

# The Era Of Open Experience

Zhipeng Huang, LF&AI Foundation Board Rep



## Outline

- Community: LFAI&Data Foundation Introduction
- Technology: General Trends of LLM Development
- Opportunity: Era of Open Experience with OpenRLGR

## Community: LFAI&Data Foundation Introduction



# LFAI&Data Foundation

GOSIM

**THE LINUX FOUNDATION PROJECTS**

OPEA Virtual Week, April 14-18 REGISTER NOW →

**LF AI & DATA**

- ABOUT
- PROJECTS
- RESOURCES
- EVENTS
- BLOGS
- JOIN
- MODEL OPENNESS FRAMEWORK

X in YouTube GitHub 🔍

## LF AI & Data Foundation Stats

We are a community of innovators. We connect you with groundbreaking technical projects, companies, and developer communities that are transforming artificial intelligence, machine learning, and deep learning in open source.

67 Projects	200M+ LOC	331K+ Commits	2,920+ Contributing Organizations	100K Contributors	9 Technical Working Groups

**Trusted & Responsible AI**

- Bias & Fairness: (INCUBATING)
- Open Voice TRUSTMARK: (INCUBATING)
- Intersectional Fairness: (SANDBOX)

**Distributed Computing**

- Computing & Management: (INCUBATING) | (INCUBATING)
- Interface: (INCUBATING) | (INCUBATING)

**Natural Language Processing**

- Delta: (INCUBATING)
- RosenNLG: (SANDBOX)

**Notebook Environment**

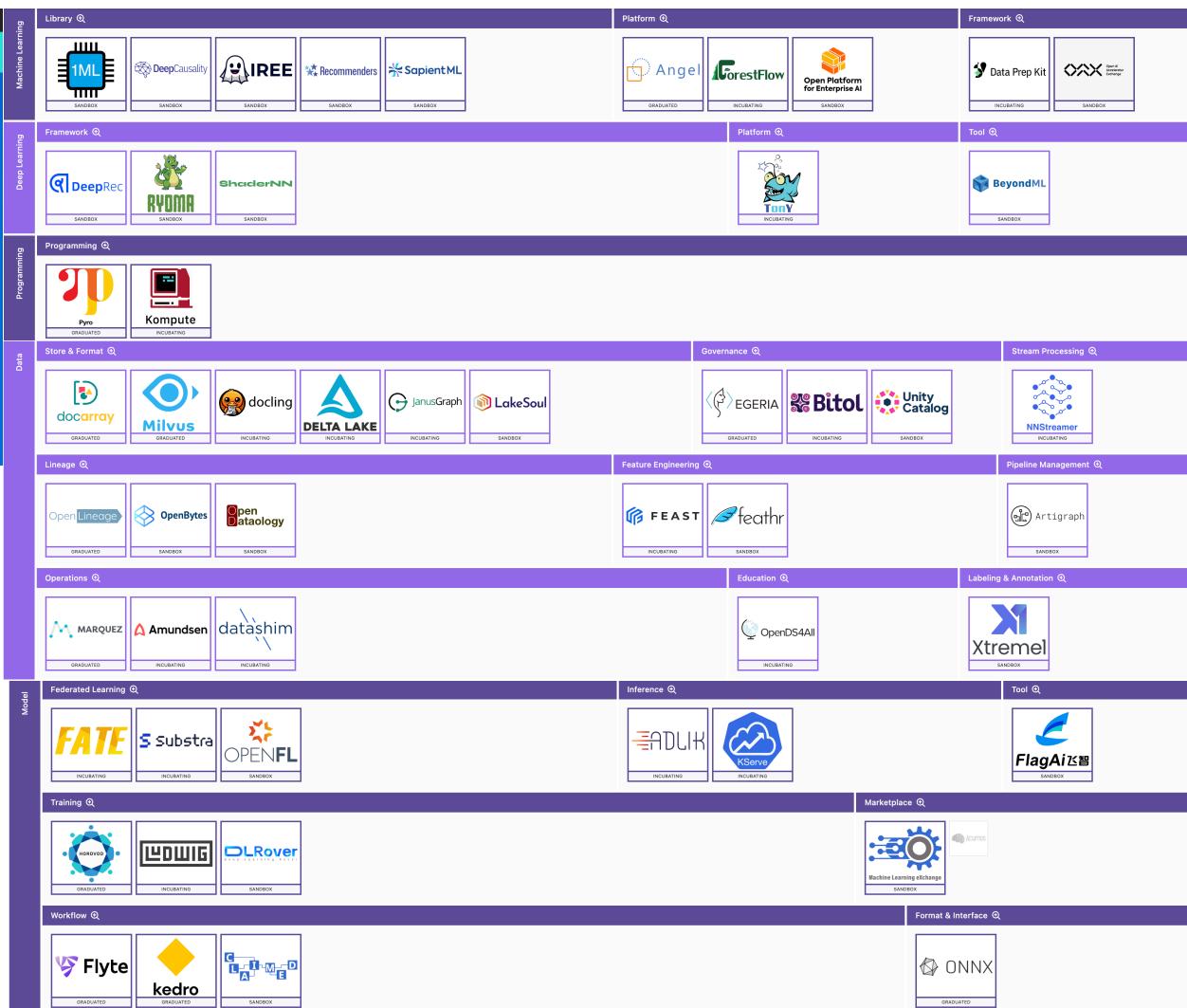
- Elyra: (SANDBOX)

**Generative AI**

- RWKV: (INCUBATING)
- Open Model Initiative: (SANDBOX)

**Tools**

- BeeAI: (INCUBATING)
- Monocle: (SANDBOX)



<https://landscape.lfai.foundation/?group=projects-and-products&view-mode=grid&project=lfaidata>  
**GOSIM AI Paris 2025**



# LFAI&Data Foundation

GOSIM

**THE LINUX FOUNDATION PROJECTS**

OPEA Virtual Week, April 14-18 REGISTER NOW →

**LF AI & DATA**

ABOUT ▾ PROJECTS ▾ RESOURCES ▾ EVENTS ▾ BLOGS ▾ JOIN MODEL OPENNESS FRAMEWORK X in YouTube Pinterest Search

The APAC Regional User Group (RUG) is a collaborative community dedicated to advancing the open-source artificial intelligence ecosystem across the Asia-Pacific region, with a particular focus on driving technological advances and market dynamics of APAC. Aligned with the LF AI & Data Foundation's mission, APAC RUG fosters knowledge sharing, networking, and collaboration among individuals and organizations passionate about AI innovation and ecosystem building, while embracing the diverse cultural and economic landscapes of the region.



亚太地区用户组 (APAC Regional User Group, RUG) 是一个合作社区，致力于推动亚太地区开源人工智能生态系统的发展，尤其关注推动亚太地区和中国的技术进步和市场动态。APAC RUG 与 LF AI & Data 基金会的使命保持一致，通过促进知识共享、网络交流和合作，汇聚热衷于人工智能创新与生态系统建设的个人和组织，同时兼顾该地区的多样化文化和经济环境。

[Join APAC RUG.](#)



The LF AI & Data Foundation Japan RUG (Regional User Group) empowers Japanese AI engineers by fostering a collaborative community. By providing community members with opportunities and incentives for discussion and knowledge sharing in Japanese, the Japan RUG facilitates these efforts. It tackles practical challenges in applying open-source AI technologies. This includes promoting best practices, cultivating Japanese AI talent, and contributing to open-source projects within and beyond the LF AI & Data Foundation. Japanese-language interaction within the Japan RUG is key to fostering a deep understanding of AI technologies and driving innovation within the global open-source AI ecosystem.

LF AI & Data Foundation Japan Regional User Group (RUG)は、協力的なコミュニティを育むことで、日本のAIエンジニアを支援します。Japan RUGは日本語で議論や知識共有ができる場を提供することでこれらを促進し、オープンソースAI技術の応用における実践的な課題に取り組んでいきます。これには、ベストプラクティスの推進、日本のAIエンジニアの育成、LF AI & Data Foundation内外におけるOSSプロジェクトへの貢献が含まれます。Japan RUGで日本語で交流することで、AI OSSへの深い理解やグローバルなオープンソースAIエコシステムにおけるイノベーションの推進につながると考えています。

[Join Japan RUG.](#)

**GOSIM AI Paris 2025**

**GENERATIVE AI COMMONS**

HOME PROJECTS ▾ RESOURCES ▾ JOIN X in YouTube Pinterest

**Workstream Leaders**

	<b>Anni Lai</b> Generative AI Commons Co-Chair <a href="#">LinkedIn</a>		<b>Arnaud Le Hors</b> Generative AI Commons Co-Chair <a href="#">LinkedIn</a>		<b>Nick Chase</b> MAD Workstream Lead <a href="#">LinkedIn</a>		<b>Sachin Varghese</b> MAD Workstream Lead <a href="#">LinkedIn</a>
	<b>Raghavan Muthuregathan</b> MAD Workstream Lead & Education and Outreach Workstream Lead <a href="#">LinkedIn</a>		<b>Susan Malaika</b> Responsible AI Workstream Lead <a href="#">LinkedIn</a>		<b>Suparna Bhattacharya</b> Responsible AI Workstream Lead <a href="#">LinkedIn</a>		<b>Ofer Hermoni</b> Education and Outreach Workstream Lead <a href="#">LinkedIn</a>

