Multimodal AI

Lecture 14.1 – Advanced Topics

Paul Liang
Assistant Professor
MIT Media Lab & MIT EECS

https://pliang279.github.io
ppliang@mit.edu
@pliang279

mit media lab

MIT EECS

Assignments for This Coming Week

Please fill out course evaluations and give us feedback!

HW5 due this Friday May 8.

For project:

- Make sure to meet with myself and TAs this week if you need advice.
- Presentations next Tuesday May 12.
- Final report due Tuesday May 19.

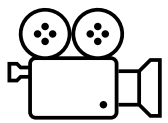
This Week: Recap and Advanced Topics

1. Recap of course material
2. Native multimodal models and mixture of experts
3. Advanced fusion techniques
4. Self-evolving multimodal agents
5. Extending human senses – touch, smell, and taste
6. Human-AI interaction and safety

What is a Modality?

Modality

Modality refers to the way in which something expressed or perceived.



from a sensor

Raw Modalities
(closest from sensor)

Abstract Modalities
(farthest from sensor)

Examples:

Raw Modalities		Abstract Modalities
Speech signal	Language	Sentiment intensity
Image	Detected objects	Object categories

multisensory intelligence

What is Multimodal?

A dictionary definition…

Multimodal: with multiple modalities

A research-oriented definition…

***Multimodal* is the science of**

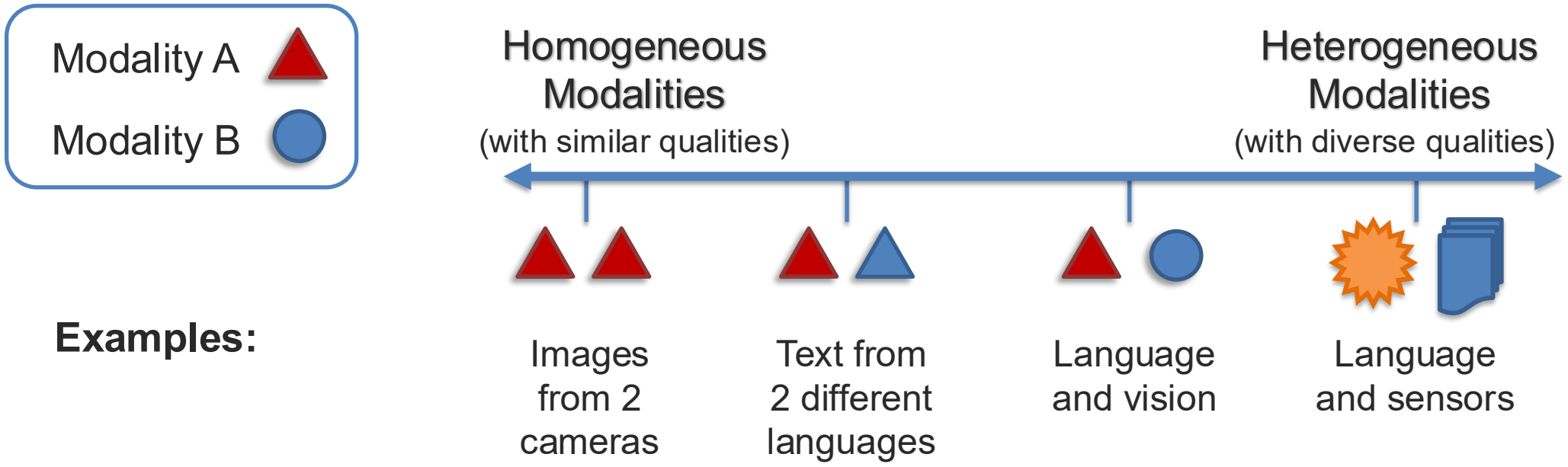
heterogeneous and interconnected data

Connected + Interacting

multisensory intelligence

Heterogeneous Modalities

Information in different modalities shows diverse qualities, structures, & representations.

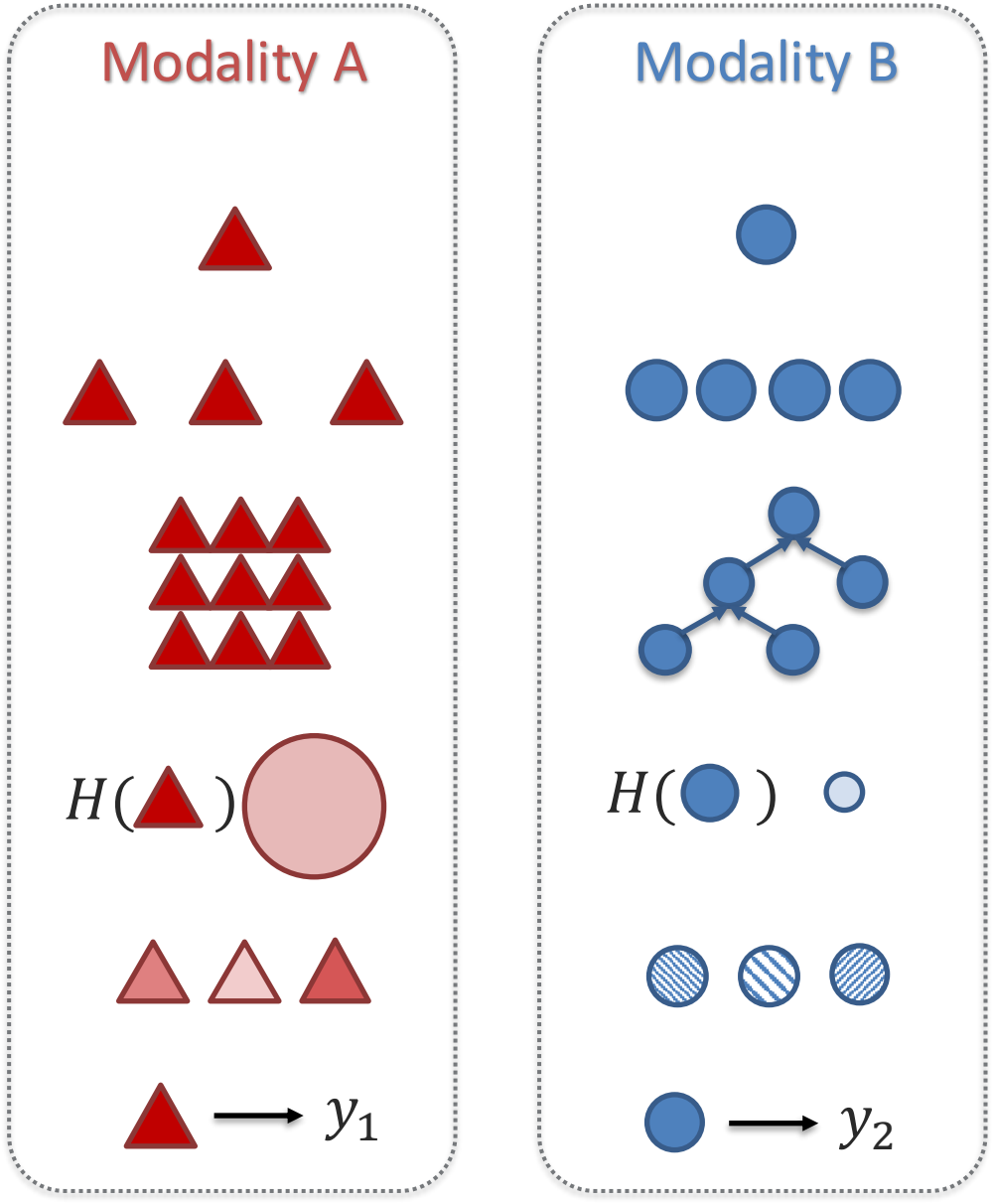


Abstract modalities are more likely to be homogeneous

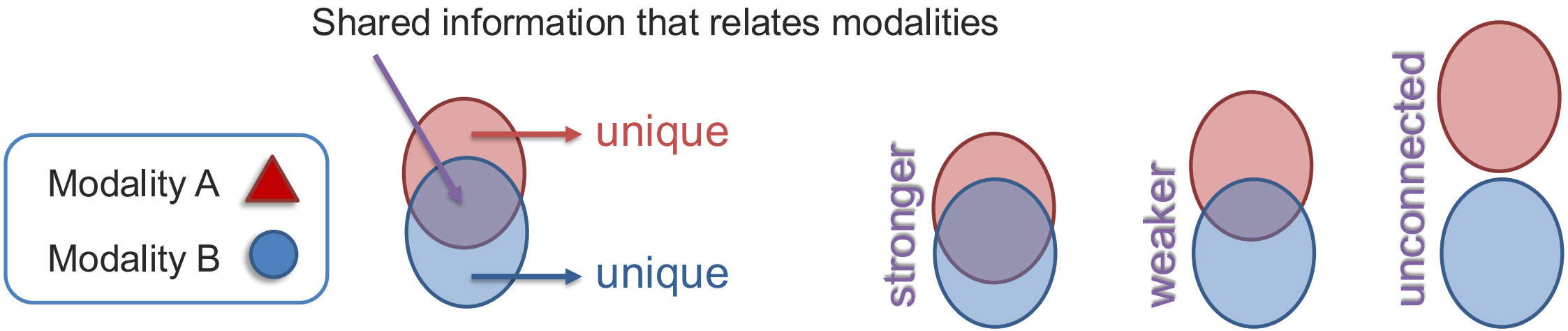
multisensory intelligence

Modality Profile

1. **Element representations:**
   Discrete, continuous, granularity
2. **Element distributions:**
   Density, frequency
3. **Structure:**
   Temporal, spatial, latent, explicit
4. **Information:**
   Abstraction, entropy
5. **Noise:**
   Uncertainty, noise, missing data
6. **Relevance:**
   Task, context dependence



Connected Modalities

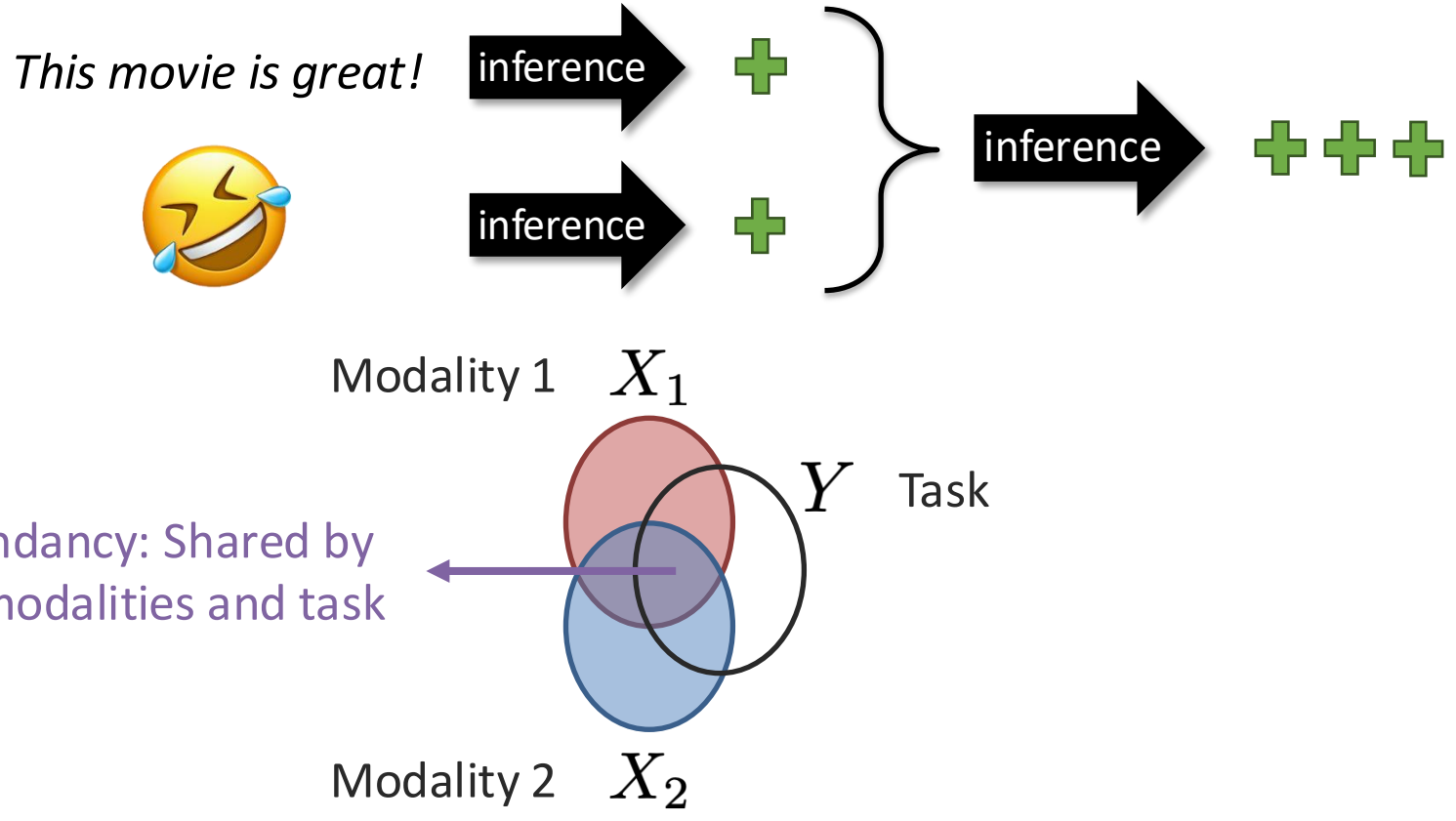


A teacup on the right of a laptop in a clean room.

multisensory intelligence

Interacting Modalities

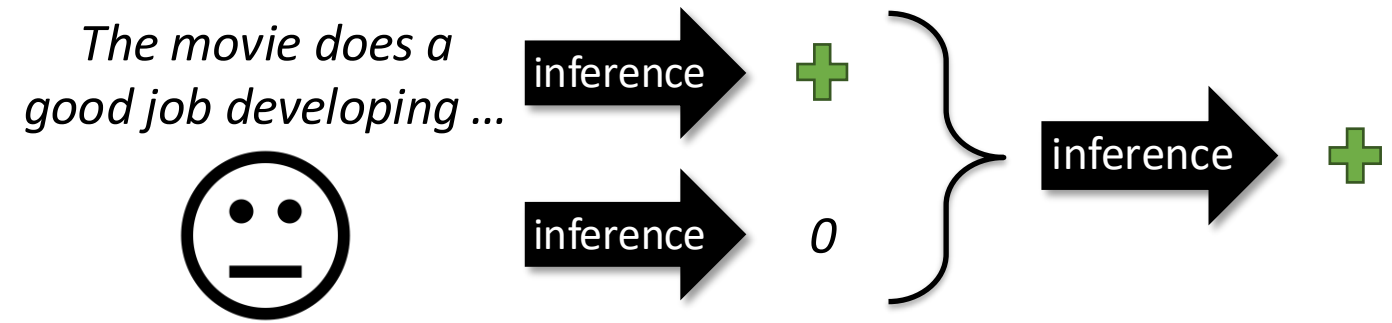
Interactions: How modalities *combine* to provide information for a task.

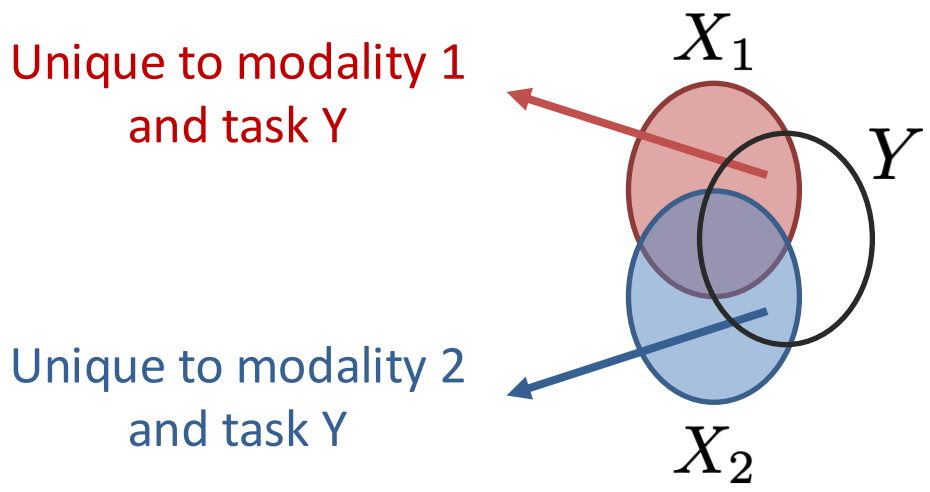


multisensory intelligence

Interacting Modalities

Interactions: How modalities *combine* to provide information for a task.

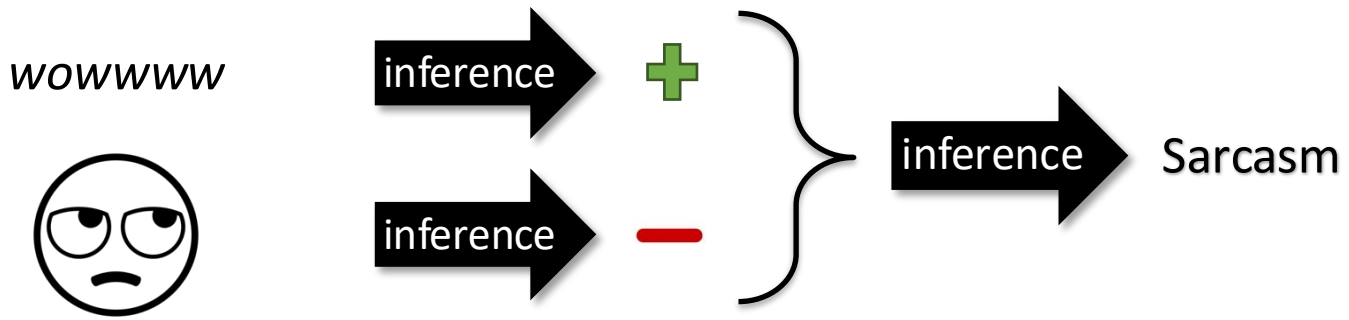


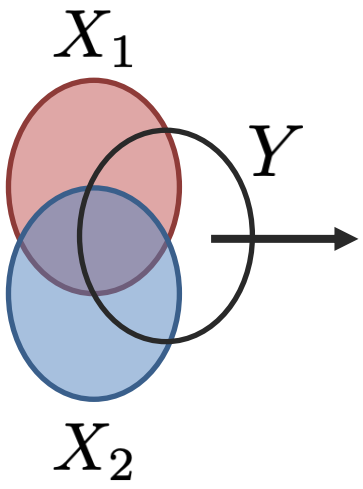


multisensory intelligence

Interacting Modalities

Interactions: How modalities *combine* to provide information for a task.





