technology have on electric vehicle range, charging speed, and adoption rates compared to previous generations? Gather relevant scientific studies, industry adoption data, and technical specifications; create visual comparisons, and summarize key scientific advancements.

Reasoned for 40 seconds ^

17 searches · 26 sources

- There are numerous recent breakthroughs in battery technology, such as silicon anodes, solid-state batteries (like those from Toyota and QuantumScape), and advancements in lithium iron phosphate (LFP) tech from companies like CATL and BYD. Additionally, there's progress with sodium-ion and lithium-metal

OpenAI o1

What impact did recent breakthroughs in battery technology have on electric vehicle range, charging speed, and adoption rates compared to previous generations? Gather relevant scientific studies, industry adoption data, and technical specifications; create visual comparisons, and summarize key scientific advancements.

Reasoned for 5 seconds ^

I'm pulling together info on scientific studies, industry data, and tech specs about recent battery tech breakthroughs in electric vehicles. Important topics include high-density lithium-ion, solid-state batteries, and advancements from Tesla and CATL.

I'm summarizing key scientific advancements and creating a visual comparison. This includes highlighting breakthroughs like lithium-ion

OpenAI o3 delivers a comprehensive, accurate, and insightful analysis of how recent battery technology breakthroughs are extending EV range, speeding up charging, and driving adoption, all backed by scientific studies and industry data. o1, while credible and on-topic, is less detailed and forward-looking, with minor inaccuracies or oversimplifications.

工具调用融合的推理有更高的latency

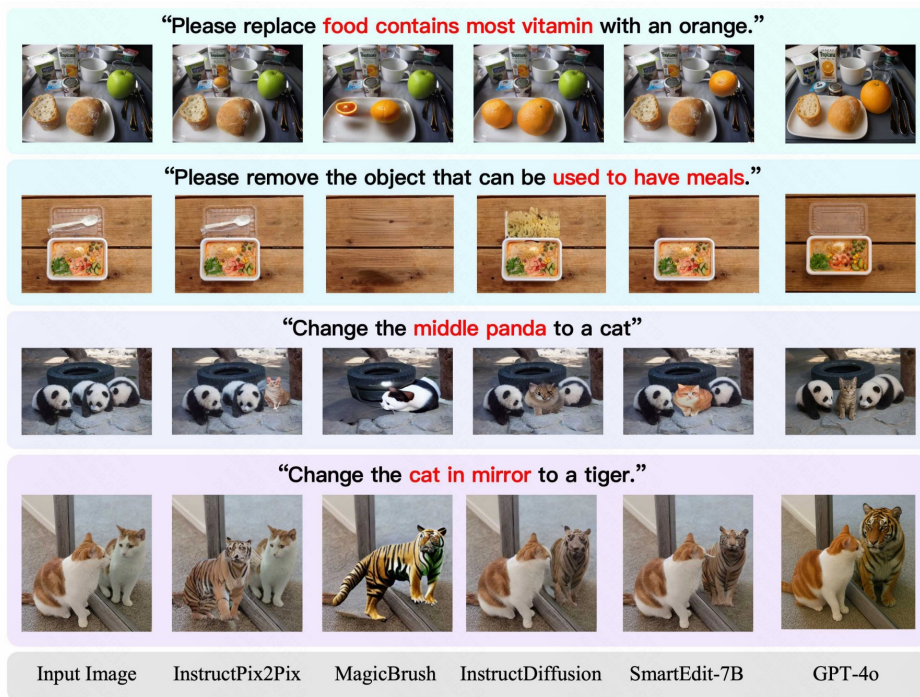


# 1、跨模态统一、兼具生成和理解能力的模型



## 为什么要跨模态生成理解统一？ GPT-4o的例子

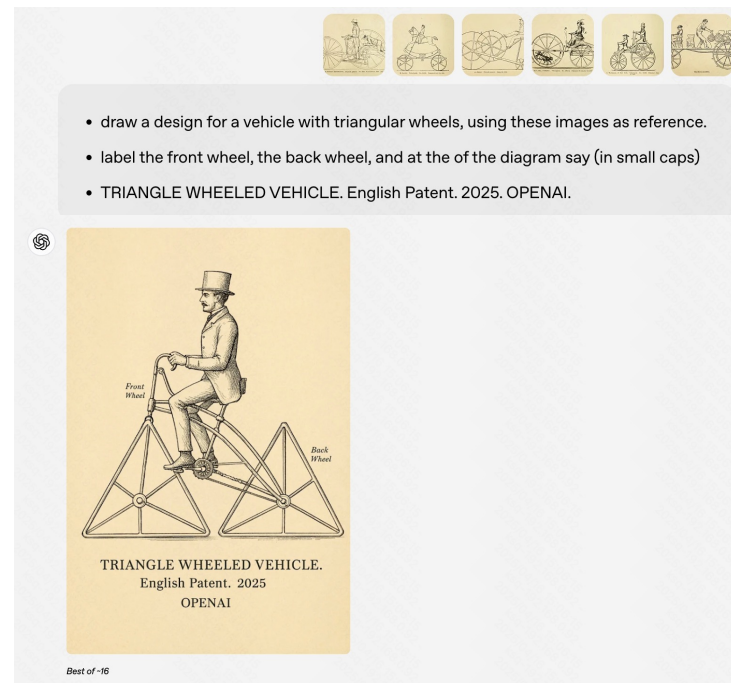
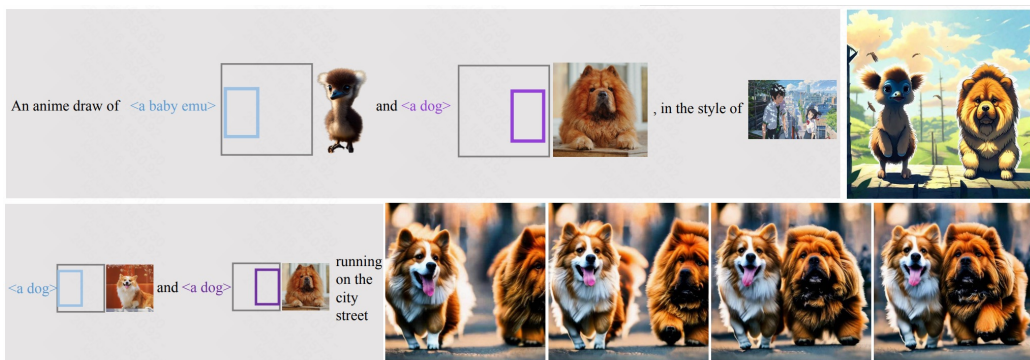
- 相较于专用的图像生成模型，统一语言和视觉建模有助于建立世界知识
- 在传统编辑任务中的指令理解与跟随能力显著增强



# 为什么要跨模态生成理解统一? GPT-4o的例子

- 统一模型天然具备长上下文学习能力

处理多图与文本混合输入时, 统一模型能够有效整合多模态信息, 展现了控制精准、主体一致性强的生成效果



# 如何将跨模态生成理解统一？模型桥接

- 模型桥接的方式能轻易实现跨模态生成

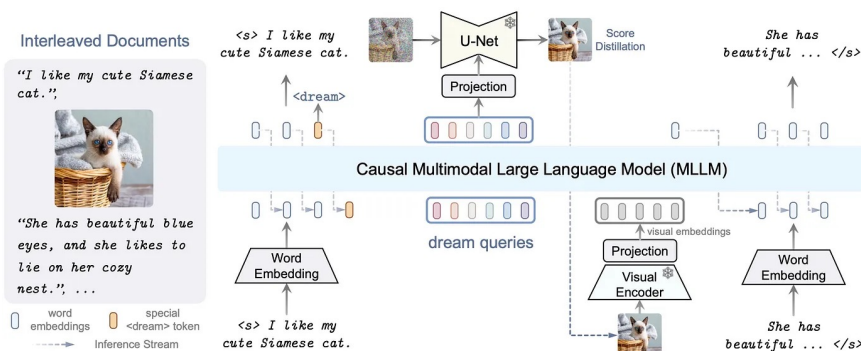
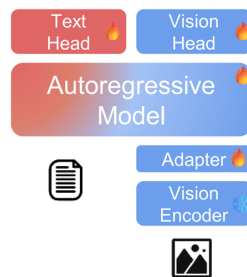


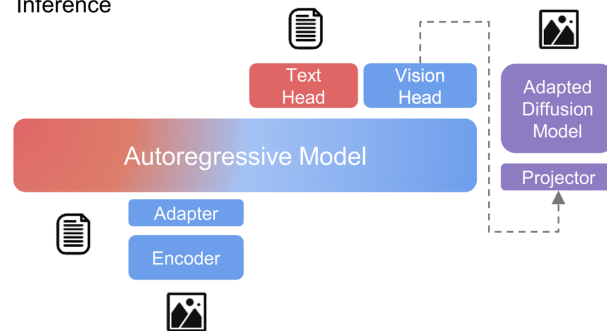
Figure 2: **Overview of our DREAMLLM framework.** Interleaved documents serve as input, decoded to produce outputs. Both text and images are encoded into sequential, discrete token embeddings for the MLLM input. A special <dream> token predicts where to generate images. Subsequently, a series of *dream queries* are fed into the MLLM, capturing holistic historical semantics. The images are synthesized by the SD image decoder conditioned on queried semantics. The synthesized images are then fed back into the MLLM for subsequent comprehension.

DreamLLM [<https://arxiv.org/abs/2309.11499>]

VPiT



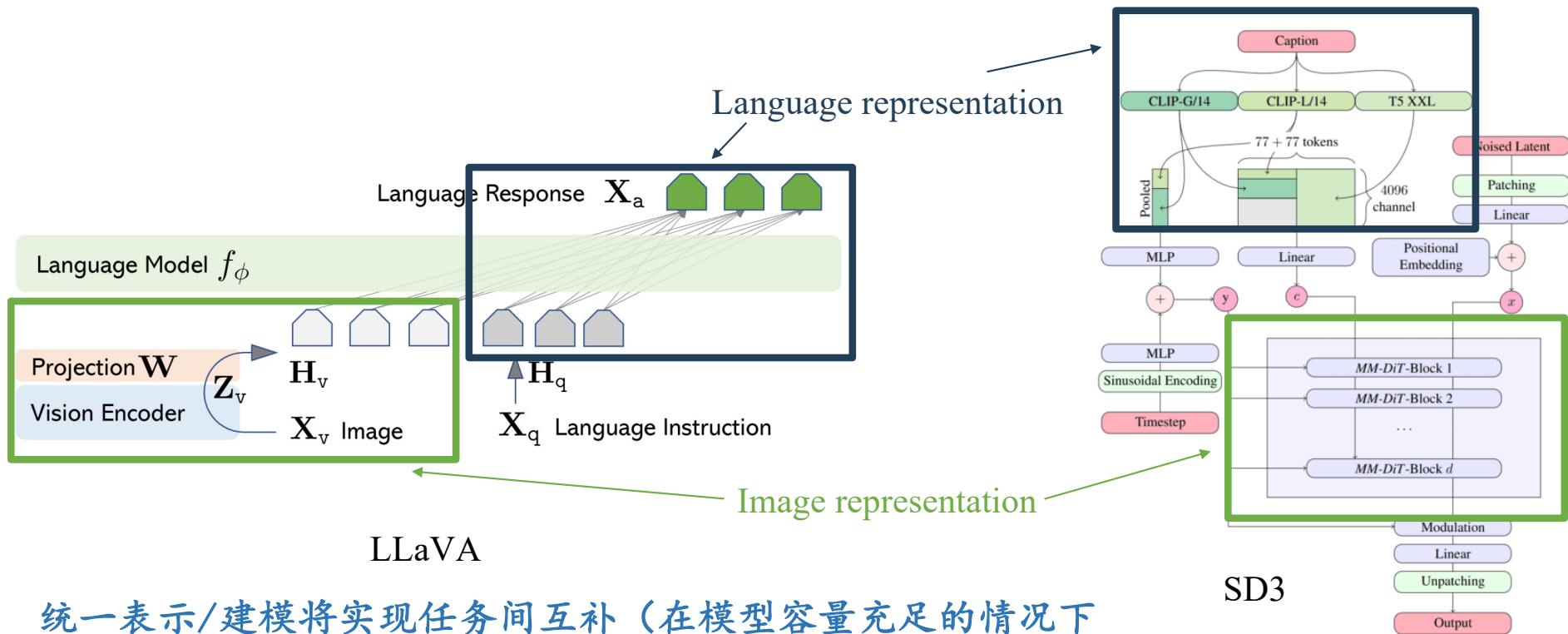
Inference



MetaMorph [<https://arxiv.org/pdf/2412.14164>]

# 如何将跨模态生成理解统一？ 模型桥接

- 扩散模型与语言模型中的 图像/文本表示冗余



统一表示/建模将实现任务间互补（在模型容量充足的情况下）

# 如何将跨模态生成理解统一？自回归

- 图像离散化，统一自回归（但是离散化会丢信息）

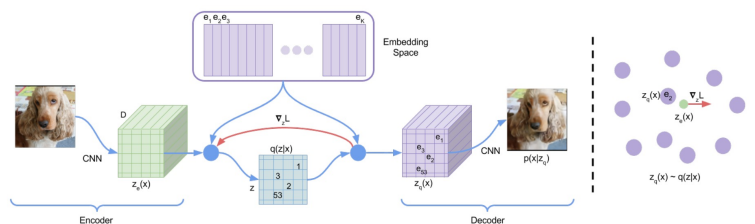
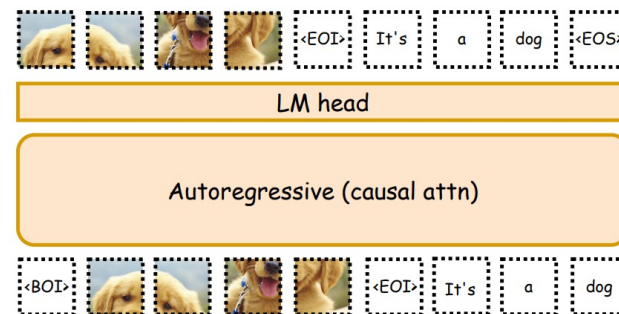
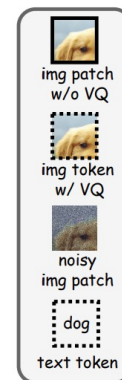


Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder  $z(x)$  is mapped to the nearest point  $e_2$ . The gradient  $\nabla_z L$  (in red) will push the encoder to change its output, which could alter the configuration in the next forward pass.