[mot.lisitopen.ai](#)

Name	Organization	Classification	Last updated	Badge
Aquila-VL-2B	Beijing Academy of Artificial Intelligence (BAAI)	Class I - Open Science Model	2025-02-25	Class I - Open Science Model Qualified
Mistral-8x7B-v0.1	Mistral	Class III - Open Model	2025-03-13	Class II - Open Tooling Model In progress (60%) Class III - Open Model Qualified
RWKV-v4-(Raven)	RWKV	Class III - Open Model	2025-02-25	Class II - Open Tooling Model In progress (50%) Class III - Open Model Qualified
RWKV-v4-World	RWKV	Class III - Open Model	2025-02-25	Class II - Open Tooling Model In progress (50%) Class III - Open Model Qualified
RWKV-v5	RWKV	Class III - Open Model	2025-02-25	Class II - Open Tooling Model In progress (50%) Class III - Open Model Qualified
Baichuan-1-13B	Baichuan AI	Class III - Open Model	2025-02-25	Class II - Open Tooling Model In progress (40%) Class III - Open Model Qualified
Mistral-7B-Instruct-v0.1	Mistral	Class III - Open Model	2025-03-13	Class II - Open Tooling Model In progress (70%) Class III - Open Model Qualified
Mistral-7B-Instruct-v0.2	Mistral	Class III - Open Model	2025-03-13	Class II - Open Tooling Model In progress (60%) Class III - Open Model Qualified
Mistral-7B-Instruct-v0.3	Mistral	Class III - Open Model	2024-10-03	Class II - Open Tooling Model In progress (60%) Class III - Open Model Qualified

## Technology: General Trends of LLM Development



# Trend 1 - Unified Architecture For MLLM GOSIM

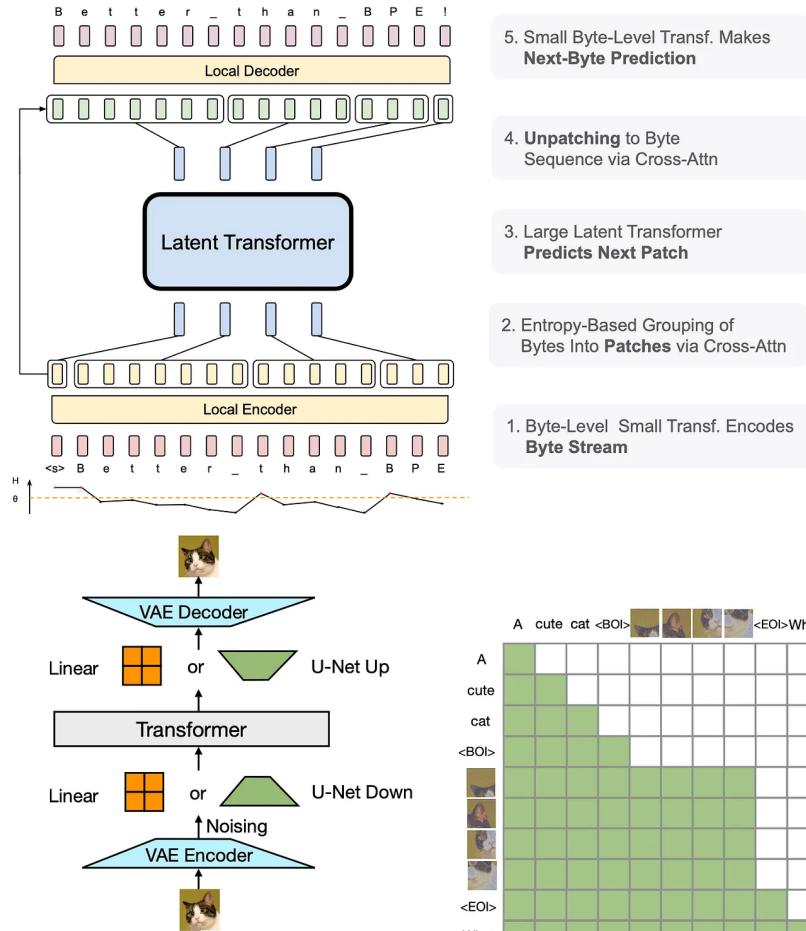


Figure 3: We convert images to and from latent representations using a pretrained VAE, and then into patch representations with either a simple linear layer or U-Net down blocks.

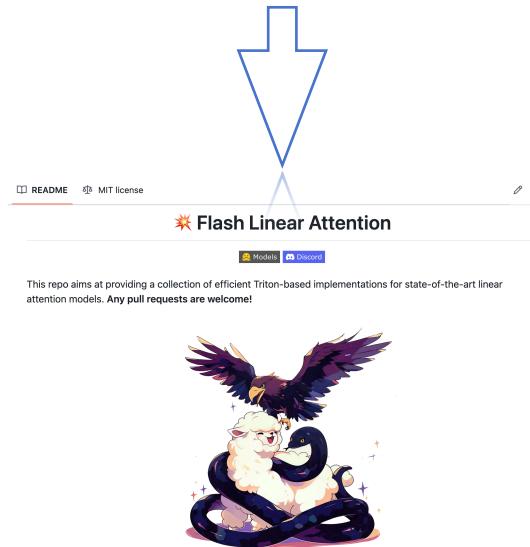
Figure 4: Expanding on the causal mask, Transfusion allows patches of the same image to condition on each other.

This field saw a turning point at H2 2024 with two papers aiming at two directions: token-level and byte-level. While both signals a significant development direction - finally **a unified architecture for generative model**, another common ground almost uncanny is that **the encoder-decoder architecture just might make a great come back this time**.

# Trend 2 - Linear, Sparse and Hybrid

GOSIM

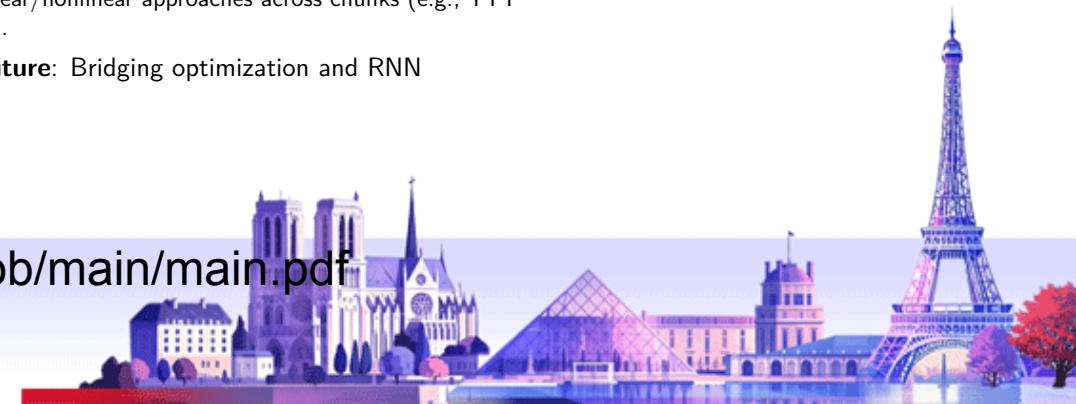
- ▶ Training: quadratic time complexity
  - ▶ Expensive for long sequence modeling (e.g., video or DNA modeling)
- ▶ Inference: linear memory complexity
  - ▶ Requires storing KV cache for each token
  - ▶ High memory burden.



Use linear recurrence for parallel training

- ▶ Gated linear RNNs (HGRN, Griffin, ...)
- ▶ State-space models (S4, Mamba, ...)
- ▶ Linear attention (RetNet, GLA, xLSTM, DeltaNet, ...)

- ▶ **Modern RNNs through the lens of online learning:**
  - ▶ (Decaying/Gated) Linear attention: Negative inner-product loss.
  - ▶ (Gated) DeltaNet: Linear regression loss.
  - ▶ TTT & Titans: Nonlinear regression losses.
- ▶ **Gradient-based optimization techniques prove valuable:**
  - ▶ Weight decay enables effective forgetting (e.g., Mamba2, Gated DeltaNet).
  - ▶ Momentum improves performance (e.g., Titans).
- ▶ **Efficient hardware utilization via:**
  - ▶ Chunkwise training for linear attention.
  - ▶ Hybrid linear/nonlinear approaches across chunks (e.g., TTT & Titans).
- ▶ **Promising future:** Bridging optimization and RNN architectures.



