Machine Learning Systems

LLMs Serving Techniques II

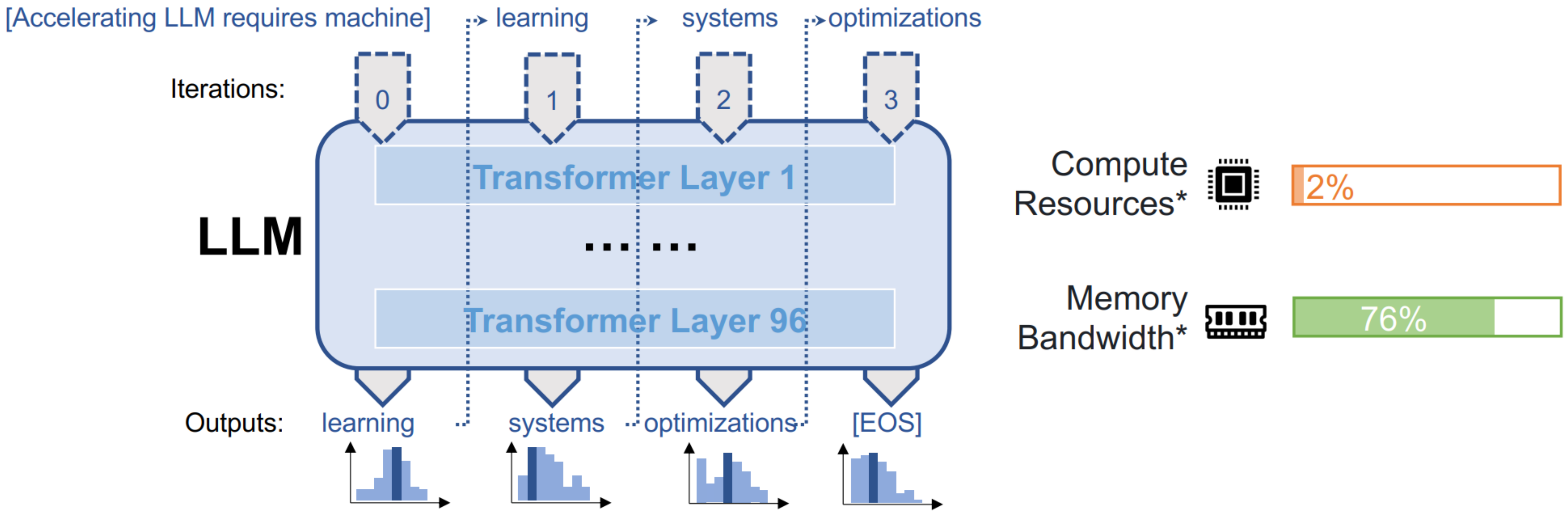
Siyuan Feng
Shanghai Innovation Institute

01

Speculative Decoding

Recall: Incremental Decoding Issues

上海创智学院
Shanghai Innovation Institute



- Limited degree of parallelism → underutilized GPU resources
- Need all parameters to decode a token → bottlenecked by GPU memory access

Measured by serving LLAMA-2-70B on 4 A100 GPUs with 4K sequence length

Tradeoffs between Different Language Models

上海创智学院
Shanghai Innovation Institute

# Parameters	175B	13B	2.7B	760M	125M
TriviaQA	**71.2**	57.5	42.3	26.5	6.96
PIQA	**82.3**	79.9	75.4	72.0	64.3
SQuAD	**64.9**	62.6	50.0	39.2	27.5
latency	20 s	7.6s	2.7s	1.1s	**0.3s**
# A100s	10	1	1	1	**1**