Synergy: Emerging information from multimodal interaction

multisensory intelligence

What is Multimodal? → *Why is it hard?* → *What is next?*

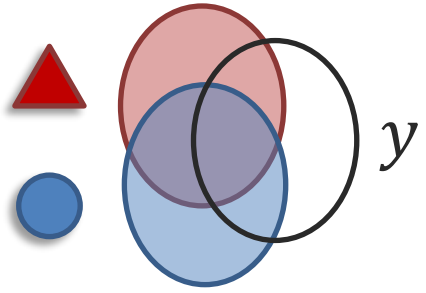
Heterogeneous

+

Connected

+

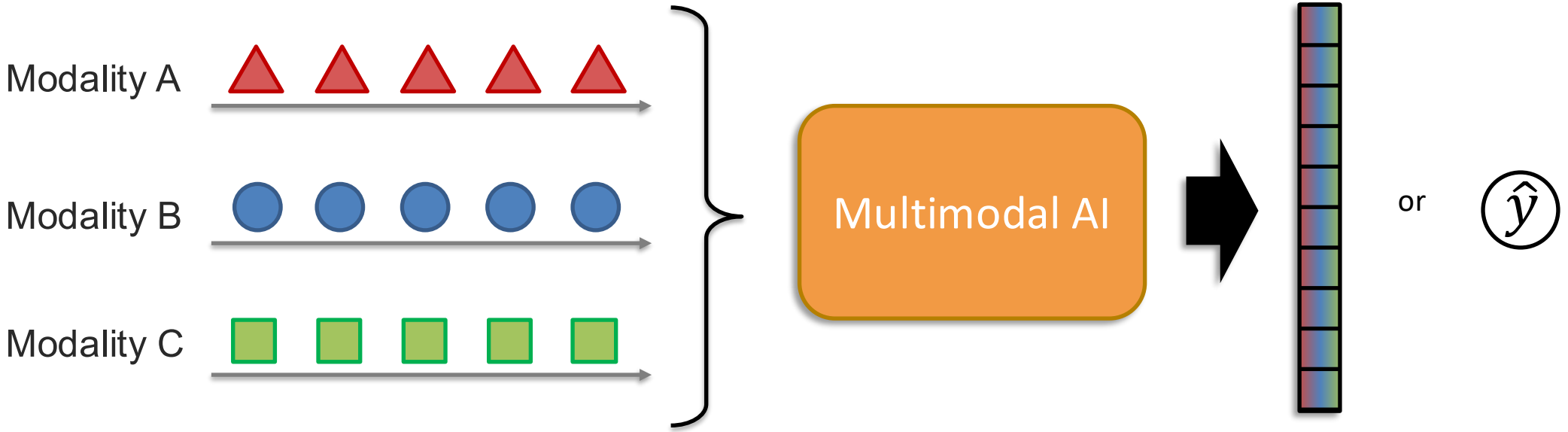
Interacting



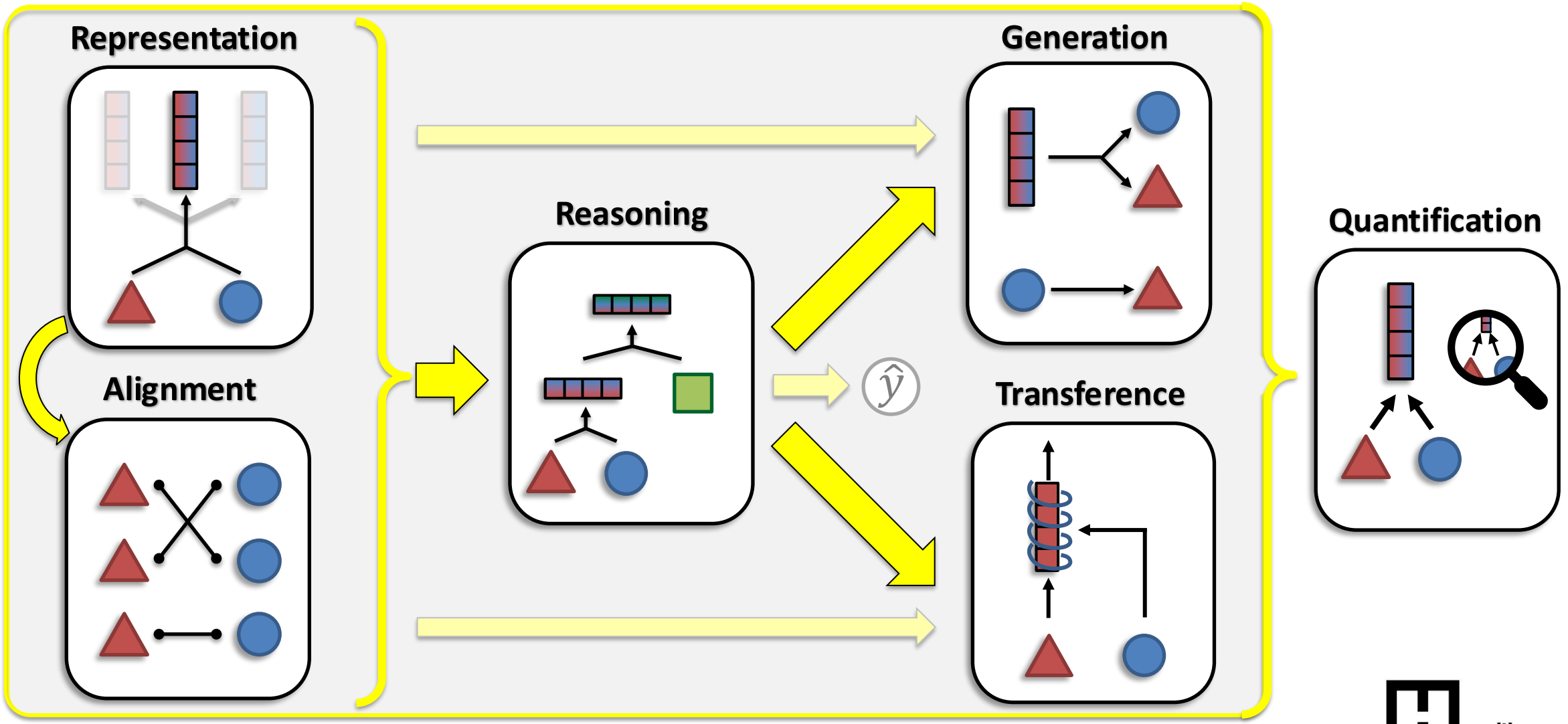
Multimodal is the scientific study of heterogeneous and interconnected data ☺

multisensory intelligence

Multimodal AI Challenges



Summary of Core Multimodal Challenges

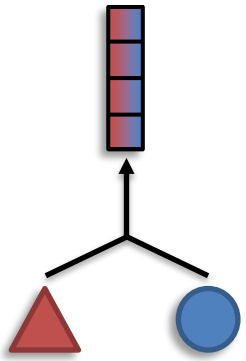


Challenge 1: Representation

Definition: Learning representations that reflect cross-modal interactions between individual elements, across different modalities.

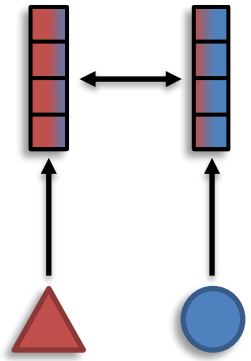
Sub-challenges:

Fusion



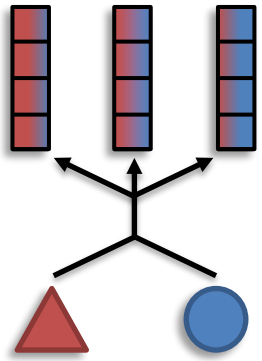
modalities > # representations

Coordination



modalities = # representations

Fission



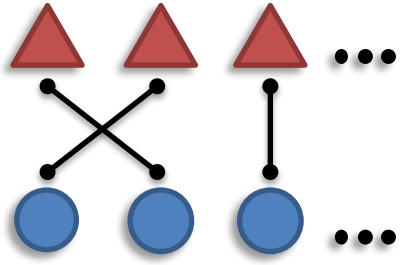
modalities < # representations

Challenge 2: Alignment

Definition: Identifying and modeling cross-modal connections between all elements of multiple modalities, building from the data structure.

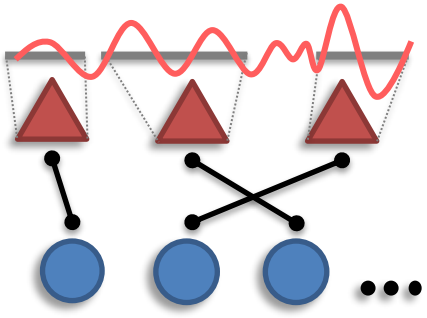
Sub-challenges:

Discrete connections



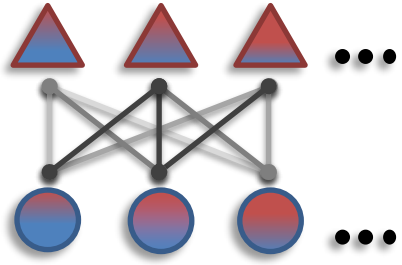
Explicit alignment
(e.g., grounding)

Continuous alignment



Granularity of individual elements

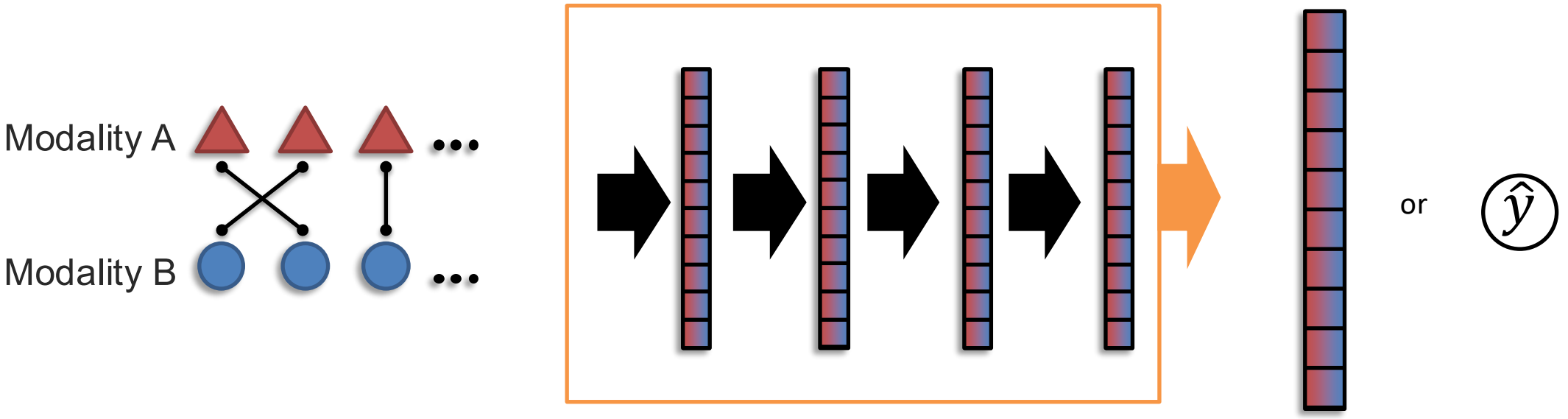
Contextualized representation



Implicit alignment + representation

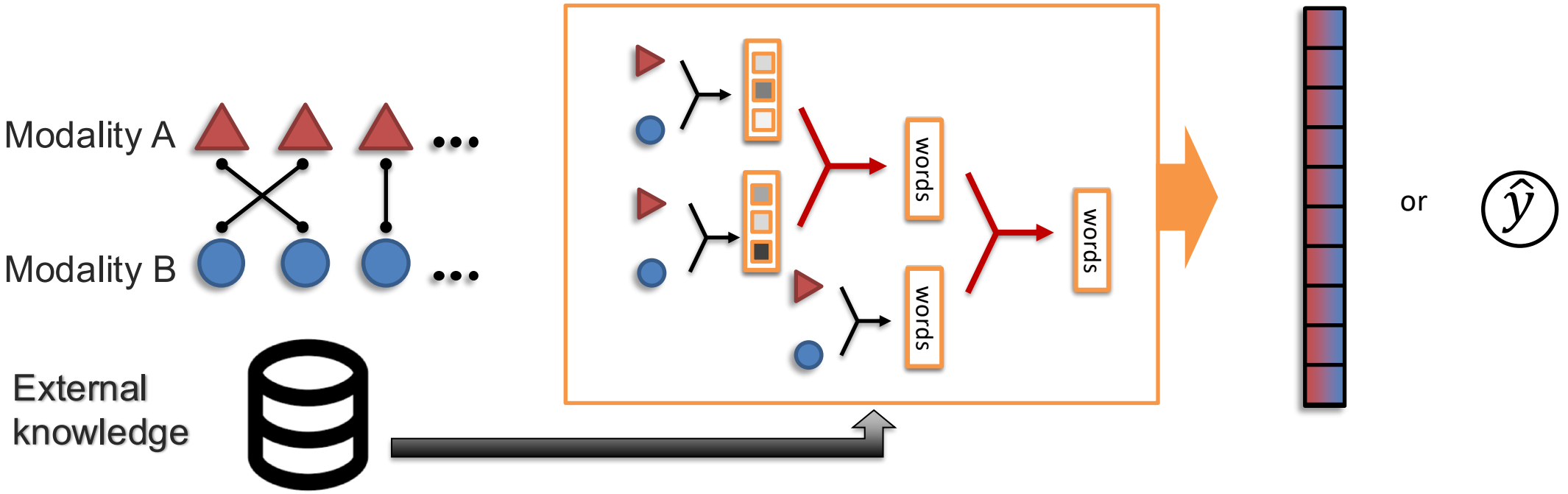
Challenge 3: Reasoning

Definition: Combining knowledge, usually through multiple inferential steps, exploiting multimodal alignment and problem structure.



Challenge 3: Reasoning

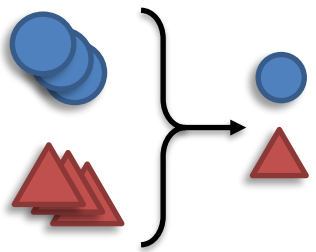
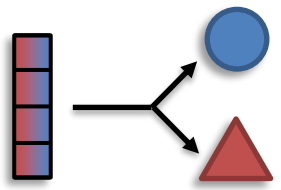
Definition: Combining knowledge, usually through multiple inferential steps, exploiting multimodal alignment and problem structure.



Challenge 4: Generation

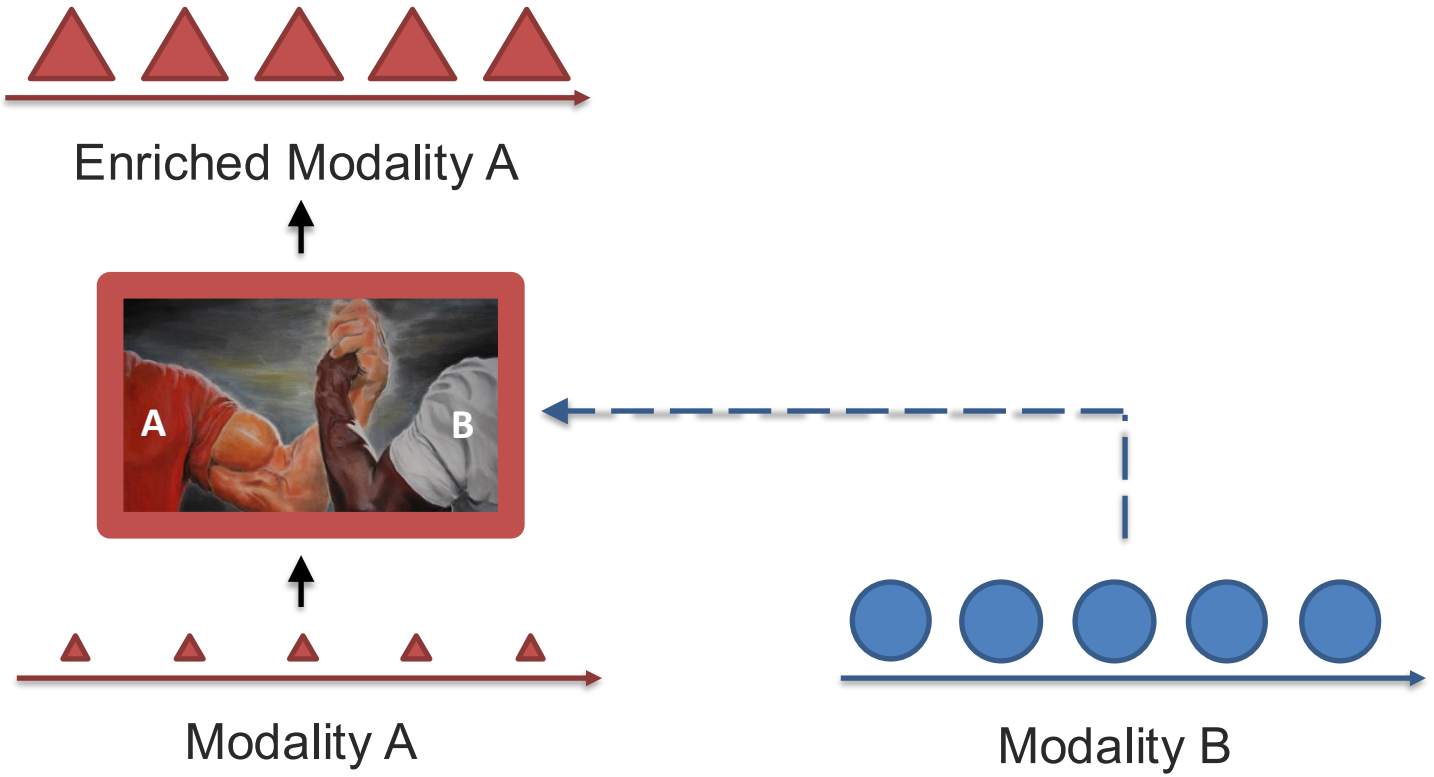
Definition: Learning a generative process to produce raw modalities that reflects cross-modal interactions, structure, and coherence.

Sub-challenges:

	Summarization	Translation	Creation
			
Information: (content)	Reduction	Maintenance	Expansion
	□ > □	□ = □	□ < □

Challenge 5: Transference

Definition: Transfer knowledge between modalities, usually to help the target modality which may be noisy or with limited resources.

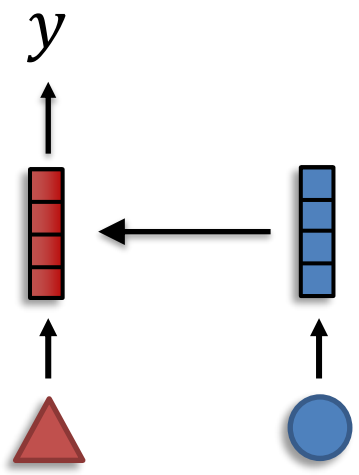


Challenge 5: Transference

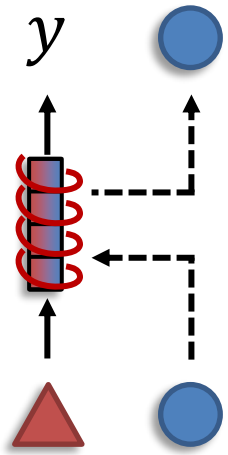
Definition: Transfer knowledge between modalities, usually to help the target modality which may be noisy or with limited resources.

