Knowledge Augmented FMs

Presented by

Amir Shariatmadari, Guangzhi Xiong, Sabit Ahmed, Shiyu Feng

University of Virginia | ENGINEERING
Department of Computer Science

Overview

- Retrieval-Augmented Generation for Large Language Models: A Survey
  - Retrieval-Augmented Generation for AI-Generated Content: A Survey
- A Survey of Table Reasoning with Large Language Models
- Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models
- A Comprehensive Study of Knowledge Editing for Large Language Models

Retrieval-Augmented Generation for Large Language Models: A Survey

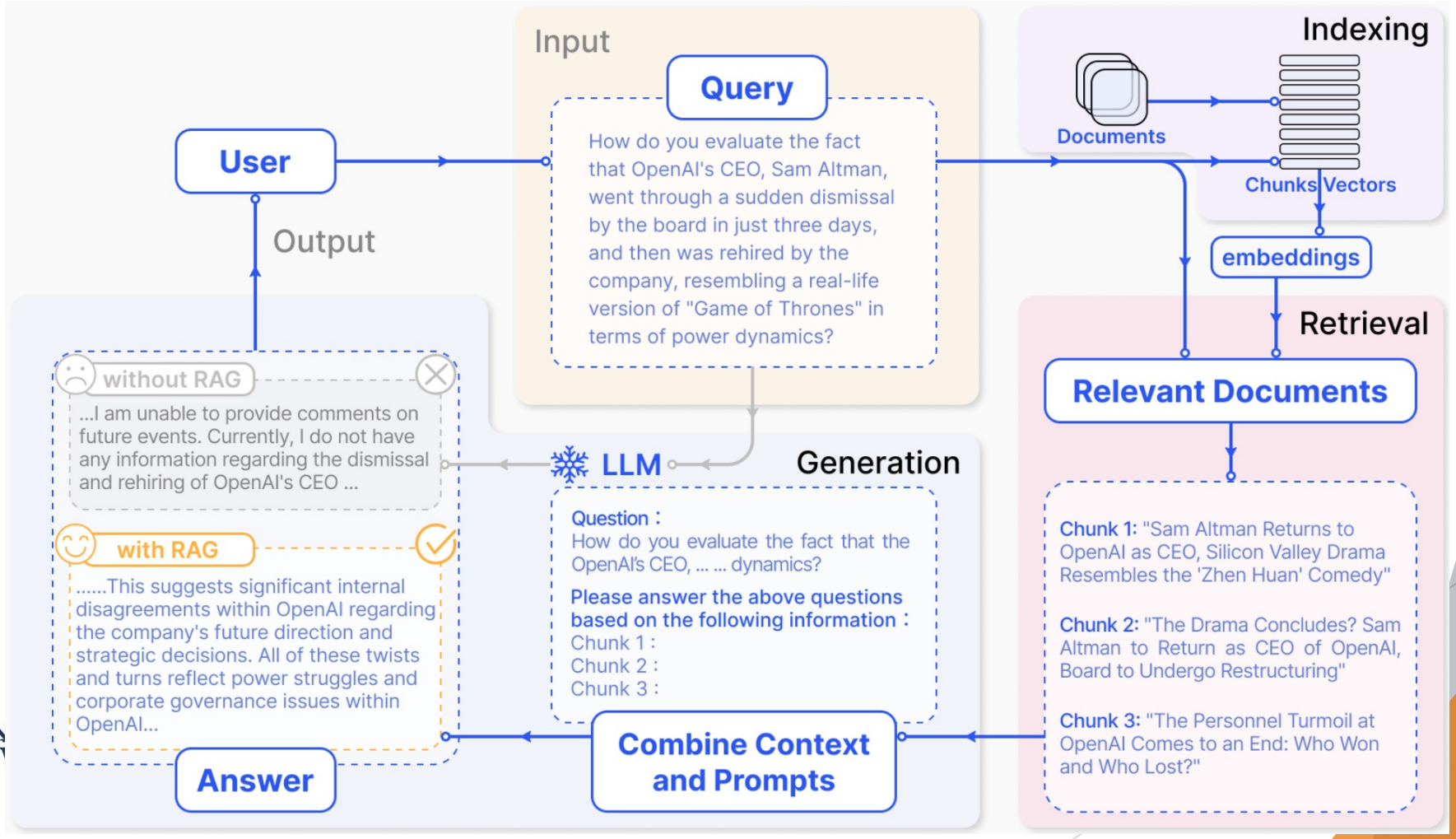
Tongji University, Fudan University

Presenter: Guangzhi Xiong, hhu4zu

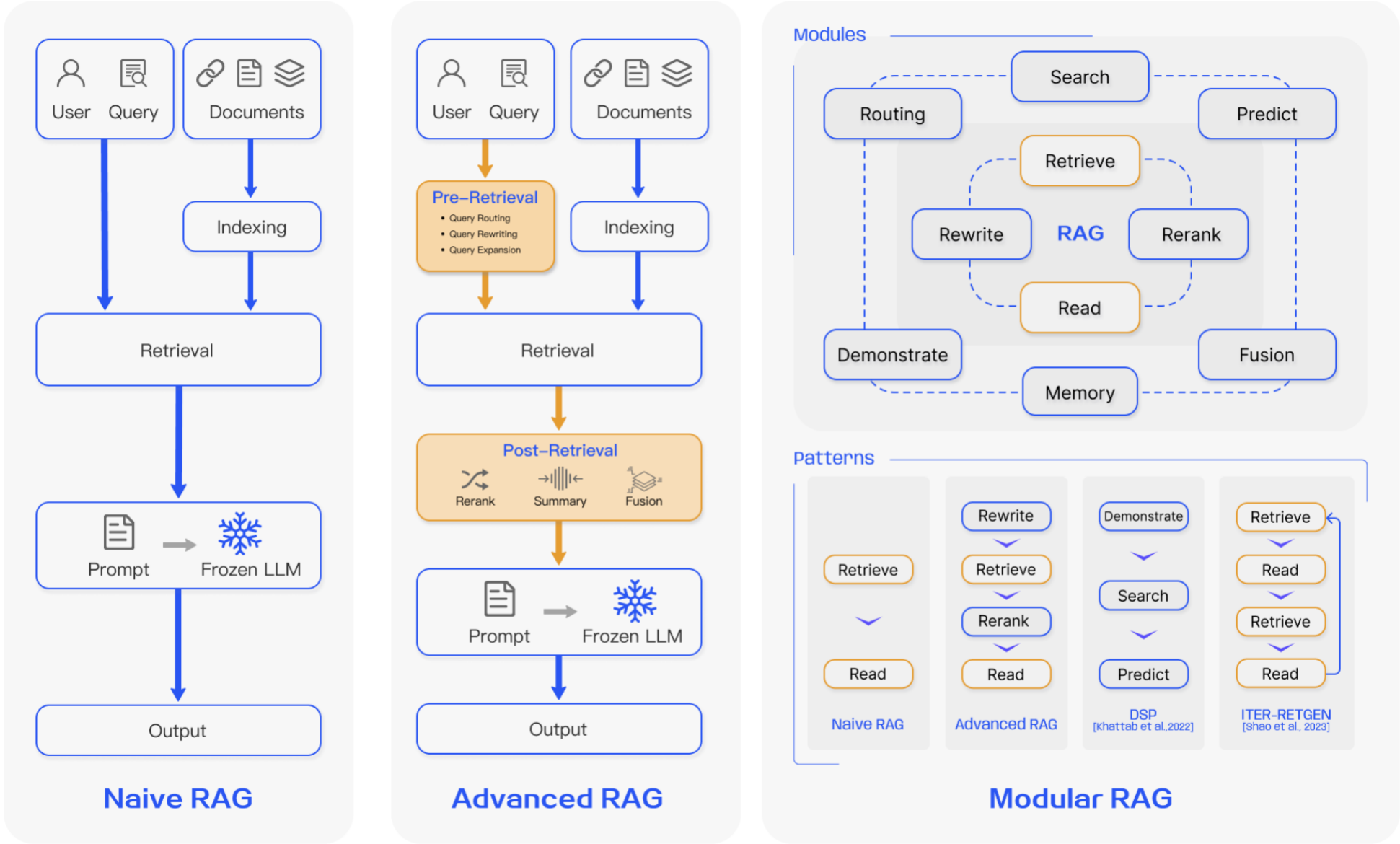
University of Virginia | ENGINEERING Department of Computer Science

What is Retrieval-Augmented Generation

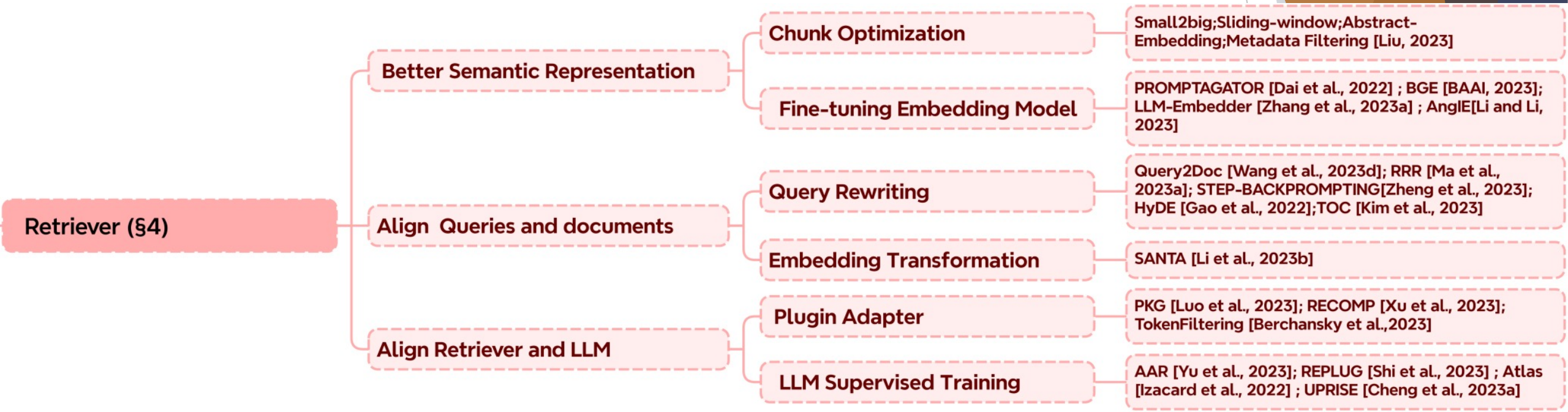
- Retrieval-Augmented Generation (RAG) can incorporate knowledge from external databases, which enhances the accuracy and credibility of the models.