Comparing multiple GPT-3 models*

Large models

- Pro: better generative performance
- Con: slow and expensive to serve

Small models

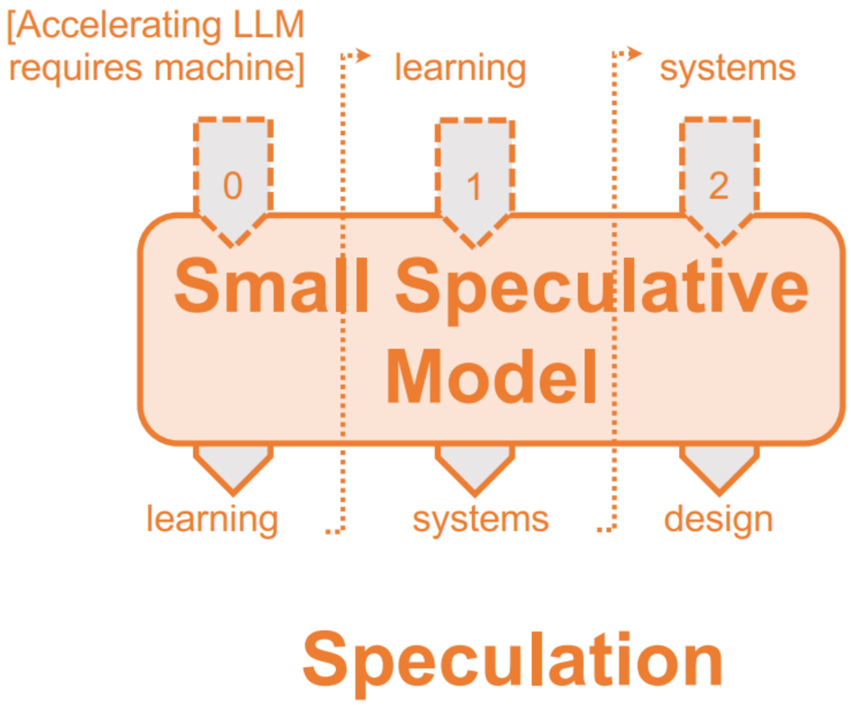
- Pro: cheap and fast
- Con: less accurate