Sub-challenges:

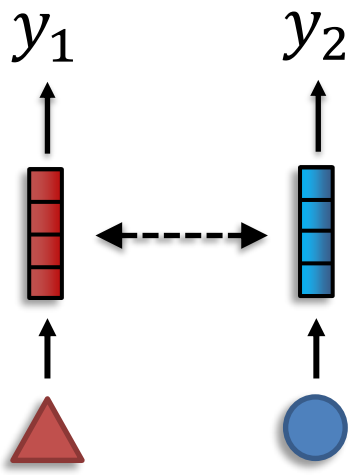
Transfer



Co-learning



Model Induction

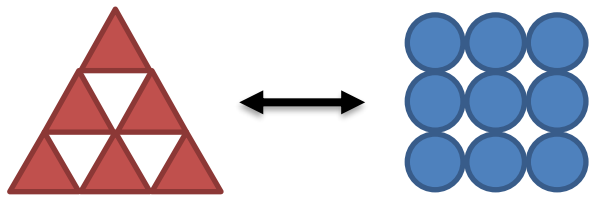


Challenge 6: Quantification

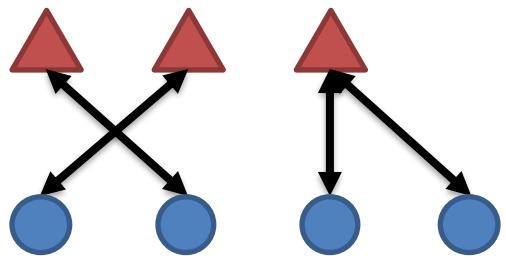
Definition: Empirical and theoretical study to better understand heterogeneity, cross-modal interactions, and the multimodal learning process.

Sub-challenges:

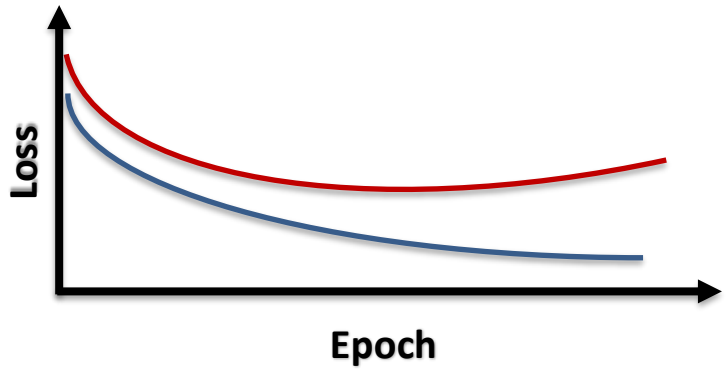
Heterogeneity



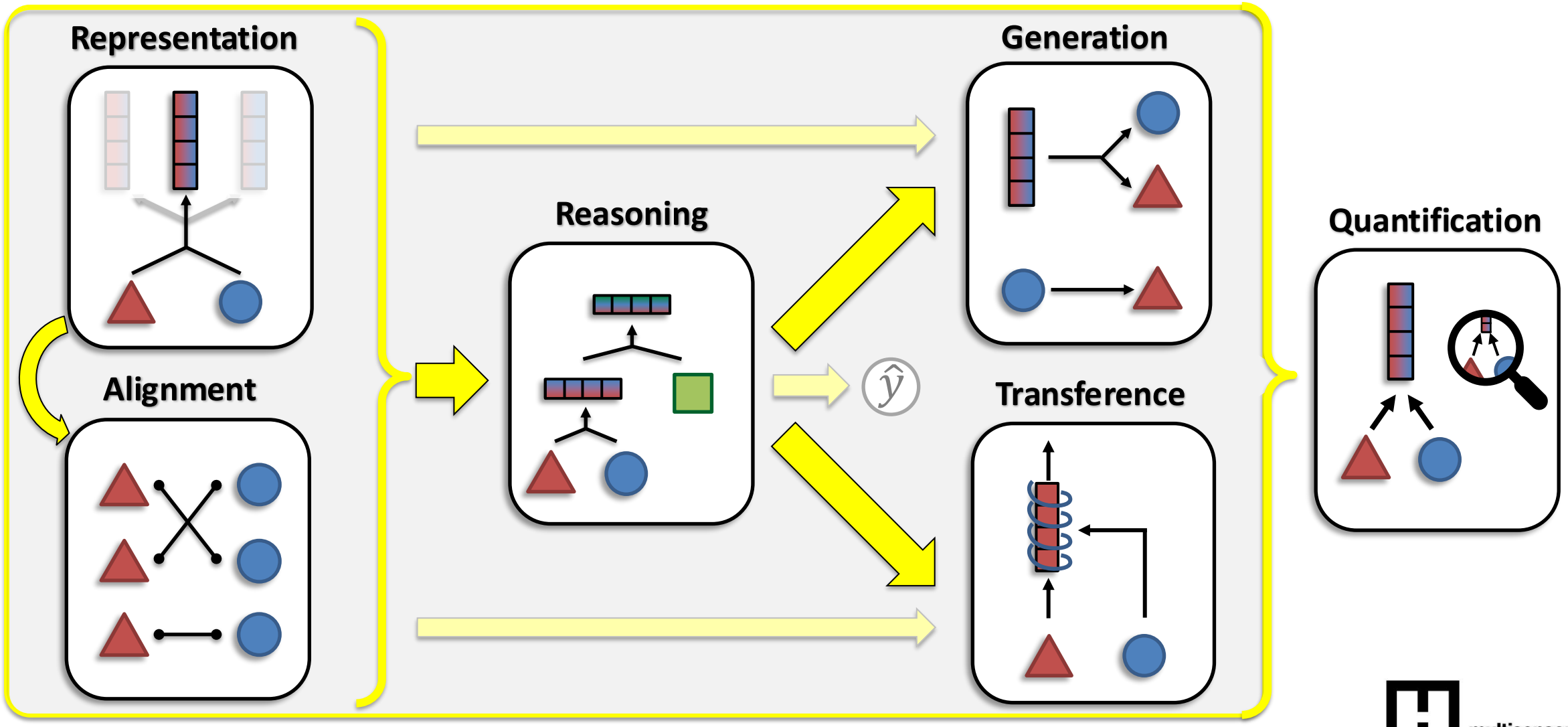
Interactions



Learning

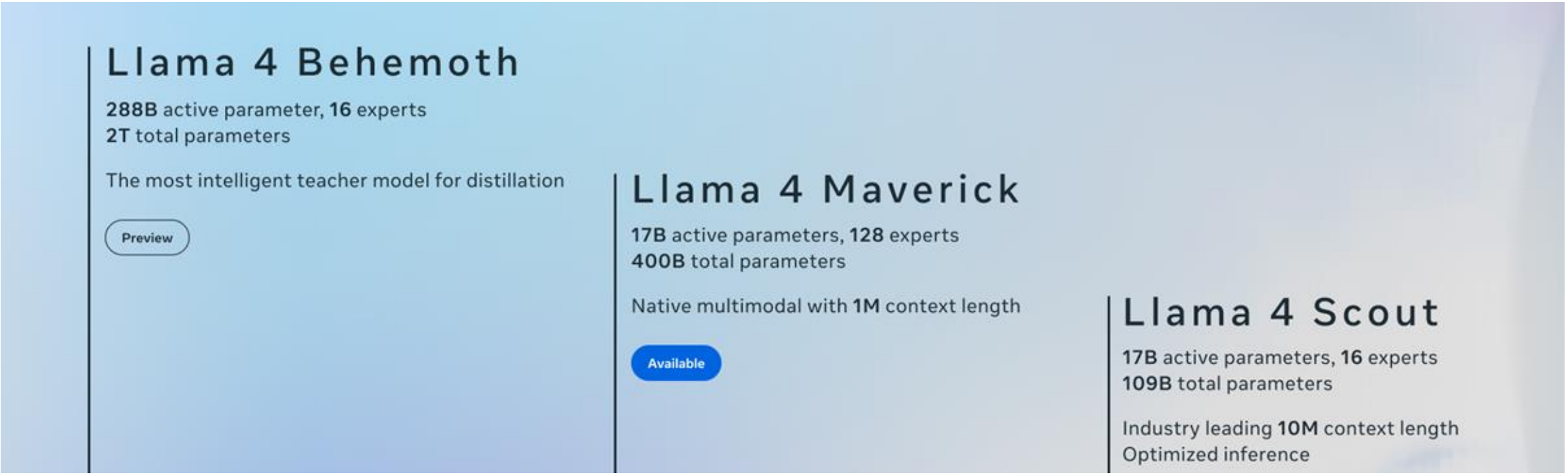


Summary of Core Multimodal Challenges



Native Multimodal Models

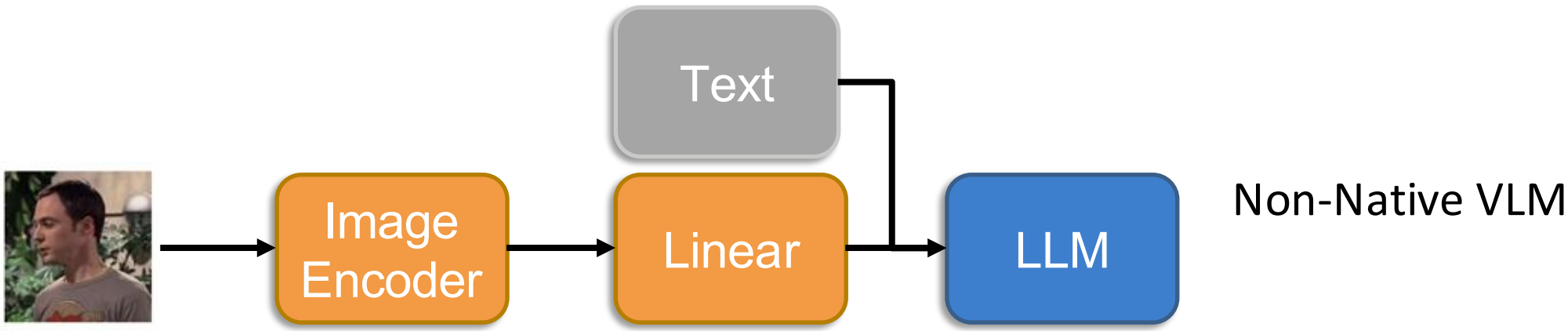
- *Native Multimodal Modals*: LLMs Trained from scratch with multimodal input (instead of finetuning a trained unimodal LLM)
- Largest public model now: 109B - 2T parameters



[The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation, https://ai.meta.com/blog/llama-4-multimodal-intelligence/]

Native Multimodal Models

- *Background*
  - **Non-native VLMs***:* Image encoder paired with frozen trained LLM. The image encoder can either be frozen or trained. Most VLMs now use this structure.

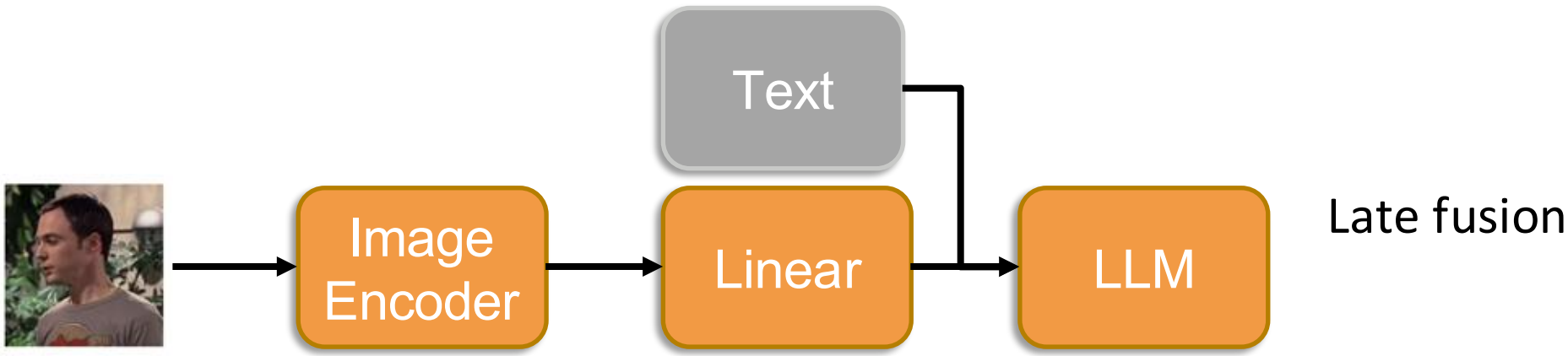


Most current VLMs use this architecture.

[Scaling Laws for Native Multimodal Models, https://arxiv.org/abs/2504.07951]

Native Multimodal Models

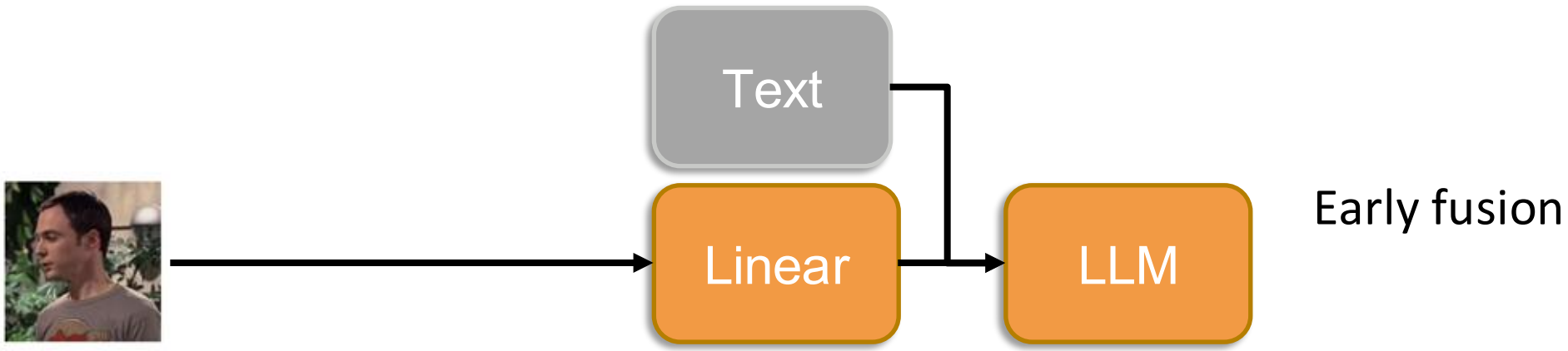
- *Background*
  - Non-native VLMs*:* Image encoder paired with frozen trained LLM. The image encoder can either be frozen or trained. Most VLMs now use this structure.
  - Native Multimodal Modals: LLMs Trained from scratch with multimodal input
    - **Late fusion**: Image patches -> Image Encoder -> Linear -> LLM.
    - Early fusion: Image patches -> Linear -> LLM (No image encoder!)



[Scaling Laws for Native Multimodal Models, https://arxiv.org/abs/2504.07951]

Native Multimodal Models

- *Background*
  - Non-native VLMs*:* Image encoder paired with frozen trained LLM. The image encoder can either be frozen or trained. Most VLMs now use this structure.
  - Native Multimodal Modals: LLMs Trained from scratch with multimodal input
    - Late fusion: Image patches -> Image Encoder -> Linear -> LLM.
    - **Early fusion**: Image patches -> Linear -> LLM (No image encoder!)



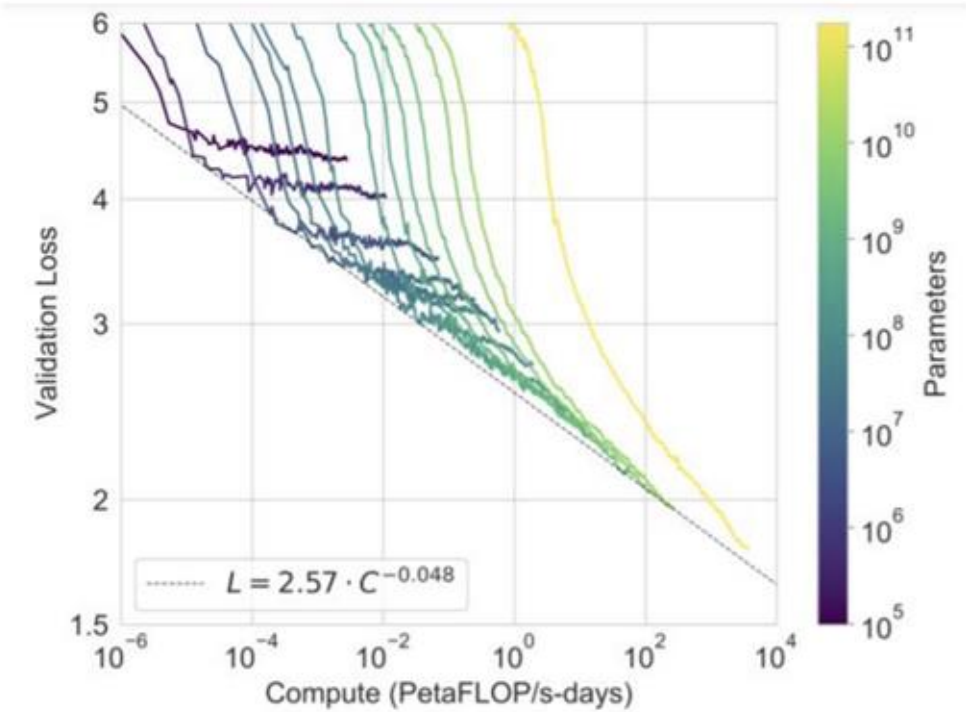
[Scaling Laws for Native Multimodal Models, https://arxiv.org/abs/2504.07951]

multisensory intelligence

Scaling Laws

- Bigger model allows models to reach a better performance given sufficient compute
- Over training models getting popular nowadays

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion



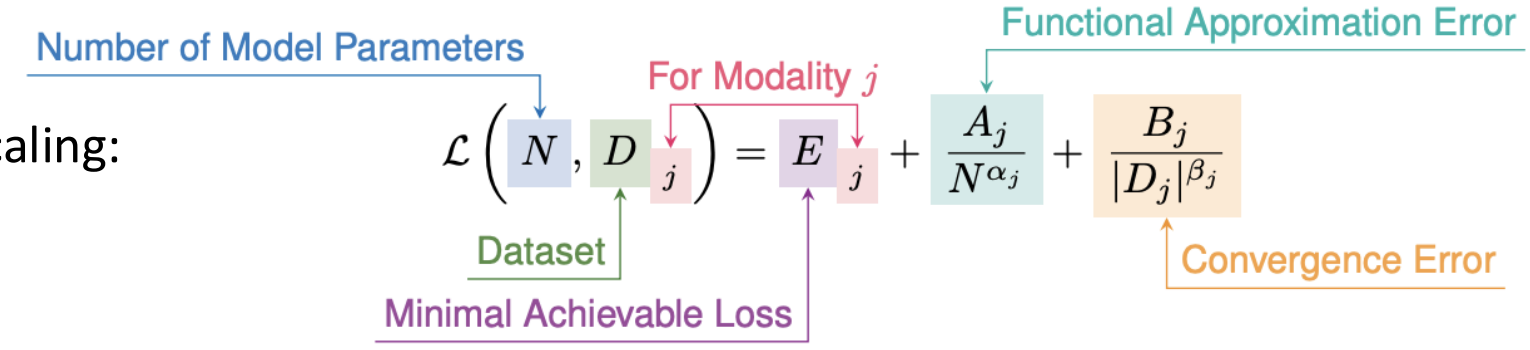
	Training Data	Params	Context length	GQA	Token count	Knowledge cutoff
Llama 3	A new mix of publicly available online data.	8B	8k	Yes	15T+	March, 2023
		70B	8k	Yes		December, 2023

Credit: https://web.stanford.edu/class/cs224n/slides_w25/cs224n-2025-lecture09-pretraining.pdf, https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct

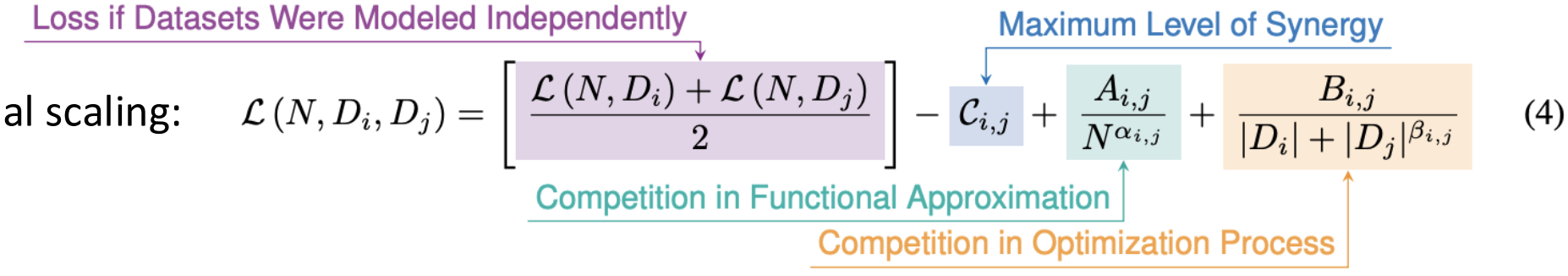
Scaling Laws for Native Multimodal Models

- Studies how multimodal models scale (with model parameters and dataset size) as compared to their unimodal counterparts.