VQ-VAE编码丢失细节

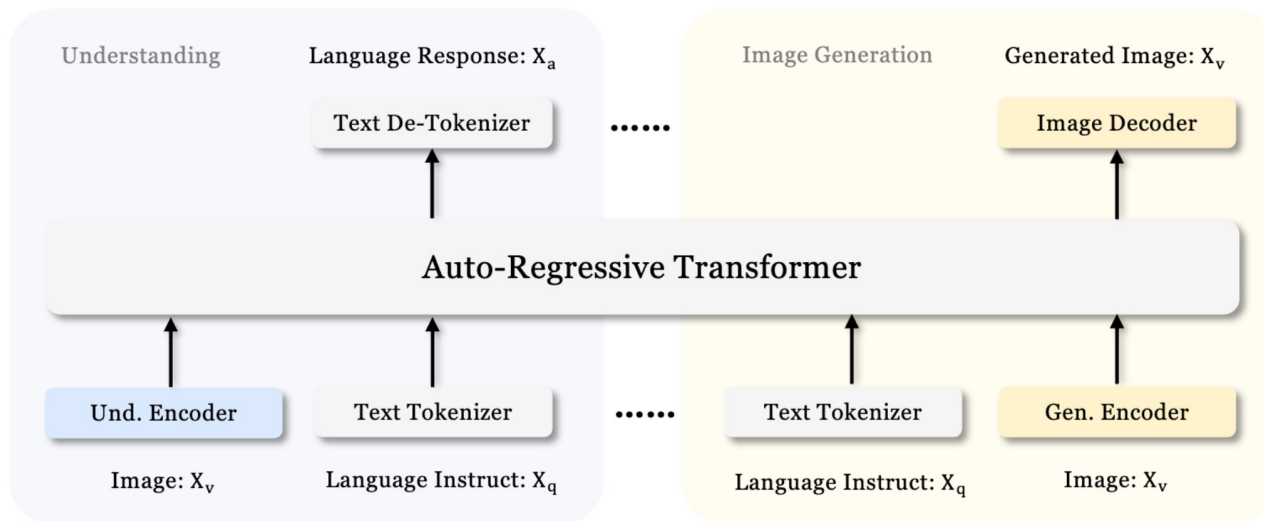


Chameleon, EMU3, etc.



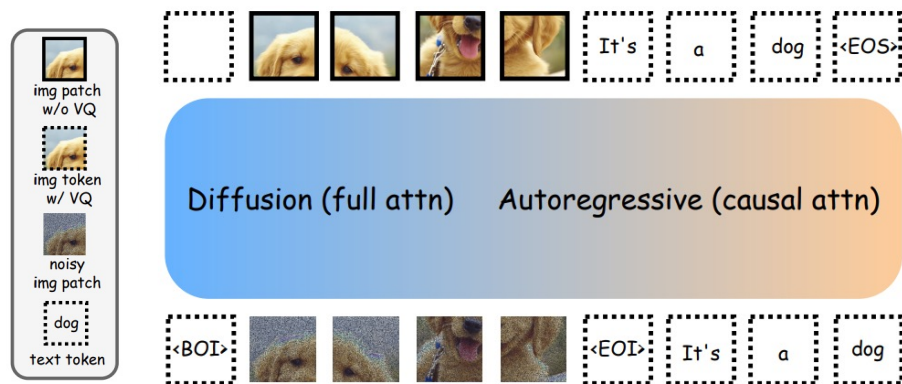
## 如何将跨模态生成理解统一？自回归

- 图像离散化，统一自回归（但是离散化会丢信息）
  - DeepSeek Janus-Pro: 为图像理解和生成分别使用连续和离散编码器



## 如何将跨模态生成理解统一？ 参数共享的图像扩散+文本自回归

- 对于图像，扩散建模



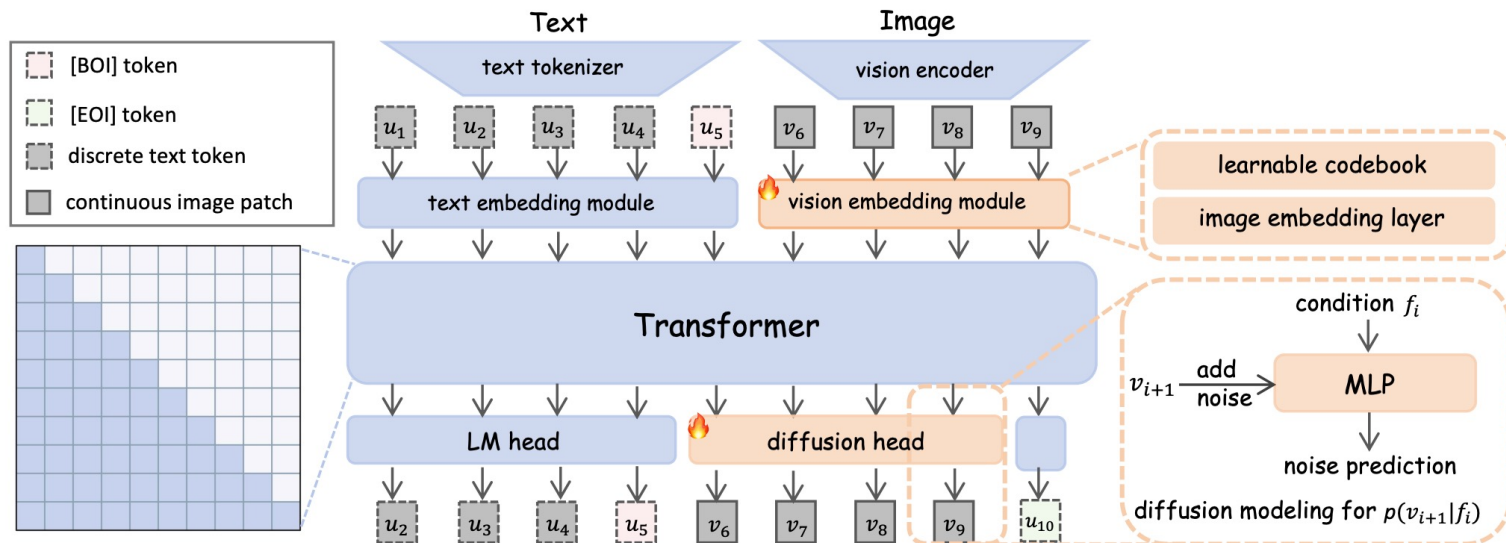
**Transfusion**

图文交错训练效率低：

- 图-文-图-文-图，只能在最后一张图算loss

图像生成不能使用KV Cache

# 如何将跨模态生成理解统一? Orthus!



- 自回归**Transformer**主干（拥抱**KV Cache**）
- 处理离散的文本**token**和**连续的图像feature**（基于连续VAE）
- 基于线性层定义的**language head**和**diffusion MLP**来分别生成文和图（逐**token/patch**）

# 如何将跨模态生成理解统一? Orthus!

- 从离散图像特征到连续特征:

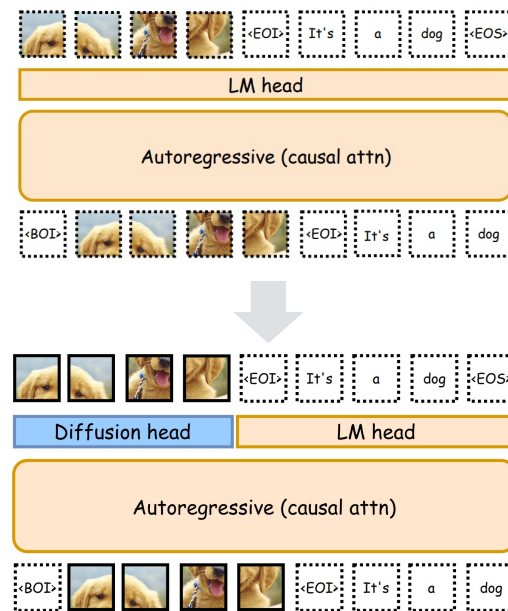
$$h_i = \sum_j w_j \mathbb{1}_{\tilde{v}_i=j}, \tilde{v}_i = \arg \min_{j \in \{1, \dots, K\}} d(\mathbf{v}_i, \mathbf{c}_j)$$

$$\Rightarrow h_i = \sum_j w_j \frac{e^{-d(\mathbf{v}_i, \mathbf{c}_j)/\tau}}{\sum_{k=1}^K e^{-d(\mathbf{v}_i, \mathbf{c}_k)/\tau}}$$

- 自回归统一模型 (如: **Chameleon**) :  $\tau=0$

- Orthus:  $\tau=1$**

- 从  $\tau=0$  的模型冷启动
  - 72个A100 GPU hours即可得到Orthus-7B-base
- 将涉及的VQ-VAE调成了VAE



Model	PSNR $\uparrow$	SSIM [63] $\uparrow$
VQ-VAE	23.7	0.80
Ours	26.1	0.84

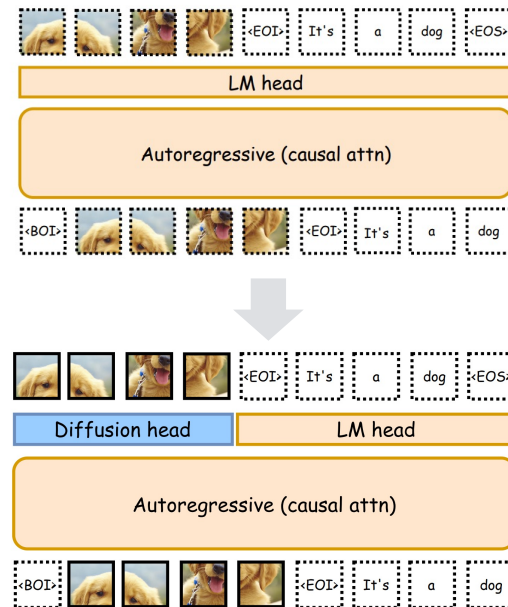
# 如何将跨模态生成理解统一? Orthus!

- **Diffusion head**训练:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\epsilon, t} [\|\epsilon - \epsilon_{\theta}(\sqrt{\alpha_t} \mathbf{v}_{i+1} + \sqrt{1 - \alpha_t} \epsilon, t, \mathbf{f}_i)\|_2^2]$$

- 从文到图/图到文等不同任务学习有价值信号
  - 1:1混合LlaVA-v1.5-665K指令微调数据和高质量文生图数据JourneyDB、LAION-COCO-aesthetic (recaptioned from ShareGPT-4v)

$$\mathcal{L}_{\text{Orthus}} = \mathcal{L}_{\text{ar}} + \lambda \mathcal{L}_{\text{diff}}$$





## Orthus: 文生图/图生文量化结果

- 在多个图像理解指标上超越了现有混合理解生成模型**Chameleon**和**Show-o**，并在文到图生成的

### GenEval 指标上超过SDXL

Table 3. Comparison with state-of-the-arts on visual generation benchmarks. Model using external pre-trained diffusion model is marked with \* and Chameleon<sup>†</sup> is post-trained with the same dataset as Orthus. The results in **bold** and underline are the best and second-best results, respectively.

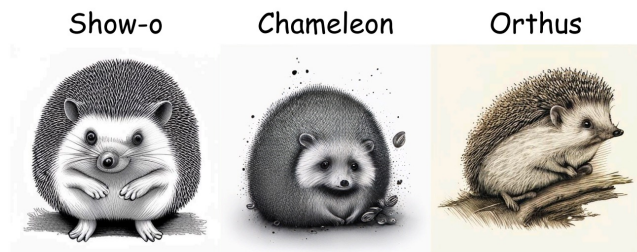
Type	Model	Res.	GenEval	HPS
Gen. Only	SDv1.5 (Rombach et al., 2022)	512	0.43	27.0
	SDv2.1 (Rombach et al., 2022)	512	0.50	27.2
	DALL-E (Ramesh et al., 2022)	512	0.52	26.9
	Emu3-Gen (Wang et al., 2024)	512	0.54	-
	SDXL (Podell et al., 2023)	512	0.55	30.9
	SD3(d=30) (Esser et al., 2024)	512	0.64	-
Und. & Gen.	SEED-X* (Ge et al., 2024)	448	0.49	-
	LWM (Liu et al., 2024e)	256	0.47	26.1
	Show-o (Xie et al., 2024)	256	0.53	27.3
	Transfusion (Zhou et al., 2024)	256	<b>0.63</b>	-
	Chameleon <sup>†</sup>	512	0.43	26.9
	<b>Orthus (Ours)</b>	512	<u>0.58</u>	<b>28.2</b>

Table 2. Evaluation on visual understanding benchmarks. Und. and Gen. denote “understanding” and “generation”, respectively. Models using external pre-trained diffusion models are marked with \* and Chameleon<sup>†</sup> is post-trained with the same dataset as Orthus. The results in **bold** and underline are the best and second-best results, respectively. The results correspond to the exact match accuracy.