Language Models are Few-Shot Learners. Arix.2005.14165

Speculative Decoding

1. Use a small speculative model (SSM) to predict the LLM's output

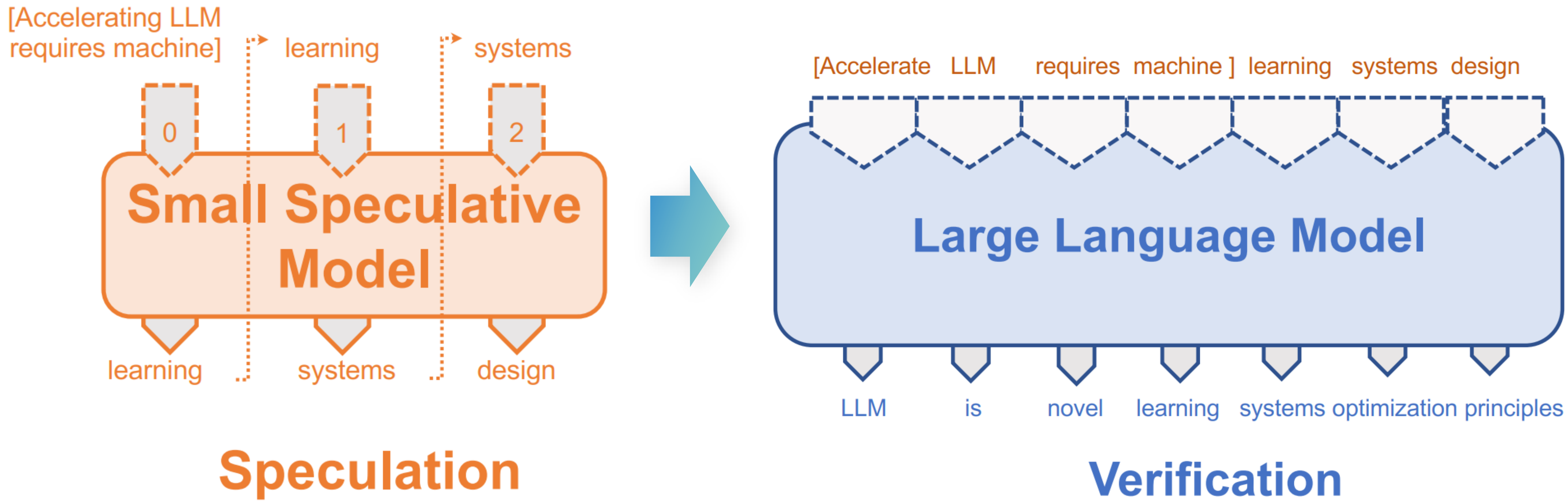
- SSM runs much faster than LLM



Speculative Decoding

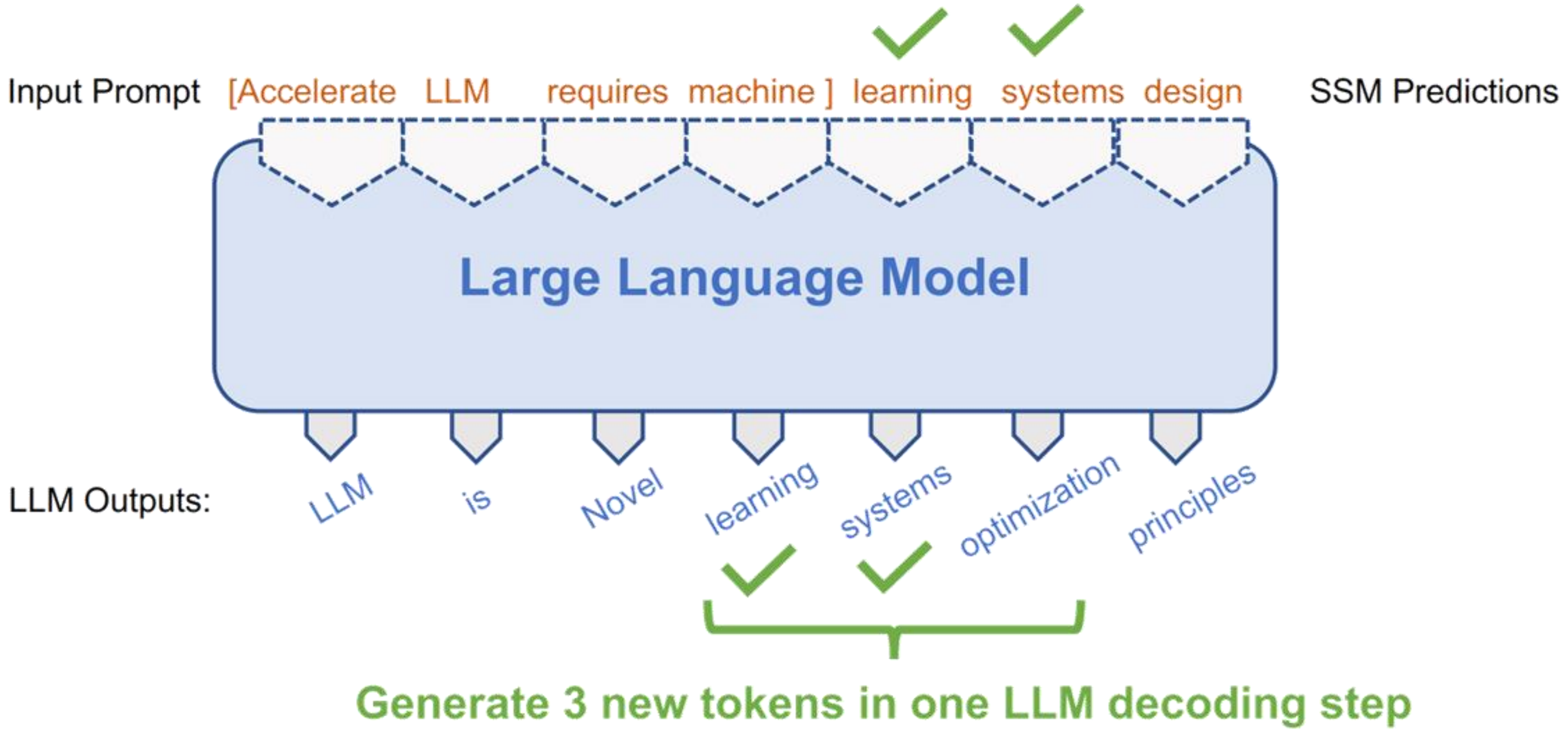
1. Use a small speculative model (SSM) to predict the LLM's output
- SSM runs much faster than LLM