Unimodal scaling:



Multimodal scaling:

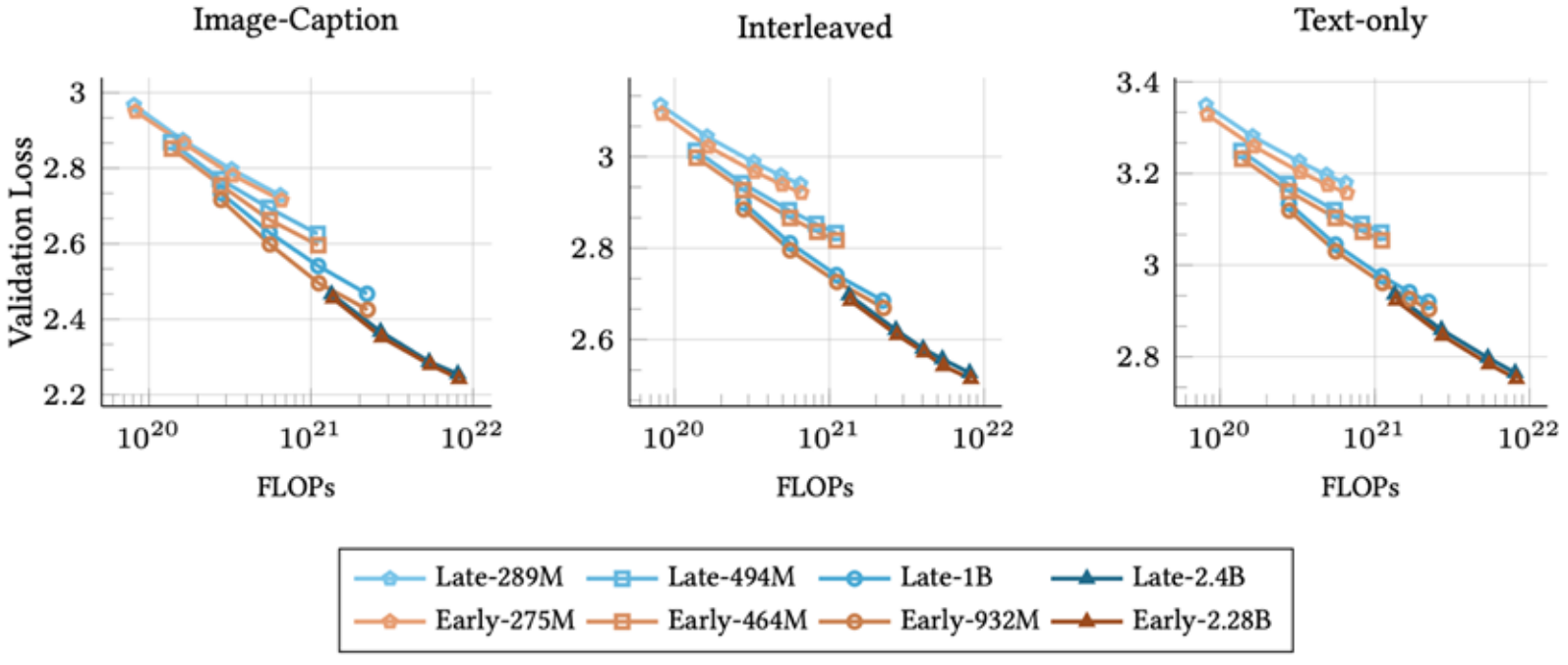


[Scaling Laws for Generative Mixed-Modal Language Models, https://arxiv.org/abs/2301.03728]

multisensory intelligence

Scaling Laws for Native Multimodal Models

- Early fusion models hold small advantage on small scales.
- On larger scales, **both architectures perform similarly.** (We don't actually need image encoders!)
- **NMMs scale similarly to unimodal LLMs**, with slightly varying scaling exponents depending on the target data type and training mixture

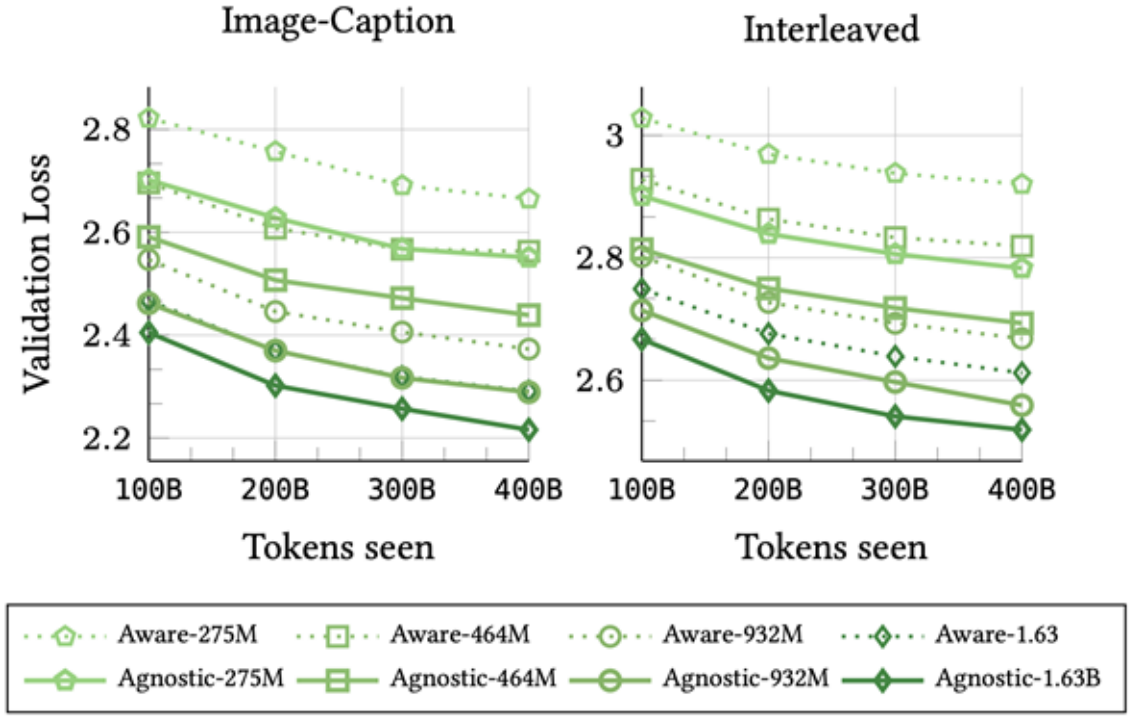


[Scaling Laws for Native Multimodal Models, https://arxiv.org/abs/2504.07951]

multisensory intelligence

Scaling Laws for Native Multimodal Models

- Sparse structure like **MOE significantly benefits NMMs** at the same inference cost
- In an MOE structure, Modality-aware design (having separate image/text experts) performs **worse** than modality-agnostic design (unified experts for both image/text tokens)

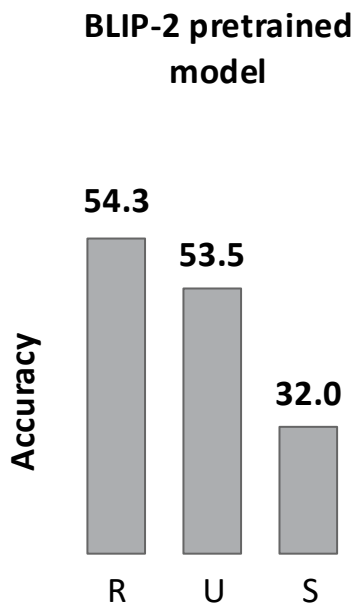


[Scaling Laws for Native Multimodal Models, https://arxiv.org/abs/2504.07951

multisensory intelligence

One model for everything?

Video sarcasm detection



Y: Sarcasm

X_ℓ: Spoken language

It's just a privilege to watch your mind at work.

X_{av}: Audio + visual

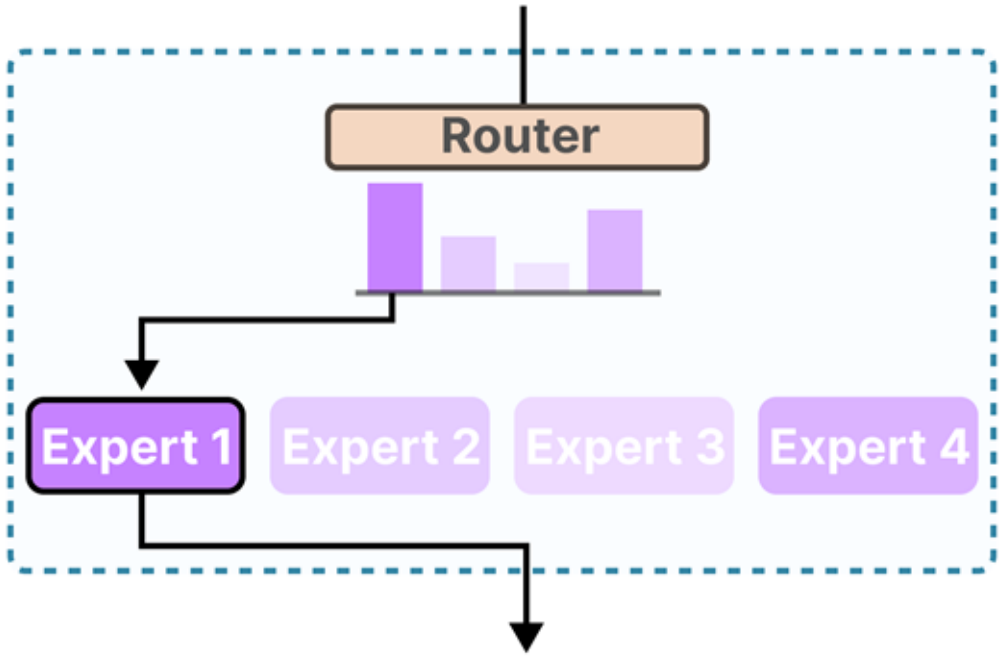
Neutral tone + straight face

[Yu et al., MMOE: Enhancing Multimodal Language Models with Mixture of Multimodal Interaction Experts. EMNLP 2024]

multisensory intelligence

Efficient Training: Mixture of Experts

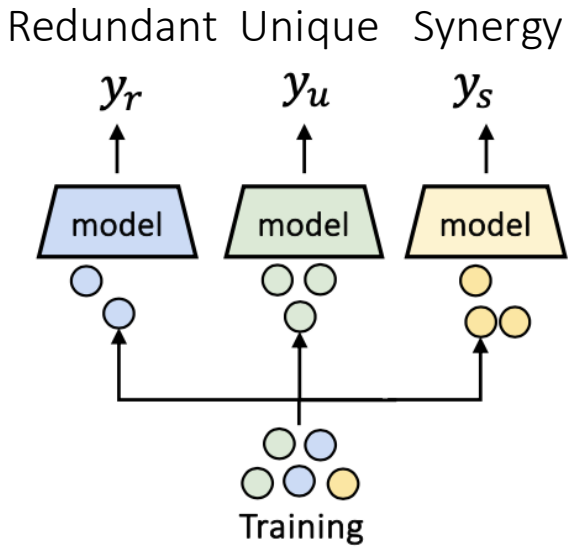
- Train multiple parallel networks (experts) simultaneously
- During each forward pass, only activate k experts
- Saves compute & GPU memory
- Deepseek R1: 671B, only 37B activated, performance on par with OpenAI o1-mini



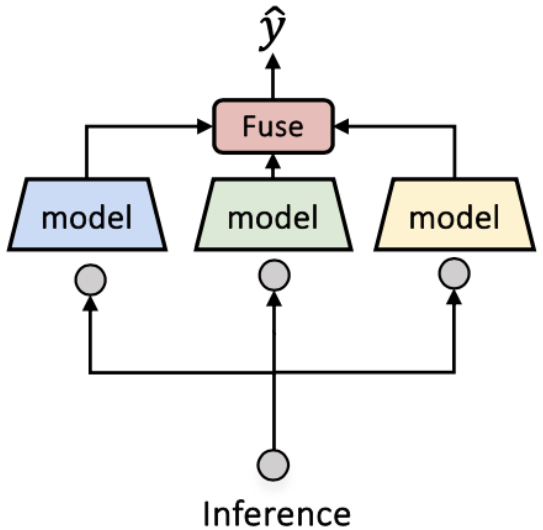
A Visual Guide to Mixture of Experts (MoE), https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

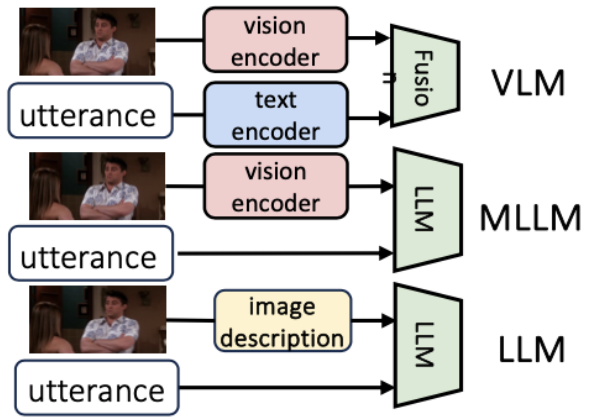
Mixture of Multimodal Interaction Experts

One model for everything -> specialized models for each interaction



Datapoint-level quantification of interactions





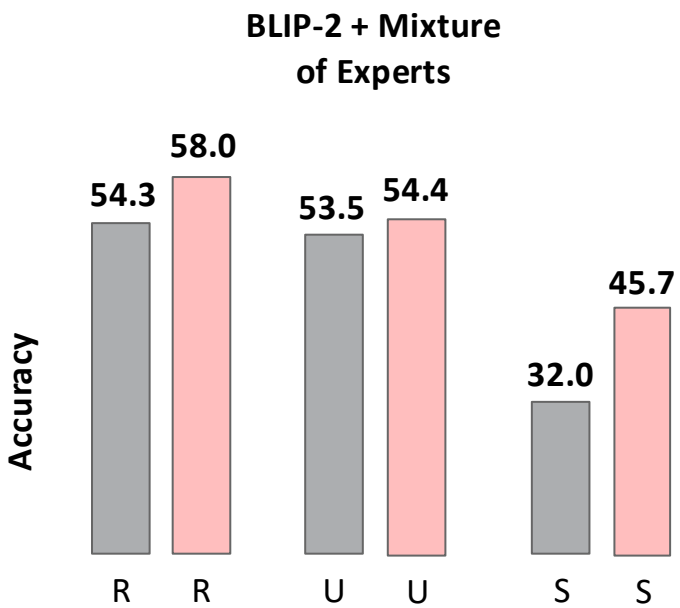
[Yu et al., MMOE: Enhancing Multimodal Language Models with Mixture of Multimodal Interaction Experts. EMNLP 2024]

multisensory intelligence

Mixture of Multimodal Interaction Experts

One model for everything -> specialized models for each interaction

Video sarcasm detection



The car is as fast as a cheetah.

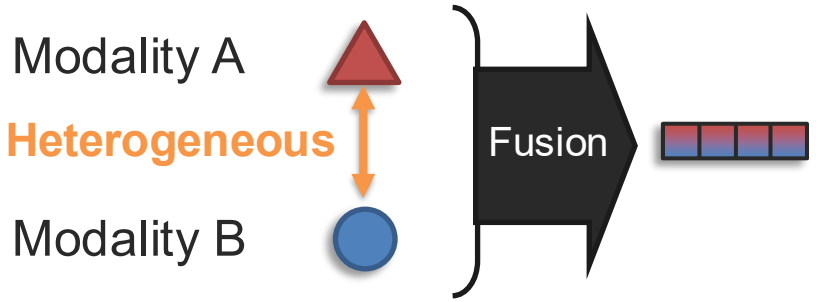
[Yosef et al., EMNLP 23]

[Hessel et al., ACL 23]