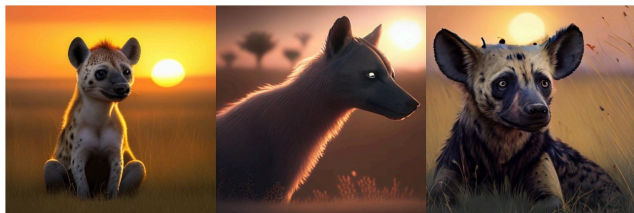
Type	Model	# Params	POPE↑	MME-P↑	VQAv2↑	GQA↑	MMMU↑
Und. Only	LlaVa (Liu et al., 2024d)	7B	76.3	809.6	-	-	-
	LlaVA-v1.5 (Liu et al., 2024b)	7B	85.9	1510.7	78.5	62.0	35.4
	InstructBLIP (Dai et al., 2023)	7B	-	-	-	49.2	-
	Qwen-VL-Chat (Bai et al., 2023)	7B	-	1487.5	78.2	57.5	-
	Emu3-Chat (Wang et al., 2024)	8B	85.2	1243.8	75.1	60.3	31.6
	InstructBLIP (Dai et al., 2023)	13B	78.9	1212.8	-	49.5	-
Und. and Gen.	Emu* (Sun et al., 2023)	13B	-	-	52.0	-	-
	NExT-GPT* (Wu et al., 2013)	13B	-	-	<b>66.7</b>	-	-
	Gemini-Nano-1 (Team et al., 2023)	1.8B	-	-	62.7	-	26.3
	Show-o (Xie et al., 2024)	1.3B	73.8	948.4	59.3	48.7	25.1
	LWM (Liu et al., 2024e)	7B	75.2	-	55.8	44.8	-
	Chameleon <sup>†</sup>	7B	77.8	1056.9	57.8	49.6	26.7
	<b>Orthus (Ours)</b>	7B	<b>79.6</b>	<b>1265.8</b>	<u>63.2</u>	<b>52.8</b>	<b>28.2</b>

Model	Res.	GenEval↑	HPSv2↑	POPE↑	MME↑	GQA↑
Orthus	512	0.58	28.2	79.6	1265.8	52.8
VILA-U	256	0.40	25.3	83.9	1336.2	58.3
Janus	384	0.61	27.8	87.0	1338.0	59.1

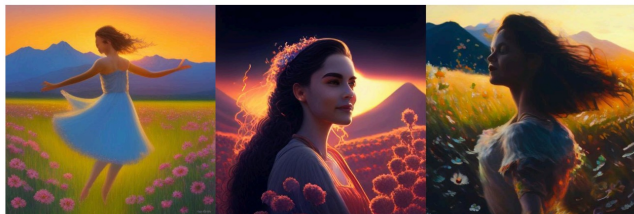
# Orthus: 文生图可视化结果



A detailed ink illustration of a hedgehog.



A hyena fursona sits in a savannah sunset amidst the grass.



Oil painting portrait of a young woman in a field of flowers at sunset with mountains in the background.



## Orthus: ablation结果

Table 4. Comparisons of the performance of Orthus via separate training and unified training across multimodal benchmarks.

Type	$\mathcal{L}_{\text{diff}}$	$\mathcal{L}_{\text{ar}}$	POPE $\uparrow$	MME-P $\uparrow$	GQA $\uparrow$	GenEval $\uparrow$
Und. only	$\times$	$\checkmark$	78.7	1244.2	51.9	-
Gen. only	$\checkmark$	$\times$	-	-	-	0.56
Und. & Gen.	$\checkmark$	$\checkmark$	<b>79.6</b>	<b>1265.8</b>	<b>52.8</b>	<b>0.58</b>

同时从文到图和图到文数据学习可以实现1+1>2

Table 5. Ablation study on the choice of vision embedding modules on visual understanding tasks.

Type	POPE $\uparrow$	MME-P $\uparrow$	VQAv2 $\uparrow$	GQA $\uparrow$	MMMU $\uparrow$
softmax	<b>78.7</b>	<b>1244.2</b>	<b>60.8</b>	<b>51.9</b>	<b>28.0</b>
argmin	77.6	1064.8	57.9	50.1	26.7
linear	70.4	800.7	50.3	44.5	22.3

连续的图像特征对于视觉理解任务必要，但要避免冷启动



# Orthus: 图文交错生成结果 (图文->图、demo+图->图、图文->图文图文)

### In-domain editing

Add a rainbow.

Make the suit red.

### Out-of-domain editing

Remove the cat.

Make it snow.

### In-context editing

According to the following examples:

output:

Generate an output for the input:

### Text-to-Image Generation

Please continue this story:

One sunny morning, George the curious monkey and his friend decided to visit the bustling city park. They started their adventure by exploring a colorful playground.

After a while, George, decided to take a break and sat on the grass, enjoying the peaceful trees and the buildings.

Please continue this story:

On a snowy day, George, the small figure in red winter gear, was excited to help his friend, the man in the yellow hat and red scarf. They were outside their blue house.

After shoveling, the man in the orange hat stood contentedly in the snowy landscape, his smile bright against the overcast sky.

George, was also out in the snowy landscape. He wore a red jacket, hat, and gloves, wondering if more snow was on the way.

### Image-to-Text Generation

Please continue this story:

Startled, George stood in the dark forest, his large, menacing shadow cast on the trees. His flashlight lay on the ground, projecting the frightening silhouette.

But George was not one to be scared easily. He stood inside a lively cave, ready for an adventure. The hint of light peering behind him cast a soft glow, making his eyes sparkle with curiosity.

In the shadowy forest, George spotted a red ball. He quickly grabbed it, excited about his new find. Trees and bushes, illuminated by a light source, surrounded him.

With a smile on his face, George stood in the cave with his arms outstretched. He was enjoying his adventure, surrounded by stalactites and a glowing light.

Table 1. Comparisons of CLIP similarities (Ruiz et al., 2023; Gal et al., 2022) between editing-specific diffusion models and Orthus on the test dataset of Instruct-Pix2Pix.

Model	-T↑	-I↑	-D↑
PnP (Tumanyan et al., 2023)	0.156	0.76	0.023
SDEdit (Meng et al.)	0.229	0.84	0.047
I-Pix2Pix (Brooks et al., 2023)	0.233	<b>0.88</b>	0.045
<b>Orthus (Ours)</b>	<b>0.238</b>	0.87	<b>0.049</b>

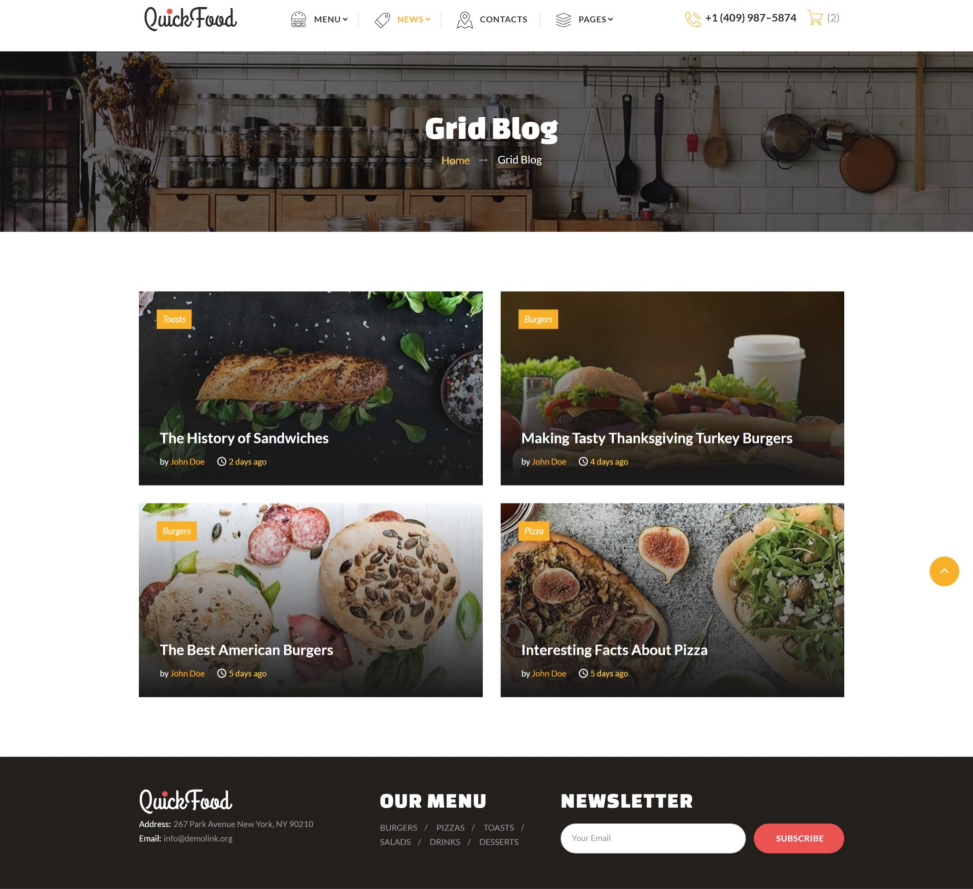
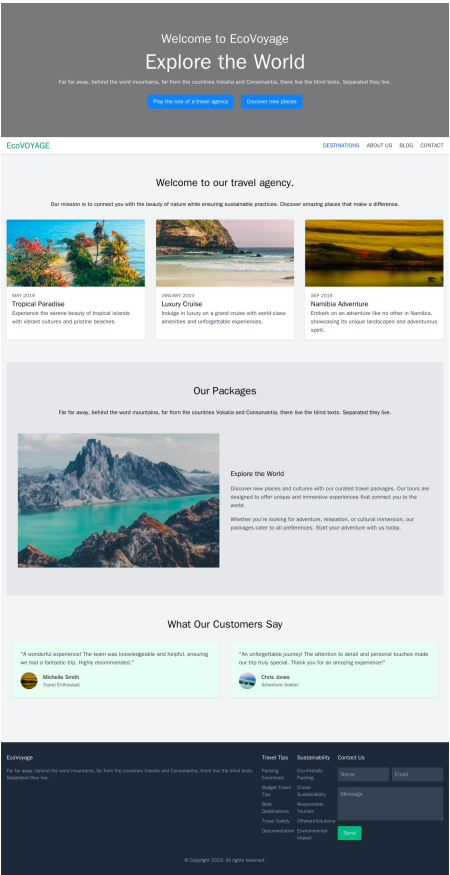
## 图像编辑能力指标

	Orthus	Show-o	NEXt-GPT	MiniGPT-5	GILL	SEED-X
OpenING-IVD ↑	6.3	5.1	5.2	5.3	6.2	8.0

## 图文交错生成指标



Orthus: 图文交错的HTML网页生成

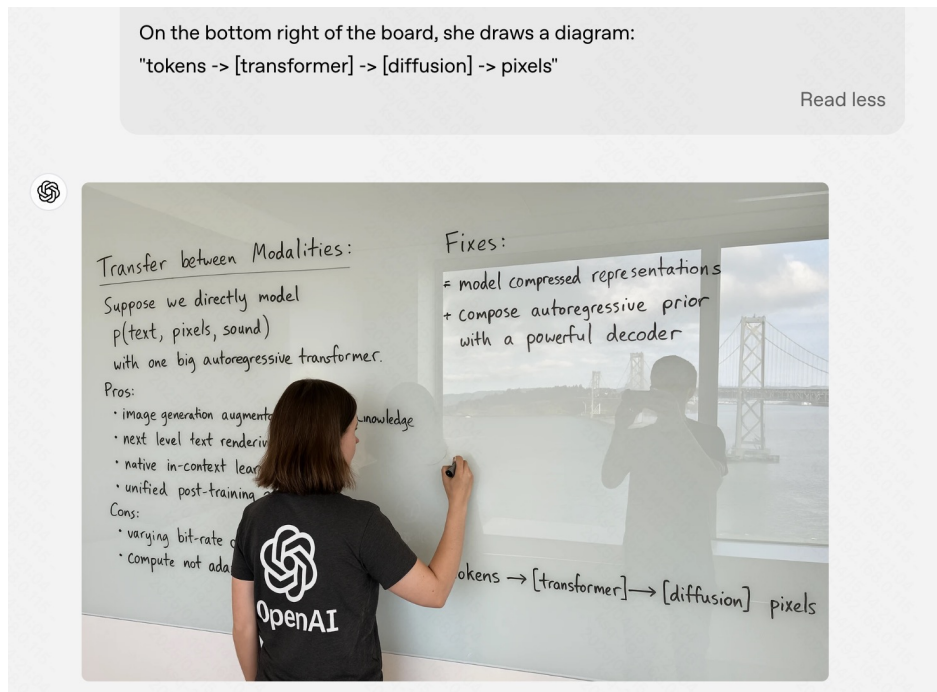


## 如何将跨模态生成理解统一？

GPT-4o的彩蛋: **tokens**  $\rightarrow$  [transformer]  $\rightarrow$  [diffusion]  $\rightarrow$  **pixels**

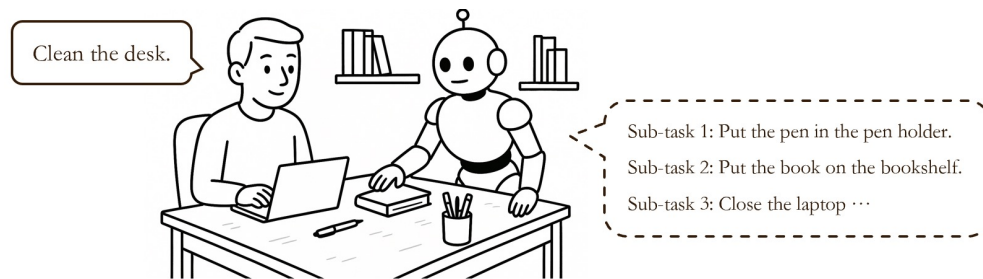
结合自回归模型的语义建模优势和扩散模型的细节建模优势

- 与Orthus大思路一致
  - 如何改进生成效率？Diffusion forcing？



## 跨模态交错生成模型在文/图数据外的应用

- **Vision-language-action (VLA)!**
- **VLA**为什么需要跨模态交错生成？
  - **long-horizon**任务：高层次目标，需要多个步骤的解决方案