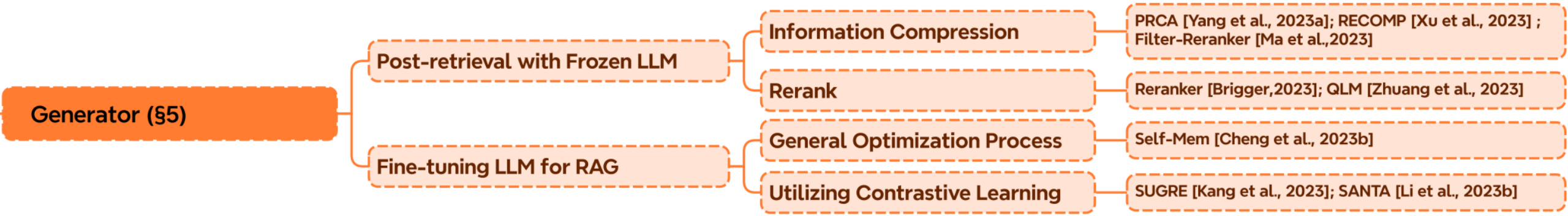
RAG Frameworks



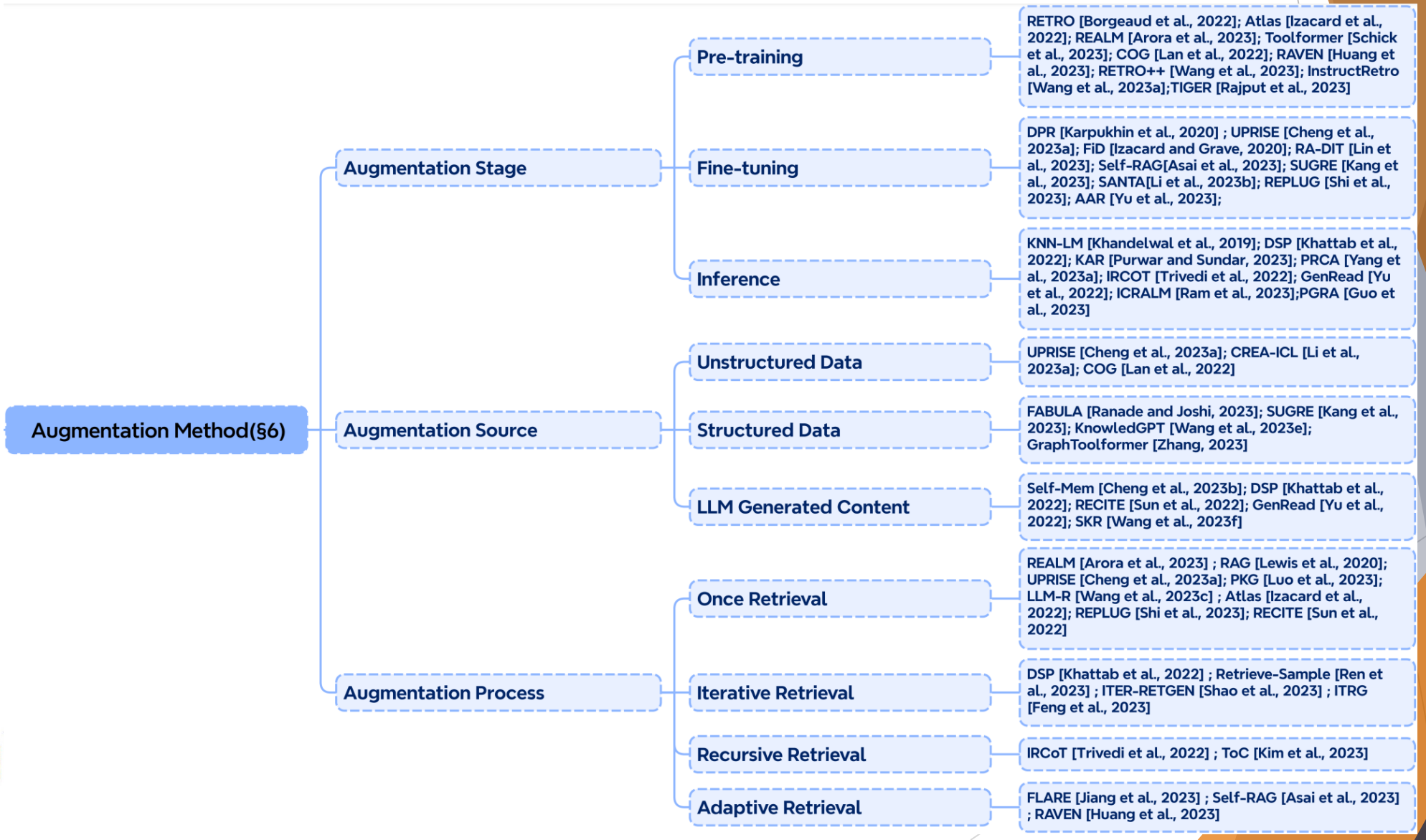
Taxonomy of RAG Techniques: Retrieval



Taxonomy of RAG Techniques: Generation



Taxonomy of RAG Techniques: Augmentation



RAG vs. Fine-tuning

	Feature Comparison	RAG	Fine-Tuning
↑	Knowledge Updates	Directly updating the retrieval knowledge base ensures that the information remains current without the need for frequent retraining, making it well-suited for dynamic data environments.	Stores static data, requiring retraining for knowledge and data updates.
↑	External Knowledge	Proficient in leveraging external resources, particularly suitable for accessing documents or other structured/unstructured databases.	Can be utilized to align the externally acquired knowledge from pretraining with large language models, but may be less practical for frequently changing data sources.
↑	Data Processing	Involves minimal data processing and handling.	Depends on the creation of high-quality datasets, and limited datasets may not result in significant performance improvements.
↓	Model Customization	Focuses on information retrieval and integrating external knowledge but may not fully customize model behavior or writing style.	Allows adjustments of LLM behavior, writing style, or specific domain knowledge based on specific tones or terms.
↑	Interpretability	Responses can be traced back to specific data sources, providing higher interpretability and traceability.	Similar to a black box, it is not always clear why the model reacts a certain way, resulting in relatively lower interpretability.

RAG vs. Fine-tuning (cont.)

	Feature Comparison	RAG	Fine-Tuning
⟷	Computational Resources	Depends on computational resources to support retrieval strategies and technologies related to databases. Additionally, it requires the maintenance of external data source integration and updates.	The preparation and curation of high-quality training datasets, defining fine-tuning objectives, and providing corresponding computational resources are necessary.
↓	Latency Requirements	Involves data retrieval, which may lead to higher latency.	LLM after fine-tuning can respond without retrieval, resulting in lower latency.
↑	Reducing Hallucinations	Inherently less prone to hallucinations as each answer is grounded in retrieved evidence.	Can help reduce hallucinations by training the model based on specific domain data but may still exhibit hallucinations when faced with unfamiliar input.
⟷	Ethical and Privacy Issues	Ethical and privacy concerns arise from the storage and retrieval of text from external databases.	Ethical and privacy concerns may arise due to sensitive content in the training data.

RAG Evaluation

Table 2: Summary of metrics applicable for evaluation aspects of RAG

	Context Relevance	Faithfulness	Answer Relevance	Noise Robustness	Negative Rejection	Information Integration	Counterfactual Robustness
Accuracy	✓	✓	✓	✓	✓	✓	✓
EM					✓		
Recall	✓						
Precision	✓			✓			
R-Rate							✓
Cosine Similarity			✓				
Hit Rate	✓						
MRR	✓						
NDCG	✓						