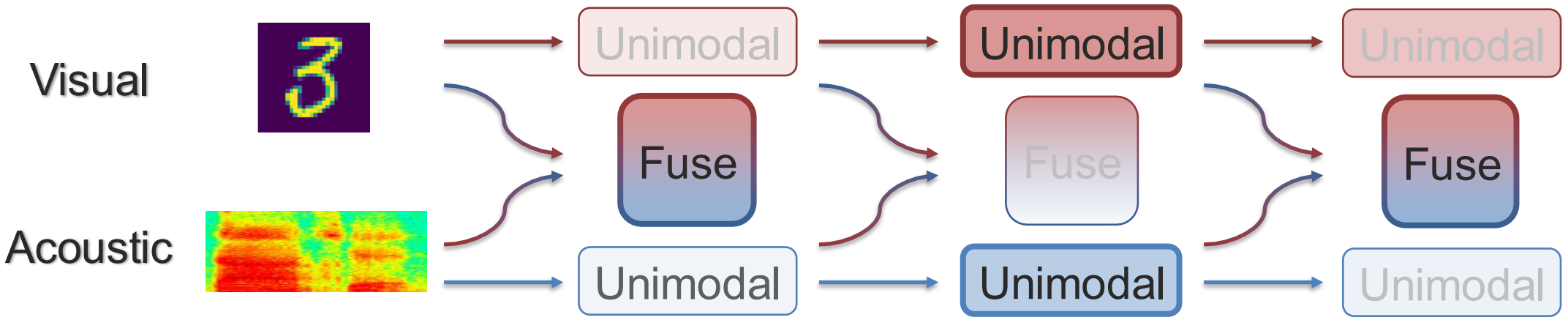
[Yu et al., MMOE: Enhancing Multimodal Language Models with Mixture of Multimodal Interaction Experts. EMNLP 2024]

multisensory intelligence

Dynamic Early Fusion



Idea: Deciding when to fuse in early fusion

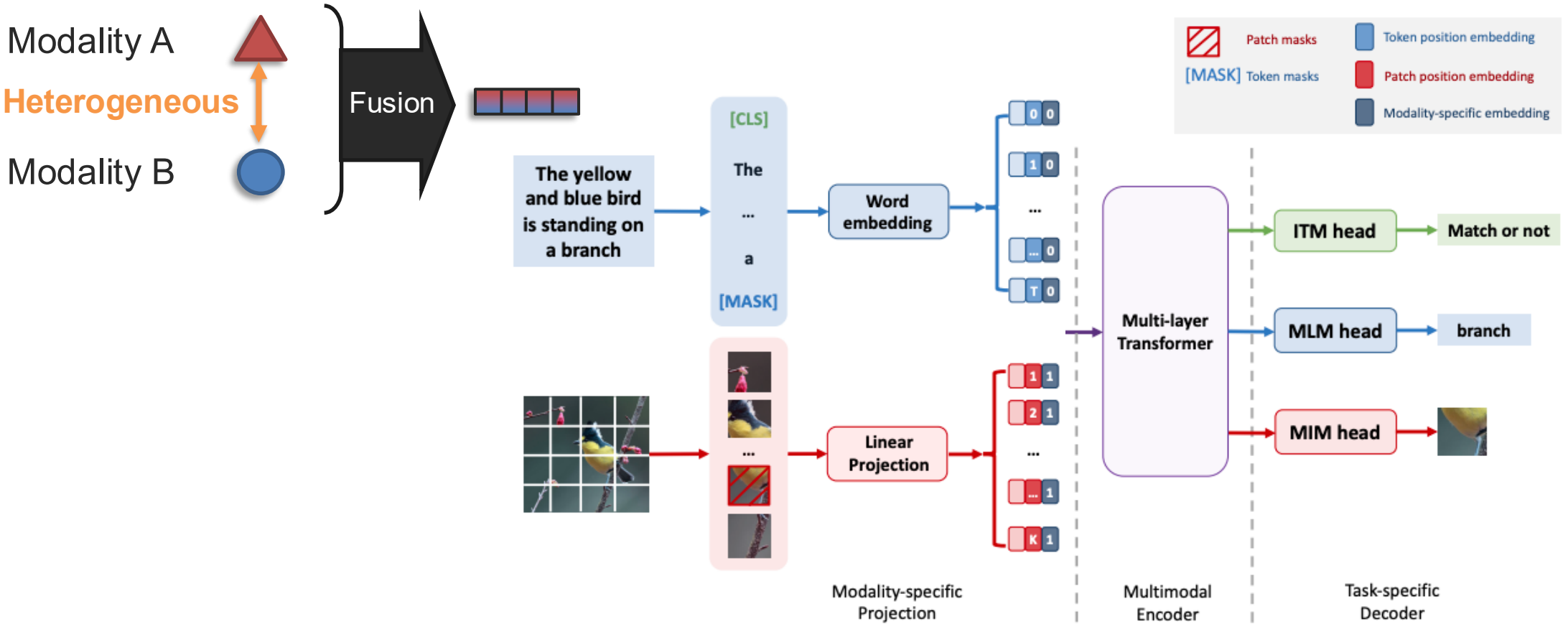


[Xue and Marculescu, Dynamic Multimodal Fusion, arxiv 2022]
[Xu et al., MUFASA: Multimodal Fusion Architecture Search for Electronic Health Records. AAAI 2021]
[Liu et al., DARTS: Differentiable Architecture Search. ICLR 2019]

multisensory intelligence

Fusion with Heterogeneous Modalities

Example: From feature fusion to early fusion

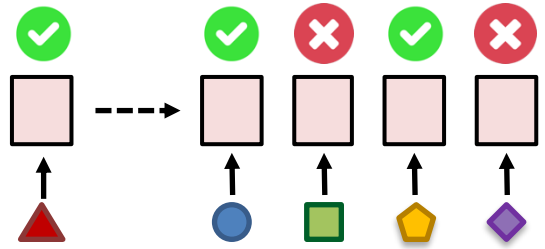


[Liang et al., High-modality Multimodal Transformer. TMLR 2022]
[Gui et al., Training Vision-Language Transformers from Captions. arxiv 2022]

Fusion with Heterogeneous Modalities

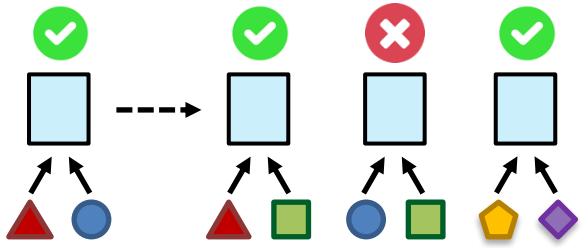
Information transfer, transfer learning perspective

1a. Estimate modality heterogeneity via transfer

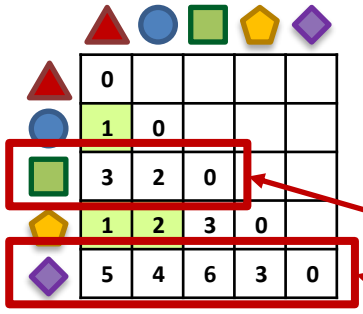


(Implicitly captures heterogeneity)

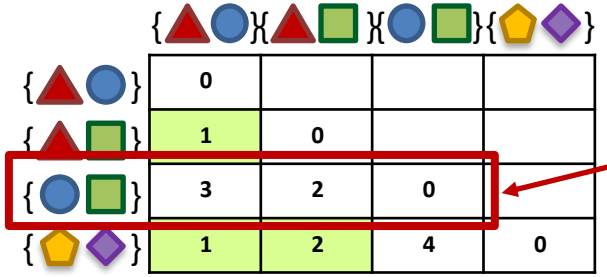
1b. Estimate interaction heterogeneity via transfer



2a. Compute modality heterogeneity matrix



2b. Compute interaction heterogeneity matrix



3. Determine parameter clustering

$$\mathbb{U}_1 = \{U_1, U_2, U_4\}$$
$$\mathbb{U}_2 = \{U_3\}$$
$$\mathbb{U}_3 = \{U_5\}$$

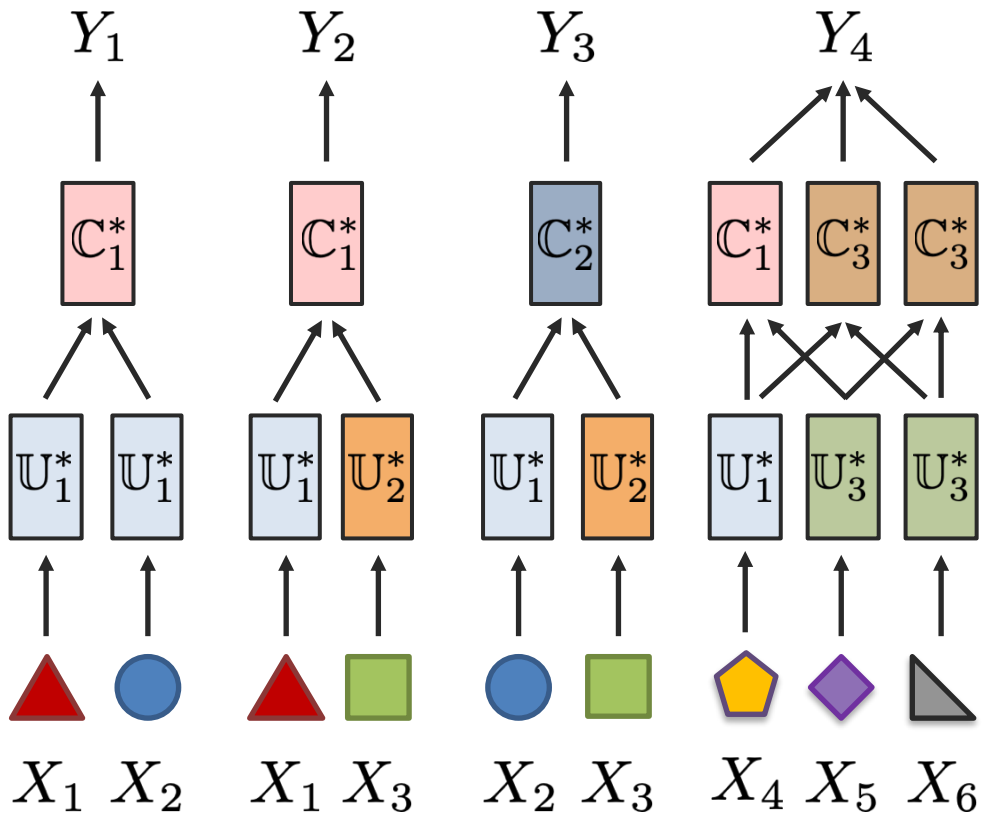
$$\mathbb{C}_1 = \{C_{12}, C_{13}, C_{45}\}$$
$$\mathbb{C}_2 = \{C_{23}\}$$

[Zamir et al., Taskonomy: Disentangling Task Transfer Learning. CVPR 2018]
[Liang et al., HighMMT: Quantifying Modality & Interaction Heterogeneity for High-Modality Learning. TMLR 2022]

multisensory intelligence

Fusion with Heterogeneous Modalities

Information transfer, transfer learning perspective

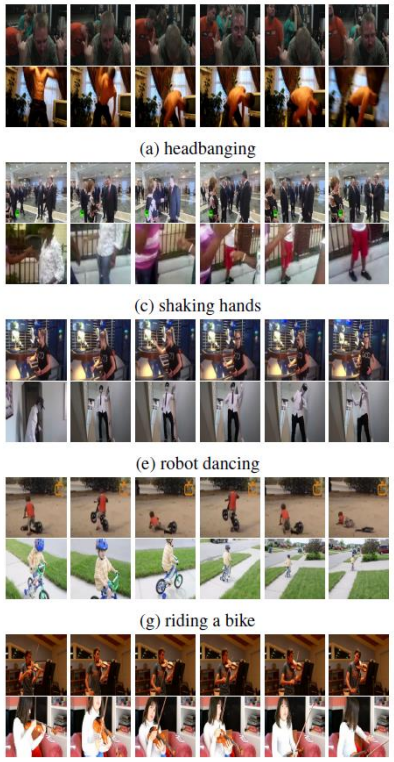


[Zamir et al., Taskonomy: Disentangling Task Transfer Learning. CVPR 2018]
[Liang et al., HighMMT: Quantifying Modality & Interaction Heterogeneity for High-Modality Learning. TMLR 2022]

multisensory intelligence

Nonlinear Fusion

Kinetics dataset



Adding more modalities should always help?

Modalities: **RGB** (video clips)
A (Audio features)
OF (optical flow - motion)

Dataset	Multi-modal	V@1	Best Uni	V@1	Drop
Kinetics	A + RGB	71.4	RGB	**72.6**	-1.2
	RGB + OF	71.3	RGB	**72.6**	-1.3
	A + OF	58.3	OF	**62.1**	-3.8
	A + RGB + OF	70.0	RGB	**72.6**	-2.6

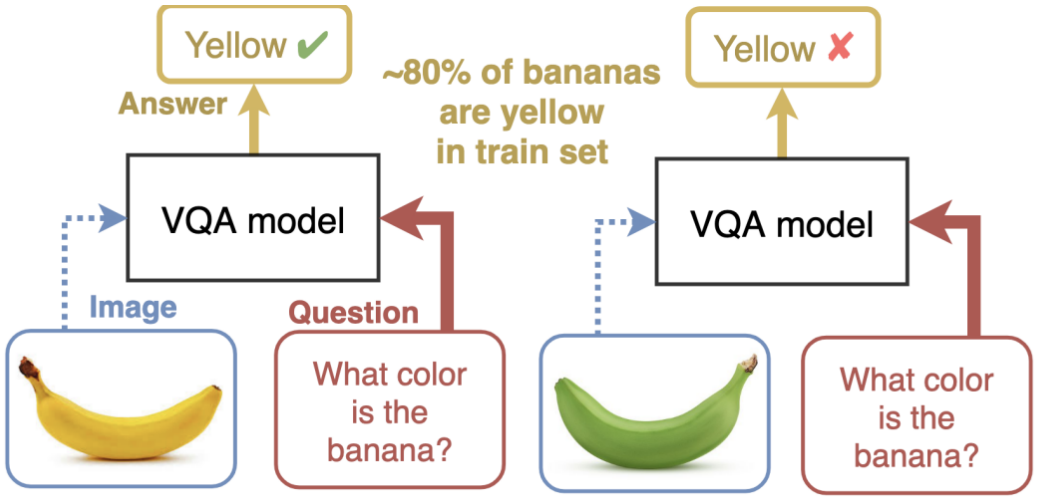
But sometimes multimodal doesn't help! **Why?**

[Wang et al., What Makes Training Multi-modal Classification Networks Hard? CVPR 2020]
[Wu et al., Characterizing and Overcoming the Greedy Nature of Learning in Multi-modal Deep Neural Networks. ICML 2022]

multisensory intelligence

Unimodal Biases

Finding: VQA models answer the question without looking at the image

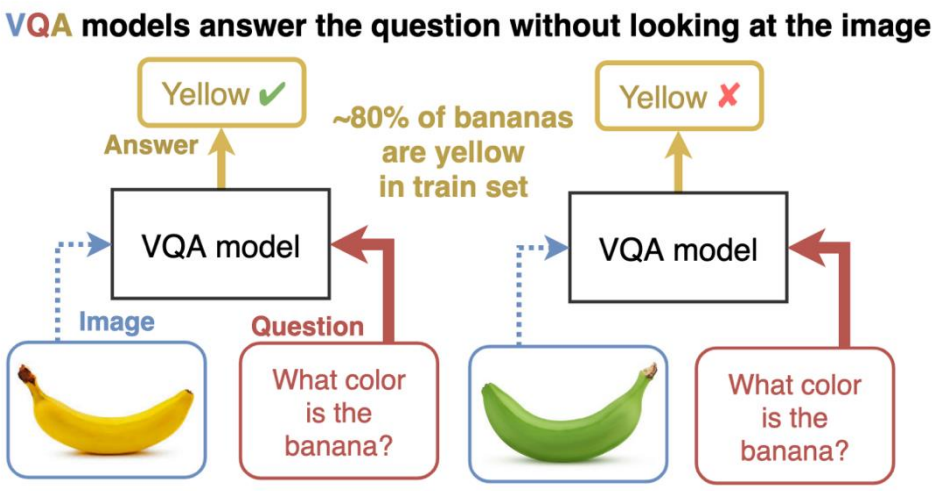


Finding: Image captioning models capture spurious correlations between gender and generated actions.

[Goyal et al., Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. CVPR 2017]
[Hendricks et al., Women also Snowboard: Overcoming Bias in Captioning Models. ECCV 2018]

multisensory intelligence

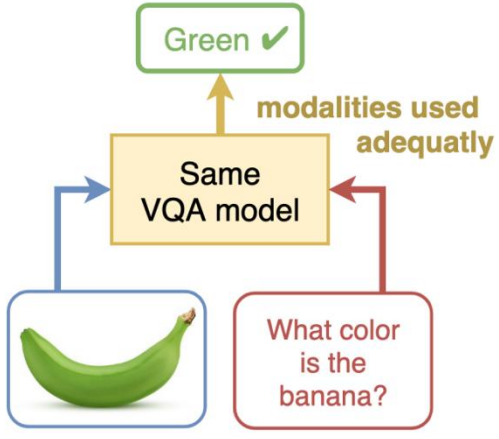
Unimodal Biases



Balancing modalities

Balancing training

Not the case when trained with RUBi



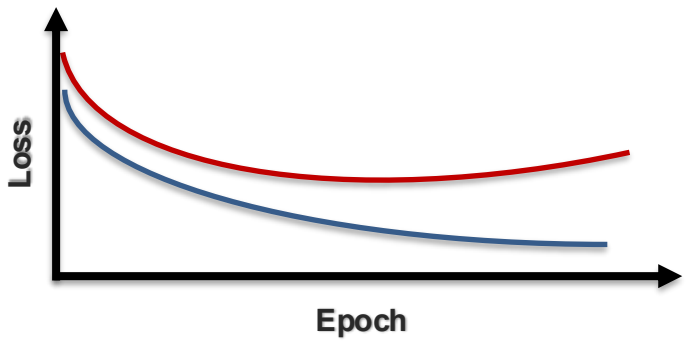
[Javaloy et al., Mitigating Modality Collapse in Multimodal VAEs via Impartial Optimization. ICML 2022]
[Goyal et al., Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. CVPR 2017]

multisensory intelligence

Optimization Challenges

2 explanations for drop in performance:

1. Multimodal networks are more prone to overfitting due to **increased complexity**
2. Different modalities overfit and generalize at **different rates**



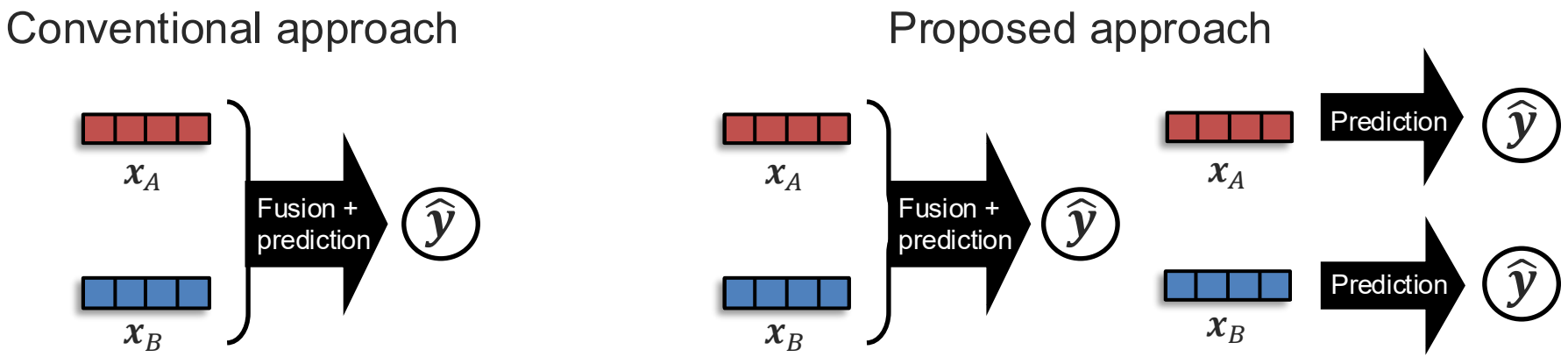
Key idea 1: compute overfitting-to-generalization ratio (OGR)

→ Gap between training and valid loss

OGR wrt each modality tells us how much to train that modality

[Wang et al., What Makes Training Multi-modal Classification Networks Hard? CVPR 2020]
[Wu et al., Characterizing and Overcoming the Greedy Nature of Learning in Multi-modal Deep Neural Networks. ICML 2022]

Optimization Challenges



Key idea 2: Simultaneously train unimodal networks to estimate OGR wrt each modality