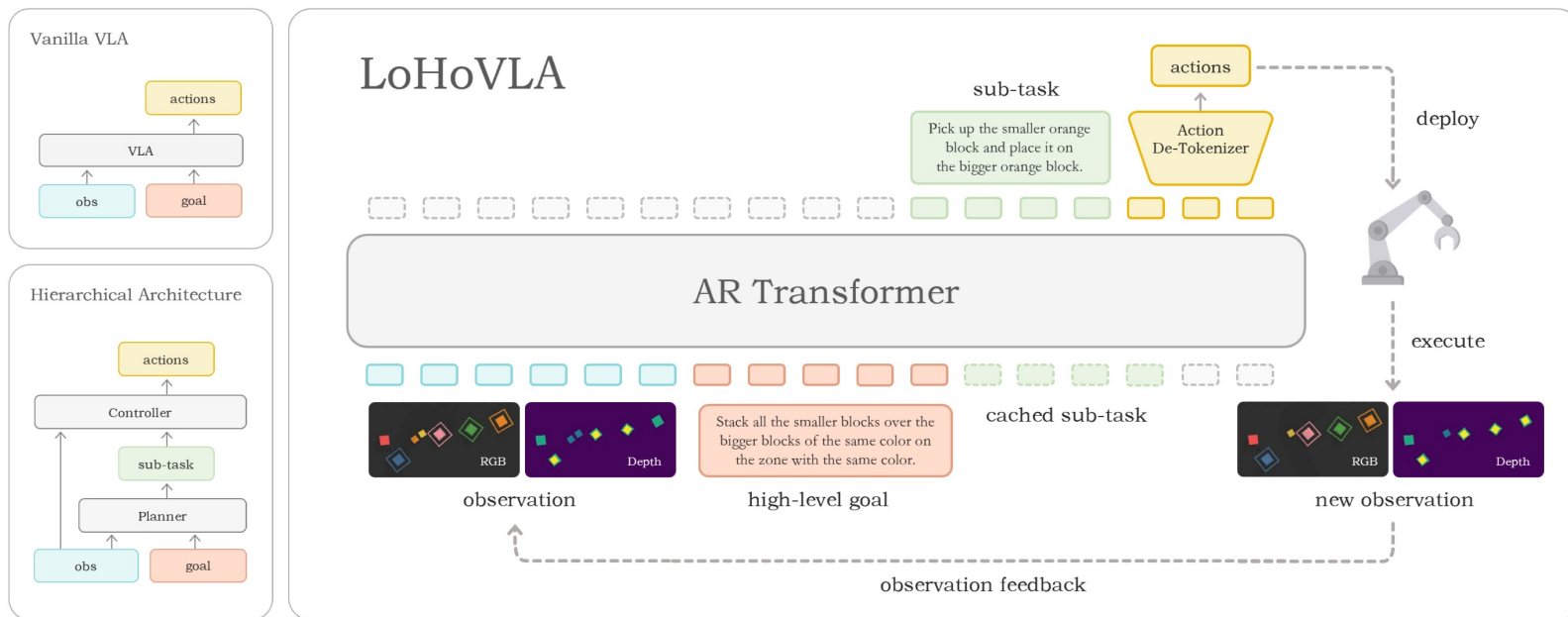


- 要求兼顾**高层任务规划**（**目标**->**子任务**）与**低层动作控制**（**子任务**->**动作**）
- 子任务（文本/图像）和动作（末端执行器位置、夹持器开合）模态不同



## LoHoVLA: 面向长时程具身任务的统一VLA模型

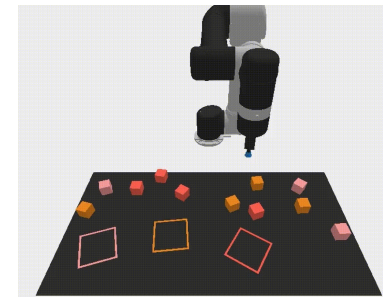
- 普通**VLA**模型：只能生成动作，隐式规划子任务（规划能力弱）
- 层次化架构：**Planner**规划子任务，**Controller**生成动作（模块冗余，次优协调）
- **LoHoVLA**：使用同一个模型完成子任务规划和动作控制（规划能力强，泛化性好）



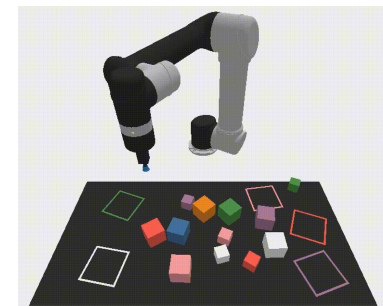
# LoHoVLA显著提高模型的规划能力和未见任务的泛化性能

Table 2: Comparison of the average award (%) and success rate (%) on LoHoRavens benchmark. Bold entries indicate the highest success rates, underlined entries indicate the second-highest.

Tasks		Vanilla VLA	LoHoRavens		LoHoVLA
			Explicit feedback	Implicit feedback	
Seen Tasks	A	<b>79.0</b> / 79.0	67.3 / -	67.3 / -	<u>77.5</u> / 77.5
	B	14.9 / 0.0	31.4 / -	<u>37.0</u> / -	<b>97.8</b> / 91.5
	C	<u>26.8</u> / 0.5	18.0 / -	22.1 / -	<b>34.9</b> / 22.5
	D	32.3 / 3.0	30.4 / -	<u>33.2</u> / -	<b>35.8</b> / 11.5
	E	<u>22.1</u> / 3.5	9.6 / -	8.2 / -	<b>85.1</b> / 81.0
Unseen Tasks	F	<u>52.1</u> / 9.0	28.5 / -	21.1 / -	<b>86.1</b> / 41.0
	G	6.8 / 0.0	<u>21.9</u> / -	14.7 / -	<b>40.1</b> / 25.0
	H	7.3 / 0.0	<u>13.2</u> / -	5.2 / -	<b>16.7</b> / 7.5
	I	<u>43.1</u> / 1.5	12.8 / -	11.7 / -	<b>77.2</b> / 52.0
	J	<u>38.6</u> / 10.5	27.4 / -	27.2 / -	<b>43.6</b> / 22.0
	K	<u>58.2</u> / 33.0	4.0 / -	6.8 / -	<b>73.8</b> / 54.5



"Move all blocks of a color that occur in even numbers to the same colored zone."



"Stack blocks of the same color in the zone with same color, with bigger blocks underneath."



## 2、大模型并行推理算法

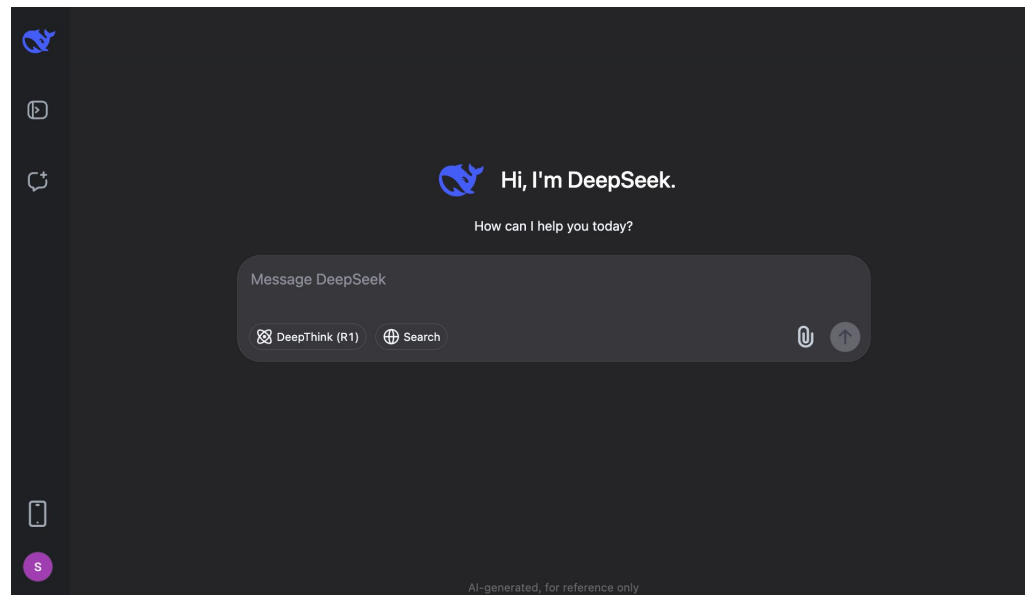
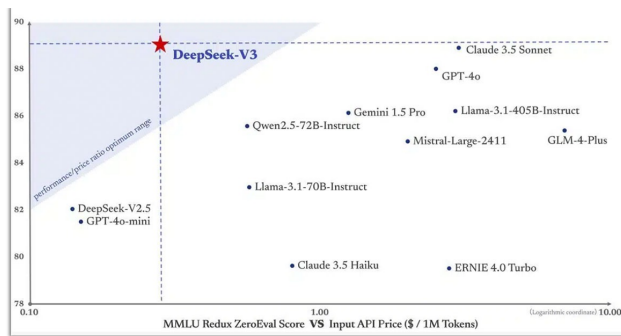
# DeepSeek给我们上的一课：只有模型好不够，**降低成本**才是应用的关键

reminder that @deepseek\_ai spent just over \$5.57 million to train their DeepSeek-V3 model, which is what META pays five senior AI researchers in one year.

LITERALLY HOW????

Training Costs	Pre-Training
in H800 GPU Hours	2664K
in USD	\$5.328M

DeepSeek v3的训练成本仅等于META五位研究员的年薪



OpenAI o1: \$60.00 per 1M output tokens  
 DeepSeek R1: \$2.19 per 1M output tokens

DeepSeek v3的API调用成本也大大低于几大公司竞品

# DeepSeek如何降低成本?

- 模型&算法侧: DeepSeekMoE & MLA & NSA

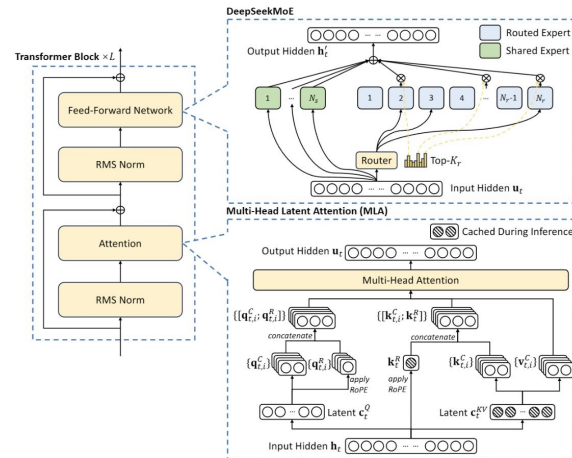
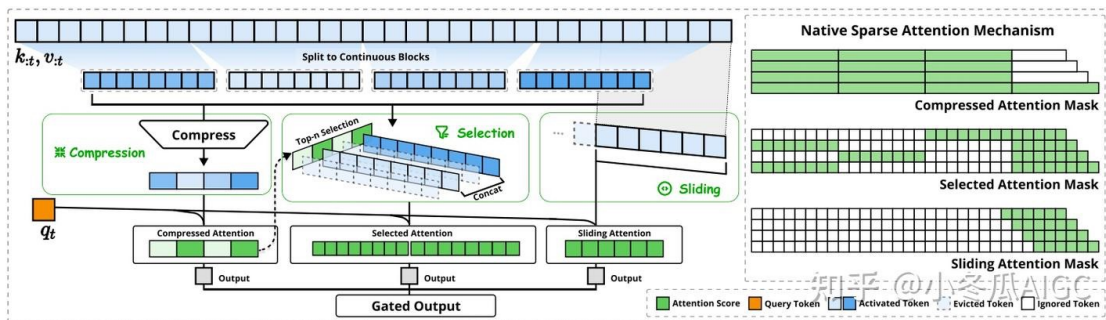
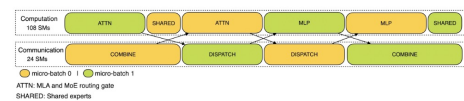


Figure 1 | Illustration of the architecture of DeepSeek-V2. MLA ensures efficient inference by significantly reducing the KV cache for generation, and DeepSeekMoE enables training strong models at an economical cost through the sparse architecture.

- 底层实现侧: 专家并行、计算/通信重叠、负载均衡、极致代码优化、etc.

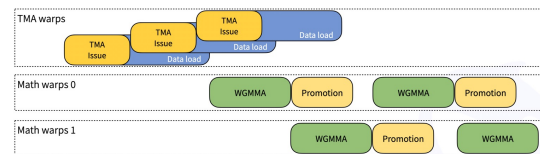
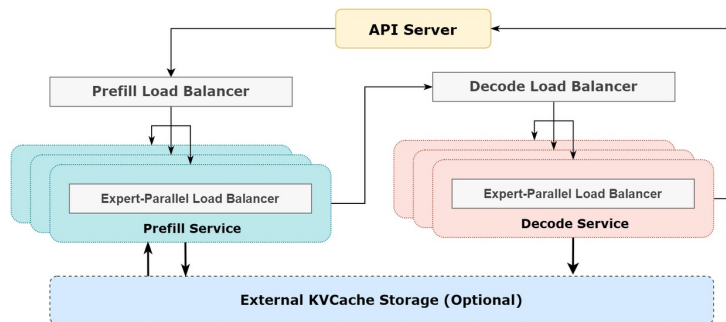
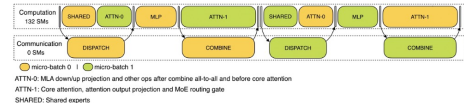
## Computation-Communication Overlapping

Large-scale cross-node EP introduces significant communication overhead. To mitigate this, we employ a dual-batch overlap strategy to hide communication costs and improve overall throughput by splitting a batch of requests into two microbatches. During the prefilling phase, these two microbatches executed alternately and the communication cost of one microbatch is hide behind the computation of the other.



## Communication-Computation Overlapping during Prefilling Phase

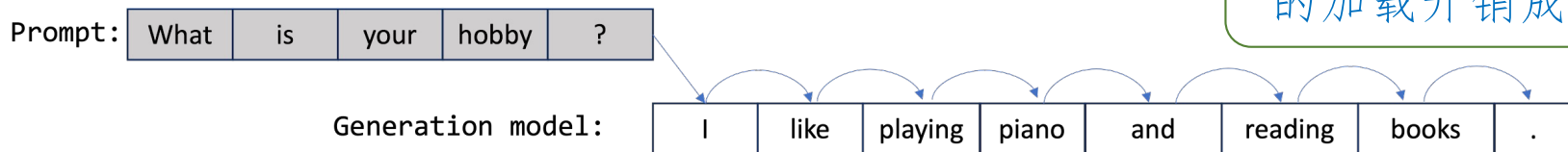
During the decoding phase, the execution durations of different stages are unbalanced. Hence, we subdivide the attention layer into two steps and use a 5-stage pipeline to achieve a seamless communication-computation overlapping.