Quality Scores

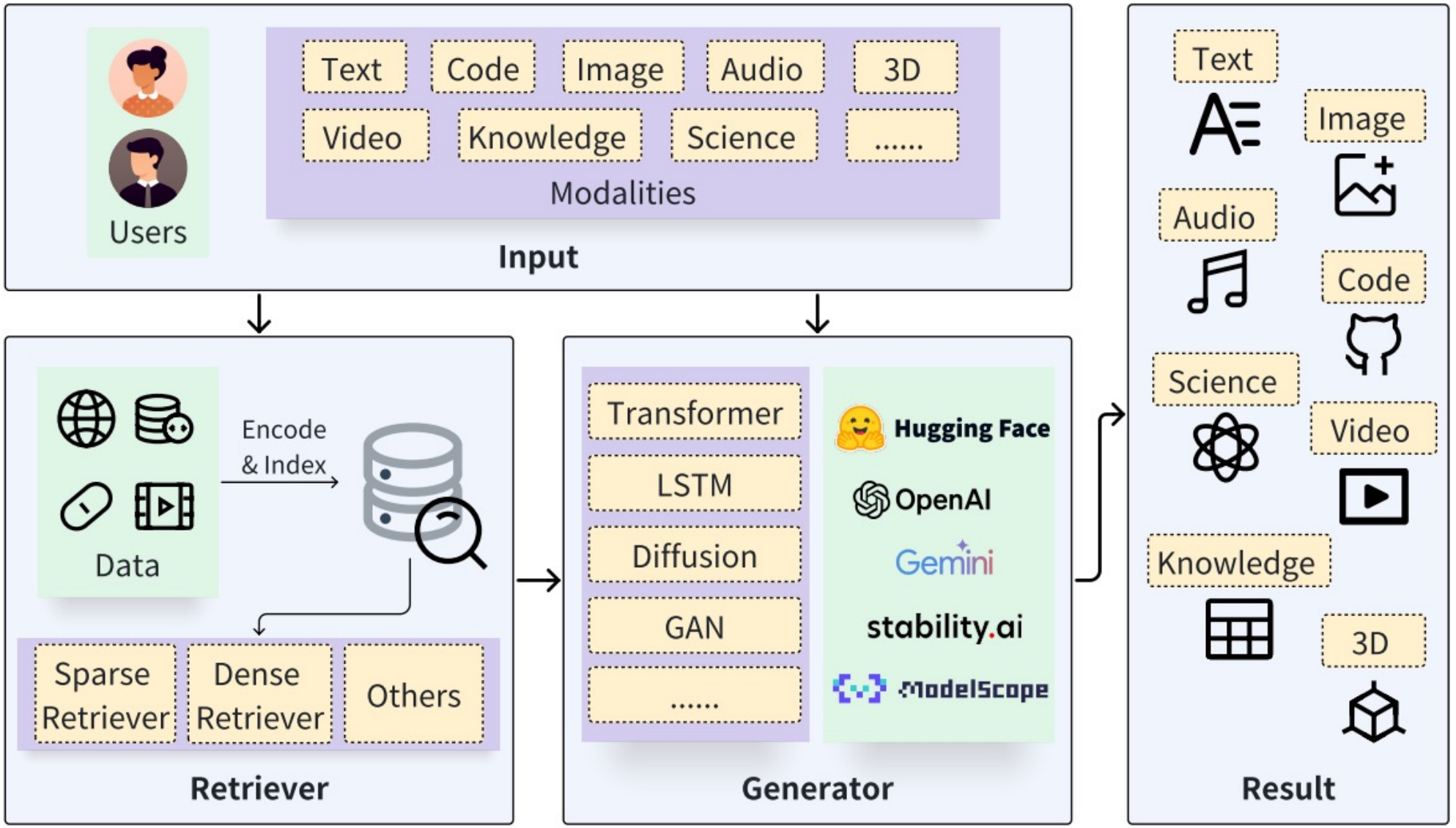
Required Abilities

Retrieval-Augmented Generation for AI-Generated Content: A Survey

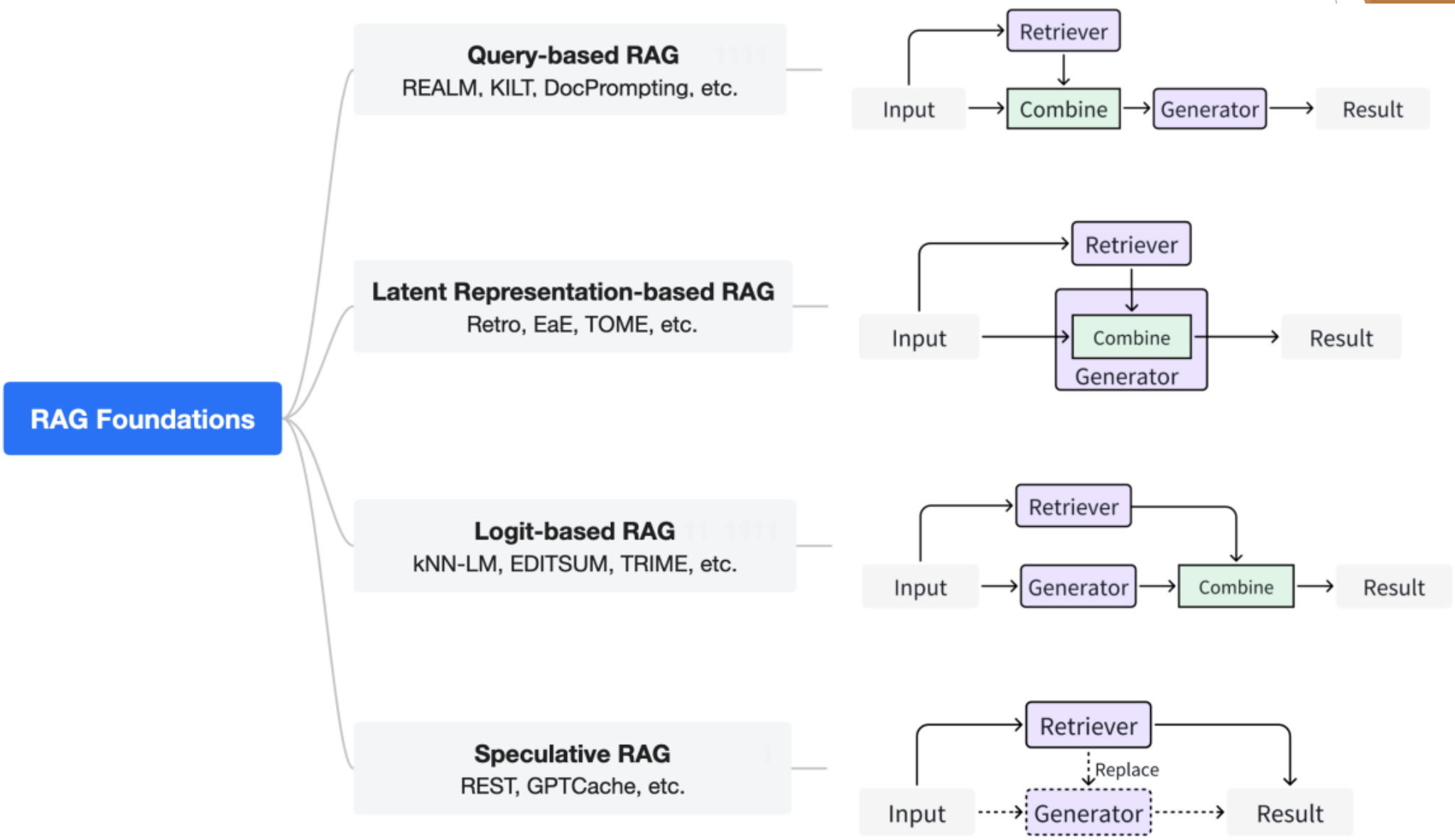
Peking University

Presenter: Guangzhi Xiong, hhu4zu

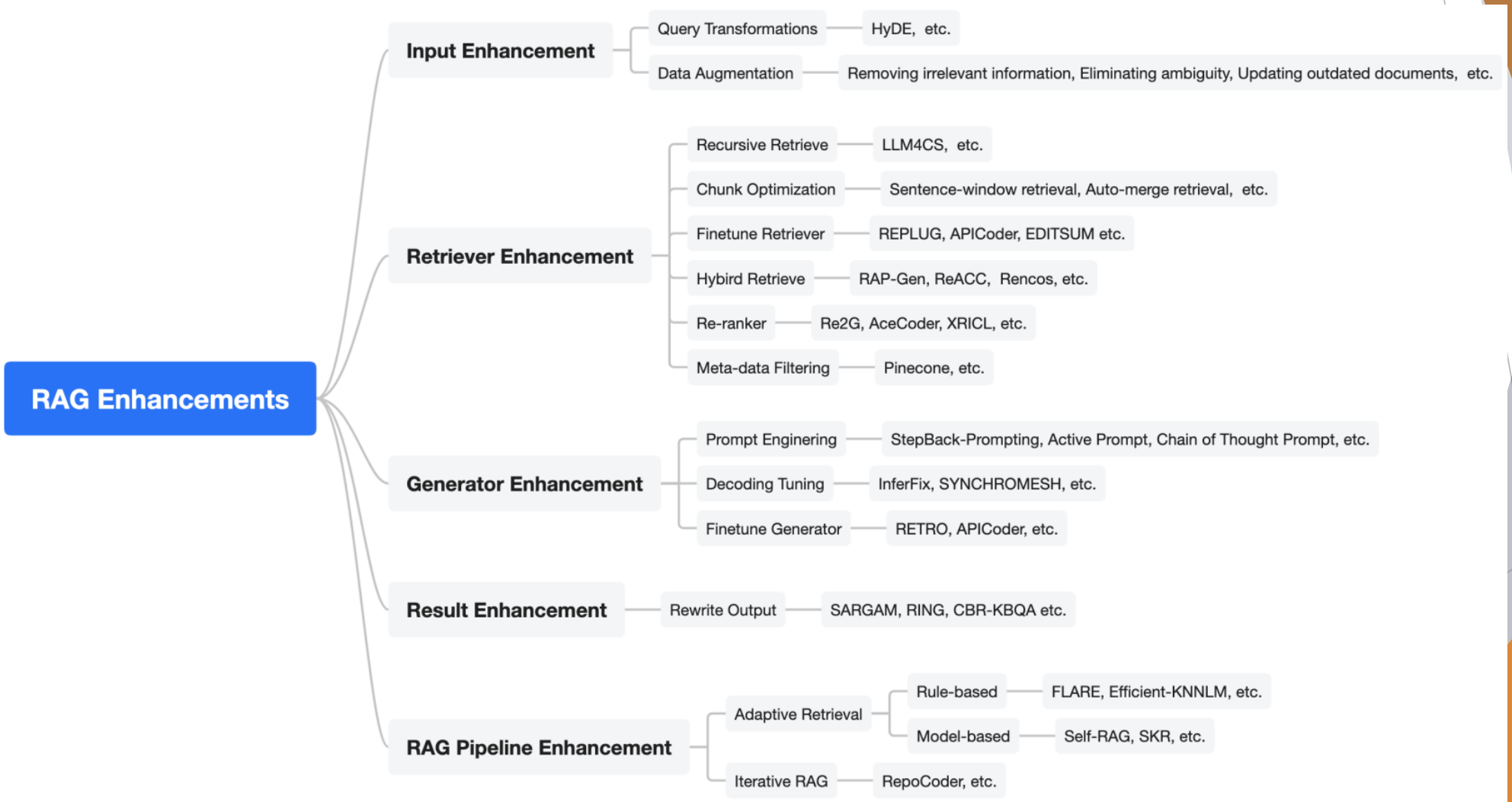
RAG Architecture



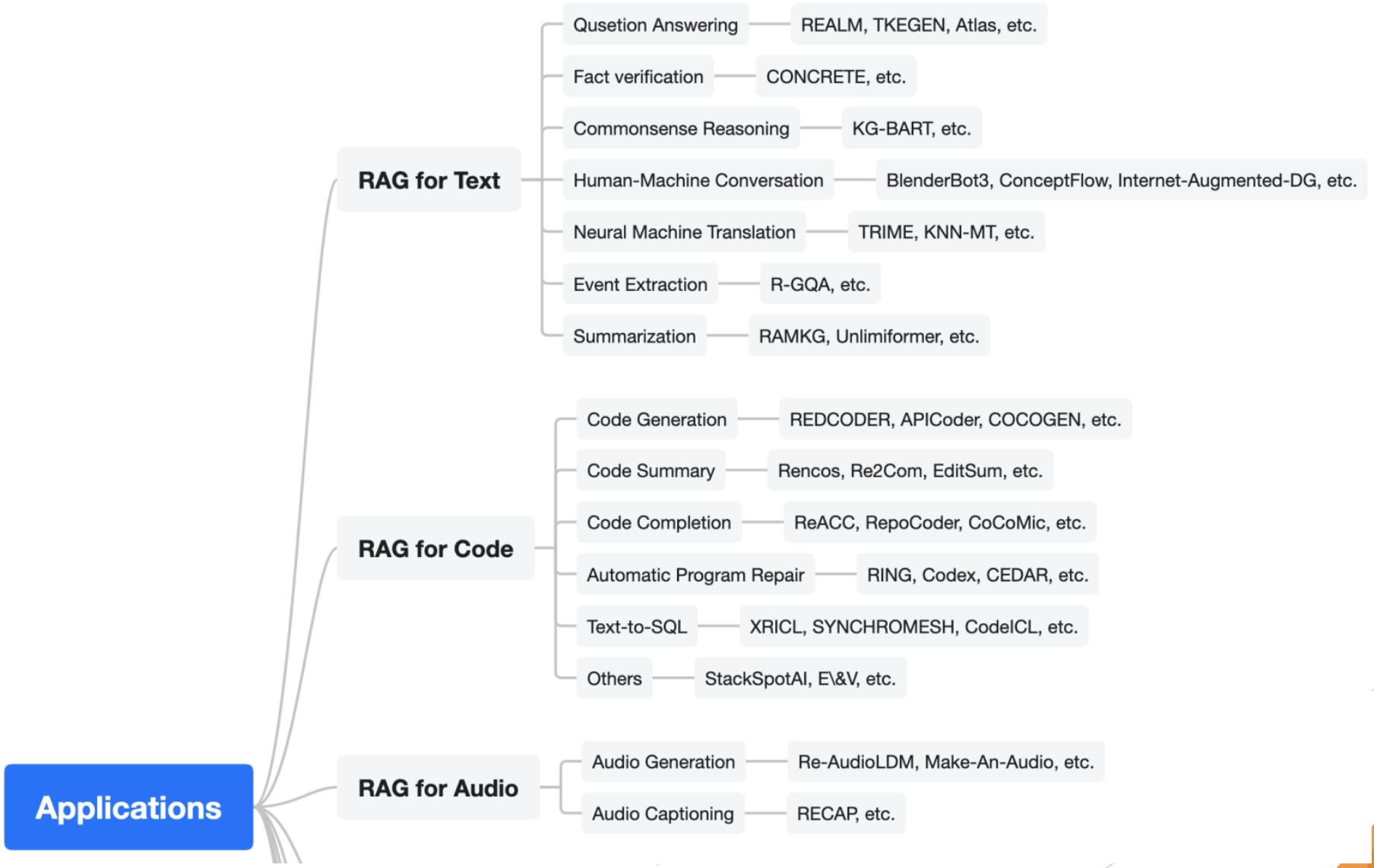
Taxonomy of RAG Foundations



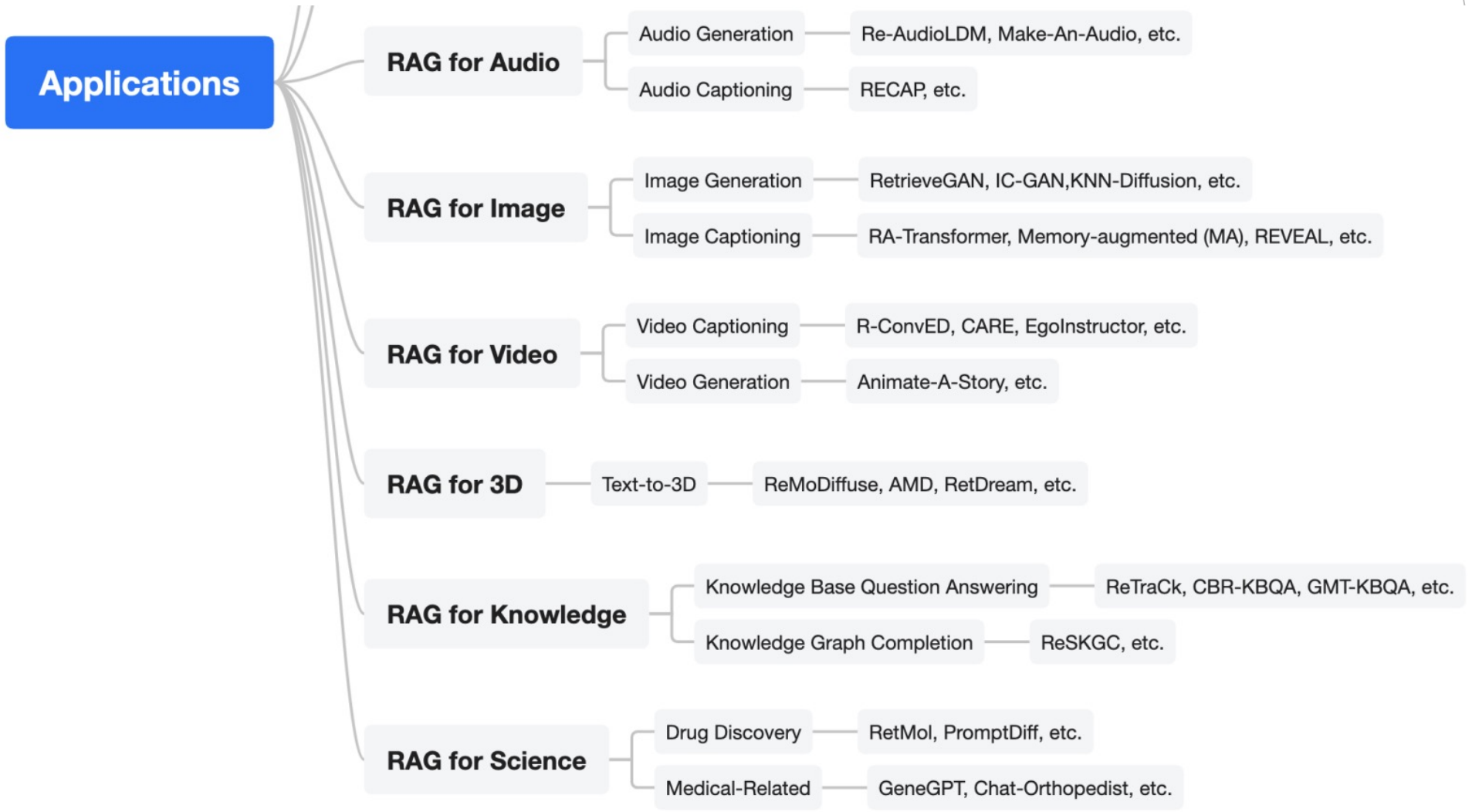
Taxonomy of RAG Enhancements



Taxonomy of RAG Applications



Taxonomy of RAG Applications



A Survey of Table Reasoning with Large Language Models

Xuanliang Zhang, Dingzirui Wang, Longxu Dou, Qingfu Zhu, Wanxiang Che

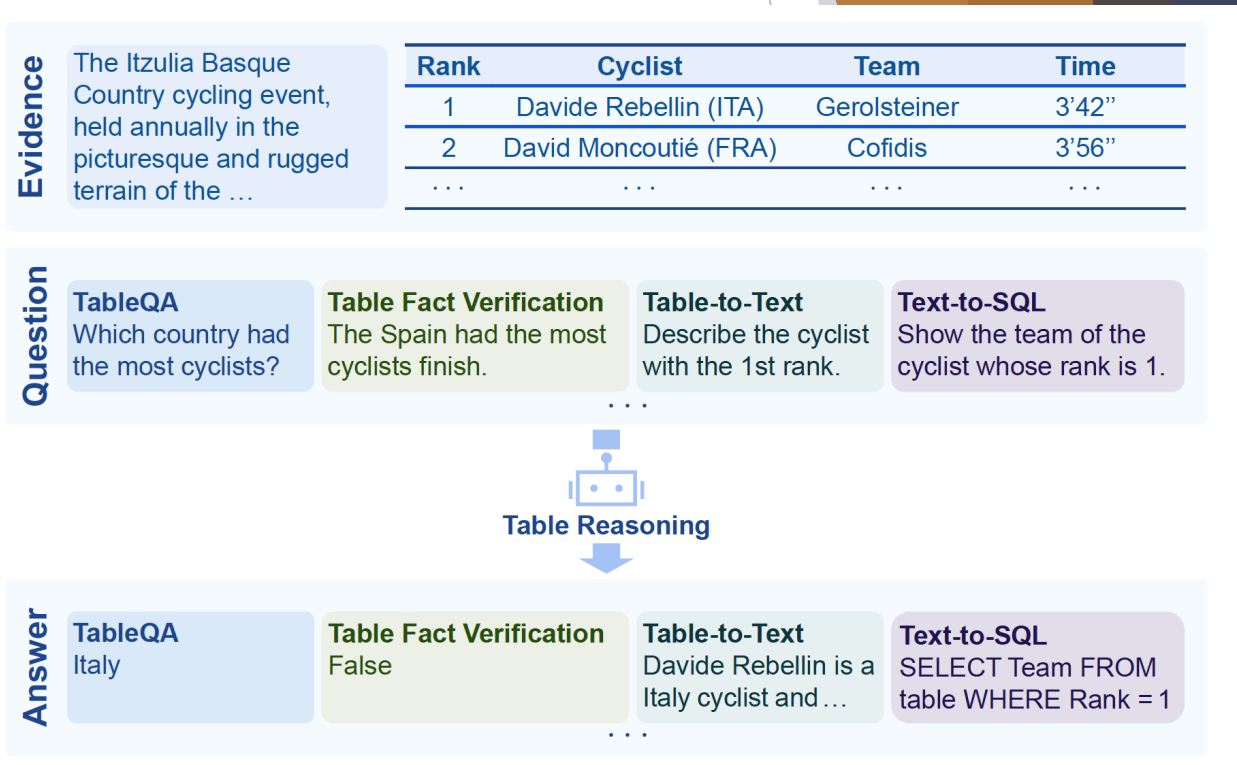
Harbin Institute of Technology, China

Presenter: Shiyu Feng, eus5fy

University of Virginia | ENGINEERING Department of Computer Science

Introduction to Table Reasoning

- Table reasoning aims to generate accurate answers from tables based on users requirements
- Table reasoning task improves the efficiency of obtaining and processing data from massive amounts of tables



University of Virginia | ENGINEERING
Department of Computer Science

The Rise of LLMs and their Advantages

- Traditional methods relied on rule-based systems or neural networks
- With LLMs' vast knowledge and language understanding capabilities, LLMs excel at table reasoning

Key Advantages of LLMs in Table Reasoning:

- Instruction following ability benefits structure understanding
- Step-by-step reasoning capability benefits schema linking
- Reduced annotation requirements

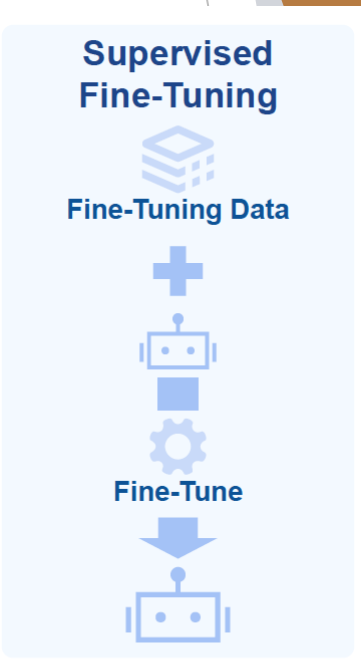
Techniques for Improving Performance in LLM era