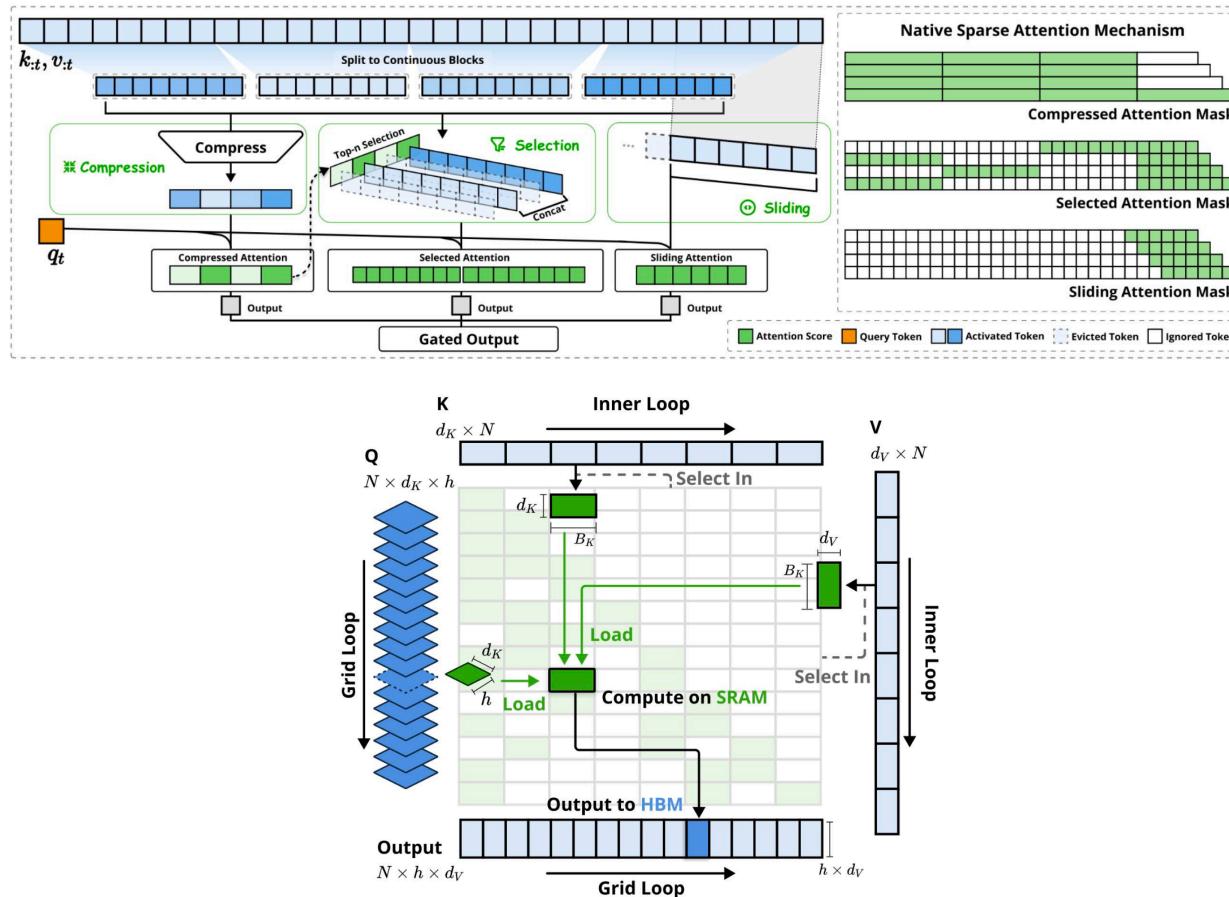
<https://github.com/sustcsonglin/linear-attention-and-beyond-slides/blob/main/main.pdf>

**GOSIM AI Paris 2025**

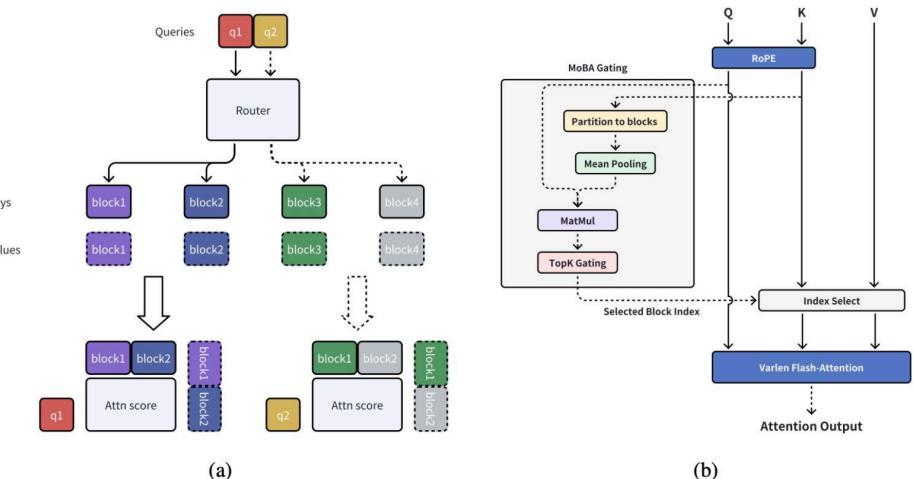
# Trend 2 - Linear, Sparse and Hybrid

GOSIM

DeepSeek-NSA



Kimi-MOBA



<https://github.com/sustcsonglin/linear-attention-and-beyond-slides/blob/main/main.pdf>

**GOSIM AI Paris 2025**

# Trend 2 - Linear, Sparse and Hybrid

GOSIM

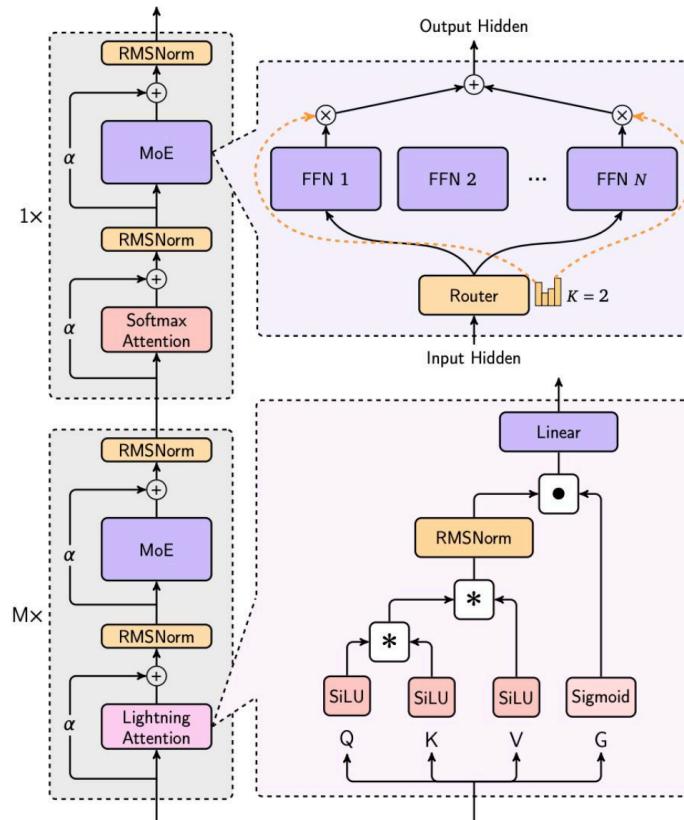


Figure 3 | The architecture of MiniMax-Text-01.



Tri Dao ✅ @tri\_dao · Jan 15

Hybrid linear-softmax attention working very well at large scale and long-context! As we've seen with multiple models now, you only need a couple of (full) attention layers

MiniMax (official) ✅ @MiniMax\_AI · Jan 14

MiniMax-01 is Now Open-Source: Scaling Lightning Attention for the AI Agent Era

We are thrilled to introduce our latest open-source models: the foundational language model MiniMax-Text-01 and the visual multi-...

MiniMax-01 (MiniMax et al. 2025) used hybrid attention: **7/8** linear attention layers + **1/8** softmax attention layers, with simple linear attention using data-independent decay: Lightning-Attention (Qin et al. 2024b).

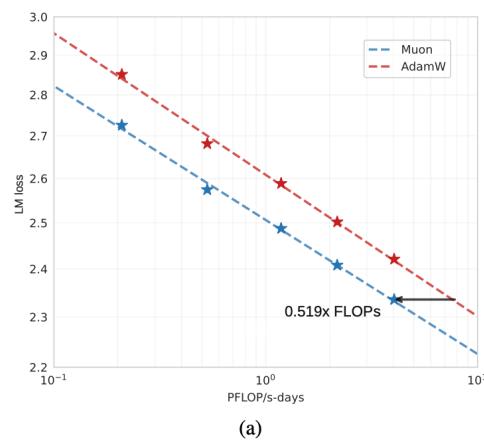
# Trend 3 - Second Order Optimization

GOSIM

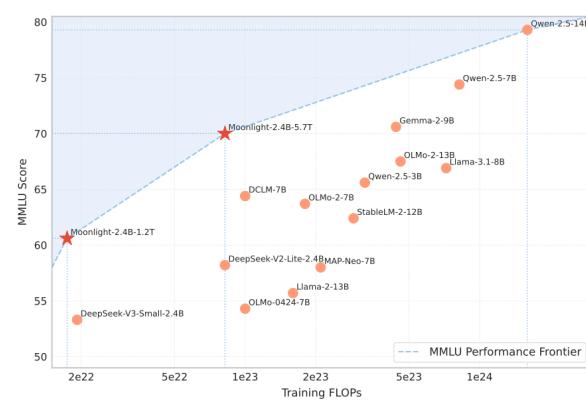
Muon, Shampoo, PSGD, MARS,...

$$W \leftarrow W - \eta \times \sqrt{\frac{\text{fan-out}}{\text{fan-in}}} \times \text{NewtonSchulz}(\nabla_W \mathcal{L}).$$

In Muon, we are exploiting the geometry of a matrix that acts as a linear operator between RMS-normed vector spaces. In particular, we build on a geometric connection between distance in weight space and distance in function space.