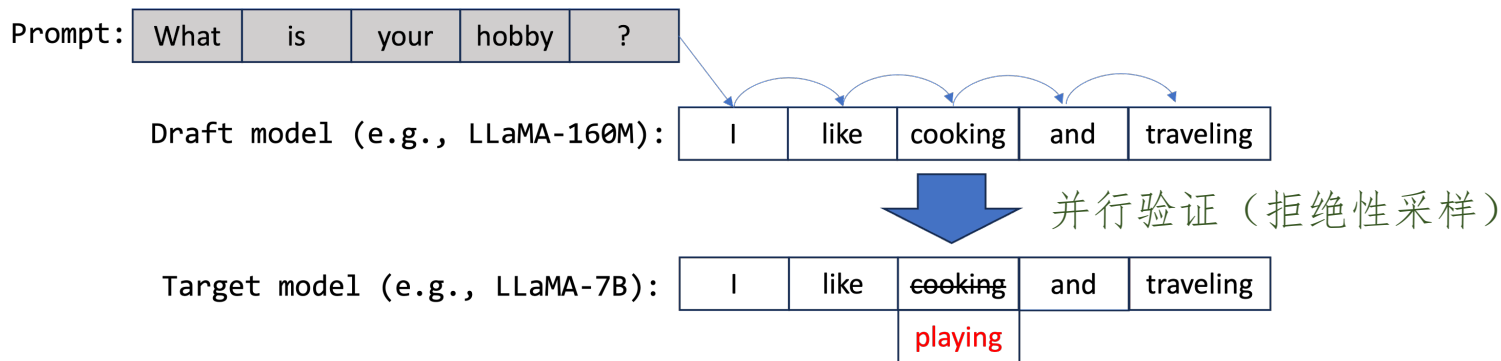


## LLMs并行推理：投机解码将大模型的计算开销卸载到小模型

**KV cache**和模型权重的加载开销成倍下降

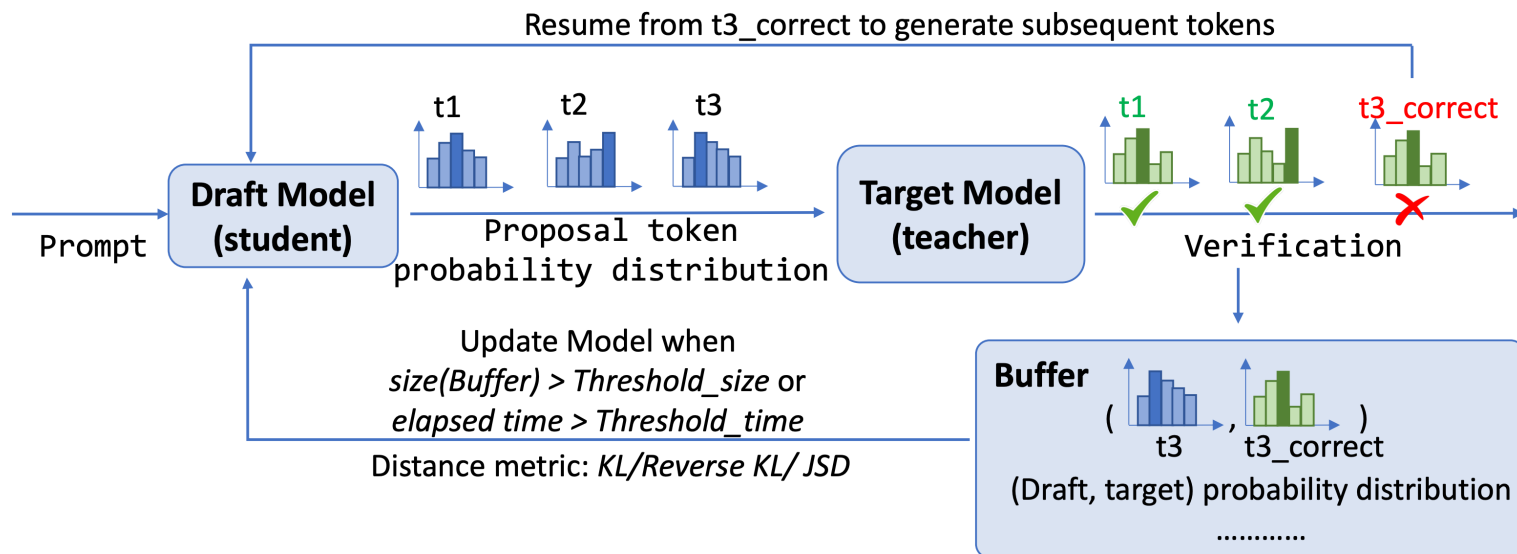


不是所有的token都“难”，可将大部分的顺序生成的负载转移到一个更小的草稿模型（draft model）



## 在线投机解码：在线蒸馏+投机解码

Open domain情况下，draft model快速适配query dist.



- 将草稿模型的错误预测和目标模型的校正结果存储在**buffer**中
- 当**buffer**打满，基于在线蒸馏损失函数更新草稿模型





# 在线投机解码：结果

Dataset	Spider	Chatbot Arena	Extra Parameters (B)
Medusa-7B	1.34×	2.03×	0.44
Medusa-7B + OSD	2.01×	2.38×	0.44
Draft model + OSD	2.17×	1.51×	0.16

超越/结合**Medusa**（一个代表性的多头LLM）

Dataset	Spider	Gsm8k	Alpaca-Finance	Code-Python
<b>Tokens with the greatest precision increase</b>	AV, SELECT, first, ⟨EOS⟩, template, SUM, G, COUNT, \n, city, WHERE, ', ;, (, IST, id	⟨EOS⟩, >>, +, To, <<, this, =, %, know, are, We, calculate, be, The, have	1, Here, (, :, provide, depends, However, goals, amount, 3, there, The, \n, personal, will	'''', (, Here, python, ', how, doc, snippet, import, based, {, Python, This, :, you
<b>Tokens with the greatest recall increase</b>	SELECT, *, FROM, (, IST, *), \n, COUNT, G, first, WHERE, ⟨EOS⟩, IN, :, MAX, ',	start, >>, <<, +, find, how, we, =, fore, To, so, \, ⟨EOS⟩, then, let	general, 1, several, This, depends, Here, provide, However, goals, over, (, If, amount, it, can	Here, This, snippet, '''', ', how, python, (, takes, Python, you, doc, an, import, def

**Token acceptance rate** 提升最多的**tokens**

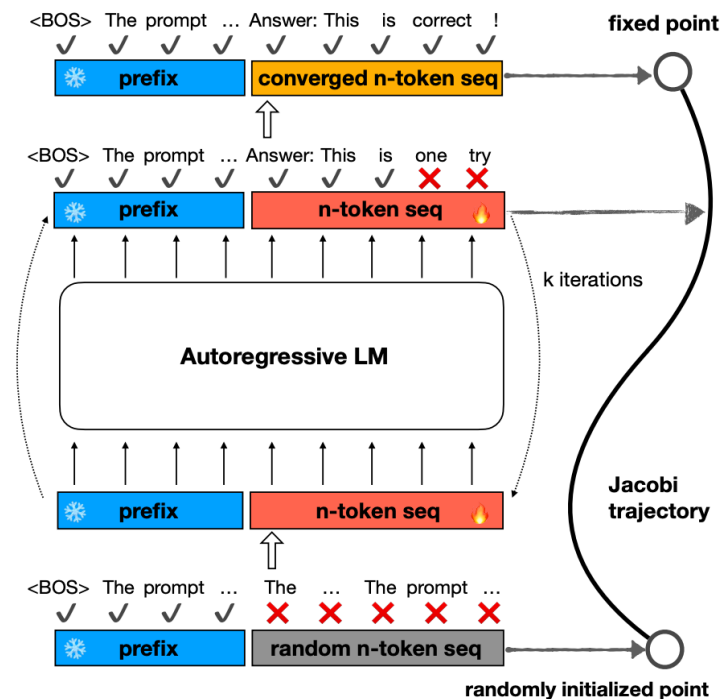
## 基于Jacobi decoding的大语言模型并行推理

- 给定大语言模型, 同时预测 **n** 个 **token** 等价于求解:

其中  $f(y_i, \mathbf{y}_{<i}, \mathbf{x}) = 0$  for  $i = 1, \dots, n$ .

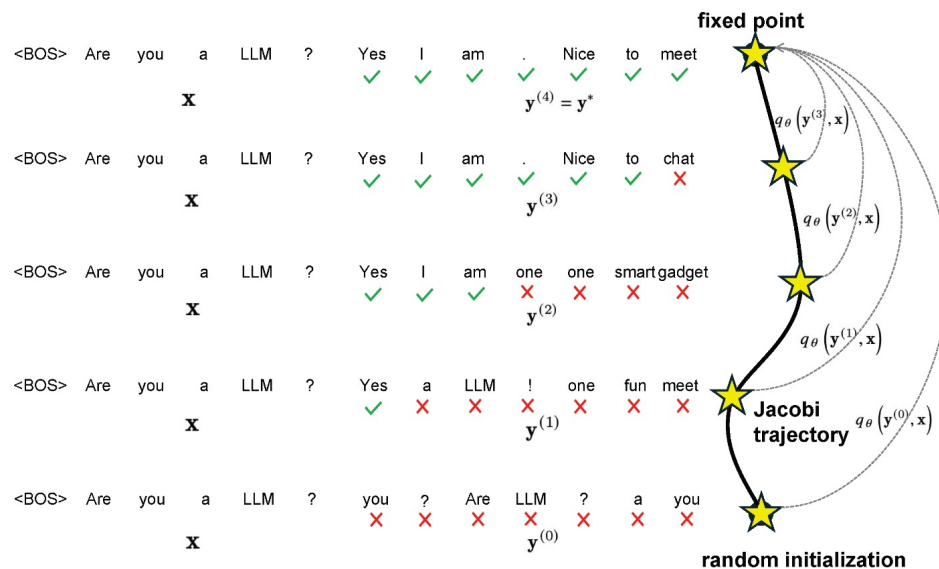
- 这  $f(y_i, \mathbf{y}_{<i}, \mathbf{x}) := y_i - \arg \max_y p(y | \mathbf{y}_{<i}, \mathbf{x})$  并行求解, 步数不超过 **n**, 生成质量可保证

- 但实际效果差 (如: 仅**1.05**倍提升)
  - 原因: 模型训练时未学过如何预测多个 **tokens**



# 一致性大语言模型 (Consistency LLMs, CLLMs)

- 通过训练习得预测 **n** 个 **tokens** 的能力
  - 从随机初始化的起点预测 **fixed point**?
    - 不行，问题太难，训练难收敛
  - 从 **Jacobi** 解码轨迹上的任意点预测 **fixed point**?
    - 可以，形成一系列从简单到困难的学习问题，有助于模型收敛



# 一致性大语言模型

$$\mathcal{L}_{GC} = \mathbb{E}_{(\mathbf{x}, \mathcal{J}) \sim \mathcal{D}, \mathbf{y} \sim \mathcal{J}} \left[ \sum_{i=1}^n D(q_{\theta}(\cdot | \mathbf{y}_{:i}^*, \mathbf{x})) || q_{\theta}(\cdot | \mathbf{y}_{:i}, \mathbf{x}) \right]$$

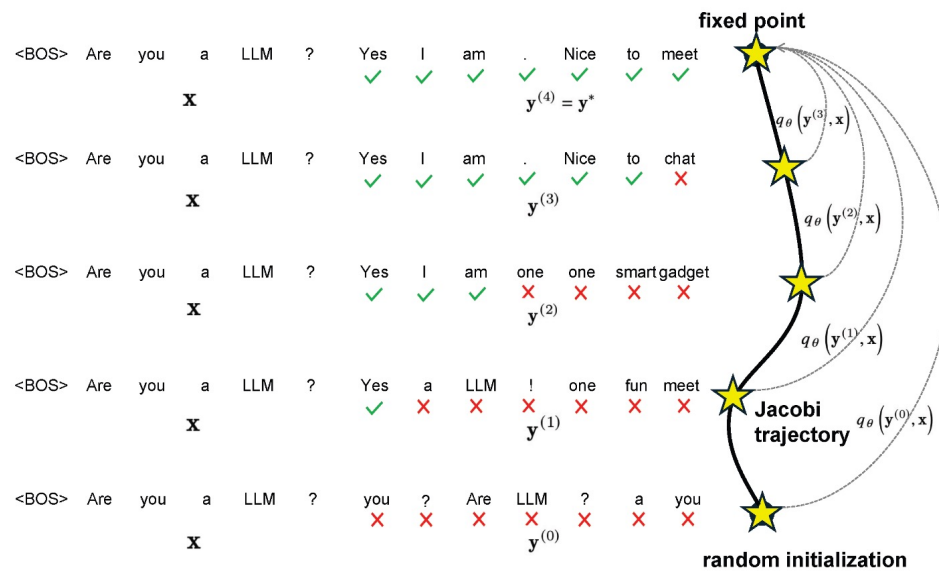
fixed point

$$\mathcal{L}_{LC} = \mathbb{E}_{(\mathbf{x}, \mathcal{J}) \sim \mathcal{D}, (\mathbf{y}^{(j)}, \mathbf{y}^{(j+1)}) \sim \mathcal{J}} \left[ \sum_{i=1}^n D(q_{\theta}(\cdot | \mathbf{y}_{:i}^{(j+1)}, \mathbf{x})) || q_{\theta}(\cdot | \mathbf{y}_{:i}^{(j)}, \mathbf{x}) \right]$$

$D(\cdot || \cdot)$  是分布间距离度量

$$\mathcal{L}_{AR} = \mathbb{E}_{(\mathbf{x}, \mathbf{l}) \sim \mathcal{D}} \left[ - \sum_{i=1}^N \log q_{\theta}(l_i | \mathbf{l}_{:i}, \mathbf{x}) \right]$$