- Supervised Fine-Tuning
- Result Ensemble
- In-Context Learning
- Instruction Design
- Step-by-Step Reasoning

Supervised Fine-tuning

- Fine-tuning LLMs on annotated data to enhance reasoning capabilities
  - Using pre-existing datasets or manually labeled data
  - Leveraging distilled data generated by other LLMs

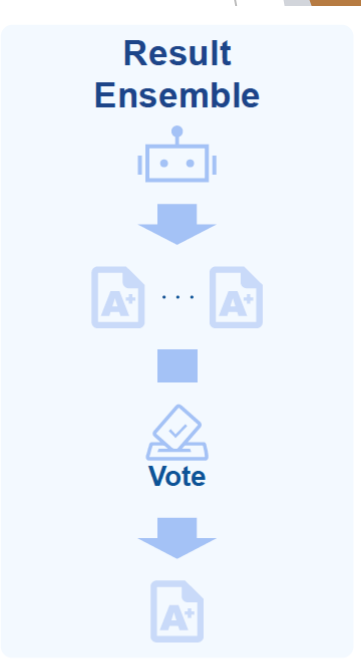
- In the LLM era, instruction-based and multi-task data fine-tune models for better generalization



Result Ensemble

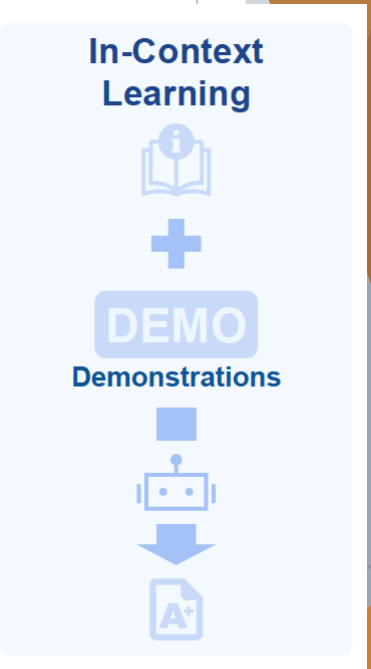
- Obtaining **diverse results** by varying prompts, models, or random seeds
- **Selecting the most suitable answer** through scoring or verification

- Compared to pre-LLM methods, LLMs can generate diverse results more effectively, often by simply changing instructions, unlike pre-LLM methods requiring aligned fine-tuning and inference instructions.



In-context Learning

- Leveraging LLMs' ability to generate expected answers using suitable prompts
- In-context learning capability of LLMs allows flexible adjustment of prompts suitable for different questions without further fine-tuning
- Reduces labeling overhead while enhancing performance



Example:ODIS

- ODIS
  - Ontology-Guided Domain-Informed Sequencing
  - using in-domain demonstrations to enhance model performance by synthesizing in-domain SQL based on SQL similarity

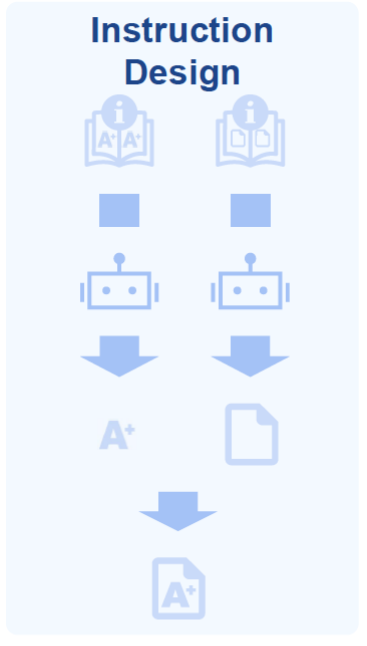
```
CreateTable+SelectCol(concert_singer)

-- Using valid SQLite, answer the following
 questions for the tables provided above.
Question: what is the name and nation of the singer
who have a song having 'Hey' in its name?
select name, country from singer where song_name like
'Hey';
Question: How many concerts are there in year 2014 or
2015?
select count(*) from concert where year = 2014 or
year = 2015;
Question: Which year has most number of
concerts?
select
```

An example prompt of 2-shot in-domain text-to-SQL
Two in-domain demonstrations are present prior to the test question

Instruction Design

- Utilizing LLMs' instruction following ability
- Instruction design involves instructing LLMs to complete decomposed sub-tasks for table reasoning.
  - Modular decomposition: Breaking tasks into sub-tasks (DATER)



Example: DATER(Decompose evidence And questions for effective Table-basEd Reasoning)

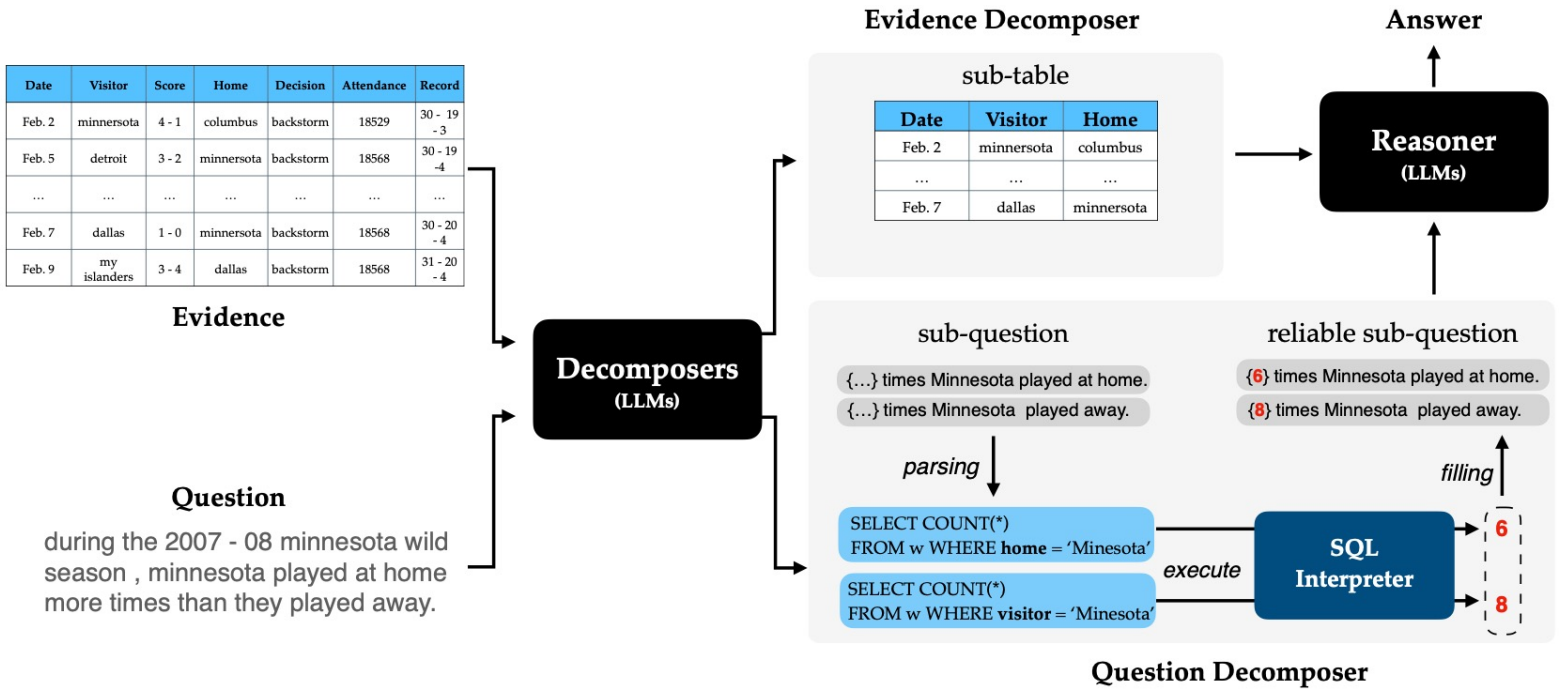
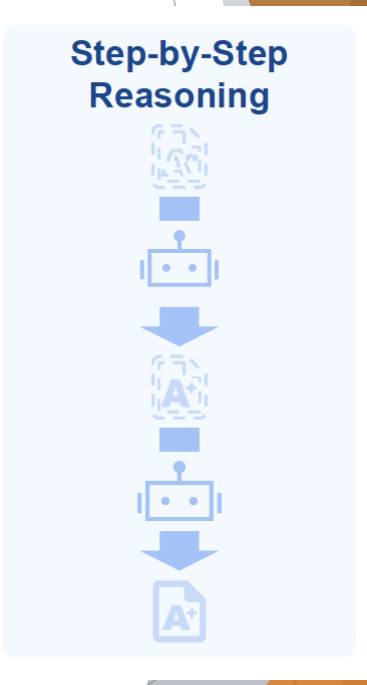


Figure 2: The overview of our Dater framework for table-based reasoning. We first use a powerful LLM (Codex) to probe sub-evidence and sub-questions by performing in-context learning. To obtain a reliable sub-question, we propose a novel "parsing-execution-filling" strategy to alleviate hallucination issues. Ultimately, the reasoner browses through the sub-evidence and sub-questions to get the final answer.

Step-by-step Reasoning

- Solving complex tasks by incorporating intermediate reasoning stages
  - Techniques like Chain-of-Table
  - Decomposing questions into simpler sub-questions or predefined operations
  - Differs from modular decomposition which breaks tasks into widely different sub-tasks.



Example: Chain-of-Table

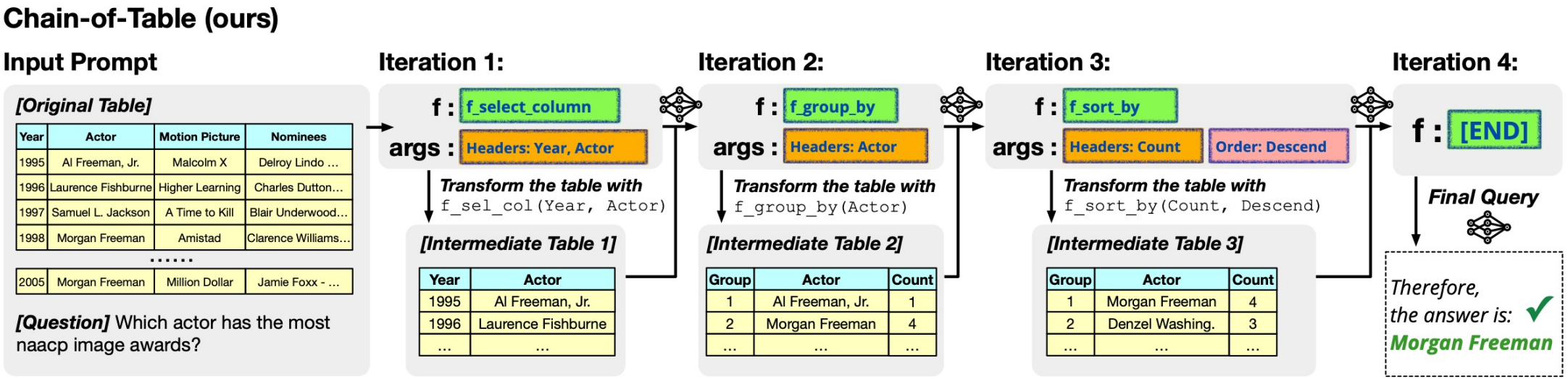


Figure 4: Illustration of the tabular reasoning process in CHAIN-OF-TABLE. This iterative process involves dynamically planning an operation chain and accurately storing intermediate results in the transformed tables. These intermediate tables serve as tabular thought process that can guide the LLM to land to the correct answer more reliably.

Future Research Directions

Improving Table Reasoning Performance

- Supervised Fine-Tuning: Establishing Diverse Training Data
- Result Ensemble: Sampling Results More Efficiently
- In-Context Learning: Optimizing Prompts Automatically
- Instruction Design: Automatically Refining Design with Verification
- Step-by-Step Reasoning: Mitigating Error Cascade in Multi-Step Reasoning

Future Research Directions

Expanding Practical Applications

- Multi-Modal: Enhancing Alignment between Image Tables and Questions
- Agent: Cooperating with More Diverse and Suitable Table Agents
- Dialogue: Backtracking Sub-tables in Multi-turn Interaction
- Retrieval-Augmented Generation: Injecting Knowledge Related to Entities

Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models

Liu et. al, 2024

Lehigh University, Microsoft Research

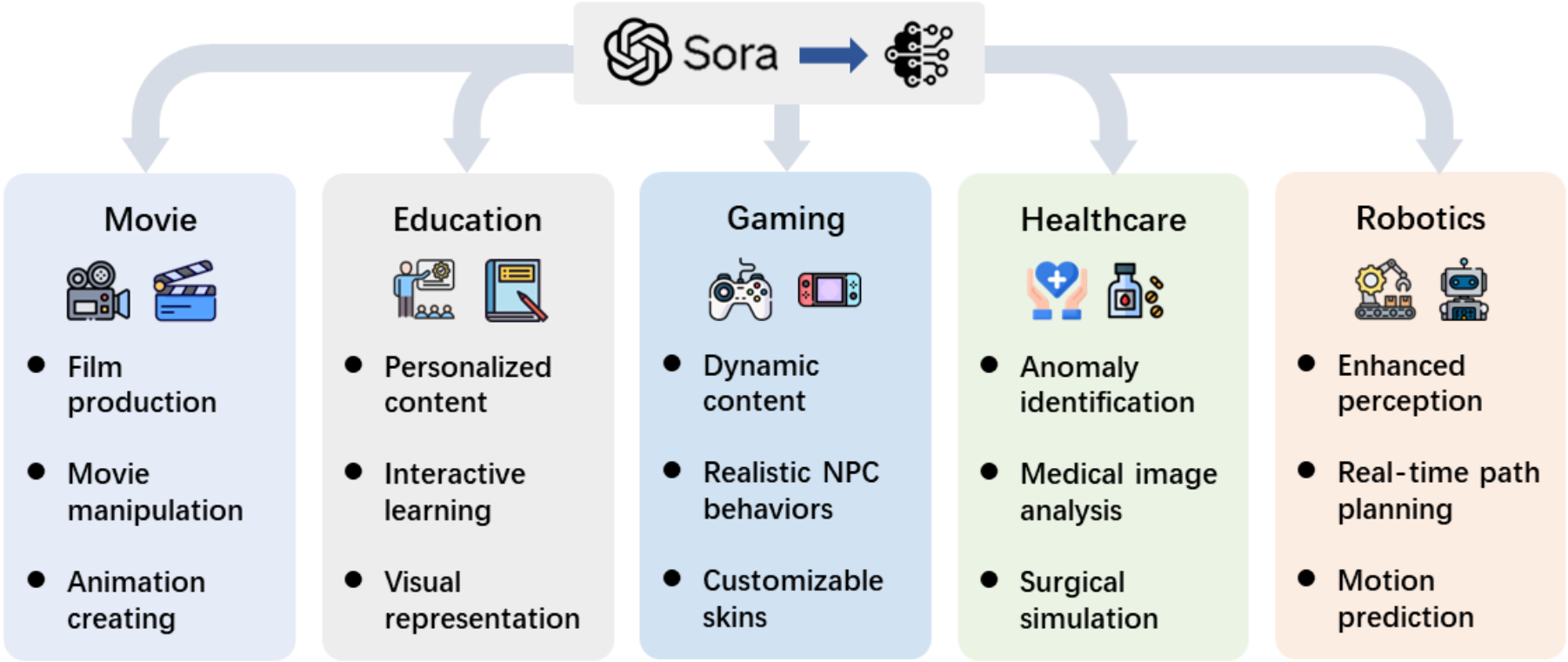
Presenter: Amir Shariatmadari (ahs5ce)

University of Virginia | ENGINEERING
Department of Computer Science

What is SORA?