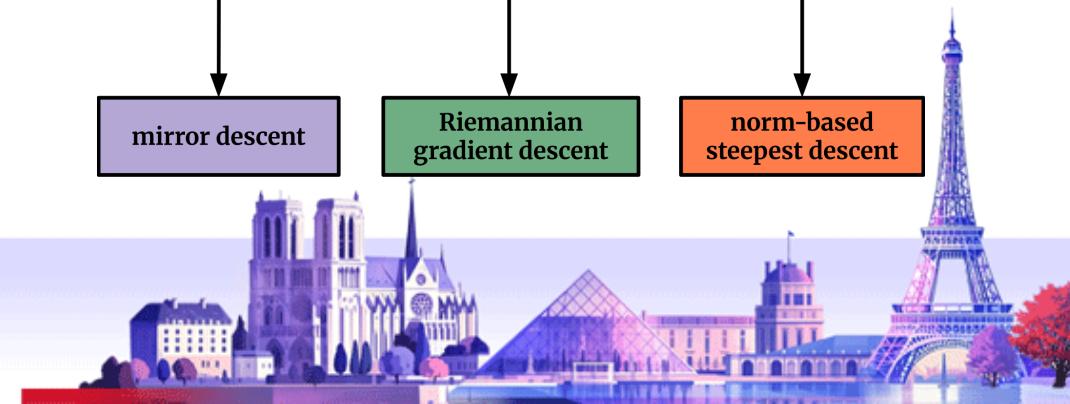
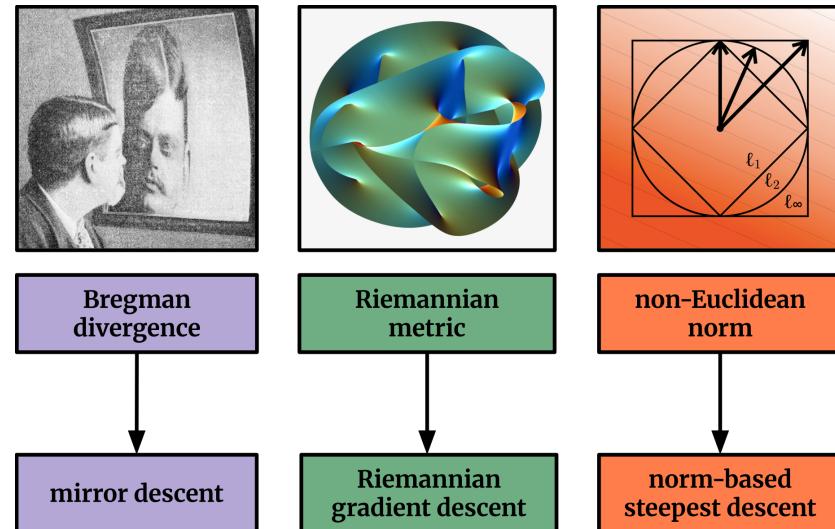
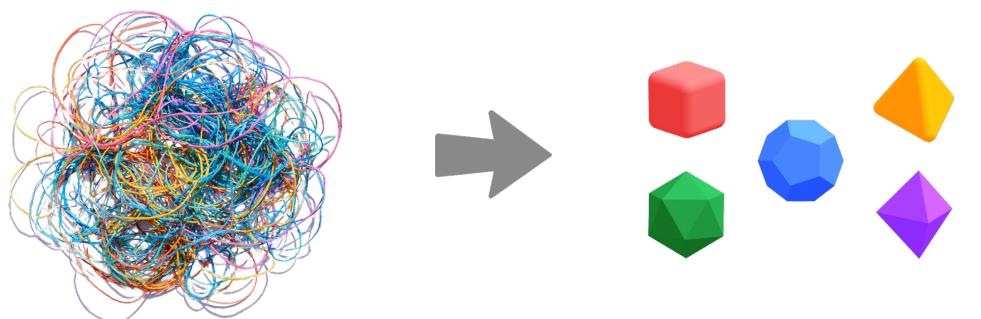


(a)



(b)

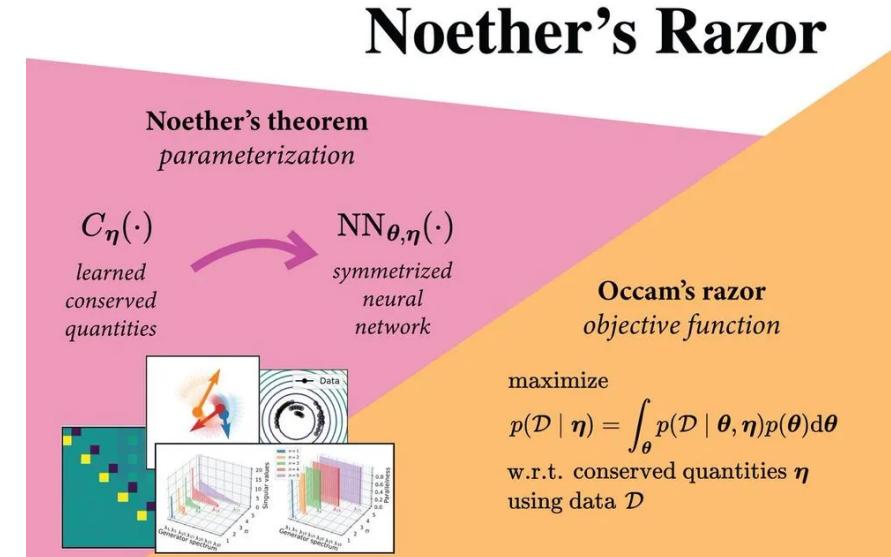
Figure 1: Scaling up with Muon. (a) Scaling law experiments comparing Muon and Adam. Muon is  $\sim 2\times$  more computational efficient than Adam with compute optimal training. (b) The MMLU performance of our Moonlight model optimized with Muon and other comparable models. Moonlight advances the Pareto frontier of performance vs training FLOPs.



# Trend 4 - From AI4S To Equivariance4ALL GOSIM



GOSIM AI Paris 2025



The only way to correctly and efficiently learn the real physical world's ever massive priors, is to have equivariant-aware transformers, to have things like the Noether's razor and GINN to work.



# Trend 5 - Build The Data Oil Rigs

GOSIM

1. **Perfect Validation Methods** - This is your MuZero self play type stuff with Chess and Go boards, synthetic math datasets based on proof assistants like Lean and Coq, "generate a test suite and then find code that passes the test suite", basically anything where you have a formal procedure/algorithm to test the correctness of your solution and you can guarantee that every training sample found by your process is correct. That these work is well understood and there's plenty of papers in the literature about making synthetic data with these methods.
2. **Backtranslation Methods** - RetroInstruct is named after these. The idea is that we start with known-good answers and then make an index over them with a general model. "Write an English summary of this code and then reverse the ordering so we train summary -> code instead of code -> summary" is an example of this. We can also make synthetic corruption passes where we start with a known good example, introduce errors to it with the pass and then teach a model to undo the errors. For example we could make a dataset where a model samples a flaw it could introduce into known good code, adds it to the code, and then reverse causality so we start with the buggy code, "diagnose" the problem with the instruction we used to put the flaw into the code in the first place, and end with the known-good code to thoroughly teach a model debugging. I think these methods are the least appreciated in the current literature and have *huge* low hanging fruit available. e.g. I think you could make the best general question and answer dataset by doing backtranslation on Wikipedia, you could make the best debugging dataset with the method I previously described, etc. There's opportunities to make expansive high quality corpora from existing royalty free data that teach models to present a much better interface to their understanding than we normally get. We can further improve the quality of the index over the answers by using weave evaluator and toolformer type methods to check our correctness against a strong models preexisting understanding.
3. **Rejection Sampling Methods** - This is where we create synthetic data by rejection sampling from the model based on some reward modeling or classifier. Notably the reward model or classifier doesn't actually need to be the model as the one we're sampling from, so we can in principle distill the knowledge from another model into a dataset by rejection sampling from our text prior to extend it. In the weave MCTS setup we're using minimal data from outside the model (i.e. user prompt + grading rubric) so this is where we have to worry about our model becoming ungrounded from reality. Technically multi-agent setups are a form of this (<https://x.com/sebkrier/status/1790016617897681170>) even if they tightly constrain the models action space and context. I think the current literature is slowly starting to appreciate these methods but there's still a ton of low hanging fruit here that I'd like to execute on.
4. **Careful Prompting** - We can think of getting the right output from our models as a search process which the model has n bits of in its branching space. e.g. Maybe we need to divide our search space in half 32 times to get the right answer reliably. When we write a careful prompt we're essentially finding a program in the models latent space which can reliably traverse n bits of the hypothesis space to land in the right ballpark. This means that even without rejection sampling or filtering it's possible to generate useful synthetic data just by finding a careful prompt that does something well and recontextualizing its answers to more normal prompt contexts.