自回归损失防止模型退化





# 一致性大语言模型：结果



聊天，**Vicuna-7B**，**2.4倍加速**



数学，**Abel-7B-001**，**3倍加速**



代码，**Deepseek-coder-7B**，**3.4倍加速**



# 一致性大语言模型：结果

- 至多3.6倍加速

相比于Medusa2、Eagle3，不

需要模型架构上的改变

生成质量极少损失

Methods	Speed (tokens/s)	Speedup	Metric	Size
GSM8K				
Fine-tuned LLaMA2-7B (Chem et al.)				
+ AR	43.5	1.0×	59.1	6.7B
+ Jacobi	45.7	1.1×	59.1	
+ lookahead	74.8	1.7×	59.1	
CLLM-LLaMA2-7B				
+ AR	43.5	1.0×	56.4	6.7B
+ Jacobi	132.4	3.0×	56.4	
+ lookahead	125.2	2.9×	56.4	
Medusa-2 + LLaMA2-7B				
+ typical	70.2	1.6×	51.3	8.3B
Fine-tuned LLaMA2-7B + distilled LLaMA-160m				
+ speculative	73.8	1.7×	59.1	6.8B
ShareGPT (MT-Bench)				
Fine-tuned LLaMA2-7B				
+ AR	37.6	1.0×	6.5	6.7B
+ Jacobi	39.9	1.1×	6.5	
+ lookahead	60.8	1.6×	6.5	
CLLM-LLaMA2-7B				
+ AR	36.7	1.0×	6.4	6.7B
+ Jacobi	88.4	2.4×	6.4	
+ lookahead	95.0	2.5×	6.4	
Medusa-2 + LLaMA2-7B				
+ typical	102.5	2.7×	6.4	8.3B
Fine-tuned LLaMA2-7B + distilled LLaMA-160m				
+ speculative	51.3	1.4×	6.5	6.8B

Methods	Speed (tokens/s)	Speedup	Metric	Size
Spider				
Fine-tuned Deepseek-7B				
+ AR	38.0	1.0×	70.0	6.7B
+ Jacobi	39.5	1.0×	70.0	
+ lookahead	55.3	1.5×	70.0	
CLLM-Deepseek-7B				
+ AR	38.0	1.0×	69.3	6.7B
+ Jacobi	127.4	3.4×	69.3	
+ lookahead	135.2	3.6×	69.3	
Medusa-2 + Deepseek-7B				
+ typical	104.2	2.7×	66.4	8.3B
Fine-tuned Deepseek-7B + distilled LLaMA-160m				
+ speculative	66.8	1.8×	70.0	6.8B
Code-Search-Net Python				
Fine-tuned Deepseek-7B				
+ AR	40.1	1.0×	60.4	6.7B
+ Jacobi	43.2	1.1×	60.4	
+ lookahead	68.0	1.7×	60.0	
CLLM-Deepseek-7B				
+ AR	38.5	1.0×	59.2	6.7B
+ Jacobi	102.1	2.5×	59.2	
+ lookahead	115.7	2.9×	59.2	
Medusa-2 + Deepseek-7B				
+ typical	128.0	3.2×	48.3	8.3B
Fine-tuned Deepseek-7B + distilled LLaMA-160m				
+ speculative	59.3	1.5×	60.4	6.8B

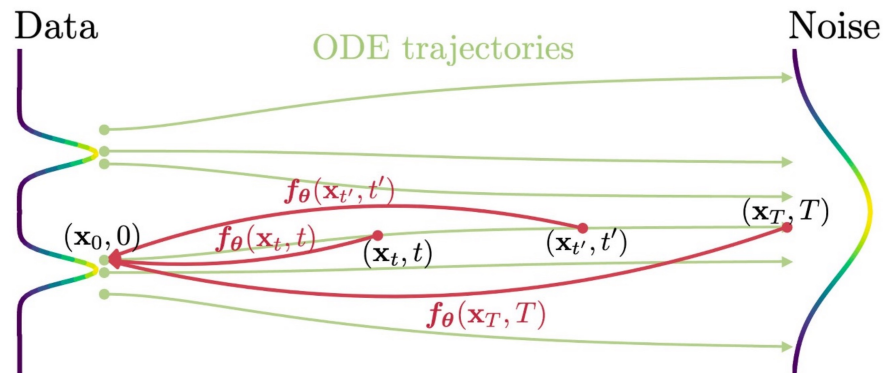
# 一致性蒸馏 (Consistency Distillation)

- 概率流ODE定义了从噪声到数据的一一映射  
=>直接建模此映射  $\forall t \in [0, T] : \mathbf{f}_\theta(\mathbf{x}_t, t) = \mathbf{x}_0$

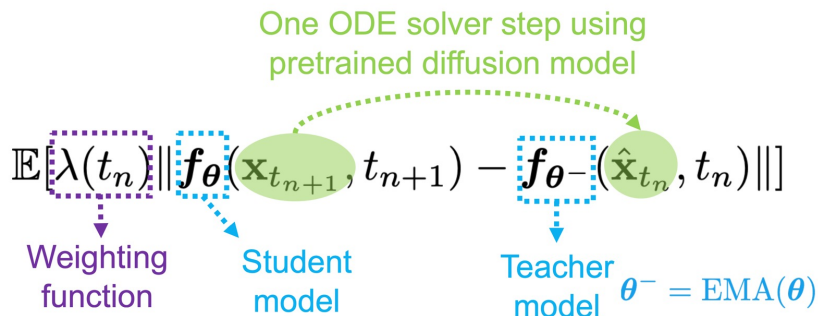
- 模型参数化: 需保证边界条件 ( $t=0$ )  

$$\mathbf{f}_\theta(\mathbf{x}, t) = c_{\text{skip}}(t)\mathbf{x} + c_{\text{out}}(t)F_\theta(\mathbf{x}, t)$$

$$c_{\text{skip}}(0) = 1 \quad c_{\text{out}}(0) = 0$$



- 训练:



## Algorithm 1 Multistep Consistency Sampling

**Input:** Consistency model  $\mathbf{f}_\theta(\cdot, \cdot)$ , sequence of time points  $\tau_1 > \tau_2 > \dots > \tau_{N-1}$ , initial noise  $\hat{\mathbf{x}}_T$

$\mathbf{x} \leftarrow \mathbf{f}_\theta(\hat{\mathbf{x}}_T, T)$

**for**  $n = 1$  **to**  $N - 1$  **do**

Sample  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$\hat{\mathbf{x}}_{\tau_n} \leftarrow \mathbf{x} + \sqrt{\tau_n^2 - \epsilon^2} \mathbf{z}$

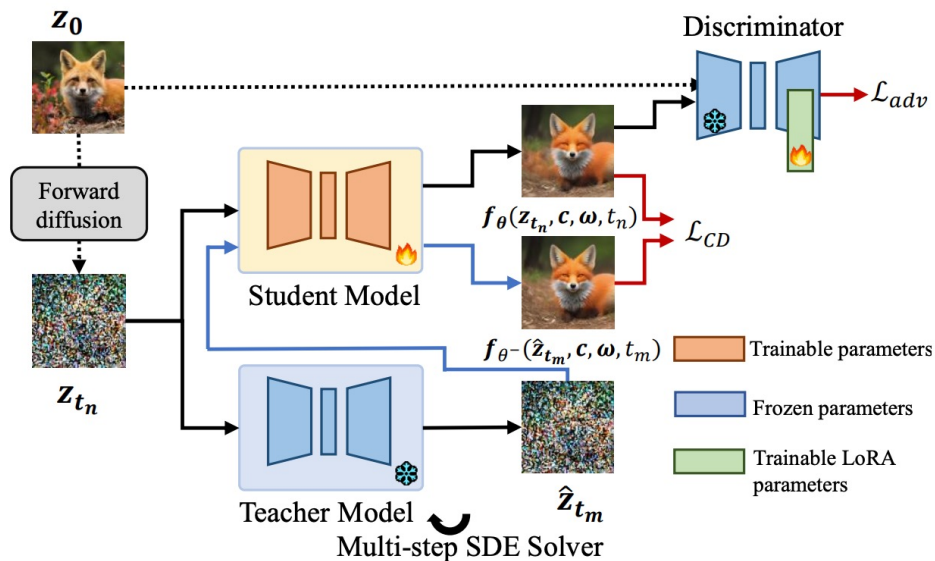
$\mathbf{x} \leftarrow \mathbf{f}_\theta(\hat{\mathbf{x}}_{\tau_n}, \tau_n)$

**end for**

**Output:**  $\mathbf{x}$



# 随机一致性蒸馏



$$\min_{\theta} \mathcal{L}_{CD}(\theta) = \mathbb{E}_{n, z_{t_n}} \left[ \lambda(t_n) \left\| f_{\theta}(z_{t_n}, t_n) - f_{\theta^-}(\hat{z}_{t_m}, t_m) \right\|_2^2 \right]$$

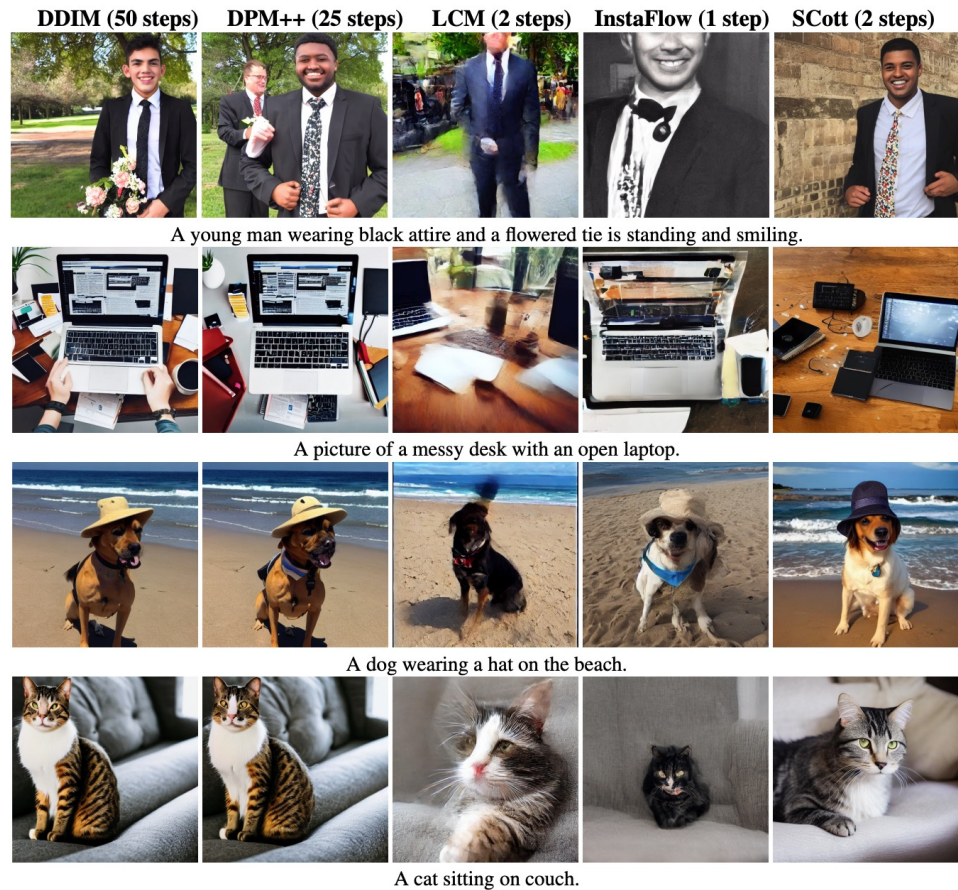
- 基于随机微分方程（SDE）采样器构建蒸馏教师模型
- 多步SDE采样矫正累计误差
- 引入的噪声可以视作数据增广

# 随机一致性蒸馏：结果

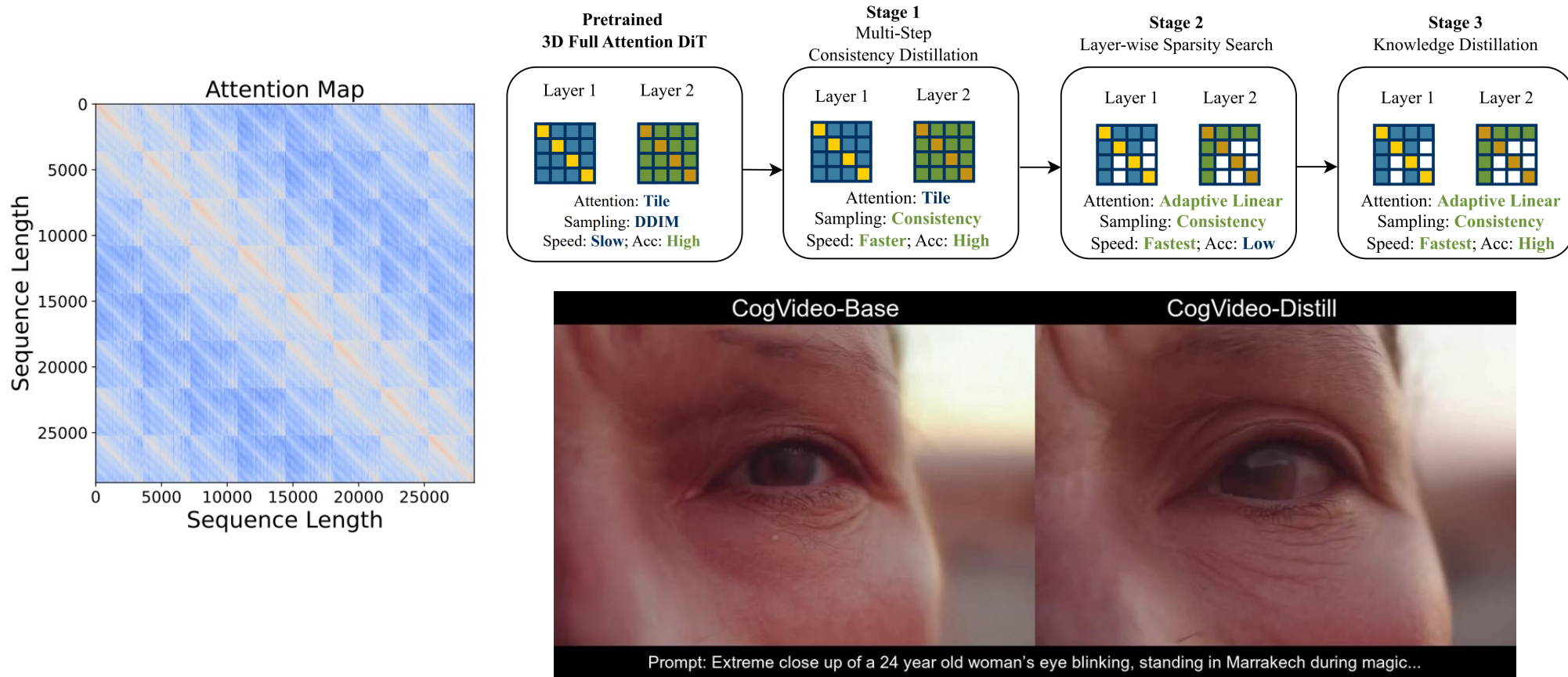
- 两步采样达到21.9的FID
  - 显著超越InstaFlow 和UFOGen



Figure 1:  $512 \times 512$  resolution images generated by SCott using 2 sampling steps. SCott is trained based on Realistic-Vision-v51.