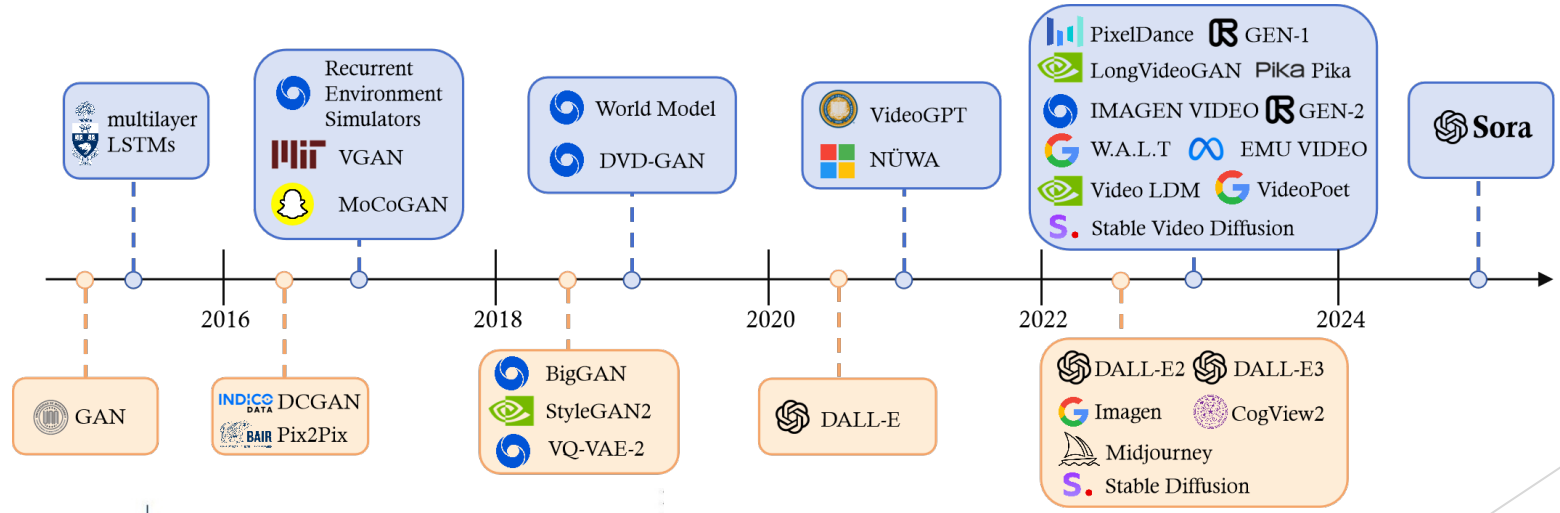
- Prompt: Historical footage of California during the gold rush.
- Generated Video:

What can Sora be used for?

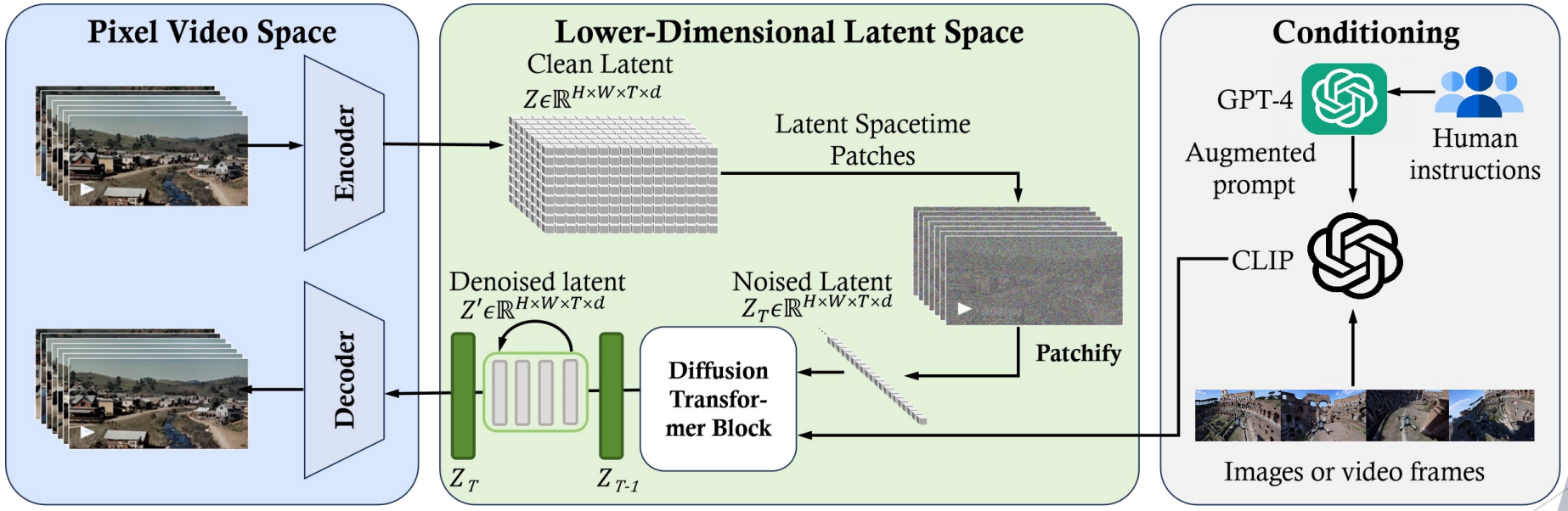


History of Generative Video

- GANs and VAEs were a turning point for generative AI for vision domain.
- Vision Transformer (ViT) and Swin Transformer proved successful (NLP success applied to vision domain)
- Diffusion Models
  - Mathematically based method of converting noise into images with U-Nets



How does Sora work? (High level overview)



How does Sora work? (High level overview)

- Sora essentially is a diffusion transformer that can deal with varying dimension image/video size.
- (1) Time-space compressor projects the original video into a latent space.
- (2) ViT inputs the tokenized latent representation and outputs a denoised latent representation.
- (3) The conditioning mechanism inputs user instructions augmented by LLMs or visual prompts to guide diffusion model to generated themed/styled videos.

Data Preprocessing: Variable Durations, Resolutions, Aspect Ratios

- Sora is able to deal with varying video duration, resolutions, and aspect ratios.

(a) Vertical

(b) Square

(c) Horizontal

Data Preprocessing: Variable Durations, Resolutions, Aspect Ratios

- Training on native sizes improves composition and framing in generative video.
  - More natural looking and coherent video
- Richard Sutton's THE BITTER LESSON:
  - Computation > human designed features.

(a) Training on videos that are cropped to squares leads to unnatural compositions and framing.

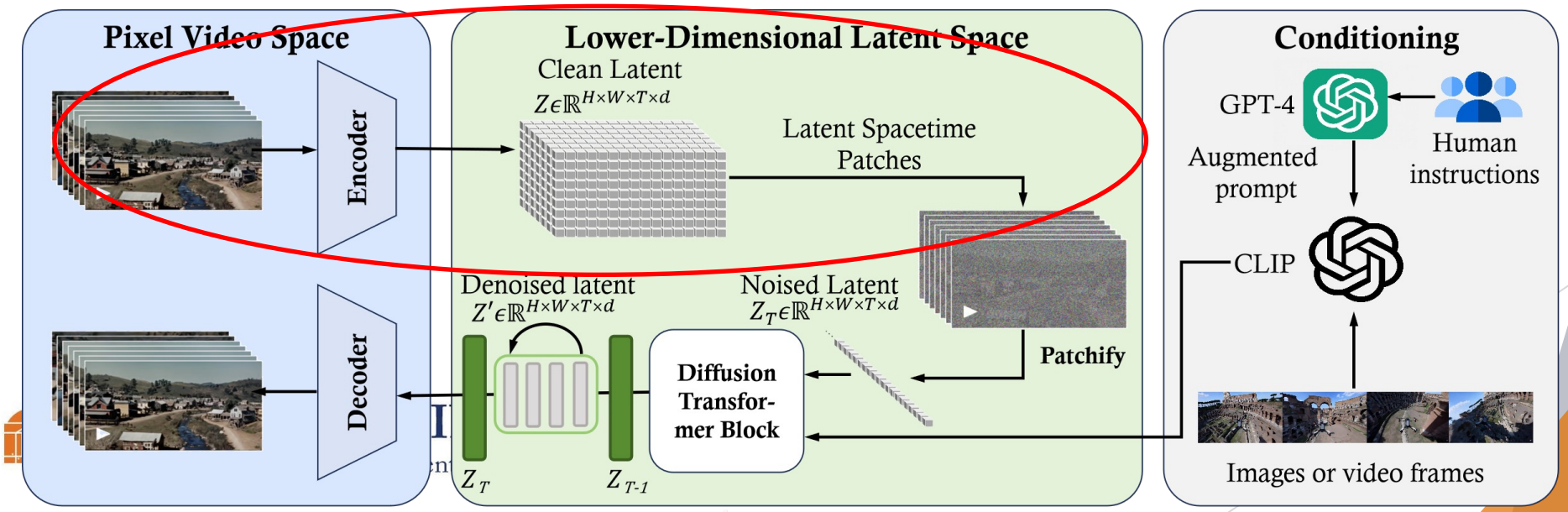
(b) Training in native sizes improves framing.

Data Preprocessing: Unified Visual Representation

- How is Sora able to deal with diverse visual inputs?
- All visual data is transformed into a unified representation.
- Open AI report discusses this broadly, so the authors attempt to reverse engineer this process to explain how it works.

Data Preprocessing:Video Compression Network

- Video compression network's objective is to reduce the dimensionality of the raw data to produce a lower dimensional latent representation.
- Sora either uses VAE or Vector Quantized-VAE
- Using a VAE to map raw data into a fix-sized latent space without resizing.
- Authors show two different implementations to do this.



Video Compression Network: Spatial-patch compression

- Splits video frames into fixed-size patches before encoding.
- Spatial tokens are organized temporally to create a spatial-temporal latent space.
- Temporal dimension cannot be fixed so only a specific number of frames can be sampled or a long universally extended input length must be defined for subsequent processing.

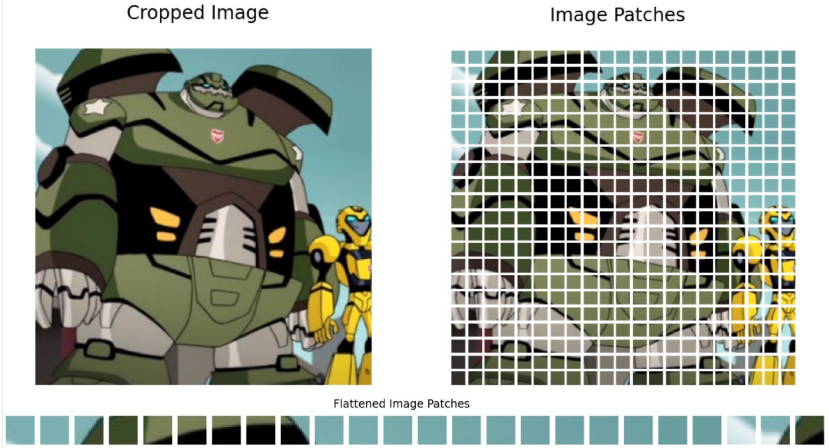


Figure 8: ViT splits an image into fixed-size patches, linearly embeds each of them, adds position embeddings, and feeds the resulting sequence of vectors to a standard Transformer encoder.

Video Compression Network: Spatial-patch compression

- Considers spatial and temporal dimensions of data.
- Instead of statically considering frames, it captures changes across frames.