➕ Reweight multimodal loss using unimodal OGR values

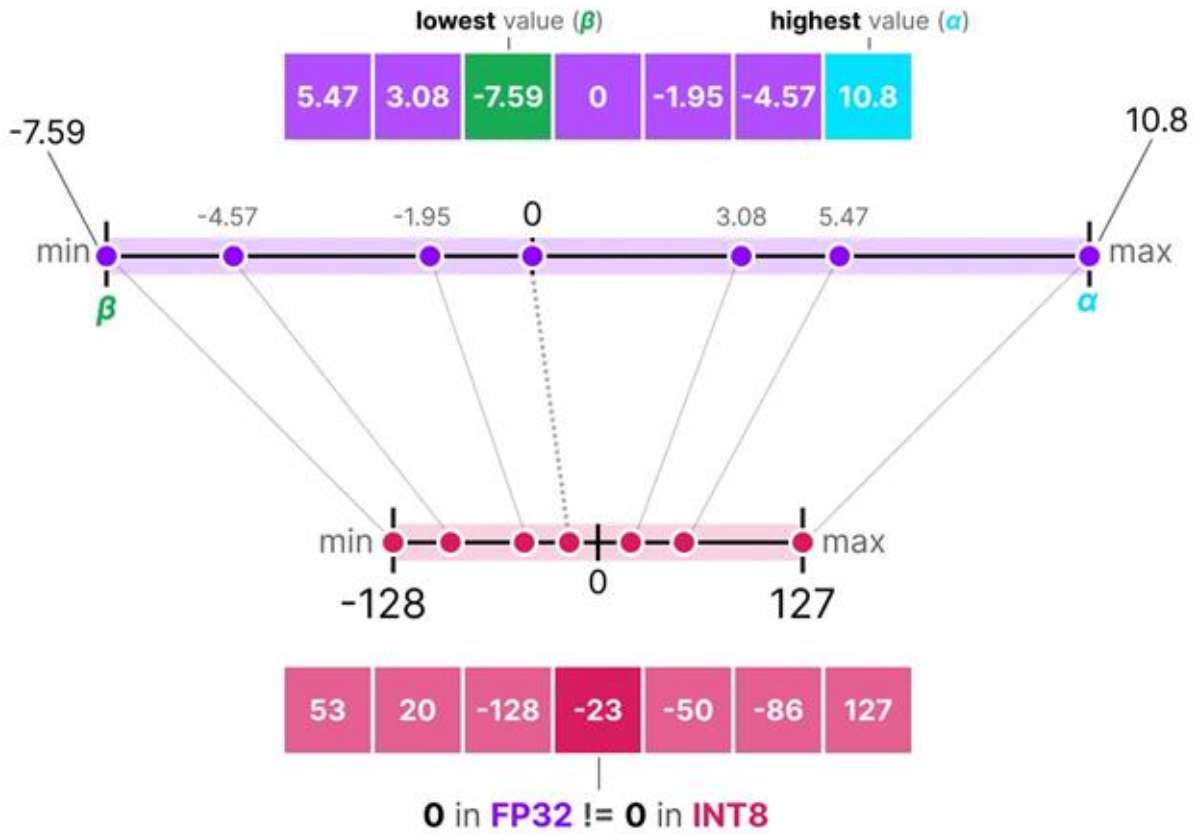
➡ Allows to better balance generalization & overfitting rate of different modalities

[Wang et al., What Makes Training Multi-modal Classification Networks Hard? CVPR 2020]
[Wu et al., Characterizing and Overcoming the Greedy Nature of Learning in Multi-modal Deep Neural Networks. ICML 2022]

multisensory intelligence

Efficient Inference: Quantization

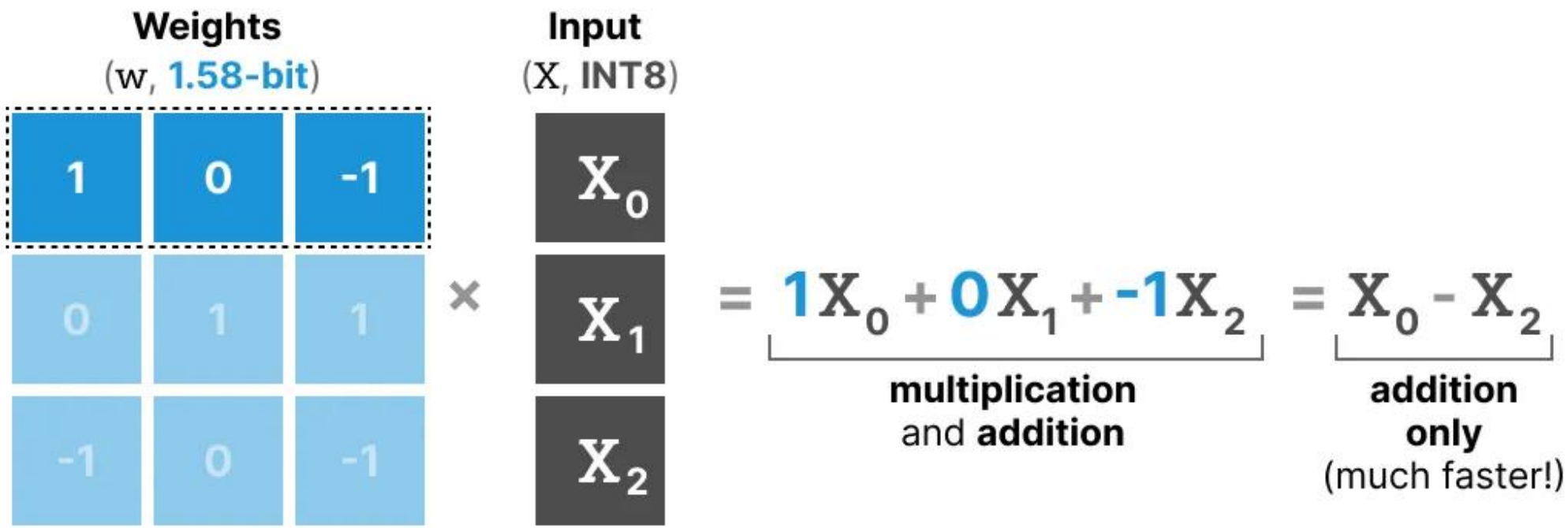
- Range Clipping
- Scale & Shift
- Convert to lower bits
- Calibration



A Visual Guide to Quantization, https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-quantization

Efficient Inference: Quantization

- Quantize weights to ternary {-1, 0, 1}
- New hardware needed to be truly 1.58 bits / log2(3)

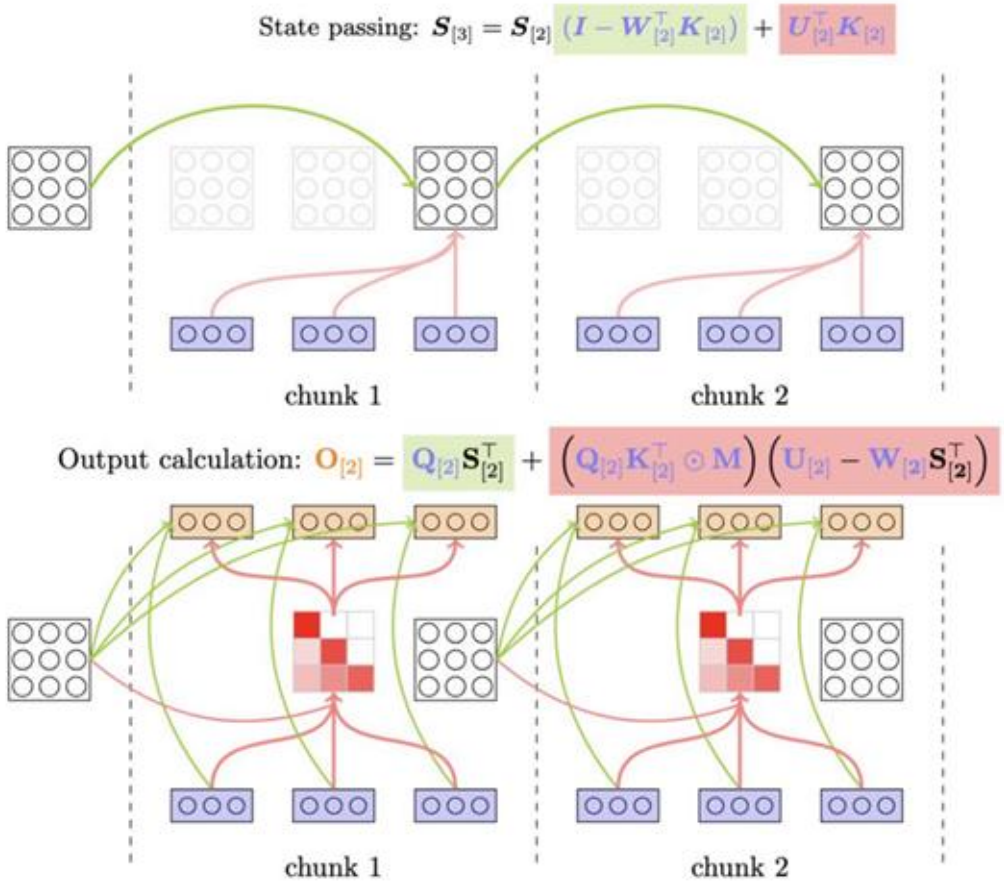


A Visual Guide to Quantization, https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-quantization

Linear Attention

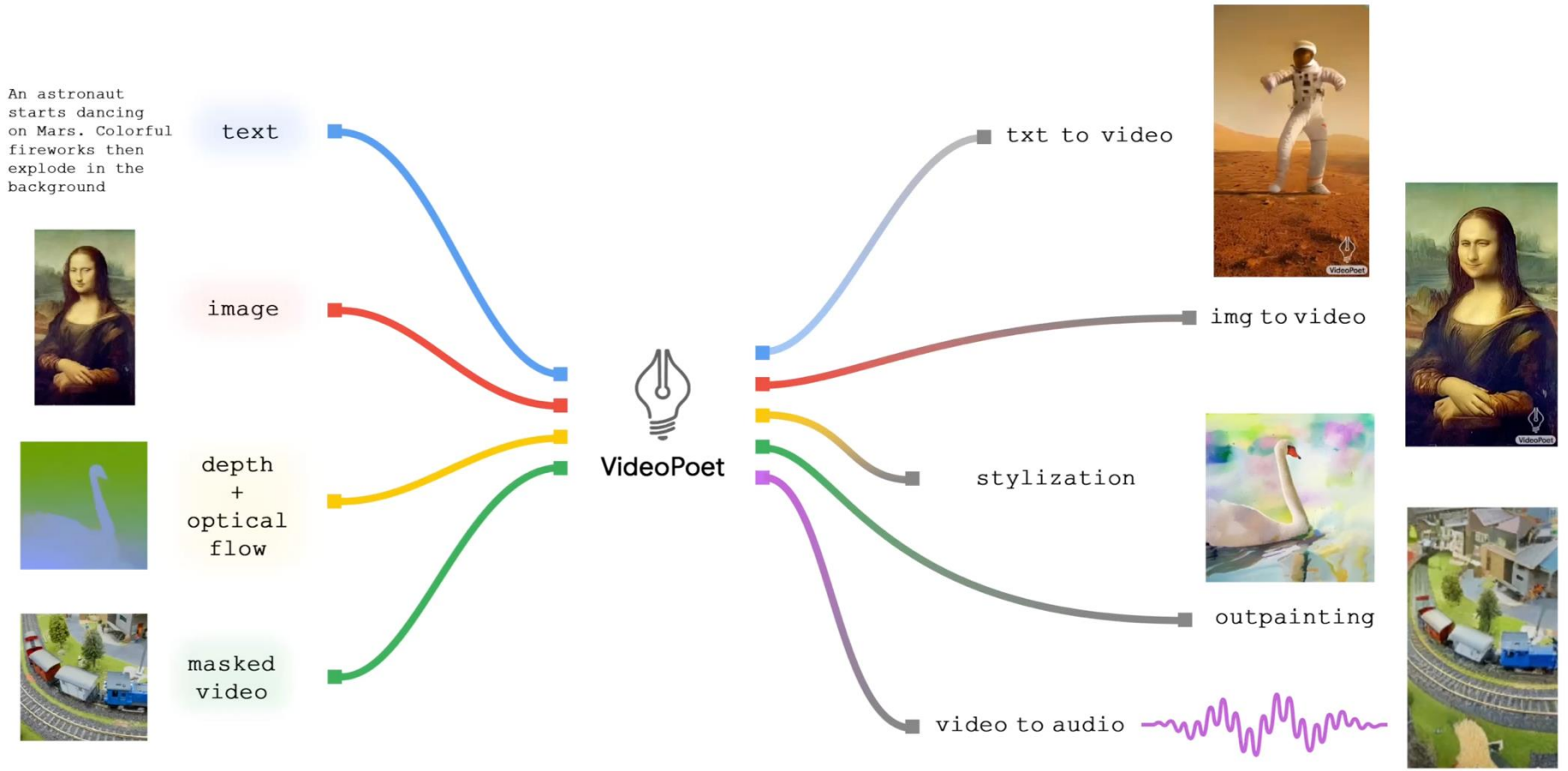
- Time Complexity of attention scales at O(n^2)
- Solution: Linear Attention
- Key idea: Combine Transformer Style Full Attention with RNN
- Split sentence into chunks, run attention within chunk, then combine them in RNN style aggregation
- Promising performance on par with full attention

$$h = \text{softmax}\left(\frac{XW_q W_k^T X^T}{\sqrt{d}}\right) XW_v$$



Any-to-any Models

Multimedia, content creation, creativity and the arts



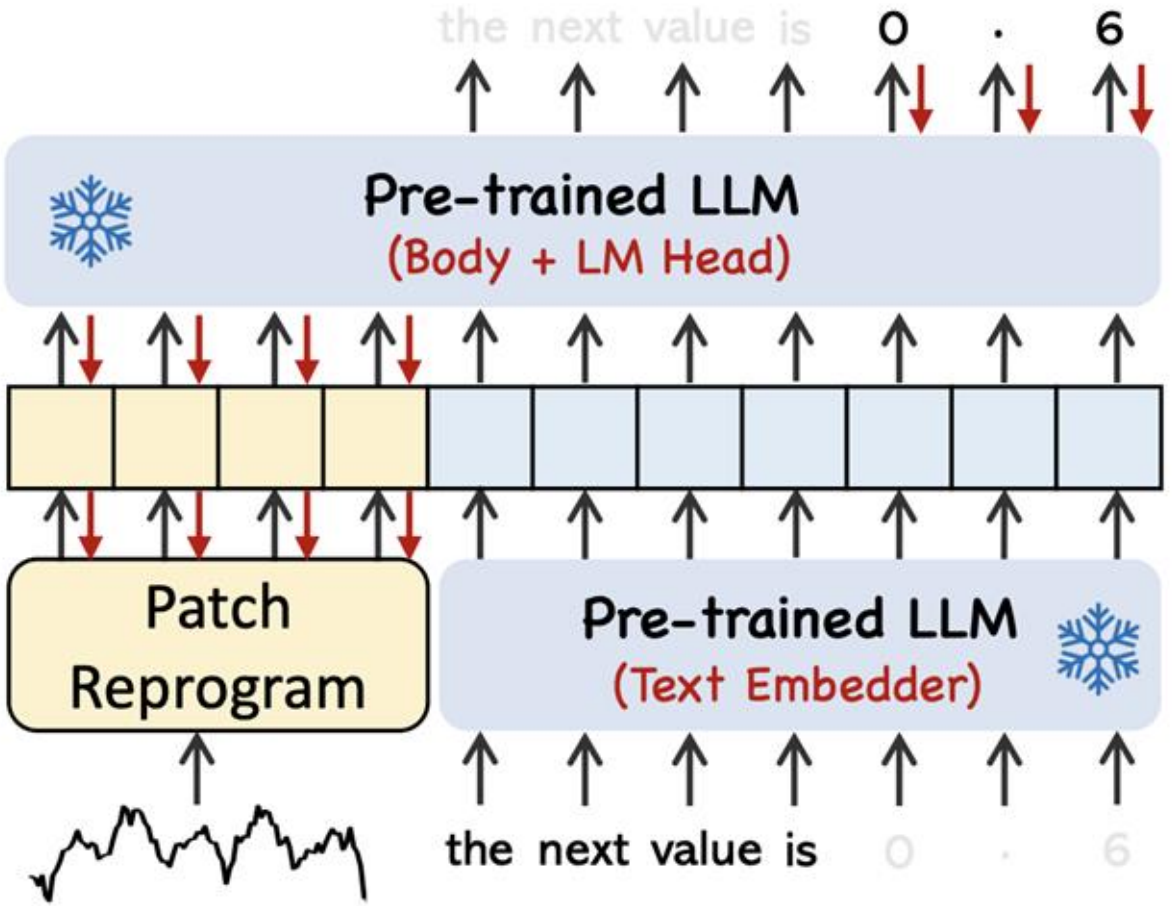
[Kondratyuk et al., VideoPoet: A Large Language Model for Zero-shot Video Generation. ICML 2024]

multisensory intelligence

Time-series Multimodal Models

- Typically trained & aligned the same way as vision language models (alignment + instruction tuning)
- Works for both analysis and prediction

- Example: Time-LLM, OneFitsAll

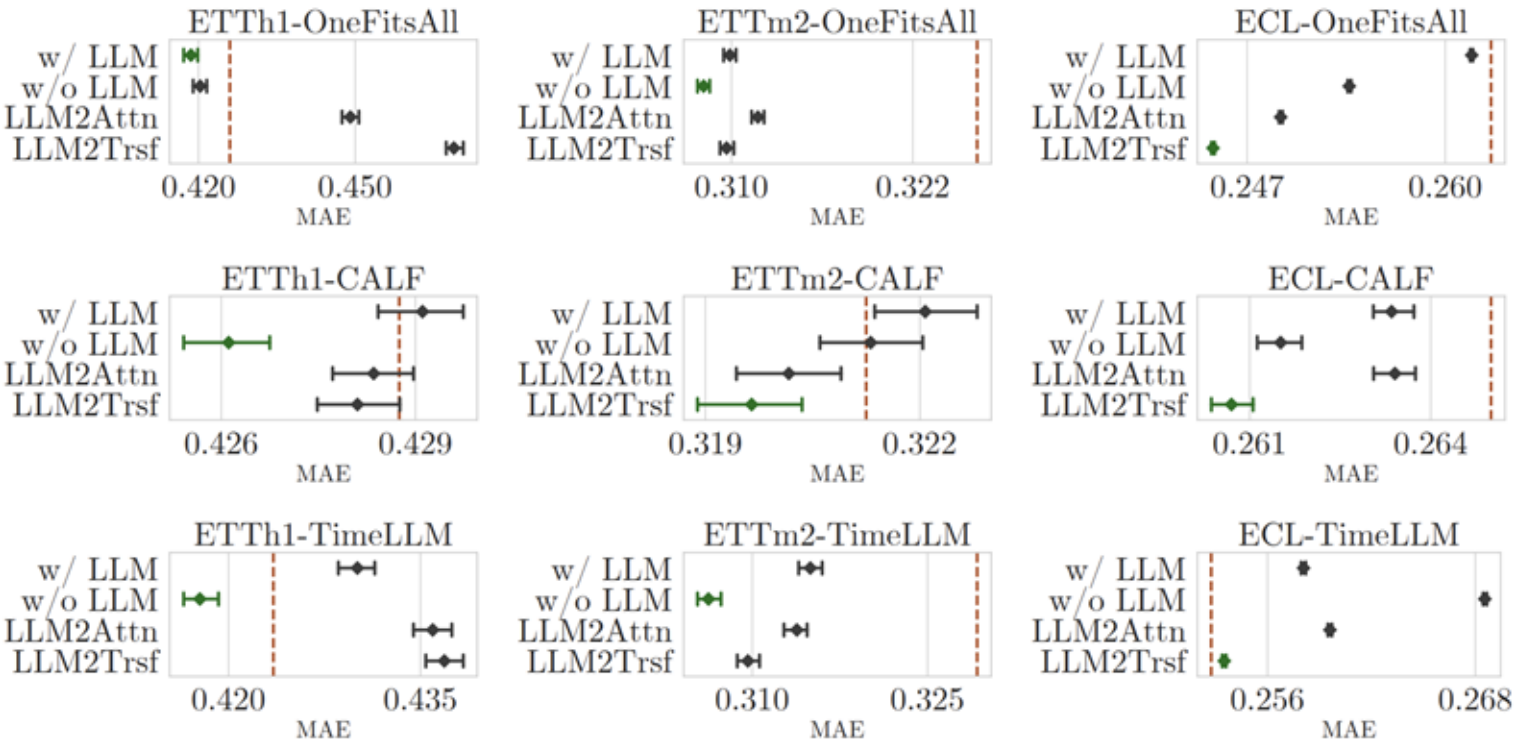


[Time-LLM: Time Series Forecasting by Reprogramming Large Language Models, https://arxiv.org/abs/2310.01728]

Time-series Multimodal Models

- But some current time series LLMs have questionable performance. Replacing LLM with a simple attention layer doesn't significantly degrade performance (sometimes even better).