# 一致性蒸馏+注意力稀疏：5-10倍视频生成加速





### 3、高效的深度推理模型





# 深度推理: the new frontier of AIGC?

## deepseek-r1开源复现方法整理

yeyan, 中国地质大学 工程硕士

deepseek-r1持续火热, 估计会掀起一波复现其训练过程的热潮, 先简单整理下目前看到的。

目录 方法

### \* 1.1 open-r1

由huggingface组建, 目前刚上线2周, 发布了最新进展open-r1/update-1, 在MATH-500任务上接近deepseek的指标, 可以在open-r1/open-r1-eval-leaderboard查看指标的排行。

Model	MATH-500 (HF lighteval)	MATH-500 (DeepSeek Reported)
DeepSeek-R1-Distill-Qwen-1.5B	81.6	83.9
DeepSeek-R1-Distill-Qwen-7B	91.8	92.8
DeepSeek-R1-Distill-Qwen-14B	94.2	93.9
DeepSeek-R1-Distill-Qwen-32B	95.0	94.3
DeepSeek-R1-Distill-Llama-8B	85.8	89.1
DeepSeek-R1-Distill-Llama-70B	93.4	94.5

### 1.2 mini-deepseek-r1

用 GRPO 和倒计时游戏复制出一个简单版本的 R1。

在大约 50 步时, 模型学会了正确的格式, 即<think>...</think><n></n><answer>...</answer>;在 100 步时, 解方程的成功率约为 25%, 并且模型开始用文字进行“推理”;在 200 步时, 收敛变慢, 成功率约为 40%。模型开始学习一种新的“格式”, 它通过尝试不同的组合并检查结果来解方程。这种方式类似于编程解决问题的方式;在 450 步时, 解方程的成功率为 50%, 性能仍然在缓慢提升, 并且模型保持了从 200 步开始的新格式。

### \* 1.3 open-thoughts

目标是整理一个推理数据集, 用于训练最先进的小型推理模型, 使其在数学和代码推理基准测试中超越 DeepSeek-R1-Distill-Qwen-32B 和 DeepSeek-R1-Distill-Qwen-7B。

目前已在以下领域生成数据: 代码 (Code) 领域; 数学 (Math) 领域; 科学 (Science) 领域; 谜题 (Puzzle) 领域。

目前已经发布了 openthinker 开源模型。

### \* 1.4 TinyZero

是在倒计时和乘法任务中对 DeepSeek R1 Zero 进行复制。基于 verl\* 构建, 通过强化学习... 3B 基础语言模型 (LM) 自行发展出自验证和推理能力。可以低于 30 美元的价格体验到“Ahh moment”。

User: Using the numbers [19, 36, 55, 7], create an equation that equals 65.  
Assistant: Let me solve this step by step.  
<think> We need to find an equation using the numbers 19, 36, 55, and 7 exactly once, with basic arithmetic operations, that equals 65. One possible combination is  $55 + 36 - 19 + 7$ . Let's check:  $55 + 36 = 91$ ,  $91 - 19 = 72$ , and  $72 + 7 = 79$ . However, this doesn't equal 65. Another combination is  $55 + 36 + 7 - 19$ . Let's check:  $55 + 36 = 91$ ,  $91 + 7 = 98$ , and  $98 - 19 = 79$ . This also doesn't equal 65. After trying different combinations, I found that  $55 + 36 - 19 + 7 = 79$ , which is close but not equal to 65. Finally, I found that  $55 + 36 - 19 + 7 = 79$ , which is still not equal to 65. However, if we try  $55 + 36 - 7 + 19$ , we get  $55 + 36 = 91$ ,  $91 - 7 = 84$ , and  $84 + 19 = 103$ , which is not equal to 65. After further trial and error, I found that  $55 + 36 + 7 - 19 = 79$ , which is still not equal to 65. However, if we try  $55 + 36 - 7 - 19$ , we get  $55 + 36 = 91$ ,  $91 - 7 = 84$ , and  $84 - 19 = 65$ . This equals 65. </think>  
<answer>  $55 + 36 - 7 - 19$  </answer>

知乎 @yeyan

### \* 1.5 simpleRL-reason

DeepSeek-R1 和 Kimi-k1.5 使用简单的强化学习算法来学习新兴的长思维链 (CoT) 和自我反思模式, 并取得了良好的结果, 其中没有使用 MCTS 和奖励模型。然而, 他们的实验是基于大规模强化学习环境中的大型模型。目前尚不清楚小型模型是否能表现出类似的行为, 需要多少数据, 以及定量结果与其他方法相比如何。simpleRL-reason 重现了 DeepSeek-R1-Zero\* 和 DeepSeek-R1 用于复杂数学推理的训练, 从 Qwen-2.5-Math-7B\* (基础模型) 开始, 并且仅使用来自原始数学数据集的 8K (查询、最终答案) 示例, 平均获得了近 20 个百分点的提升。

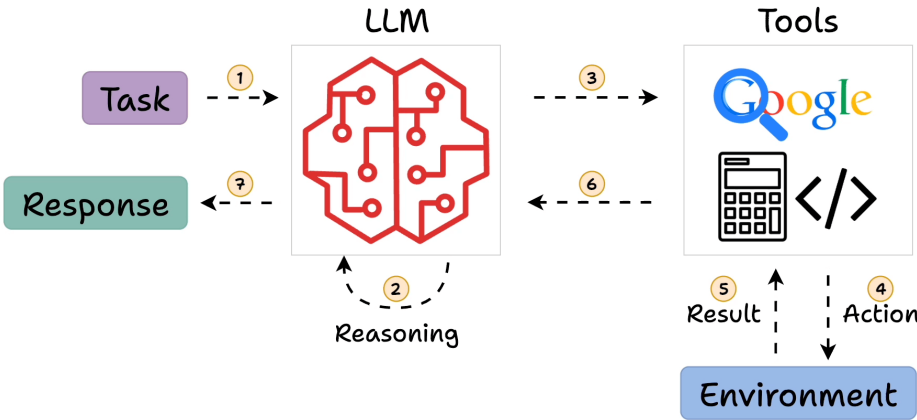
All results are in pass@1 accuracy

	AIME 2024	MATH 500	AMC	Minerva Math	OlympiadBench	Avg.
Qwen2.5-Math-7B-Base	16.7	52.4	52.5	12.9	16.4	30.2
Qwen2.5-Math-7B-Base + 8K MATH SFT	3.3	54.6	22.3	32.7	19.6	26.5
Qwen-2.5-Math-7B-Instruct	13.3	79.8	50.6	34.6	40.7	43.8
Llama-3.1-70B-Instruct	16.73	64.6	30.1	35.3	31.9	35.7
rStar-Math-7B	26.7	78.4	47.5	-	47.1	-
Eurus-2-7B-PRIME	26.7	79.2	57.8	38.6	42.1	48.9
Qwen2.5-7B-SimpleRL-Zero	33.3	77.2	62.5	33.5	37.6	48.8
Qwen2.5-7B-SimpleRL	26.7	82.4	62.5	39.7	43.3	50.9

知乎 @yeyan

### 1.6 RAGEN

RAGEN 是用于训练智能体模型的 DeepSeek-R1 (-Zero) 方法的首次复现。主要在 gym-sokoban (400 个训练步骤) 和 gym-100k 训练。



推理能力是Agent够不够“聪明”的关键

开源社区对DeepSeek R1的复现如火如荼

# 推理模型加速?

- 深度推理模型：长思维链（CoT）
  - Pros:** 显著提升复杂任务性能
  - Cons:** 导致推理延迟大幅增加，Reasoning 阶段通常占用60-80%的总时间，相当于传统任务3至5倍的token量

解决方案：

- 投机“推理”机制
- CoT压缩：减少Reasoning token数量**

Problem:

James writes a 3-page letter to 2 different friends twice a week. How many pages does he write a year?

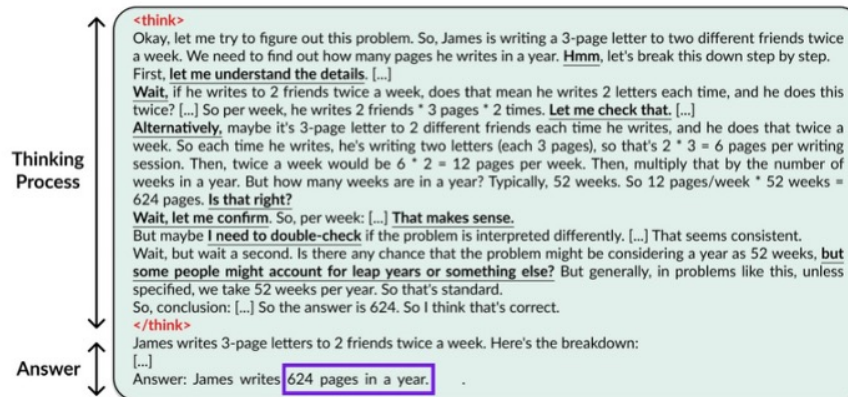
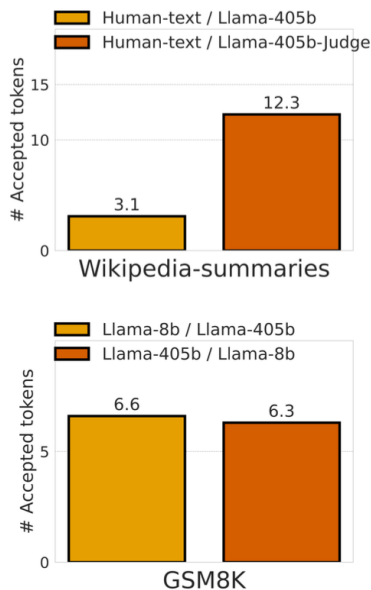


Figure from: Marjanović et al. DeepSeek-R1 Thoughtology. arXiv 2025.

## 投机解码：问题

- 标准投机解码要求输出分布对齐才能接受 **proposal**
  - 导致低的token接受率、平均单次接受token数



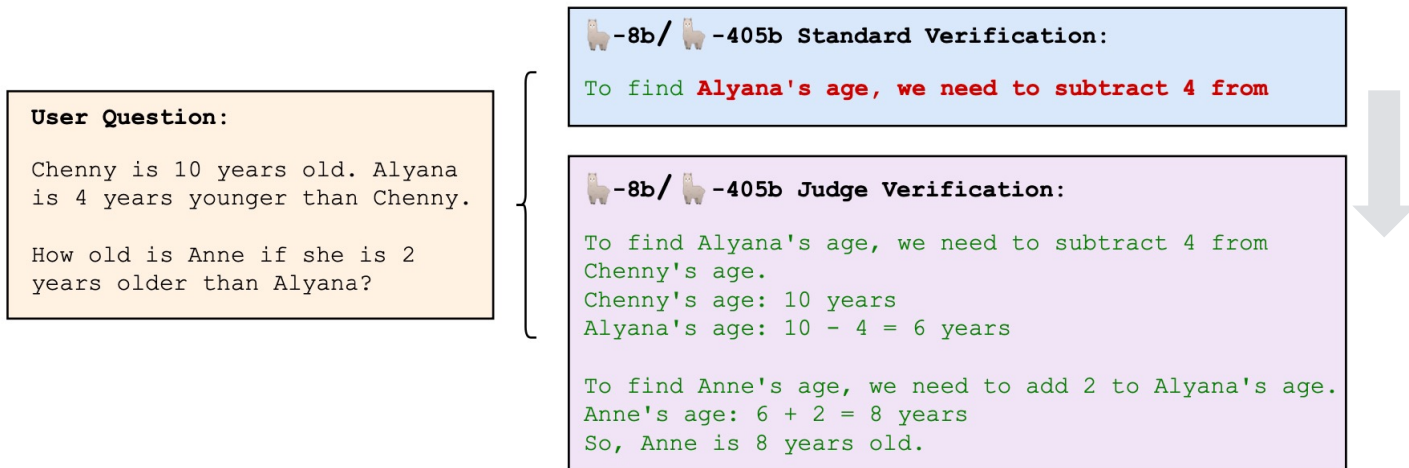
将Llama-8b/Llama-405b分别交替作为draft/target model，接受token数量基本不变（通常认为大模型输出文本质量高于小模型）

- 证明：文本质量和模型的接受率/接受token数量无关
- 即便高质量文本其在语义层面上可接受，对于上下文来说是正确的，仍会因为不符合target model分布而被拒绝，徒增迭代轮次
- 选取高质量人类文本作为draft，接受率也不高



## 新型投机解码：语义正确性与分布对齐性同等重要

- 标准投机解码要求输出分布对齐才能接受 **proposal**
  - 导致低的token接受率、平均单次接受token数
- 解决方法： **Judge Decoding**
  - 接受语义正确的proposal



# Judge Decoding: 训练简单线性层预测proposal的可接受性

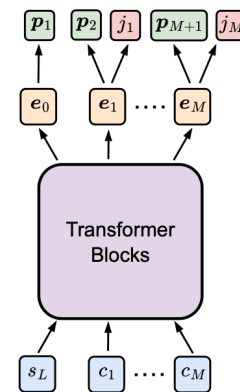
- LLMs有自动纠错能力
  - last hidden states能够有效地“标记”错误，促使模型生成后续 token 尝试纠正错误
- 构造正负样例数据集，用于训练新加线性层（回归头）
  - 16.4k参数的回归头在30k tokens上训练1.5h

**User Question:**  
What is the capital of France?  
Tell me something about the city.

**Assistant:**  
The capital of France is Berlin... No just kidding.  
The capital of France is actually Paris. [...]

**User Question:**  
What is 402 + 335?

**Assistant:**  
402 + 335 = 736 + 1 = 737



**Input Question:**  
What countries border France?

**Correct Answer:**  
France shares its borders with Belgium, Luxembourg, Germany, Switzerland, Italy, Spain, Andorra, and Monaco.