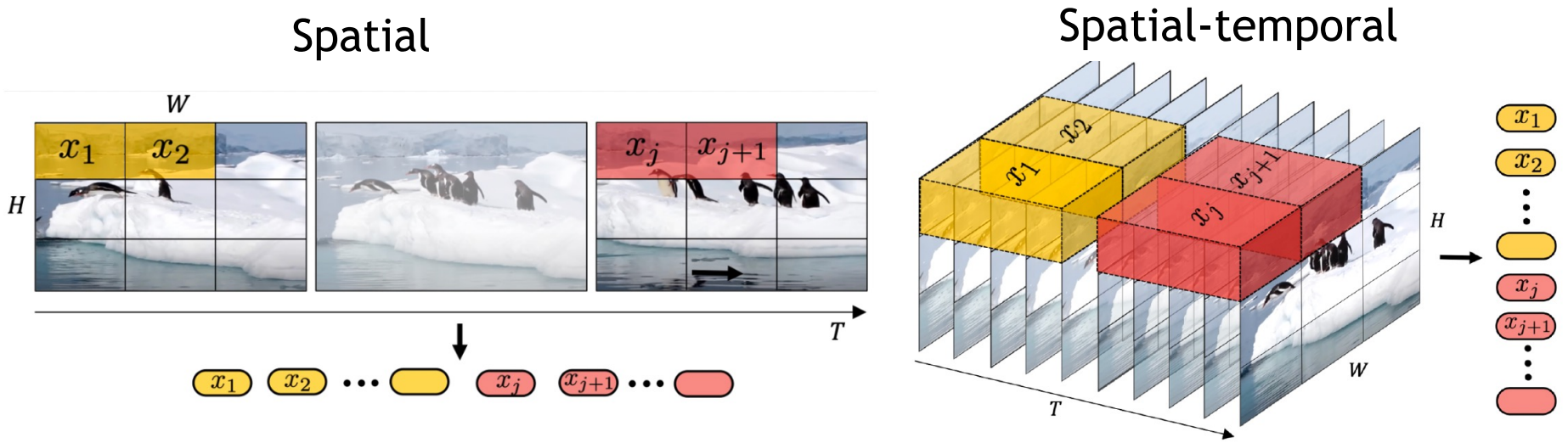
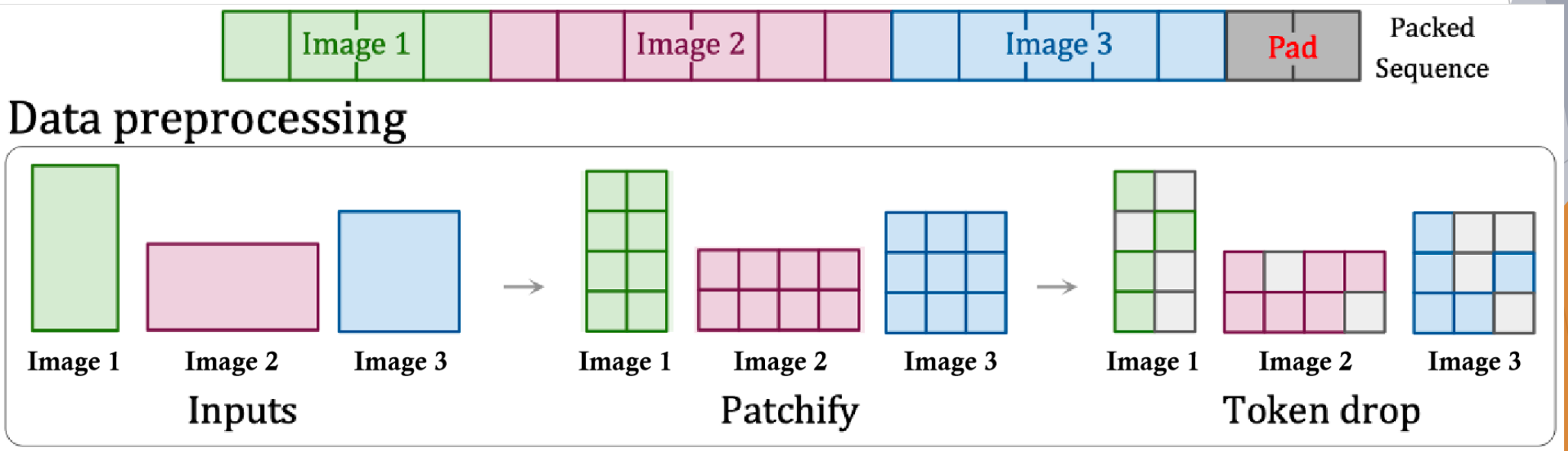


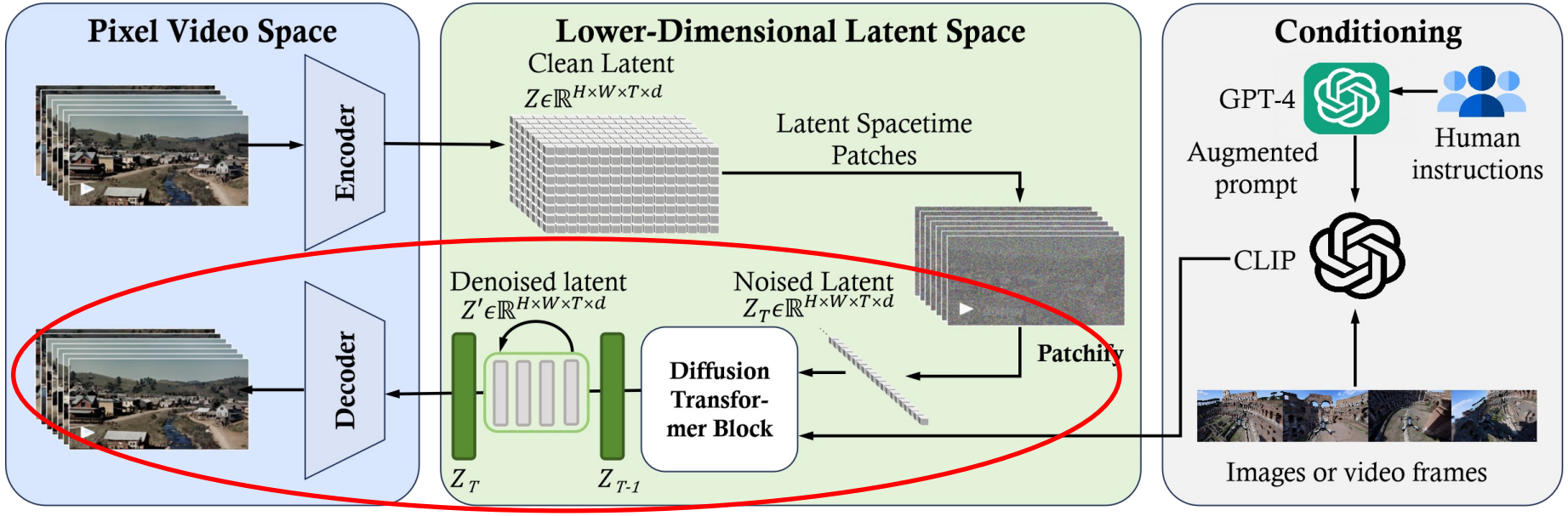
Figure 9: Comparison between different patchification for video compression. Source: ViViT [38]. (**Left**) Spatial patchification simply samples n_t frames and embeds each 2D frame independently following ViT. (**Right**) Spatial-temporal patchification extracts and linearly embeds non-overlapping or overlapping tubelets that span the spatiotemporal input volume.

Data Preprocessing: Spacetime Latent Patches

- How to handle variability in latent space dimensions?
  - Latent space dimensions are features or chunks of features in the latent space from different video types.
- Patch n' pack is likely used to handle this issue

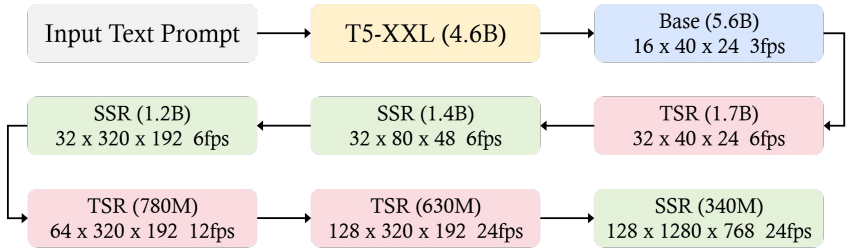


How does Sora model video generation?

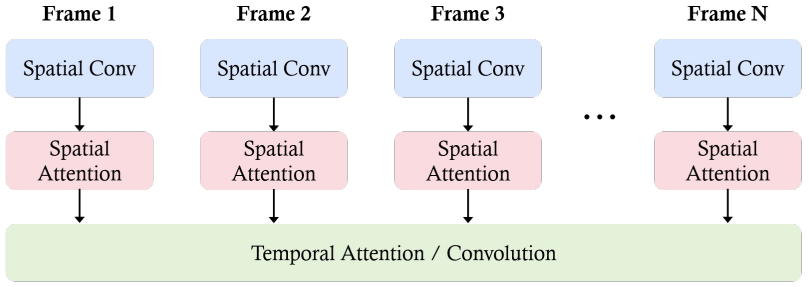


Modeling: Video Diffusion Transformer

- Imagen Video – Video generation model by Google
  - Text prompt encoded into contextual embeddings by a frozen T5 model
  - Contextual embeddings fed into a base model that generates a low-res video.
  - Cascading diffusion models progressively refine the video:
    - Each of these models has a specific task: improving resolution, enhancing spatial details, and ensuring temporal coherence, among others.



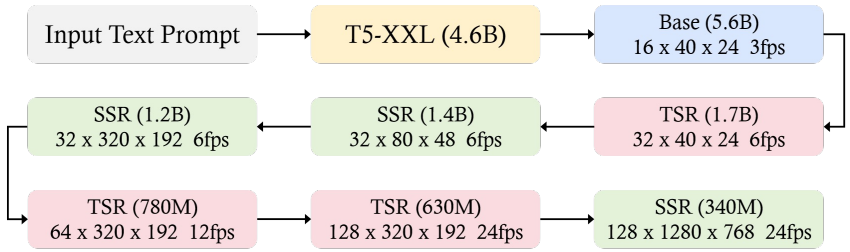
(a) **Cascaded diffusion models.** The cascaded sampling pipeline with a base diffusion model and six up-sampling models that operate spatially and temporally. The text embeddings are injected into all the diffusion models.



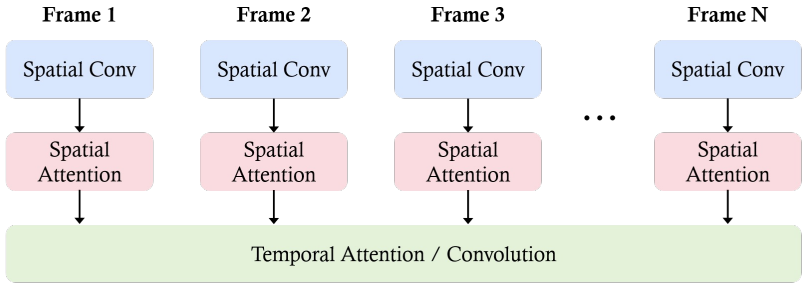
(b) **Video U-Net space-time separable block.** Spatial operations are performed independently over frames with shared parameters, whereas the temporal operation mixes activations over frames. Temporal attention is only used in the base model for memory efficiency.

Modeling: Video Diffusion Transformer

- Imagen architecture utilizes 3D U-Net architecture with temporal attention mechanisms and convolution layers to maintain the consistency and flow between frames.
- U-Net is not necessary for performance of traditional diffusion architecture.
- Adopting transformer instead of U-net is more flexible since it can allow for more training data and more model parameters.

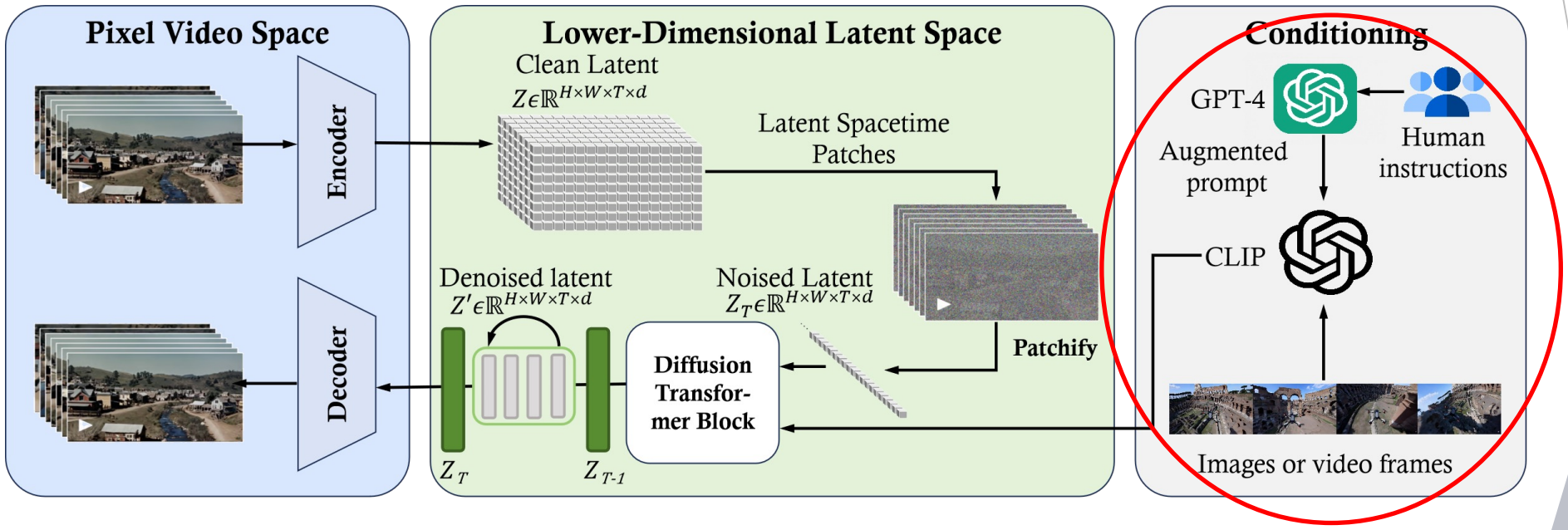


(a) **Cascaded diffusion models.** The cascaded sampling pipeline with a base diffusion model and six up-sampling models that operate spatially and temporally. The text embeddings are injected into all the diffusion models.