2. Use the LLM to verify the SSM's prediction



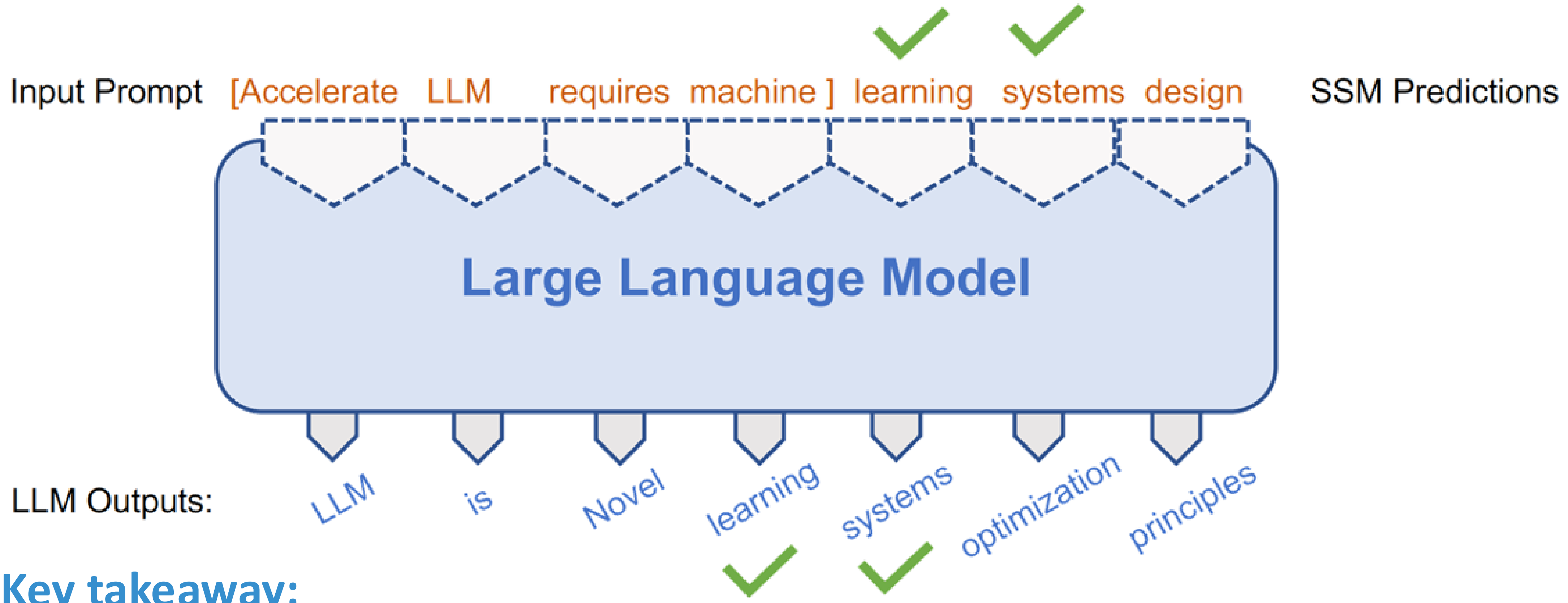
Verifying Speculative Decoding Results

上海创智学院 Shanghai Innovation Institute



Verifying Speculative Decoding Results

上海创智学院 Shanghai Innovation Institute



Key takeaway:

- LLM inference is bottlenecked by accessing model weights
- using LLM to decode multiple tokens to improve GPU utilization