**Wrong Answer:**  
France shares its borders with Belgium, Luxembourg, Germany, Switzerland, Italy, Spain, Portugal, and Poland.

**Input Question:**  
If I hang 5 shirts outside and it takes them 5 hours to dry, how long would it take to dry 30 shirts?

**Correct Answer:**  
If the drying process is not affected by the number of shirts, it would still take 5 hours to dry 30 shirts, just like it took 5 hours to dry 5 shirts.

**Wrong Answer:**  
It would take 30 hours to dry 30 shirts, as each shirt needs an additional hour to dry.



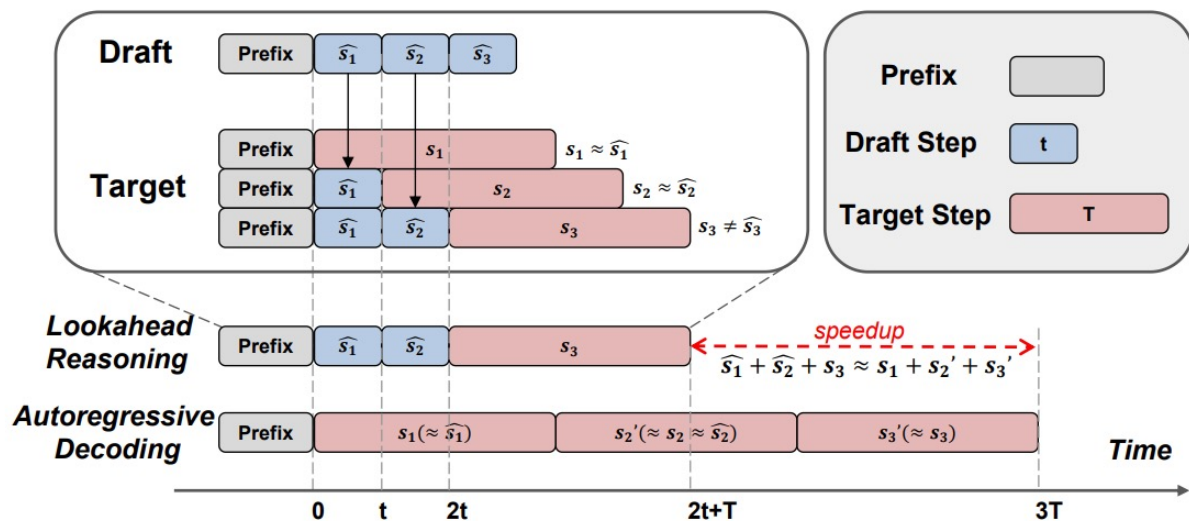
## Judge Decoding: 结果---每次接受更多tokens

- 更多的单次接受tokens数和更高的加速比，尤其适合long-cot推理

	$m_*$	HUGGINGFACE	GPT-FAST	TOKENS/S (512 + 512)
8B/70B-STANDARD	6.4	1.5×	1.7×	76.7
8B/70B-JUDGE (OURS)	18.8	2×	<b>3×</b>	<b>141.8</b>
70B-EAGLE-2	4.5	<b>3.3×</b>	1.9×	88.1
8B/405B-STANDARD	6.3	5.3×	1.78×	58.7
8B/405B-JUDGE (OURS)	19.7	<b>9.7×</b>	<b>3.9×</b>	<b>129.3</b>
405B-MEDUSA	< 6	< 6×	1.9×	108*

# Lookahead reasoning: step-level ✕ token-level parallelism

- **draft model**生成多个推理步，组成**batch**交给大模型并行验证，同时每步中基于SD并行



(a) One cycle of LOOKAHEAD REASONING



# Lookahead reasoning: step-level ✕ token-level parallelism

- 1. 4-2.1倍加速，形成了对SD进行继续**scaling**的新维度

Table 1: LOOKAHEAD REASONING’s Performance Across Datasets. Speedup is relative to the Autoregressive Decoding of the respective Target Model.

Method	Metric	Dataset						
		AIME24	AMC23	GSM8K	HumanEval	GPQA	MT-Bench	LiveCodeBench
Draft: Deepseek-R1-Distill 1.5B / Target: Deepseek-R1-Distill 32B								
Draft Model	Acc. (%)	28.5 ± 3.9	71.6 ± 4.1	77.6 ± 3.3	67.2 ± 2.4	9.6 ± 1.2	6.23 ± 1.9*	14.5 ± 1.3
Target Mode	Acc. (%)	70.8 ± 5.2	95.6 ± 2.3	91.8 ± 1.9	96.9 ± 0.8	63.3 ± 2.2	8.17 ± 1.2*	48.9 ± 1.3
SpecReason	Acc. (%)	58.3 ± 5.7	90.6 ± 2.6	85.9 ± 2.2	94.5 ± 1.5	57.0 ± 2.8	–	40.6 ± 1.5
	Apt.	0.39	0.69	0.93	0.43	0.08	–	0.25
LR(ours)	Acc. (%)	69.2 ± 8.1	94.1 ± 2.1	92.8 ± 1.8	95.5 ± 1.8	61.2 ± 2.8	8.13 ± 1.2*	49.5 ± 2.3
	Apt.	0.47	0.58	0.63	0.44	0.35	0.48	0.47
	Speedup	1.36×	1.48×	1.71×	1.27×	1.14×	1.27×	1.21×
SD	Speedup	1.53×	1.50×	1.39×	1.32×	1.48×	1.25×	1.45×
SD+LR(ours)	Speedup	1.82×	2.00×	2.11×	1.54×	1.63×	1.51×	1.58×
Draft: Qwen3 1.5B / Target: Qwen3 32B								
Draft Model	Acc. (%)	46.9 ± 8.1	84.2 ± 4.7	91.1 ± 1.6	85.4 ± 1.6	38.5 ± 1.4	7.96 ± 1.5*	28.8 ± 1.6
Target Model	Acc. (%)	80.0 ± 3.9	97.5 ± 2.0	96.6 ± 1.4	97.6 ± 0.8	68.2 ± 2.1	8.53 ± 1.1*	52.4 ± 1.4
SpecReason	Acc. (%)	68.3 ± 5.3	90.5 ± 3.9	94.5 ± 1.4	92.0 ± 2.0	66.3 ± 2.0	–	39.7 ± 1.9
	Apt.	0.75	0.92	0.95	0.91	0.46	–	0.65
LR(ours)	Acc. (%)	80.4 ± 4.1	96.4 ± 2.0	96.4 ± 1.2	97.1 ± 0.8	68.5 ± 2.4	8.46 ± 1.15*	51.7 ± 1.7
	Apt.	0.43	0.53	0.50	0.39	0.30	0.38	0.40
	Speedup	1.12×	1.22×	1.32×	1.13×	1.04×	1.10×	1.08×
SD	Speedup	1.40×	1.38×	1.32×	1.32×	1.40×	1.41×	1.25×
SD+LR(ours)	Speedup	1.49×	1.62×	1.68×	1.39×	1.44×	1.49×	1.32×



## 如何减少推理模型的生成token数目

- 两个维度：**thinking units vs. tokens**，对前者的缩减重要（冗余高）但困难

A Thinking Process from DeepSeek-R1

**Naturally Contains Multiple *Thinking Units***

Okay, so I need to [...]  $12 = x$  (*Initial Attempt*)

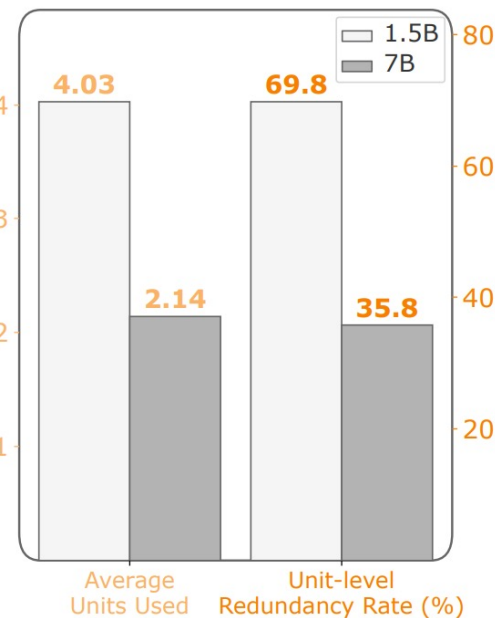
So,  $x$  equals 12? Let me check my steps to make sure I didn't make any mistakes [...] That seems to check out. (*Checking Steps*)

But wait, let me verify by plugging  $x = 12$  back into the original expressions and see if the average is indeed  $4x - 7$  [...] Hmm, so I think that's it. My answer is  $x = 12$ . (*Self-Verification*)

Wait, just to be thorough, let me check again if my initial equation was set up correctly [...] Yes, that's correct. (*Checking Again*)

Alternatively, maybe I can think of another way to approach the problem, just to confirm [...] So definitely,  $x$  is 12. So, confident now that the answer is 12. (*Another Approach*)

**\*\*Final Answer\*\***  
The value of  $x$  is  $\boxed{12}$ .







## “完成胜于完美”：冗长推理过程的multi-turn分解

- 将**long CoT**的格式进行根本改变，变成多个**turn**，每个包括一个**thinking units**和一个答案
  - 大大降低**first-token latency**
  - 将**thinkings units**的数目变得显式、**可控**

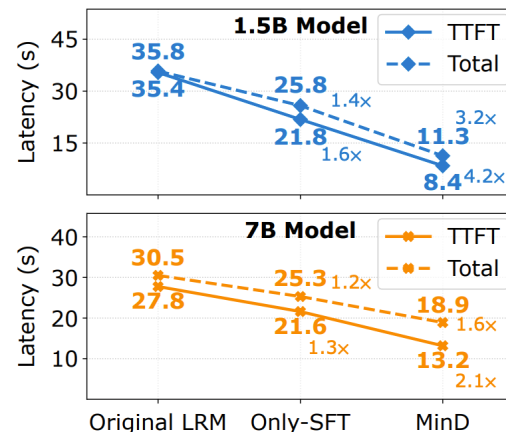
Question: Let $f(x) = 2x - 3$ and $g(x) = x + 1$ . What is the value of $g(f(5) - 1)$ ?	
DeepSeep-R1-Distill-Qwen-7B	MinD-7B
<p><b>&lt;think&gt;</b> Okay, so I need to find the value of <math>g(f(5) - 1)</math> [...] <math>g(f(5) - 1)</math> is 7. Wait, is there another way to approach this problem? [...] Both approaches lead to the same answer, 7. So, maybe I was overcomplicating it by thinking of composing functions, but it still gives the same result. Hmm, interesting. Wait, let me verify again [...] So, yes, the answer is 7. Alternatively, if I compute <math>g(f(5) - 1)</math> as follows: [...] Yep, same answer. [...] <b>&lt;/think&gt;</b> [...] Thus, the answer is <span style="border: 1px solid black; padding: 0 2px;">7</span>.</p>	<p><b>&lt;think&gt;</b> First [...] Therefore, <math>g(f(5) - 1)</math> is 7. <b>&lt;/think&gt;</b> [...] **Final Answer:** <span style="border: 1px solid black; padding: 0 2px;">7</span> <b>&lt;think&gt;</b> Wait, let me make sure I did that right [...] Yeah, that seems correct. <b>&lt;/think&gt;</b> [...] **Final Answer:** <span style="border: 1px solid black; padding: 0 2px;">7</span> <b>&lt;think&gt;</b> Let me think if there's another way to approach this [...] So, that method also gives me the same answer, which is 7. <b>&lt;/think&gt;</b> [...] **Final Answer:** <span style="border: 1px solid black; padding: 0 2px;">7</span></p>

- **GRPO**训练：提高少**turn**且正确的答案的生成概率

## “完成胜于完美”：冗长推理过程的multi-turn分解

- 在**Math**数据集训练的模型，降低70%的**token**用量；泛化到**AIME**、**AMC**、**GPQA**仍有效

	MATH-500		AIME24		AMC23		GPQA-Diamond	
	Acc.↑	Tokens↓	Acc.↑	Tokens↓	Acc.↑	Tokens↓	Acc.↑	Tokens↓
<b>1.5B</b>								
Original LRM	85.4	5389	26.7	15177	67.5	9956	32.3	9842
ThinkPrune [9]	83.2 -2.6%	1938 -64%	27.1 +1.5%	5631 -63%	73.2 +8.4%	3039 -70%	-	-
DEER [33]	73.2 -14.3%	1118 -79%	20.0 -25.1%	3302 -78%	47.5 -29.6%	2384 -76%	5.6 -82.7%	4128 -58%
MinD	82.8 -3.0%	1719 -68%	30.0 +12.4%	4856 -68%	77.5 +14.8%	2384 -76%	31.3 -3.1%	4690 -52%
<b>7B</b>								
Original LRM	93.0	3928	50.0	14107	90.0	6076	50.5	8390
Dynasor [6]	88.5 -4.8%	2591 -34%	47.7 -4.6%	8760 -38%	87.1 -3.2%	4913 -19%	-	-
DEER [33]	87.4 -6.0%	975 -75%	33.3 -33.4%	3235 -77%	82.5 -8.3%	1622 -73%	27.3 -45.9%	2265 -73%
MinD	91.6 -1.5%	2859 -27%	46.7 -6.6%	7258 -49%	95.0 +5.6%	3777 -38%	53.0 +5.0%	6845 -18%





## 总结

- 高效多模态生成方法:
  - 扩模态统一建模
  - 并行加速
  - 深度推理

感谢各位专家！ 敬请批评指正！

邮箱: [zhijied@sjtu.edu.cn](mailto:zhijied@sjtu.edu.cn)

主页: <https://thudzj.github.io/>