[https://x.com/jd\\_pressman/status/1797396190210015716](https://x.com/jd_pressman/status/1797396190210015716)

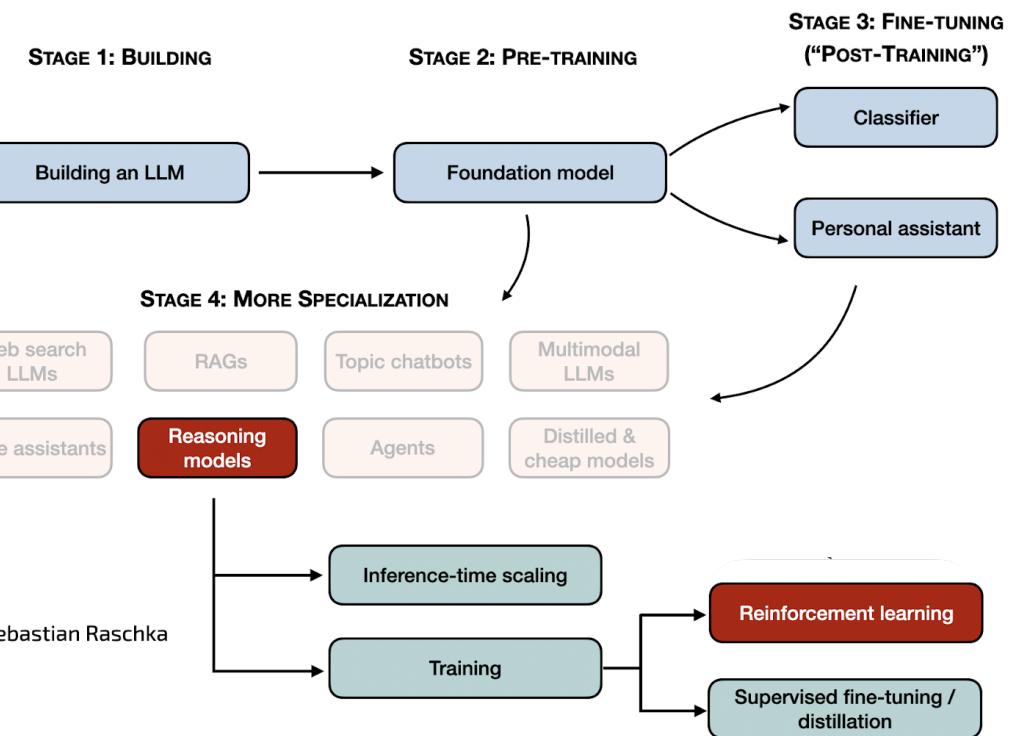
There is also good overview from DeepMind that GANs/Diffusion Models/RAG could be used for generate new data.

Other than the data itself, we will also have **more and more tools for automation of processing**. RegMix and ProX are two great examples what kind of tooling we should have more of. LeRobot from HugginFace also becomes a great open source ecosystem of building datasets and its various tooling.



# Trend 6 - Post-training enter center stage GOSIM

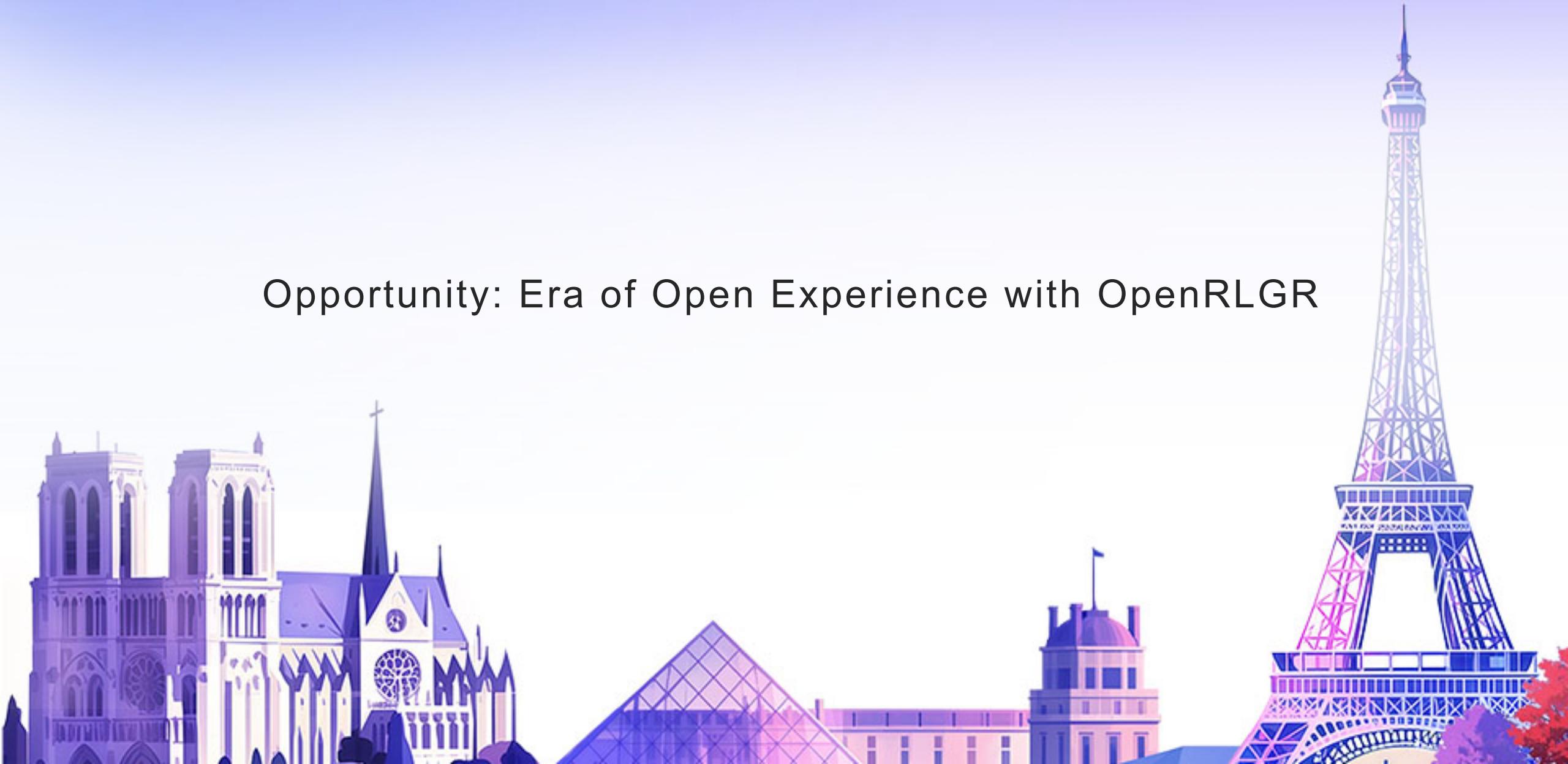
Scaling RL: “more than 10x the training compute than o1”



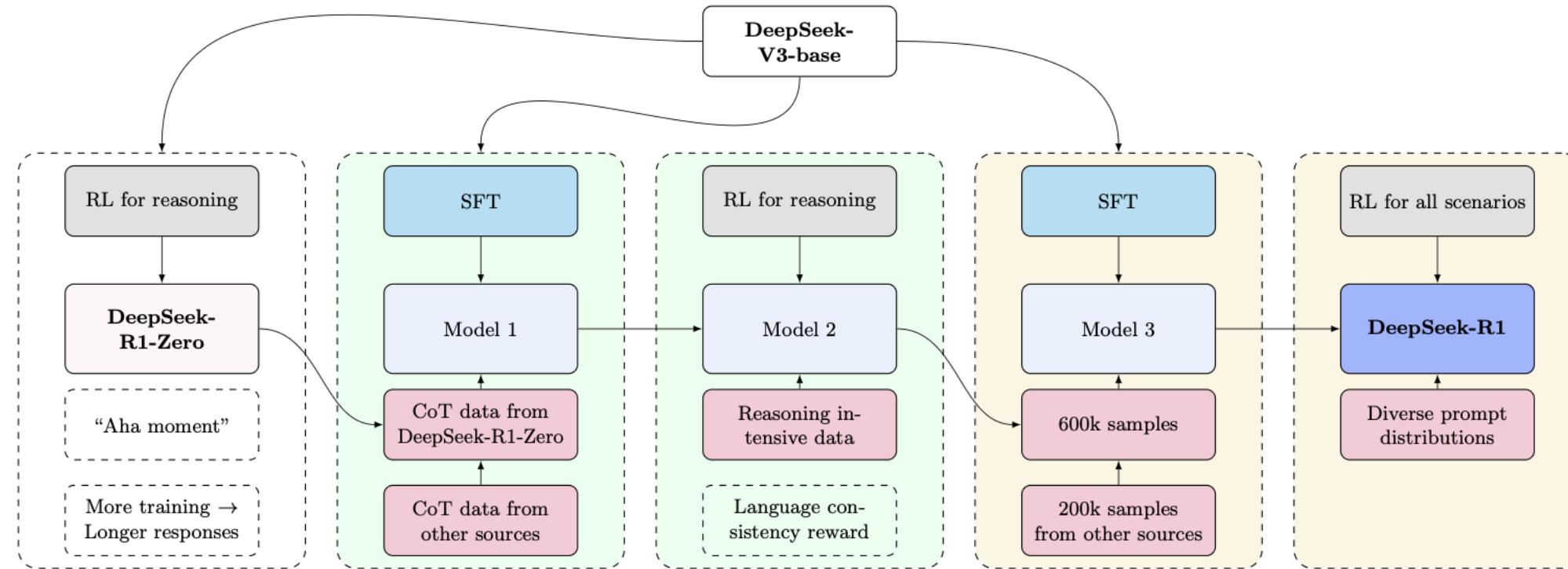
<https://sebastianraschka.com/blog/2025/the-state-of-reinforcement-learning-for-llm-reasoning.html>