(b) **Video U-Net space-time separable block.** Spatial operations are performed independently over frames with shared parameters, whereas the temporal operation mixes activations over frames. Temporal attention is only used in the base model for memory efficiency.

How does Sora follow user instructions?



How does Sora follow user instructions?

- DALLE-3 uses Contrastive Captioners (CoCa) to train an image captioner with CLIP jointly with a language model objective.
- Mismatch between user prompts and image descriptions pose a problem.
  - LLMs are used rewrite descriptions into long descriptions.
- Similar to DALLE-3, Sora uses video captioners to trained to create detailed descriptions for videos.
  - Little description
  - Likely uses VideoCoCa, which is build on top of CoCa.

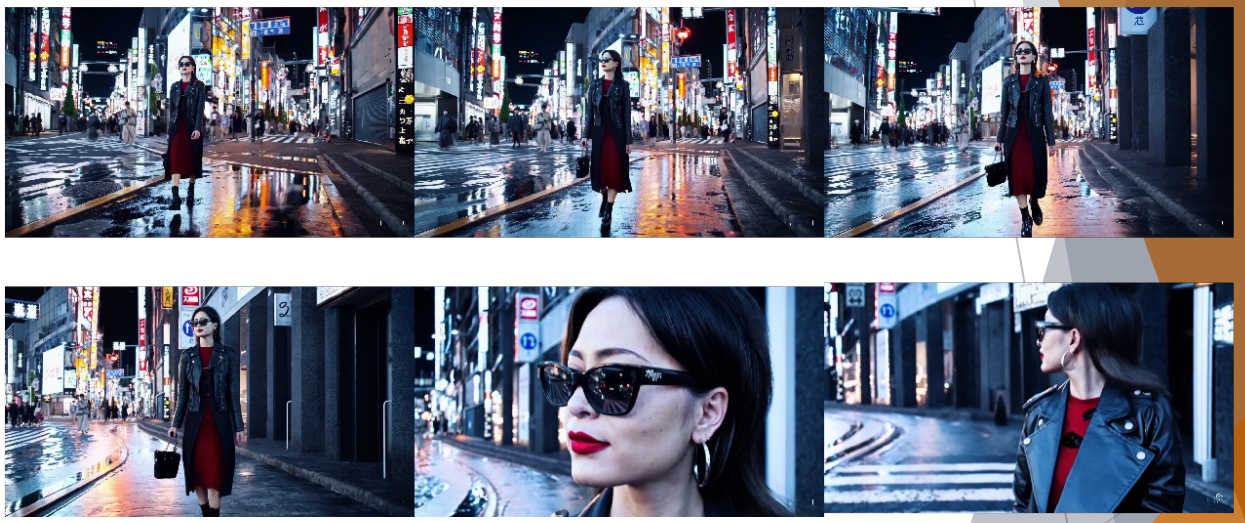
Prompt Engineering: Text Prompt

Text Prompt

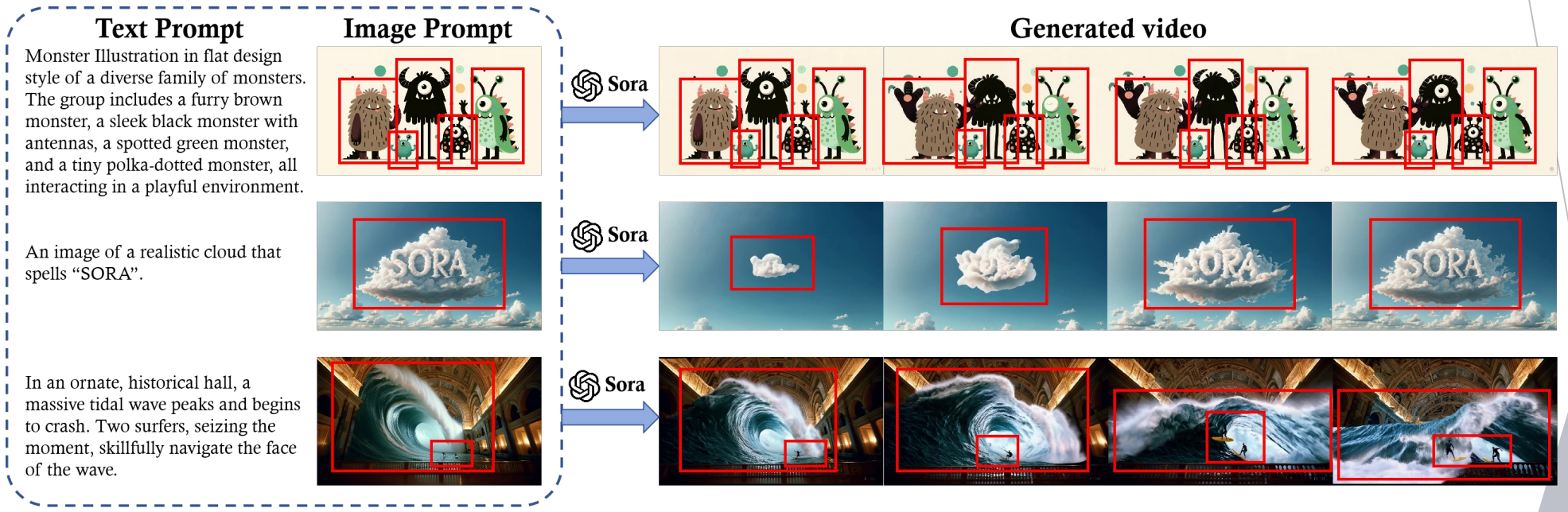
A stylish woman walks down a Tokyo street filled with warm glowing neon andanimated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights.Many pedestrians walk about.

Sora

Generated video

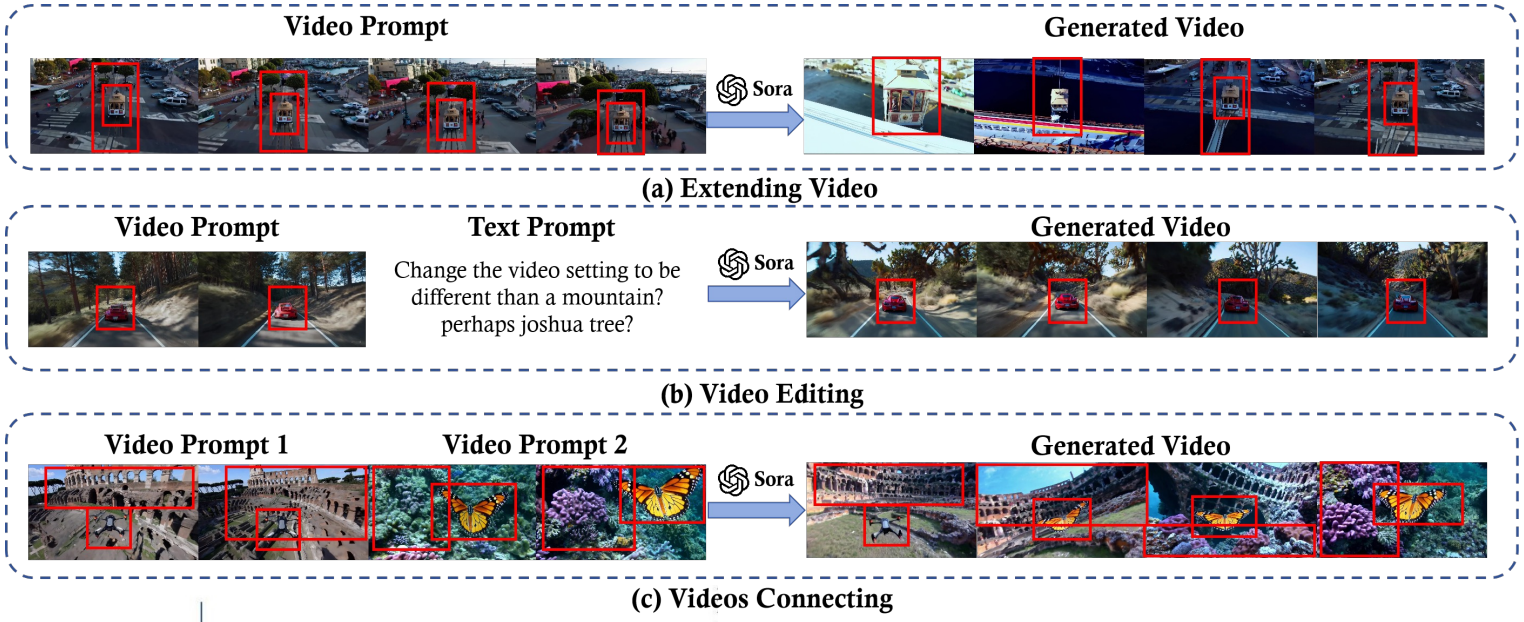


Prompt Engineering: Image Prompt



Prompt Engineering: Video Prompt

- Moonshot and Fast-Vid2Vid demonstrate that a good video prompt requires being specific and flexible so that the model gets a clear direction and objectives.



Sora's Limitations

- Lacks in physical realism, especially complex scenarios
- Spatial and temporal misunderstandings
- Limits in Human-computer interaction
- Usage limitation

Sora's Trustworthiness

- Safety Concern
  - Large multi-modal models are vulnerable to adversarial attacks due to their high dimensional nature and ability to take visual input.
- Hallucination is a problem.
- Fairness and Bias
  - How to mitigate bias in Sora from training data and make the model operate fairly?
- Privacy preservation
  - Can Sora protect user data?

Sora's Trustworthiness (continued)

- Alignment
  - It is important to ensure human intentions and model behavior are aligned.
  - RLHF used in LLMs, what will be done for Sora?
- Recommendations for Future works:
  - 'Integrated Protection of Model and External Security'
  - 'Security Challenges of Multimodal Models'
  - 'The Need for Interdisciplinary Collaboration'

A Comprehensive Study of Knowledge Editing for Large Language Models

Zhang et al. 2024

Presenter: Sabit Ahmed, bcw3zj

UNIVERSITY of VIRGINIA | ENGINEERING
Department of Computer Science

Motivation

Understanding knowledge structures in LLMs and enable knowledge editing to update outdated or harmful knowledge

Outline of this study

- Define knowledge editing problem
- Comprehensive review of existing techniques
- Propose a unified categorization criterion
- A new benchmark (**KnowEdit)** for evaluation of knowledge editing methods
- Analysis of knowledge location
- Open-source framework: **EasyEdit** for knowledge editing

UNIVERSITY of VIRGINIA | ENGINEERING
Department of Computer Science

Background

Transformers

- Encoder-decoder framework
- Equipped with self-attention module and feed-forward network
- Incorporated cross-attention layer along with self-attention in decoder

Self-Attention

$$H = \text{ATT}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V.$$

Feed-Forward Neural Net (FFN)

$$\text{FFN}(\mathbf{x}) = \text{ReLU}(\mathbf{x} \cdot W_1 + b_1) \cdot W_2 + b_2,$$

Knowledge Storing

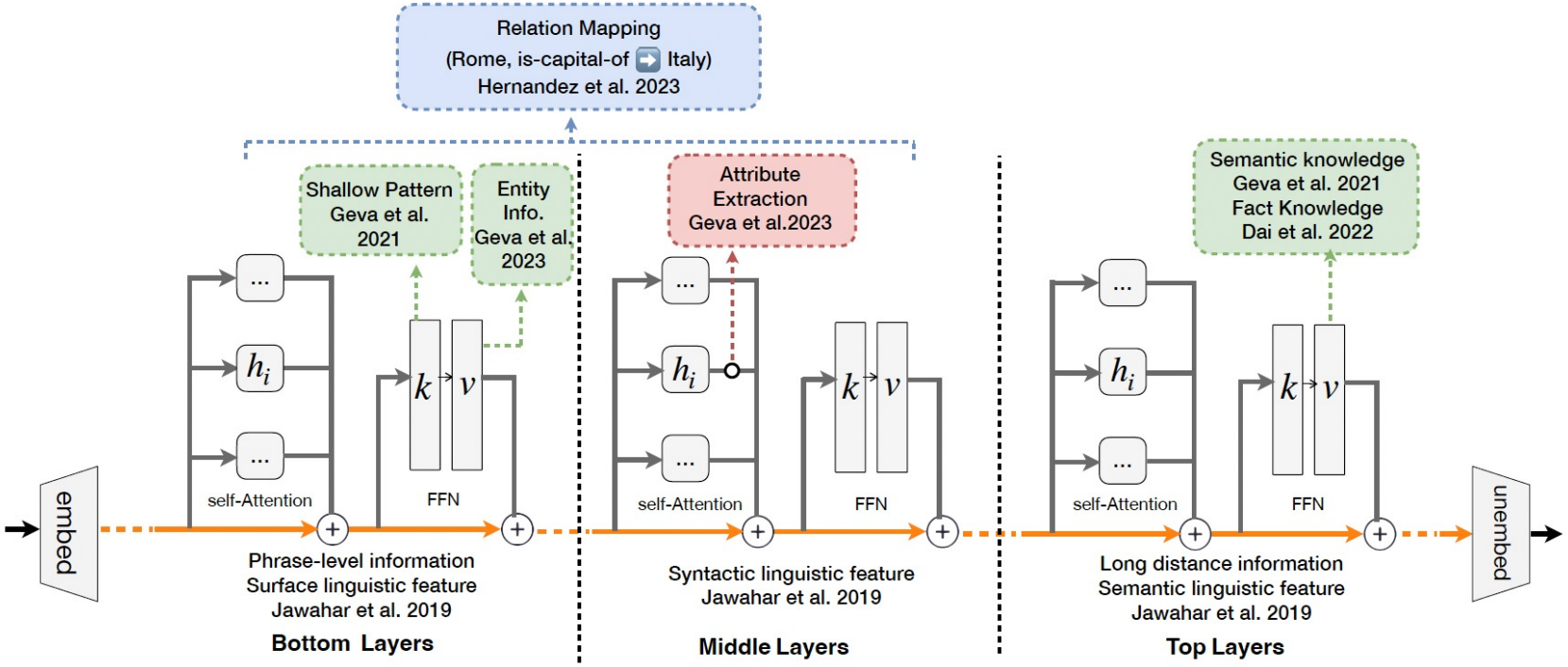


Figure 1: The mechanism of knowledge storage in LLMs. Here, we summarize the findings of current works, including: Jawahar et al. [78], Geva et al. [41], Dai et al. [39], Meng et al. [79], and Hernandez et al. [80].