* Lower is better

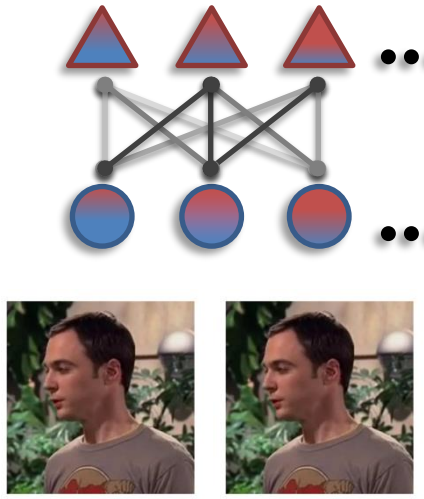


[Are Language Models Actually Useful for Time Series Forecasting? https://arxiv.org/abs/2406.16964]

Multimodal Reasoning Agents

Solving hard problems using step-by-step reasoning and action taking in multiple modalities

It's just a privilege to watch your mind at work.



Multimodal representation

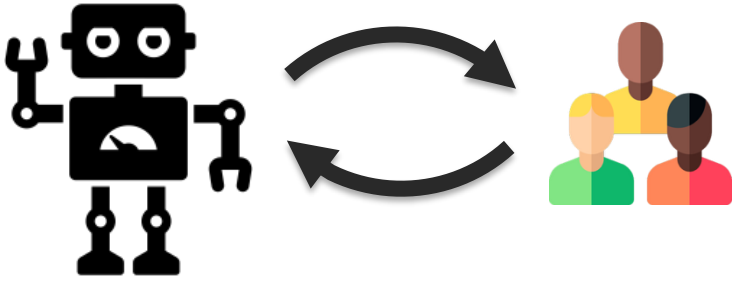
This person is being sarcastic. They seem to be close friends.

Here's a story of them in a different culture…

+

(quote previous episodes)
(highlight multimodal information)

(generate future episodes)

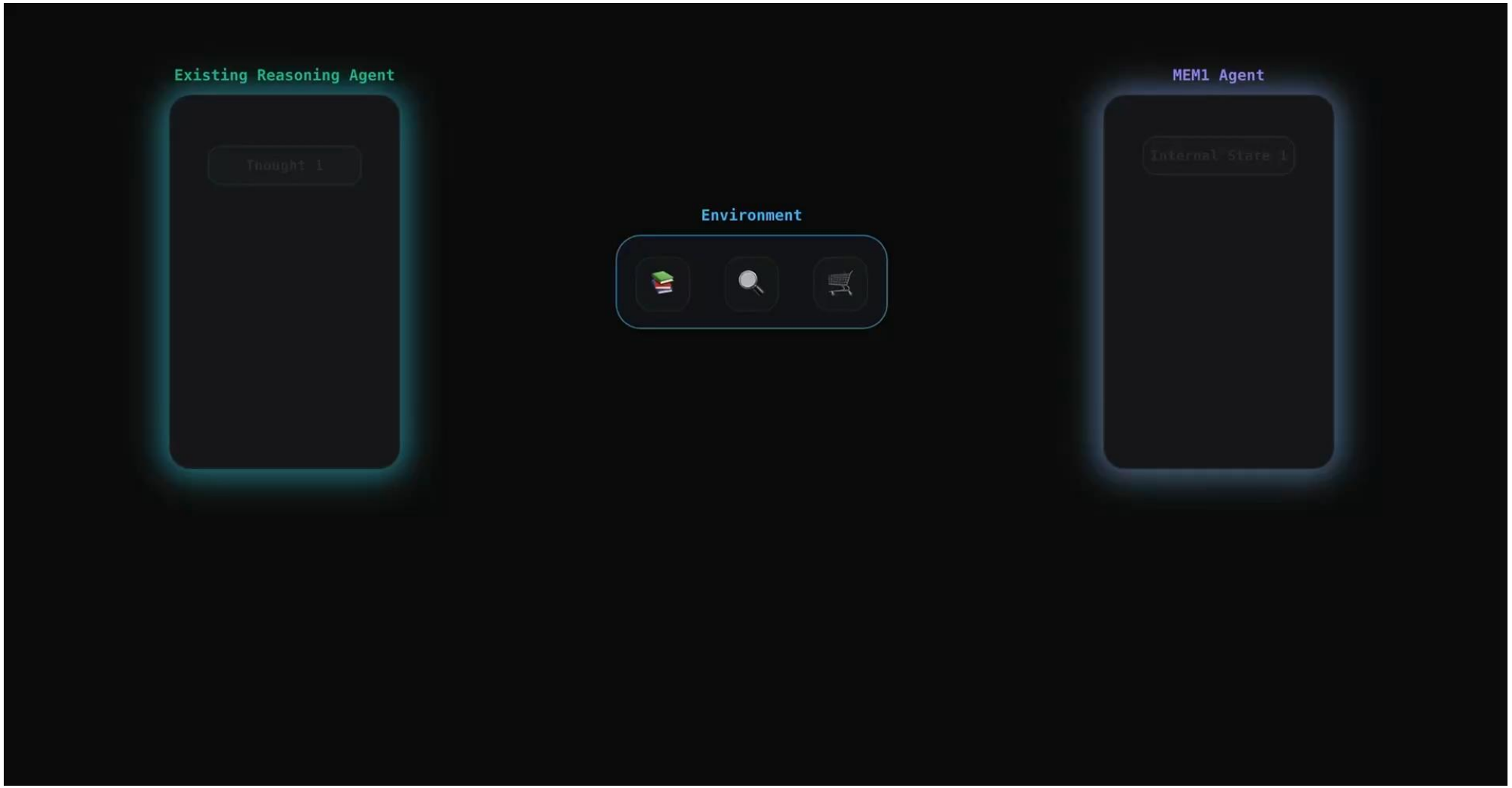


Reasoning context is getting longer and longer…

multisensory intelligence

MEM1: Memory Efficient Reasoning

https://github.com/MIT-MI/MEM1



[Zhou et al., Learning to Synergize Memory and Reasoning for Efficient Long-Horizon Agents. ICLR 2026]

Self-Evolving Agents

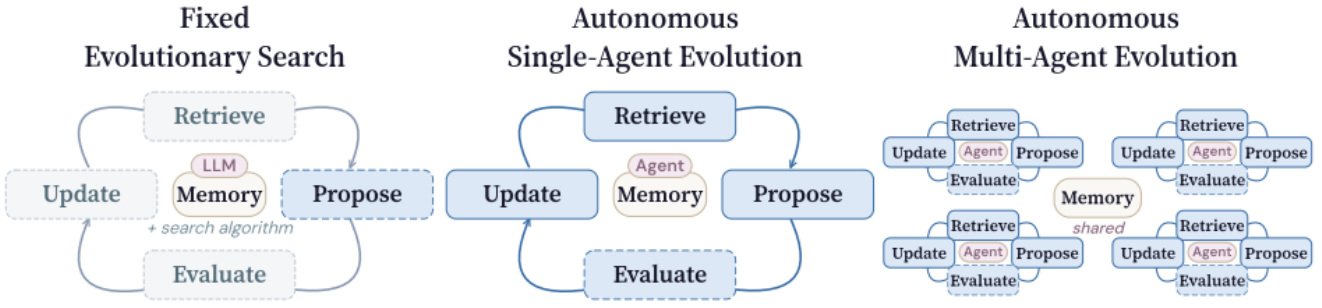
MEM1 is an agent that learns to manage its memory. Other agents manage tools, APIs, search, coding etc.

Question 1: How can agents automatically decide what functions to use for a given task?

Answer: Self-evolving agents

Question 2: How can agents automatically delegate tasks to different agents and organize themselves?

Answer: Self-evolving multi-agent systems



	Fixed Evolutionary Search	Autonomous Single-Agent Evolution	Autonomous Multi-Agent Evolution
Retrieve	Context selected by fixed rules.	Agent chooses what to inspect.	Agent reads from shared memory.
Propose	Model proposes from given context.	Agent plans, implements, and tests.	Multiple agents propose in parallel.
Evaluate	Triggered by a fixed loop.	Agent decides when to evaluate.	Each agent schedules eval.
Update	Updated by fixed rules.	Agent decides what to store.	Agents write to shared memory.

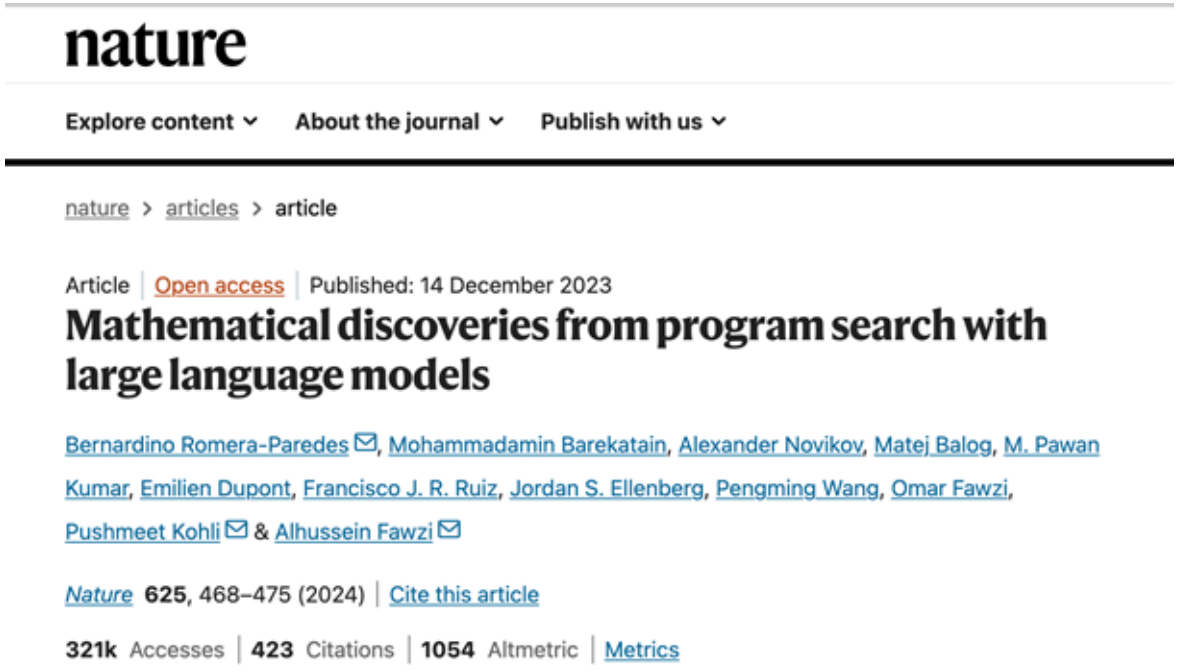
Fixed Partial autonomy Full autonomy

[Qu et al., CORAL: Towards Autonomous Multi-Agent Evolution for Open-Ended Discovery. arXiv 2026]

multisensory intelligence

Self-Evolving Agents

Agents are now helping with scientific discovery



nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › articles › article

Article | Open access | Published: 14 December 2023

Mathematical discoveries from program search with large language models

Bernardino Romera-Paredes ✉, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli ✉ & Alhussein Fawzi ✉

Nature **625**, 468–475 (2024) | Cite this article

321k Accesses | **423** Citations | **1054** Altmetric | Metrics

May 14, 2025 Science

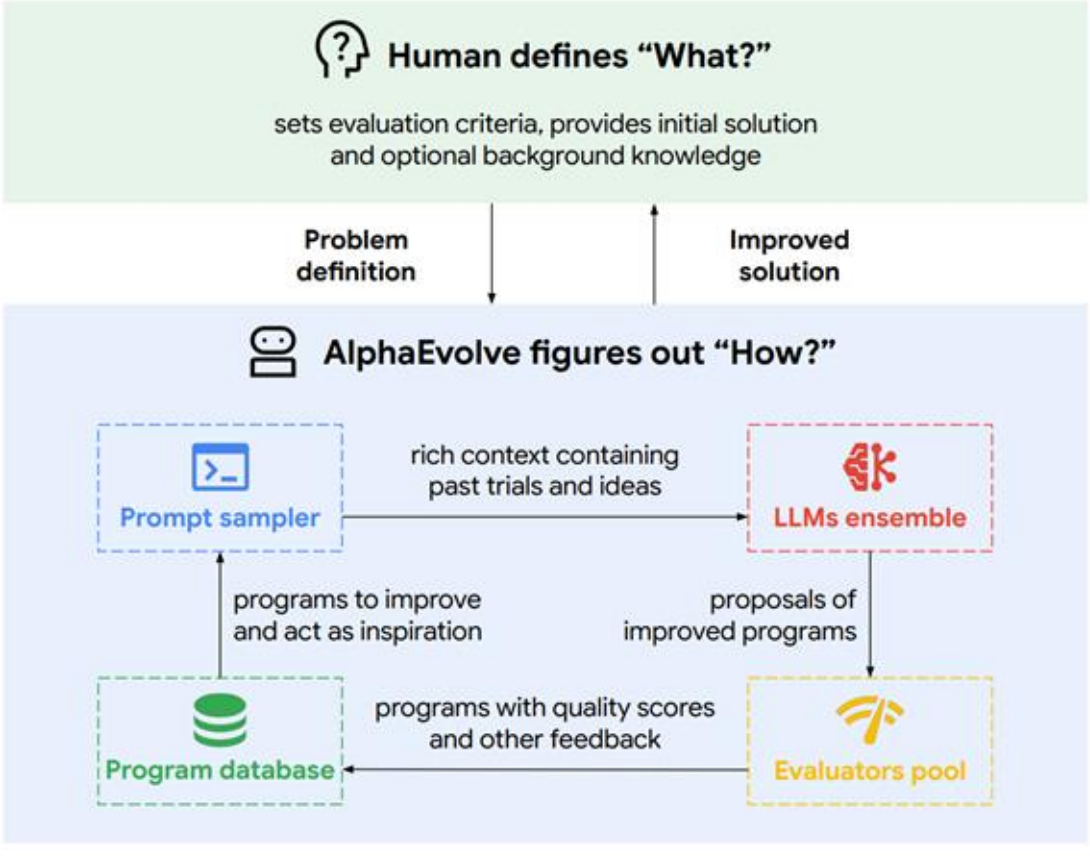
AlphaEvolve: A Gemini-powered coding agent for designing advanced algorithms

AlphaEvolve team

Share

multisensory intelligence

AlphaEvolve and OpenEvolve



```
# EVOLVE-BLOCK START
"""Image classification experiment in jaxline."""
import jax
...
# EVOLVE-BLOCK-END

...

# EVOLVE-BLOCK-START
class ConvNet(hk.Module):
  def __init__(self, num_classes): ...
  def __call__(self, inputs, is_training): ...

def sweep():
  return hyper.zipit([...])
# EVOLVE-BLOCK-END

...

def evaluate(eval_inputs) -> dict[str, float]:
  ...
  return metrics
```

(a)

```
The current model uses a simple ResNet architecture with only three ResNet blocks. We can improve its performance by increasing the model capacity and adding regularization. This will allow the model to learn more complex features and generalize better to unseen data. We also add weight decay to the optimizer to further regularize the model and prevent overfitting. AdamW is generally a better choice than Adam, especially with weight decay.

<<<<<<< SEARCH
    self._block1 = ResNetBlock(num_channels)
    self._block2 = ResNetBlock(num_channels * 2, stride=2)
    self._block3 = ResNetBlock(num_channels * 4, stride=2)
=======
    self._block1 = ResNetBlock(num_channels)
    self._block2 = ResNetBlock(num_channels, stride=1)
    self._block3 = ResNetBlock(num_channels * 2, stride=2)
    self._block4 = ResNetBlock(num_channels * 2, stride=1)
    self._block5 = ResNetBlock(num_channels * 4, stride=2)
    self._block6 = ResNetBlock(num_channels * 4, stride=1)
>>>>>>> REPLACE
<<<<<<< SEARCH
  def optimizer(self, learning_rate):
    return optax.adam(learning_rate)
=======
  def optimizer(self, learning_rate):
    return optax.adamw(learning_rate, weight_decay=1e-4)
>>>>>>> REPLACE
```

(c)

```
Act as an expert software developer. Your task is to iteratively improve the provided codebase. [...]

- Prior programs

Previously we found that the following programs performed well on the task at hand:

top_1_acc: 0.796; neg_eval_log_loss: 0.230; average_score: 0.513

"""Image classification experiment in jaxline."""
[...]
class ConvNet(hk.Module):
  """Network."""

  def __init__(self, num_channels=32, num_output_classes=10):
    super().__init__()
    self._conv1 = hk.Conv2D(num_channels, kernel_shape=3)
    self._conv2 = hk.Conv2D(num_channels * 2, kernel_shape=3)
    self._conv3 = hk.Conv2D(num_channels * 4, kernel_shape=3)
    self._logits_module = hk.Linear(num_output_classes)
[...]

- Current program

Here is the current program we are trying to improve (you will need to propose a modification to it below).

top_1_acc: 0.862; neg_eval_log_loss: 0.387; average_score: 0.624

"""Image classification experiment in jaxline."""
[...]
class ConvNet(hk.Module):
  """Network."""

  def __init__(self, num_channels=32, num_output_classes=10):
    super().__init__()
    self._conv1 = hk.Conv2D(num_channels, kernel_shape=3)
    self._block1 = ResNetBlock(num_channels)
    self._block2 = ResNetBlock(num_channels * 2, stride=2)
    self._block3 = ResNetBlock(num_channels * 4, stride=2)
    self._logits_module = hk.Linear(num_output_classes)
[...]

SEARCH/REPLACE block rules:
[...]

Make sure that the changes you propose are consistent with each other. For example, if you refer to a new config variable somewhere, you should also propose a change to add that variable.

Example:
[...]

Task
Suggest a new idea to improve the code that is inspired by your expert knowledge of optimization and machine learning.

Describe each change with a SEARCH/REPLACE block.
```

(b)

Figure 1 | *AlphaEvolve* high-level overview.