**GOSIM AI Paris 2025**

Opportunity: Era of Open Experience with OpenRLGR



# Motivation - RL Era kicked off by DeepSeek GOSIM

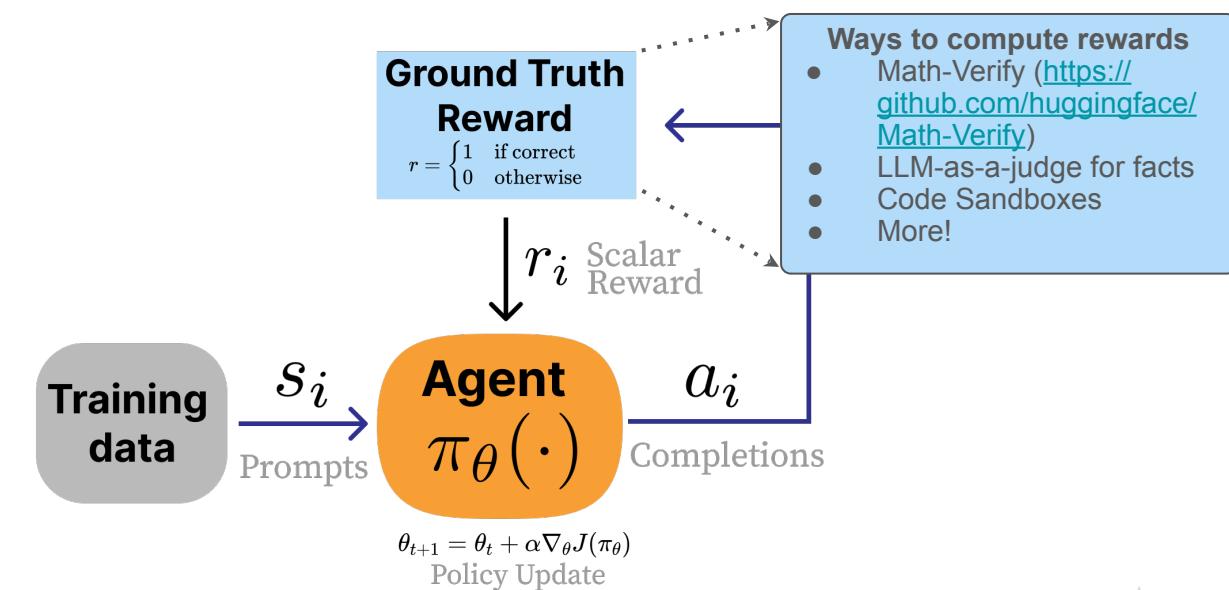
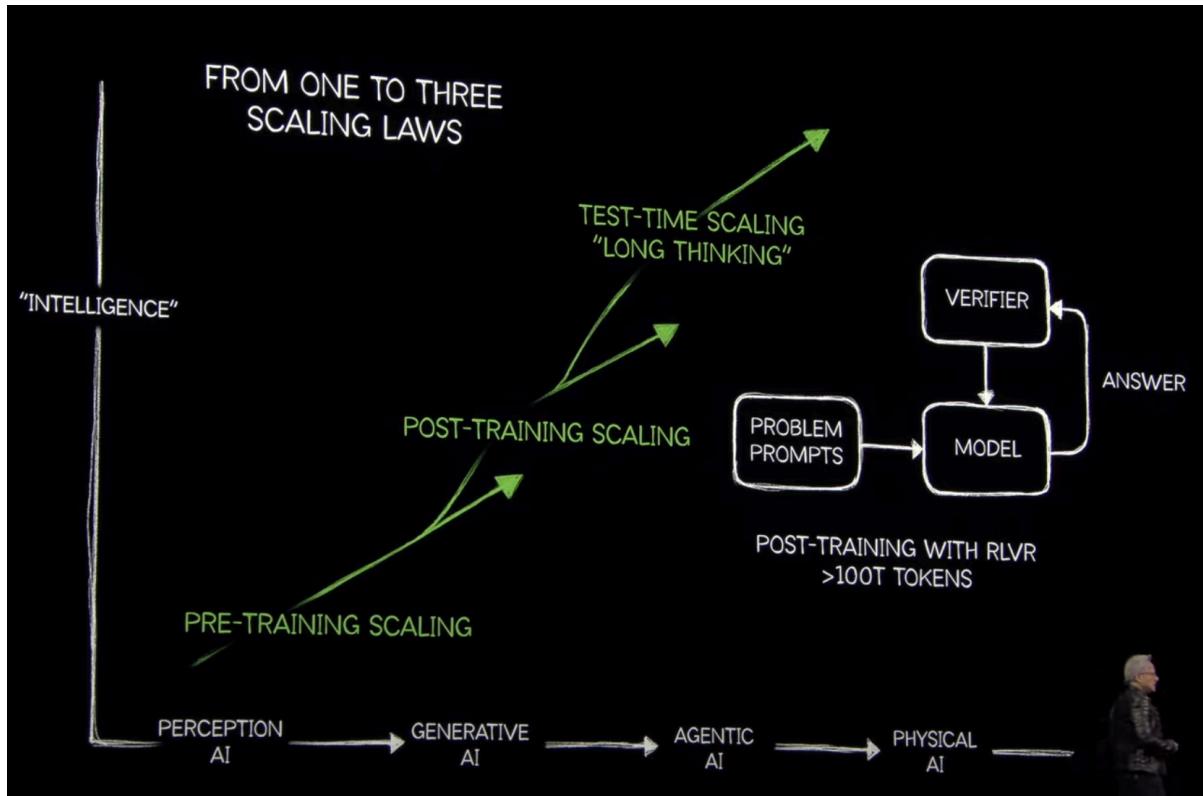


<https://arxiv.org/abs/2504.07128>

# Motivation - RL Era With RLGR

GOSIM

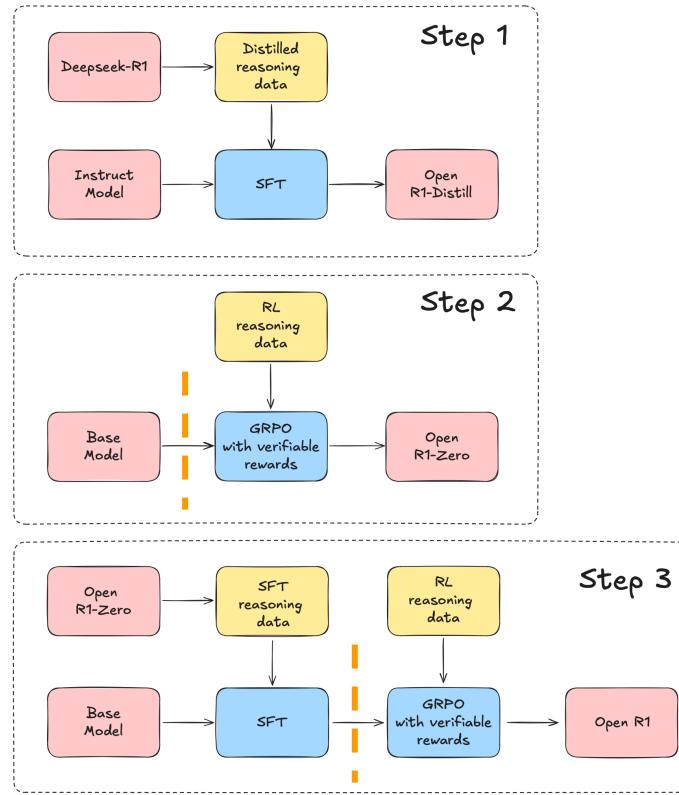
## RLGR: Reinforcement Learning With Generalized Rewards



Lambert, Nathan et al. 2024. Tülu 3.