Related Works

- Parameter-efficient Fine-tuning

  Obtaining the desired performance while updating a minimal no. of parameters

Related Works

- Parameter-efficient Fine-tuning

  Obtaining the desired performance while updating a minimal no. of parameters

- Knowledge Augmentation for LLMs

  Retrieving relevant knowledge from external sources

Related Works

- Parameter-efficient Fine-tuning

  Obtaining the desired performance while updating a minimal no. of parameters

- Knowledge Augmentation for LLMs

  Retrieving relevant knowledge from external sources

- Continual Learning

  Also known as incremental learning. Focuses on learning new information without forgetting older ones

Related Works

- Parameter-efficient Fine-tuning

  Obtaining the desired performance while updating a minimal no. of parameters

- Knowledge Augmentation for LLMs

  Retrieving relevant knowledge from external sources

- Continual Learning

  Also known as incremental learning. Focuses on learning new information without forgetting older ones

- Machine Unlearning

  Discarding undesirable (mis)behaviors

Knowledge Editing

Knowledge Editing

Efficiently modify LLMs' behaviors within specific domains while preserving overall performance across various inputs. For an original model θ, knowledge *k* and knowledge editing function *F*, the post-edited model is defined as,

$$\theta' = F(\theta, k)$$

1. Knowledge Insertion

$$\theta' = F(\theta, \{\emptyset\} \rightarrow \{k\})$$

Knowledge Editing

Knowledge Editing

Efficiently modify LLMs' behaviors within specific domains while preserving overall performance across various inputs. For an original model θ, knowledge *k* and knowledge editing function *F*, the post-edited model is defined as,

$$\theta' = F(\theta, k)$$

1. Knowledge Insertion

$$\theta' = F(\theta, \{\emptyset\} \rightarrow \{k\})$$

2. Knowledge Modification

$$\theta' = F(\theta, \{k\} \rightarrow \{k'\})$$

Knowledge Editing

Knowledge Editing

Efficiently modify LLMs' behaviors within specific domains while preserving overall performance across various inputs. For an original model θ, knowledge *k* and knowledge editing function *F*, the post-edited model is defined as,

$$\theta' = F(\theta, k)$$

1. Knowledge Insertion
$$\theta' = F(\theta, \{\emptyset\} \rightarrow \{k\})$$
2. Knowledge Modification
$$\theta' = F(\theta, \{k\} \rightarrow \{k'\})$$
3. Knowledge Erasure
$$\theta' = F(\theta, \{k\} \rightarrow \{\emptyset\})$$

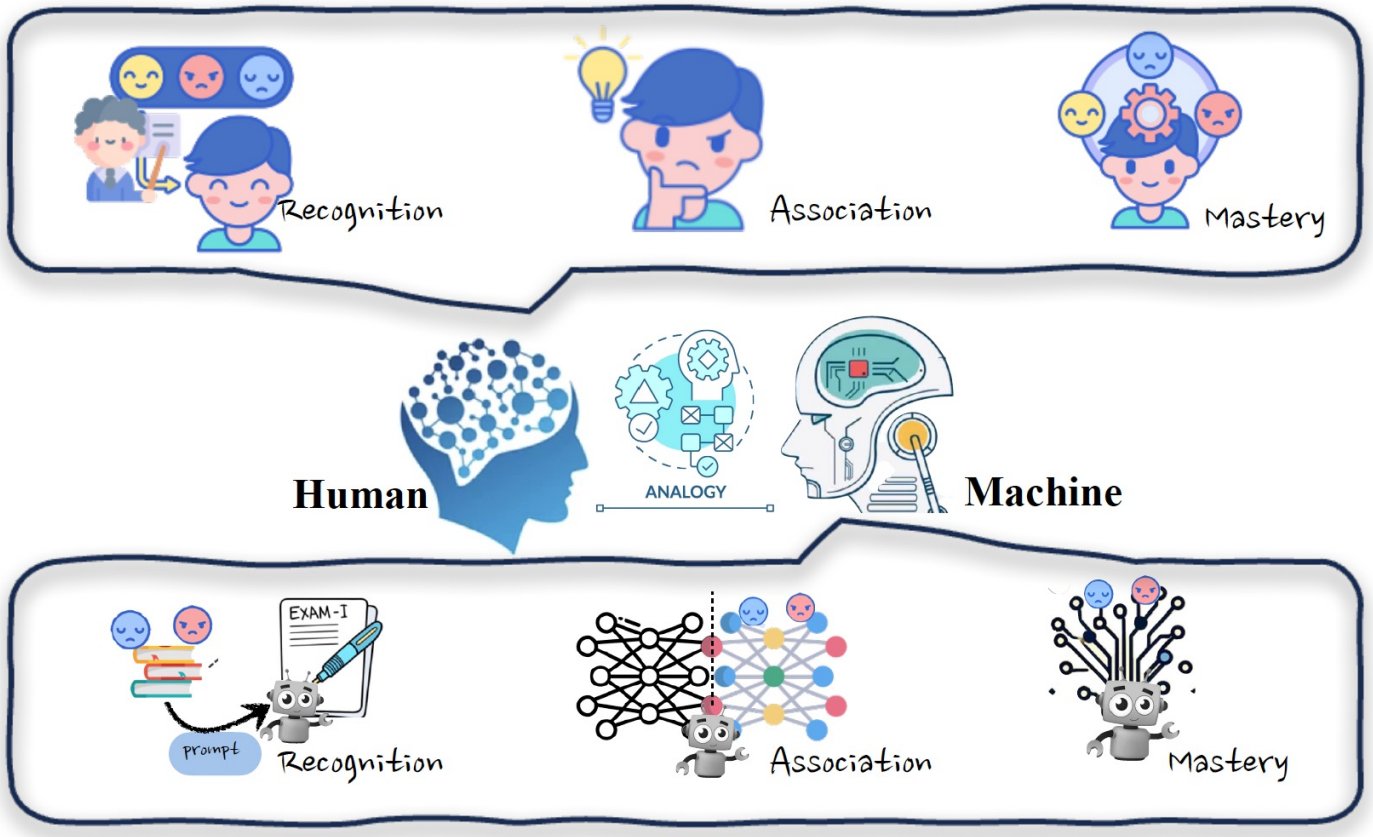


Figure 2: Applying Human Learning Phases [1–3] to Knowledge Editing in LLMs: We see an analogy of Human Learning Phases and Knowledge Editing in LLMs and categorize current knowledge editing methods based on the learning phases of humans: recognition, association, and mastery.

Benchmark Data: KnowEdit

6 datasets on knowledge editing are curated. These encompass a range of editing types, i.e., fact manipulation, sentiment manipulation and hallucination generation

Task	**Knowledge Insertion**	**Knowledge Modification**				**Knowledge Erasure**
Datasets	**WikiData**$_{recent}$	**ZsRE**	**WikiBio**	**WikiData**$_{counterfact}$	**Convsent**	**Sanitation**
Type	Fact	Question Answering	Hallucination	Counterfact	Sentiment	Unwanted Info
# Train	570	10,000	592	1,455	14,390	80
# Test	1,266	1230	1,392	885	800	80

Table 3: Statistics on the benchmark **KnowEdit**, with six selected datasets for the evaluation of knowledge editing methods. We select different knowledge types for the insertion, modification, and erasure settings.

Knowledge Editing Evaluation

- Edit Success

  Also termed as Reliability. It is the average accuracy of the edit cases

Knowledge Editing Evaluation

- Edit Success

  Also termed as Reliability. It is the average accuracy of the edit cases

- Portability

  Whether the edited model can address the effect of an edit

Knowledge Editing Evaluation

- Edit Success

  Also termed as Reliability. It is the average accuracy of the edit cases

- Portability

  Whether the edited model can address the effect of an edit

- Locality

  The edited model should not modify the irrelevant examples in out-of-scopes

Knowledge Editing Evaluation

- Edit Success

  Also termed as Reliability. It is the average accuracy of the edit cases

- Portability

  Whether the edited model can address the effect of an edit

- Locality

  The edited model should not modify the irrelevant examples in out-of-scopes

- Generative Capacity

  Generalization ability of the model after editing. Also, termed ‘fluency’

DataSet	Metric	SERAC	ICE	AdaLoRA	MEND	ROME	MEMIT	FT-L	FT
WikiData$_{recent}$	Edit Succ. ↑	98.68	60.74	65.61	76.88	85.08	85.32	71.18	31.24
	Portability ↑	63.52	36.93	47.22	50.11	37.45	37.94	48.71	15.91
	Locality ↑	100.00	33.34	55.78	92.87	66.2	64.78	63.7	3.65
	Fluency ↑	553.19	531.01	537.51	586.34	574.28	566.66	549.35	428.67
ZsRE	Edit Succ. ↑	99.67	66.01	69.86	96.74	96.57	83.07	54.65	36.88
	Portability ↑	56.48	63.94	52.95	60.41	52.20	51.43	45.02	8.72
	Locality ↑	30.23	23.14	72.21	92.79	27.14	25.46	71.12	0.31
	Fluency ↑	410.89	541.14	532.82	524.33	570.47	559.72	474.18	471.29
WikiBio	Edit Succ.↑	99.69	95.53	97.02	93.66	95.05	94.29	66.27	95.64
	Locality ↑	69.79	47.90	57.87	69.51	46.96	51.56	60.14	13.38
	Fluency ↑	606.95	632.92	615.86	609.39	617.25	616.65	604.00	589.22
WikiData$_{counterfact}$	Edit Succ. ↑	99.99	69.83	72.14	78.82	83.21	83.41	51.12	26.78
	Portability ↑	76.07	45.32	55.17	57.53	38.69	40.09	39.07	16.94
	Locality ↑	98.96	32.38	66.78	94.16	65.4	63.68	62.51	0.29
	Fluency ↑	549.91	547.22	553.85	588.94	578.84	568.58	544.80	483.71
ConvSent	Edit Succ. ↑	62.75	52.78	44.89	50.76	45.79	44.75	49.50	61.93
	Locality ↓	0.26	49.73	0.18	3.42	0.00	0.00	0.00	0.00
	Fluency ↑	458.21	621.45	606.42	379.43	606.32	602.62	607.86	546.24
Sanitation	Edit Succ. ↑	0.00	72.50	2.50	0.00	85.00	48.75	0.00	60.00
	Locality ↑	100.00	56.58	65.50	5.29	50.31	67.47	14.78	42.61
	Fluency ↑	416.29	794.15	330.44	407.18	465.12	466.10	439.10	351.39

Table: Results of 8 different knowledge editing techniques on Llama-27b-chat

Error and Case Analysis

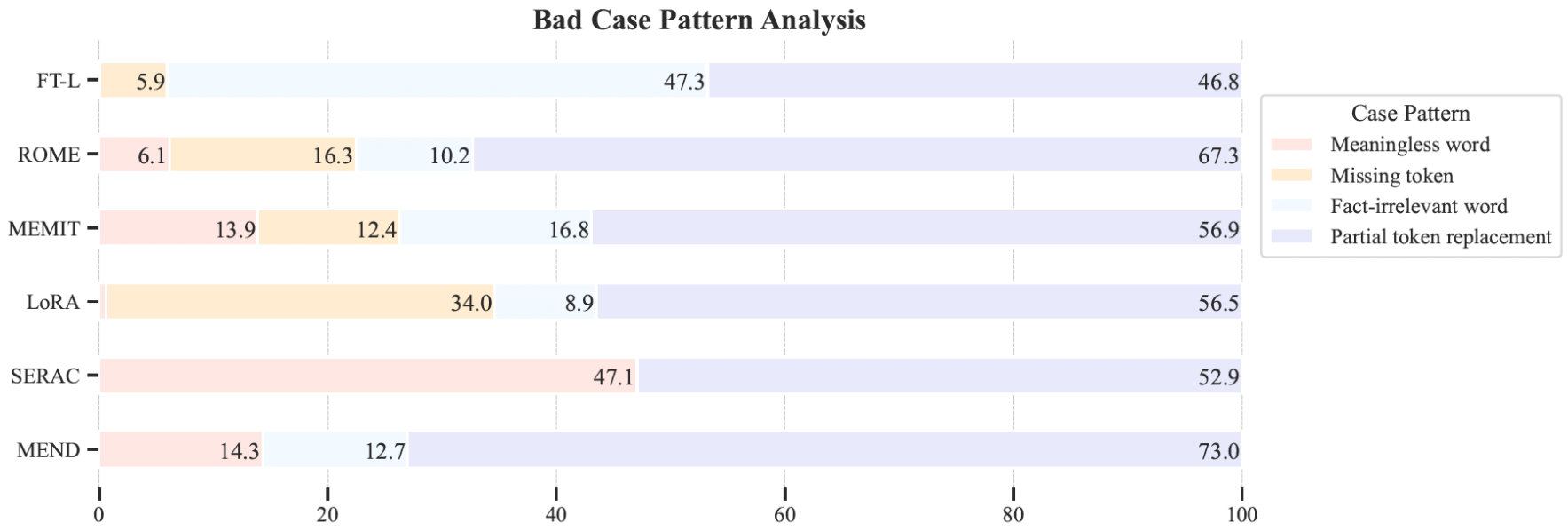


Figure 3: Bad cases statistics for different knowledge editing methods

Limitations of Knowledge Editing

- The underlying mechanism of Transformers is opaque. Therefore, it is unclear whether or not the existing knowledge editing methods are truly successful.
- Defining the boundaries of the influence of knowledge editing is challenging. It was compared with neurosurgery, where the assessment of the impact of any modifications is complex.
- Keeping pace with the dynamic and fluid nature of knowledge.

Thank you!