Problems Solved by AlphaEvolve

$$\max_{-1/2\le t\le 1/2}\int_{\mathbb{R}} f_1(t-x)f_1(x)\,dx \ge C_1\left(\int_{-1/4}^{1/4} f_1(x)\,dx\right)^2$$

1.5098 → **1.5053**

$$\|f*f\|_2^2 \le C_2\|f*f\|_1\|f*f\|_\infty$$

0.8892 → **0.8962**

$$\max_{-1/2\le t\le 1/2}\left|\int_{\mathbb{R}} f(t-x)f(x)\,dx\right| \ge C_3\left(\int_{-1/4}^{1/4} f(x)\,dx\right)^2$$

1.4581 → **1.4557**

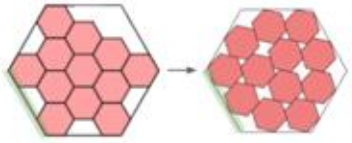
$$A(f)A(\hat{f}) \ge C_4$$

0.3523 → **0.3521**

Analysis

Hexagon outer edge

4.000 → **3.942**

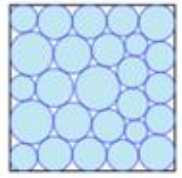


Max distance/min distance

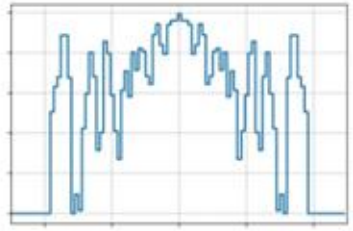
12.890 → **12.889**

Sum of radii

2.6340 → **2.6358**



Geometry



$$\sup_{x\in[-2,2]}\int_{-1}^{1} f(t)g(x+t)\,dt \ge C_5$$

0.380926 → **0.380924**

$$|A+B| \ll |A|$$

$$|A-B| \gg |A|^{C_6}$$

1.1446 → **1.1584**

Combinatorics

Evolve Solutions via Tree Search

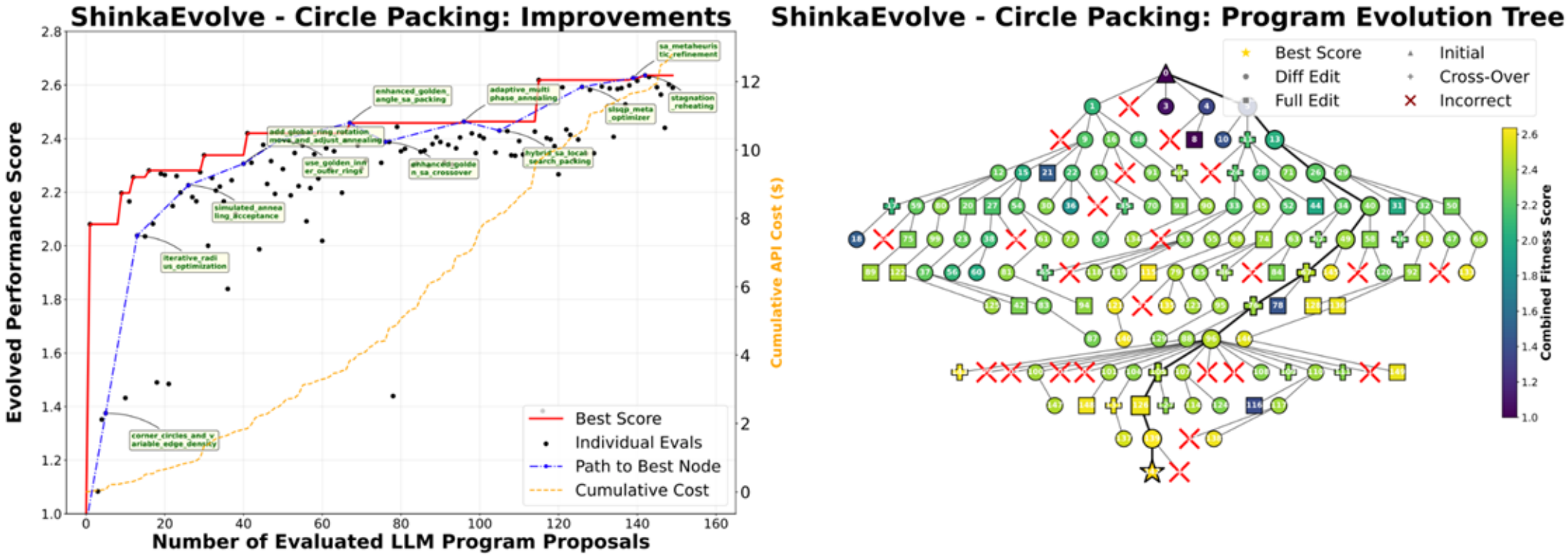
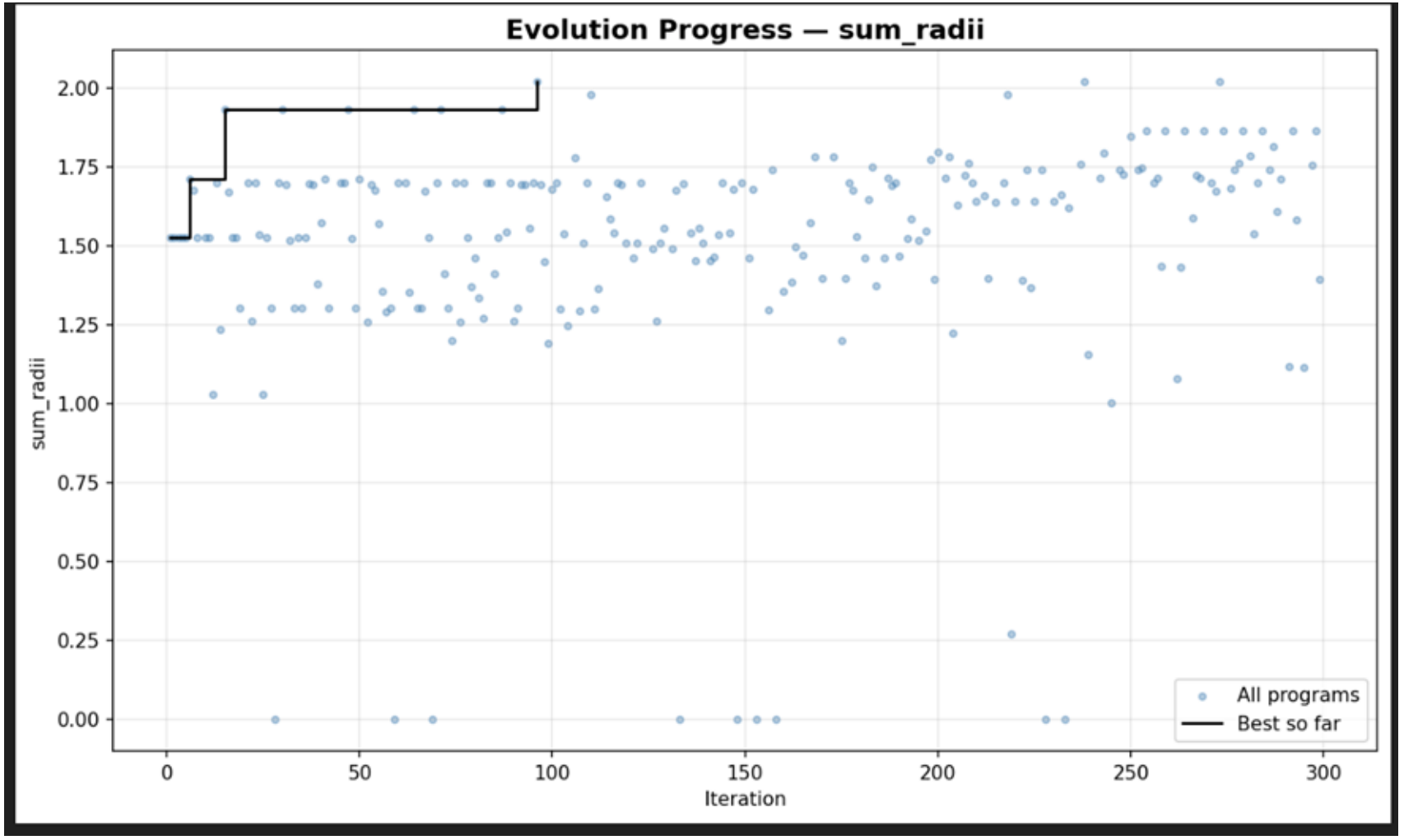


Figure 5 | **ShinkaEvolve on Circle Packing Task.** *Left:* ShinkaEvolve outperforms AlphaEvolve's solution within less than 150 program evaluations. *Right:* ShinkaEvolve's program evolution tree demonstrates the iterative composition of stepping stones into high-performing solutions.

Few Samples Lead to Improvements



multisensory intelligence

Evolving the Model

Recent works such as TTT-Discover and ThetaEvolve train the model at test-time with improvement score as the reward signal

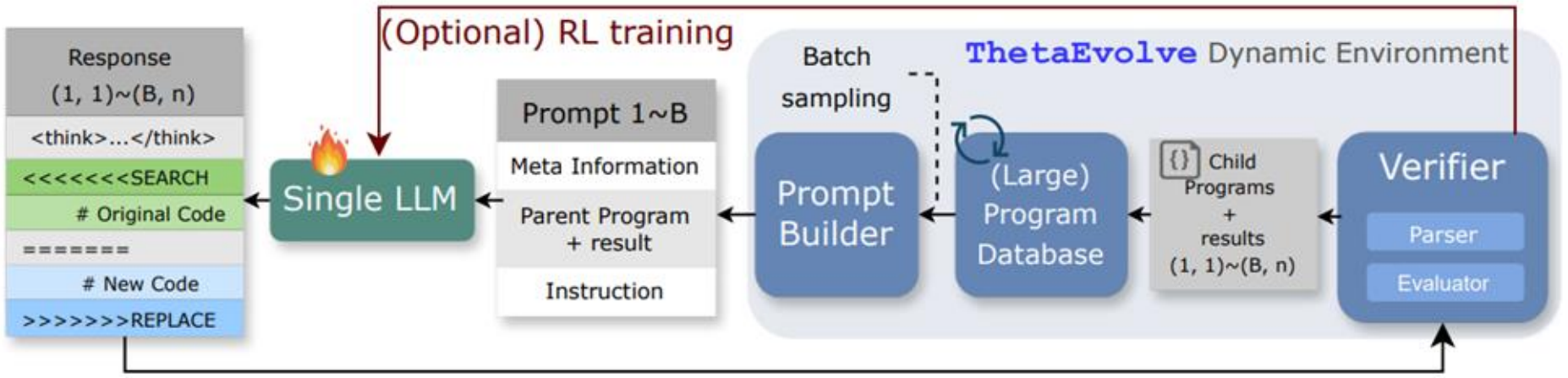
Algorithm 1 Test-Time Training to Discover (TTT-Discover)

Input: problem description d and policy π_{θ_0} with initial weights θ_0.
$R, T = \texttt{get\_env}(d)$ ▷ d induces the reward and transition functions of the environment (§2.1)
$\mathcal{H}_0 = \{(\texttt{<empty>}, R(\texttt{<empty>}), \{\})\}$ ▷ Initialize buffer with the empty solution (§2.2)
for $i = 0, 1, \ldots, N-1$ **do**
  $s_i, c_i \sim \texttt{reuse}(\mathcal{H}_i)$ ▷ Sample initial state and context with a `reuse` heuristic
  $a_i \sim \pi_{\theta_i}(\cdot \mid d, s_i, c_i)$ ▷ Sample action from policy
  $s'_i = T(a_i)$ ▷ Transition to next state
  $r_i = R(s'_i)$ ▷ Evaluate reward of next state
  $\mathcal{H}_{i+1} = \mathcal{H}_i \cup \{(s_i, a_i, s'_i, r_i)\}$ ▷ Add current attempt to buffer
  $\theta_{i+1} = \texttt{train}(\theta_i, (d, s_i, c_i, a_i, r_i))$ ▷ Improve the model weights with `train`
end for
return s_{i^*}, where $i^* = \arg\max_{i=0,1,\ldots,N-1} r_i$ ▷ Return the state with the highest reward

multisensory intelligence

Evolving the Model

Recent works such as TTT-Discover and ThetaEvolve train the model at test-time with improvement score as the reward signal

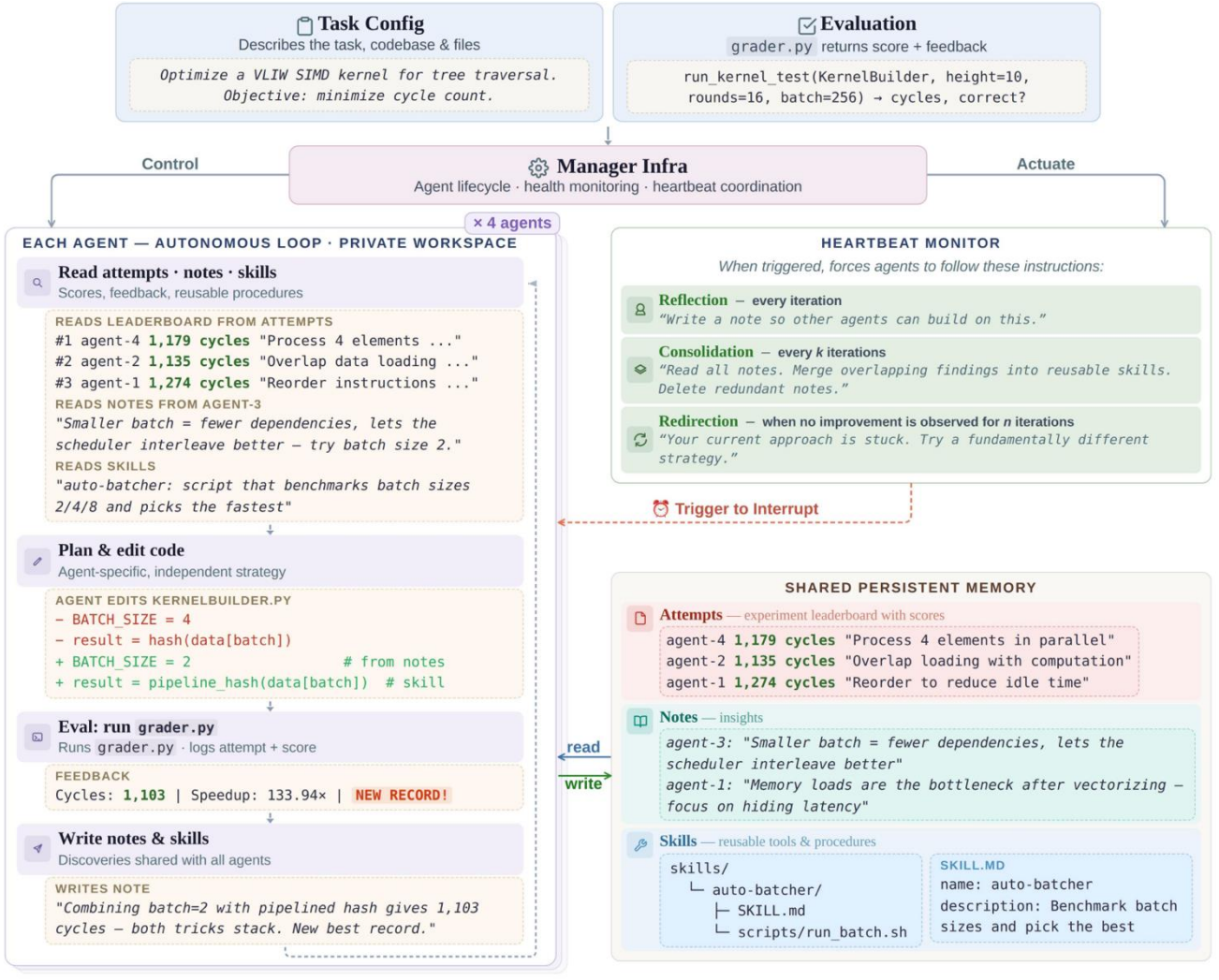


CORAL: Multi-Agent Evolution

https://github.com/Human-Agent-Society/CORAL

Key ideas:

- **Multi-agent specialization**
- **Increased autonomy**
- **Shared memory**
- **Heartbeat mechanism**

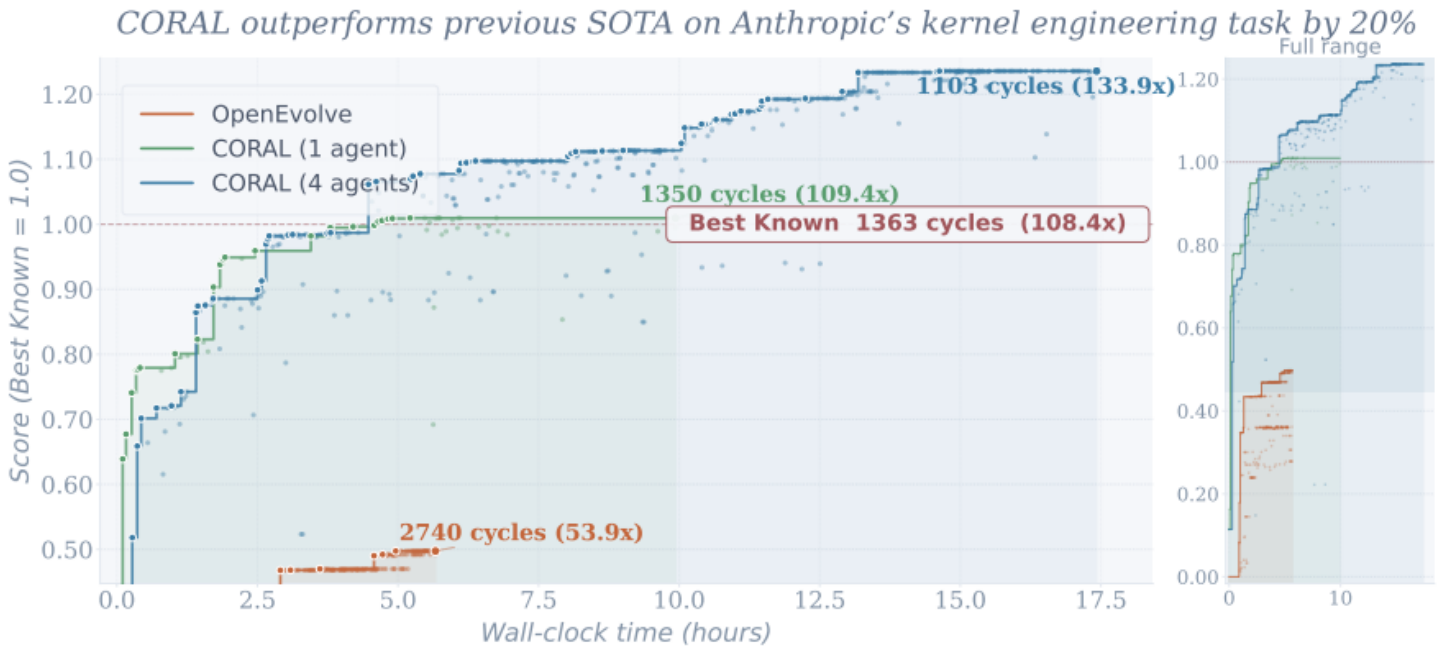


[Qu et al., CORAL: Towards Autonomous Multi-Agent Evolution for Open-Ended Discovery. arXiv 2026]

multisensory intelligence

CORAL: Multi-Agent Evolution

https://github.com/Human-Agent-Society/CORAL



	Model	Evals ↓	Best Cycles ↓	Speedup ↑	Imp. Rate ↑
• Best Known	Anthropic	–	1363	108.4x	–
• OpenEvolve	Opus 4.6	363	2740	53.9x	3% (12/363)
• CORAL (1 agent)	Opus 4.6	56	1350	109.4x	43% (24/56)
• CORAL (4 agents)	Opus 4.6	596	1103	133.9x	9% (54/596)

The task is to optimize a GPU kernel to minimize execution cycles from a 147,734-cycle baseline. Score = baseline / cycles, normalized so the previous best known result (1,363 cycles) equals 1.0. All methods use Claude Opus 4.6.

Operations research

→ Systems engineering

Scientific discovery

[Qu et al., CORAL: Towards Autonomous Multi-Agent Evolution for Open-Ended Discovery. arXiv 2026]

multisensory intelligence

Self-Evolving Agents

How agents can constantly verify their knowledge and improve their capabilities over time beyond static human supervision.

See blog post: https://quao627.github.io/blog/self-evolving-agents/

Assignments for This Coming Week

Please fill out course evaluations and give us feedback!

HW5 due this Friday May 8.

For project:

- Make sure to meet with myself and TAs this week if you need advice.
- Presentations next Tuesday May 12.
- Final report due Tuesday May 19.