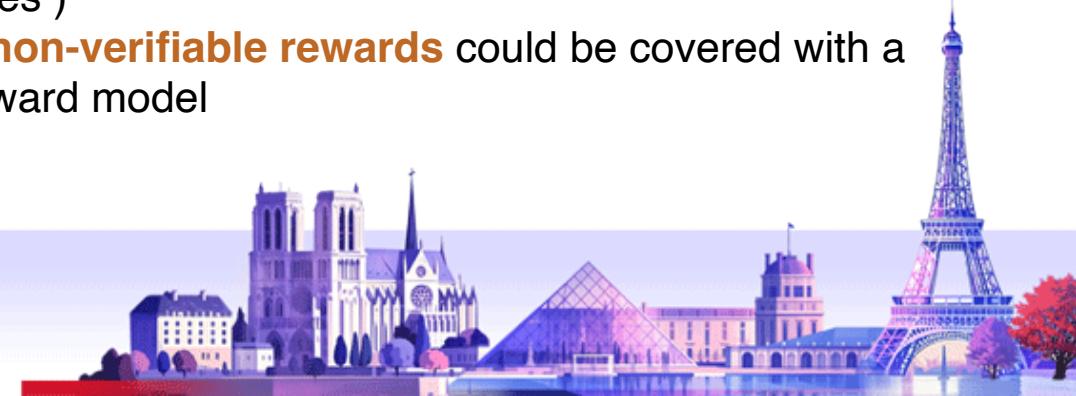
# Motivation - RL Era With RLGR

GOSIM



RL Flywheel happens right hand of the orange dotted line,  
above figure from HF's [Open-R1 project](#), left figure is  
nvidia's using R1 for generating new ops

1. DeepSeek R1's release kickstarted a new era of totally decoupled pre-training/post-training, where the latter has a signature of flywheel suited for OSS collaboration
2. Core idea from @georgejrjrjr : <https://x.com/georgejrjrjr/status/1889761173349753048>
3. TL;dr: we should collectively build an OSS ecosystem that encompass:
  - a. **Reasoning data selection** (reasoning traces in distribution, prompt filtering)
  - b. **Distillation data refinement** (pruning and its limits / pivot token data mixing laws, Re-writing reasoning traces for low loss, Reasoning condensation, Persona reasoning)
  - c. Document **prerequisites for reasoning distillation**
  - d. Document **effectiveness of LoRA/DoRA/etc**
  - e. **Build simple test-time compute levers** (collect efforts that insert pivot tokens, branching out running multiple streams in parallel, selecting and aggregating responses )
  - f. Collaborate on how **non-verifiable rewards** could be covered with a more generalized reward model



# Current Work - OpenRLGR Ontology

GOSIM

## Ontology of RL Targets

### Verifiable Rewards

Strengths: the results are verifiably, quantitatively correct.  
Weakness: many qualities and domains are not obviously amenable to verifiable correctness.

#### Math

- Arithmetic (GMS8K-like)
- Competition math (AIME-like)
- "Real math" (Lean et al)
- Figures of merit: correctness, token usage (is proof concision a concern?).

#### Code

- Interview questions (LeetCode)
- Realistic software engineering (CommitPack, SWE-Bench, SWE-Lancer)
- Transpilation
- High performance compilation (eg, [KernelBench](#))
- Code reasoning ([CruxEval](#), [CodeIO](#))
- Figures of merit: test pass rate, execution time, lines of code, token usage.

#### Verifiable Instruction following

- Format compliance
  - Language
  - Schemas
  - Tool use
  - Diffs
- String processing (e.g., AutoIF)
  - Compression / Decompression
  - Presence or absence of words, suffixes, stems, etc.
  - Palindromes
  - Text encodings (e.g., Base64)
  - Ciphers
  - Length of response
  - Parts of speech
  - Reading level
  - Corruption repair

### Gymnasia

- Terminal games (e.g., nethack)
- Console / arcade emulators (e.g., MAME)
- Board games (e.g., chess)
- Minecraft
- Puzzles (e.g., Rubik's Cubes, Mazes)
- Cybersecurity
  - Red team exercises (capture the flag: VulnHub, HITB, etc)
  - Blue team exercises (log monitoring, intrusion detection, vulnerability detection)
  - Malware analysis

### Neural rewards

Strengths: works to some degree on ~any quality expressible in words  
Weaknesses: reward hacking, eg, with listicles and length.

### Methods

#### LLM-as-Judge

- Single judge
- Panel of judges
- Adversarial adjudication

#### Reward models

- Human conditioned (RLHF)
- Constitutionally conditioned (RLAIE)
- [redacted]

#### Generative RMs

- [redacted]

### Factual (Grounded) Judgements

- Document / Search Grounded QA
- Forecasting
- Exam-style grading
- Prompt Adherence
- Output distribution mirroring ([https://arxiv.org/abs/2502.01697](#))
- Narrative adherence

### Qualitative (Ungrounded) Judgements

- Prompt quality ([https://arxiv.org/abs/2501.18578](#))
- Emotional intelligence ([https://ebench.com/](#))
- Creative writing quality ([https://ebench.com/creative\\_writing.html](#))
- Creative writing judge quality ([https://ebench.com/judgemark-v2.html](#))
- Humor ([https://ebench.com/buzzbench.html](#))
- Image quality ([https://arxiv.org/abs/2501.18096](#))
  - SVG generation
  - Text-to-image prompting
- Legality
- Audience Appropriateness
  - Age
  - Intelligence
  - Knowledge / Background
  - Background knowledge
  - Culture
- Persona adherence
  - Honesty/Humility
  - Emotional Stability
  - eXtroversion
  - Agreeableness
  - Conscientiousness
  - Openness
  - Compassion / Empathy

### Reasoning Repositories

#### Reasoning Data

- [General Reasoning](#)
- [Open Thought](#)
- [SYNTHETIC-1](#)

#### Gymnasia

- [Reasoning Gym](#)
- [DOOM-Mistral](#)

<https://github.com/OpenRLGR/ontology>, originally drafted by George and others



# Current Work - Various OSS/Research Efforts GOSIM

**RLVR in Vision Language Models: Findings, Questions and Directions**

Authors: Liang Chen · Lei Li · Haozhe Zhao · Yifan Song · Vinci · Zihao Yue · Lingpeng Kong · Qi Liu · Baobao Chang

R1-V Deep-Agent · Updated 4 days ago

Introduction

Methods

Visual RL with Verifiable Reward

SFT with Visual Reasoning Trace Distilled from R1

Experiments, Findings and Questions

Visual Counting

Geometry Reasoning

Complex Visual Reasoning with R1-Distilled Data

Future Directions

Citation

**SYNTHETIC-1**

Completed: 100.00% 2,000 / 2M samples

Contributors: [Circular map showing contributors from around the world]

Latest Samples: [List of recent samples]

Question:

Consider a 9x9 rectangular grid of alternating dark and light squares, similar to a chessboard, which starts with a dark square in the top-left corner. How many more dark squares than light squares are there in the grid? Assume how many more dark squares there are than light squares.

Answer:

Okay, let's see. I need to figure out how many more dark squares there are than light squares on a 9x9 grid that's colored like a chessboard, starting with a dark square in the top-left corner. Here, chessboards alternate colors, right? So the top-left square is dark, followed by light, dark, light, etc. Then the next row starts with light, then dark, and so on. This means that every second row starts with a light square. So we have 5 dark squares in the first row, and 4 light squares. Then the next row starts with light, then dark, and so on. This means that every second row starts with a light square. So we have 5 dark squares in the first row, and 4 light squares. Now, to find the number of dark squares minus the number of light squares. Since the grid has 5 rows and 9 columns, the total number of squares is 5\*9=45. So there are 5 dark squares and 4 light squares, i.e., 1 = 45. We need to find D = 1.

Here, maybe there's a pattern here. On a chessboard, which is a 8x8 grid, the number of dark and light squares is roughly equal. But here, we have a 9x9 grid, which is odd. So we need to consider the case where the total number of dark and light squares. Since each row has 9 squares, which is odd, the starting color of each row will be different. Let me analyze a single row. If it now starts with dark and has 9 squares, since 9 is odd, the number of dark squares will be 5 and light squares will be 4. So we have 5 dark squares and 4 light squares in this row. Now, let me check. If you start with dark and alternate, position 1,3,5,7,9 are dark (S), and 2,4,6,8 are light (L). If you start with light, position 2,4,6,8 are light (L), and 1,3,5,7 are dark (S). So each row has 5 dark squares and 4 light squares. If you start with light, position 2,4,6,8 are light (L), and 1,3,5,7 are dark (S).

<https://deepagent.notion.site/rivr-in-vlms>

<https://huggingface.co/collections/PrimeIntellect/synthetic-1-67a2c399cfdd6c9f7fae0c37>

## Verifiers: Reinforcement Learning with LLMs in Verifiable Environments

This repository contains a set of tools for reinforcement learning with LLMs in verifiable environments.