SpecInfer: Tree-based Speculative Inference & Verification

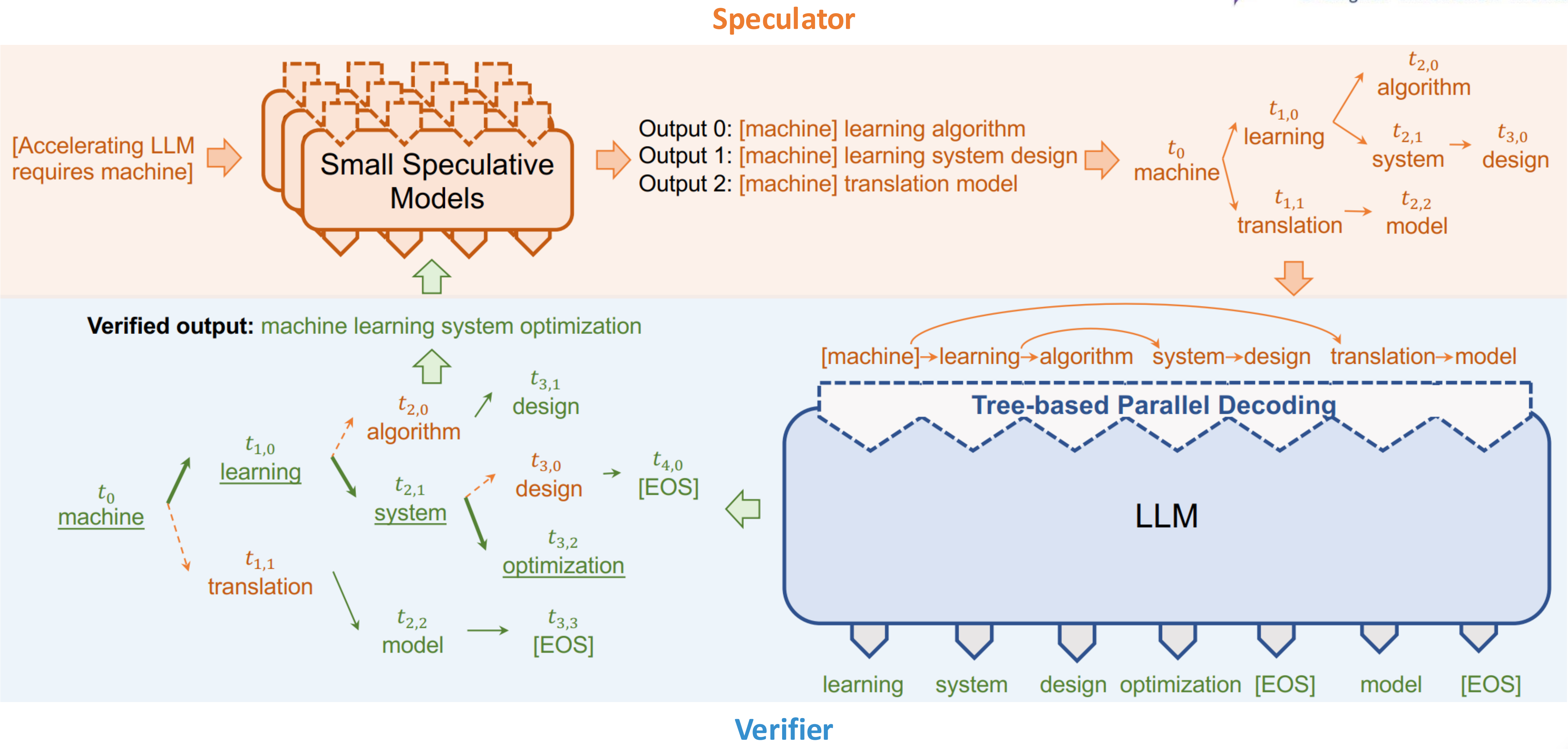
上海创智学院 Shanghai Innovation Institute

- **Key idea:** not use LLMs as incremental decoder, use them as **parallel token tree verifier**

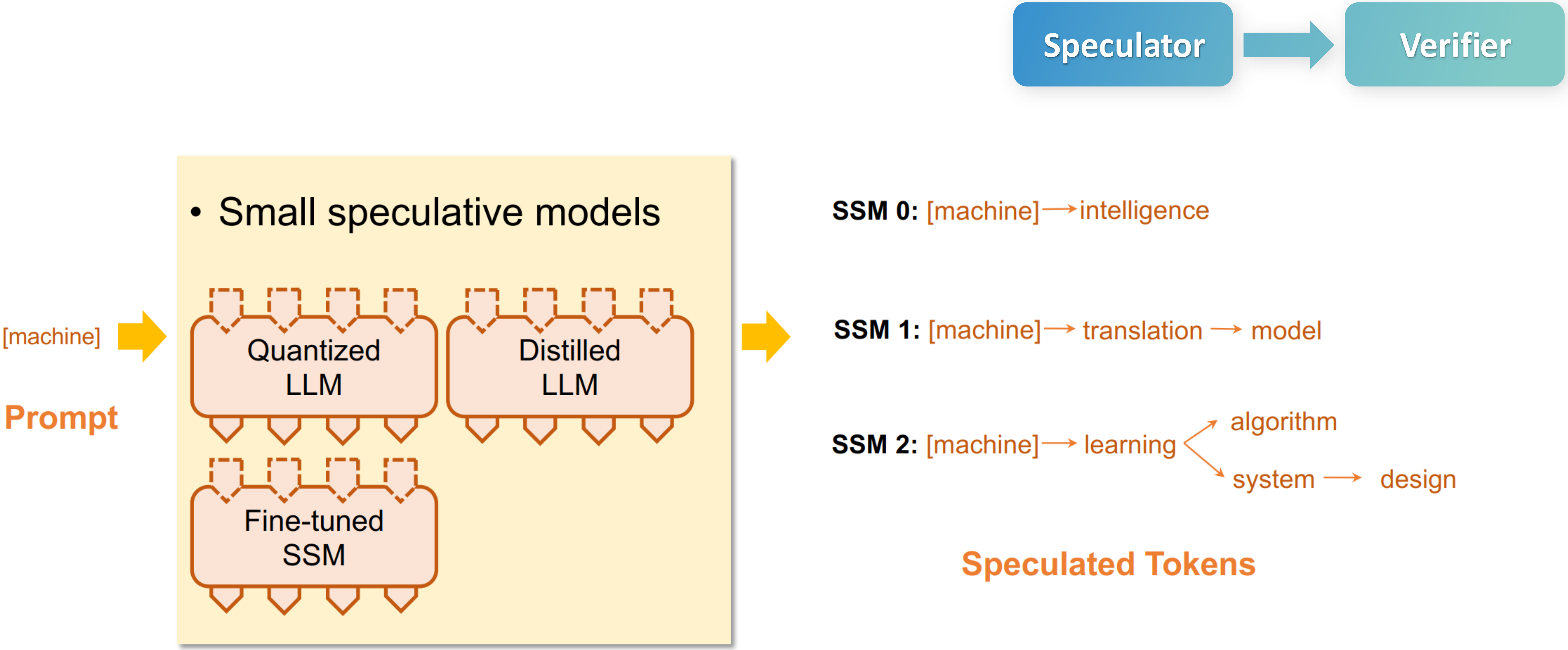
- **Better performance:** outperform existing LLM systems by **1.3-2.4x**
- **Higher efficiency:** reduce GPU memory access by **2.5-4.4x**
- **Correctness:** verification guarantees end-to-end equivalence