### Installation

PyPi [coming soon](#) once a couple more features are added, just clone it for now and run:

```
uv pip install -e .
uv pip install flash-attn --no-build-isolation
```

Ensure your wandb and huggingface-cli logins are set up (or set report\_to=None in training\_args).

Tested with Python 3.11 and this [image](#). If you encounter version issues, please confirm that you are able to run basic TRL training in your environment before opening an issue. flash-attn and liger-kernel are used for performance reasons. Recommended usage is via accelerate with DeepSpeed ZeRO 3 ([example config](#)) but torchrun works in my tests as well.

You can also use this [gist](#) from (@kalomaze) <https://github.com/kalomaze> to quickly install and run an example script (maybe outdated now idk).

<https://github.com/willccb/verifiers>

**Global Leaderboard**

Model performance across all tasks

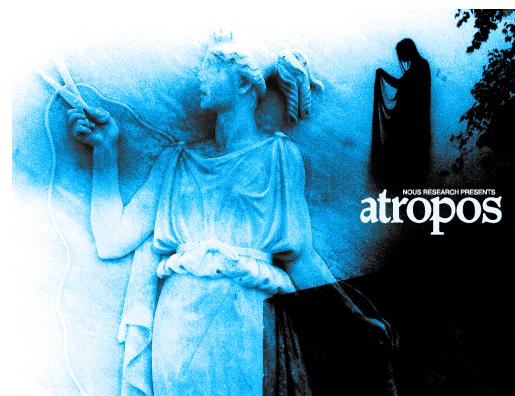
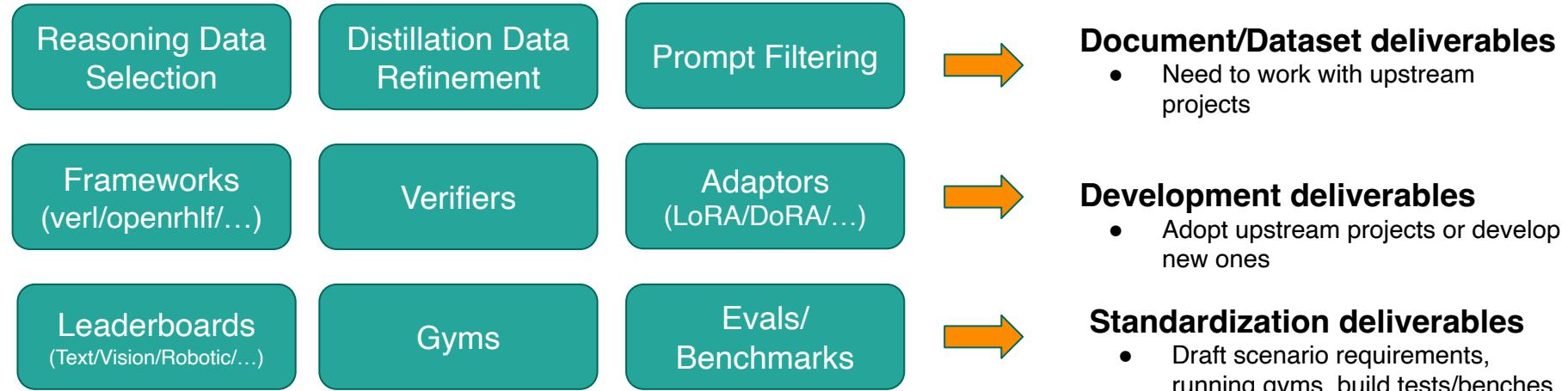
Reasoners

MODEL	ACCURACY
Claude 3.7 Sommet	68.18%
DeepSeek-R1	56.14%
Flash Thinking (01-21)	55.58%
c3-mini (01-31)	54.36%
R1-Distill (L70B)	49.07%
OpenThinker-32B	48.45%
LIMO	44.24%
DeepHermes-3-P (L8B)	41.53%
DeepScaleR-1.5B-P	34.91%

<https://gr.inc/about/>

# Proposal - Build an OSS RLGR community under LFAI&Data Foundation

GOSIM

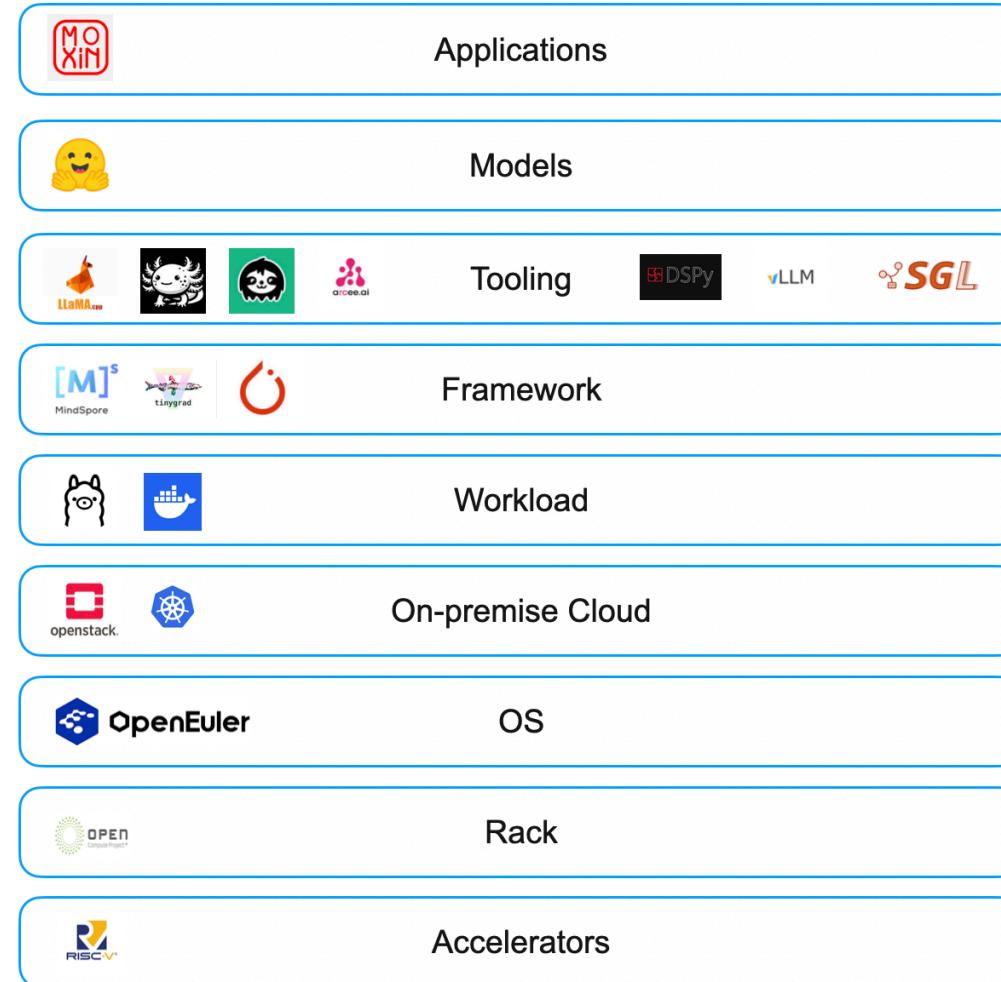


Atropos is a Language Model Reinforcement Learning Environments framework for collecting and evaluating LLM trajectories through diverse environments including:

Environment Type	Examples	Purpose
Dataset environments	GSM8K, MMLU	Evaluate and improve LLM performance on static data
Online environments	Crosswords, Hangman	Train LLMs through interactive game-based learning
RLAIF and RLHF	LLM Judge/Reward Models	Fine-tune LLMs using human feedback and alignment
Multi-Turn RL	deeprsearch, internal tool calling	Train LLMs on complex multi-step interactions

# Proposal - Build a more broader OSS Ecosystem for the era of open experience

GOSIM



There is more compute and software engineering talent outside of the labs than within them  
— @georgejrjrjr



Let's build an OSS RLGR ecosystem together

**GOSIM AI Paris 2025**

**GOSIM**

Contact

Zhipeng Huang Get verified  
@nopainkiller  
Open Source Effectively Accelerates !  
33.642931,-117.84131 [progverse.substack.com](https://progverse.substack.com) Born May 12, 1986  
Joined May 2009  
478 Following 1,065 Followers

<https://x.com/nopainkiller>

OpenRLGR

README.md  
Open Source Collective For Reinforcement Learning with Generalized Rewards  
Motivation  
The DSR1 Moment  
Since the dawn of open source (weights) LLMs, open collaboration has becomes a embarrassing pain point for the community. Although we have HuggingFace to share the model weights and datasets, collaboration was never a big thing for pre-training + RLHF era, with exception of mergekit and nanoGPT runs.  
DeepSeek R1's release kickstarted a new era of totally decoupled pre-training/post-training, where the latter has a signature of flywheel suited for OSS collaboration. Never before happened, there is actual need for open collaboration to build an OSS community for the collective good of RLGRs.

<https://github.com/OpenRLGR>



# THANK YOU