SpecIner: Accelerating LLM Serving with Tree-based Speculative Inference and Verification. ASPLOS’24

SpecInfer Workflow

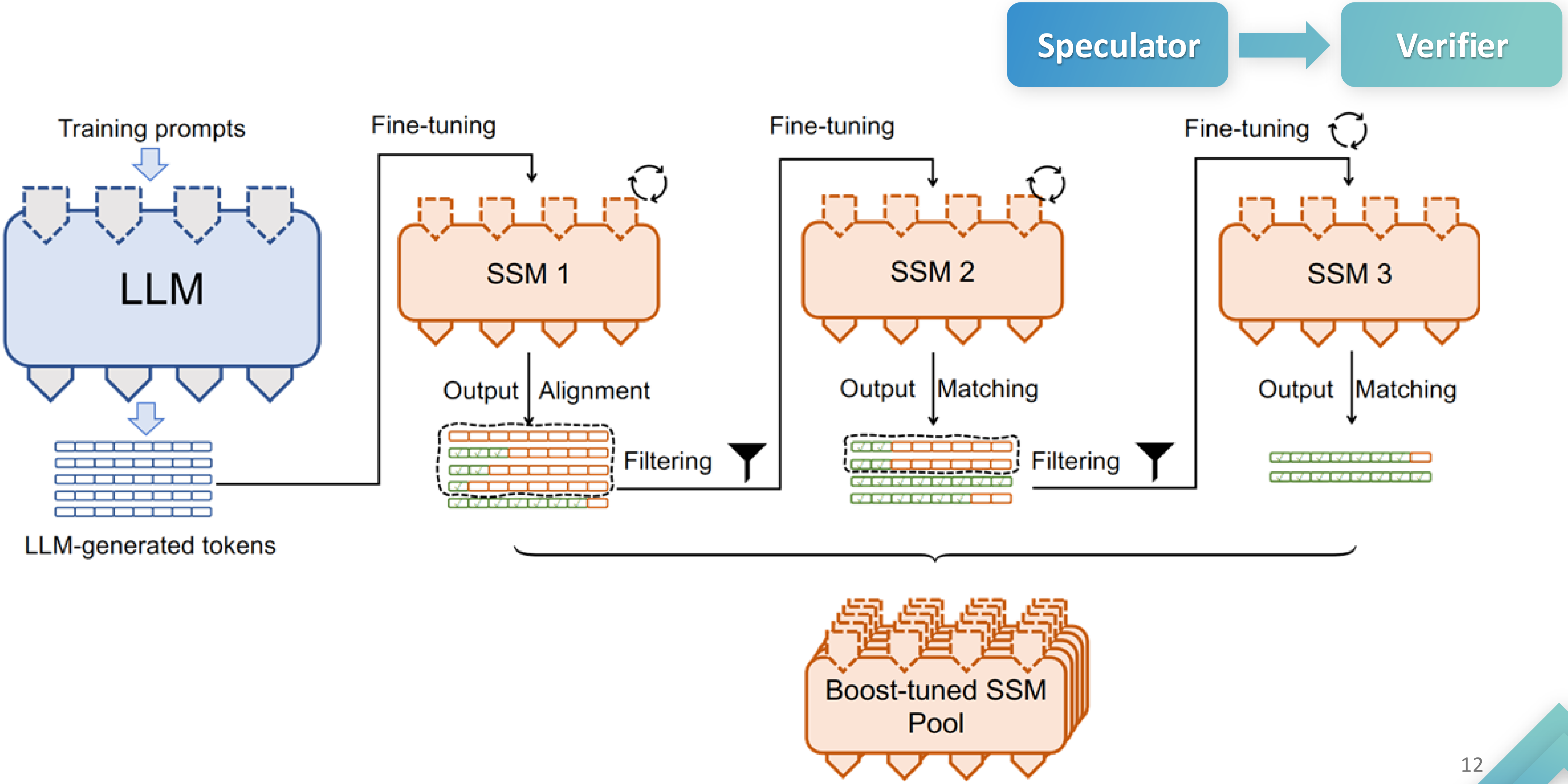


Learning-based Speculator



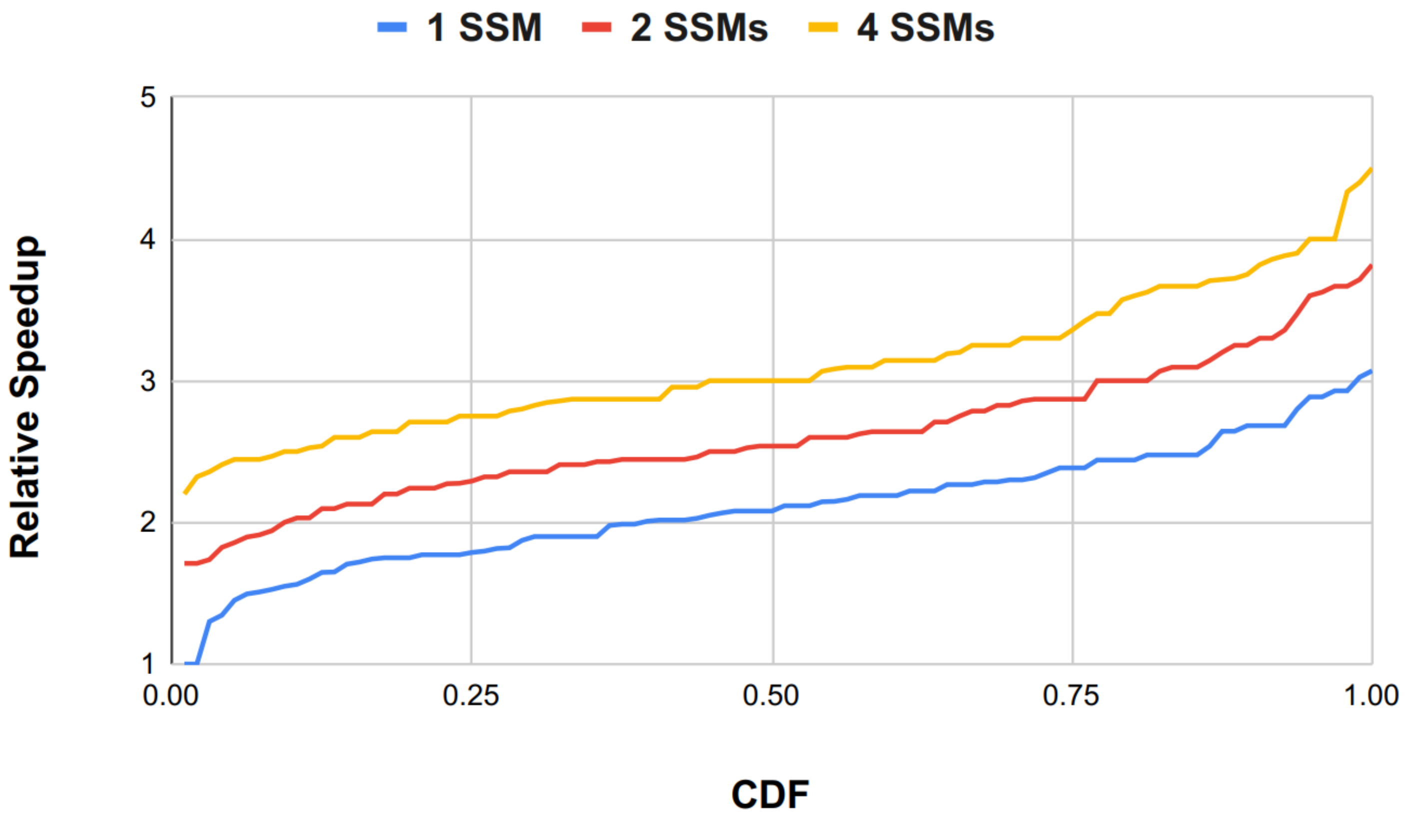
Collective Boost-Tuning

上海创智学院
Shanghai Innovation Institute



Collective Boost-Tuning Consistently Improves Performance

上海创智学院 Shanghai Innovation Institute



LLM: LLAMA-7B, SSM: LLAMA-160M, dataset: Alpaca, GPU: NVIDIA A10

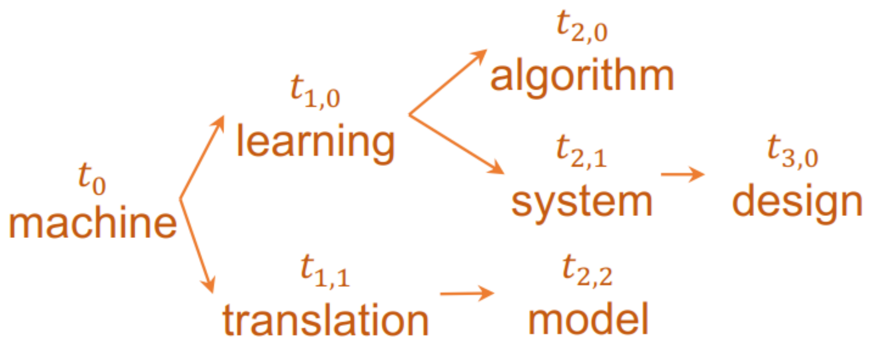
Token Tree Merge

- A compact way to represent speculated tokens

Speculator → Verifier

SSM 0: [machine] learning algorithm
SSM 1: [machine] learning system design
SSM 2: [machine] translation model

→ Tree Merge →

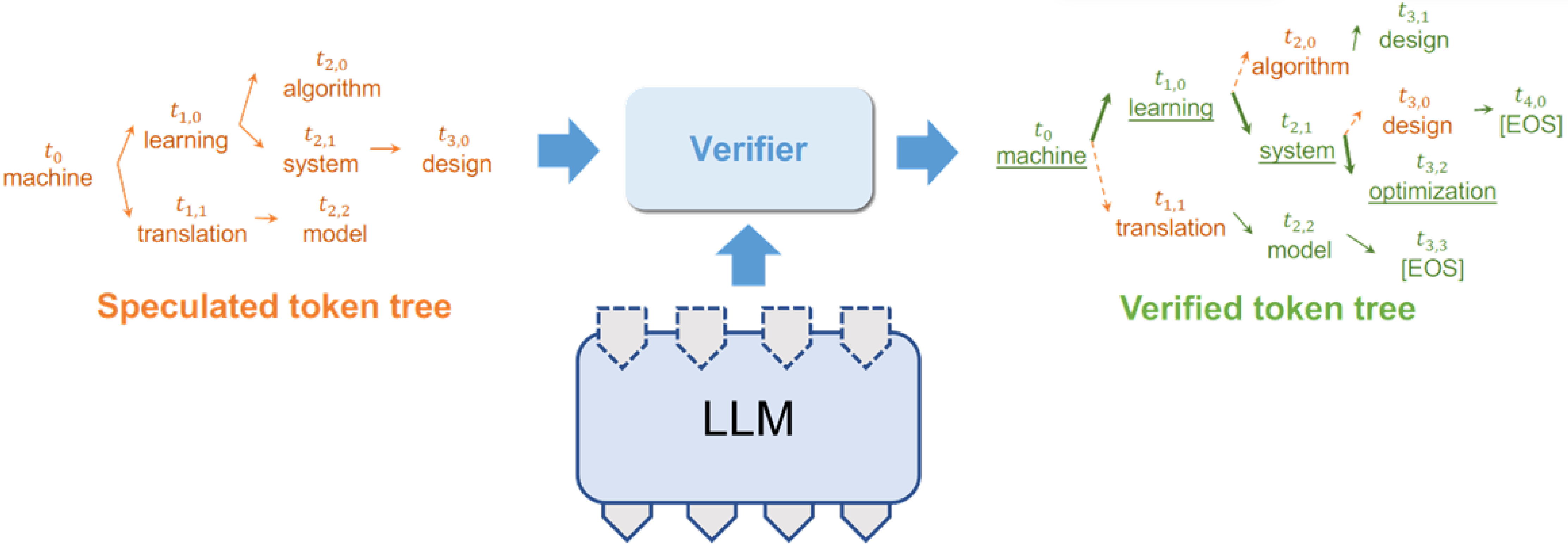


Token Sequences

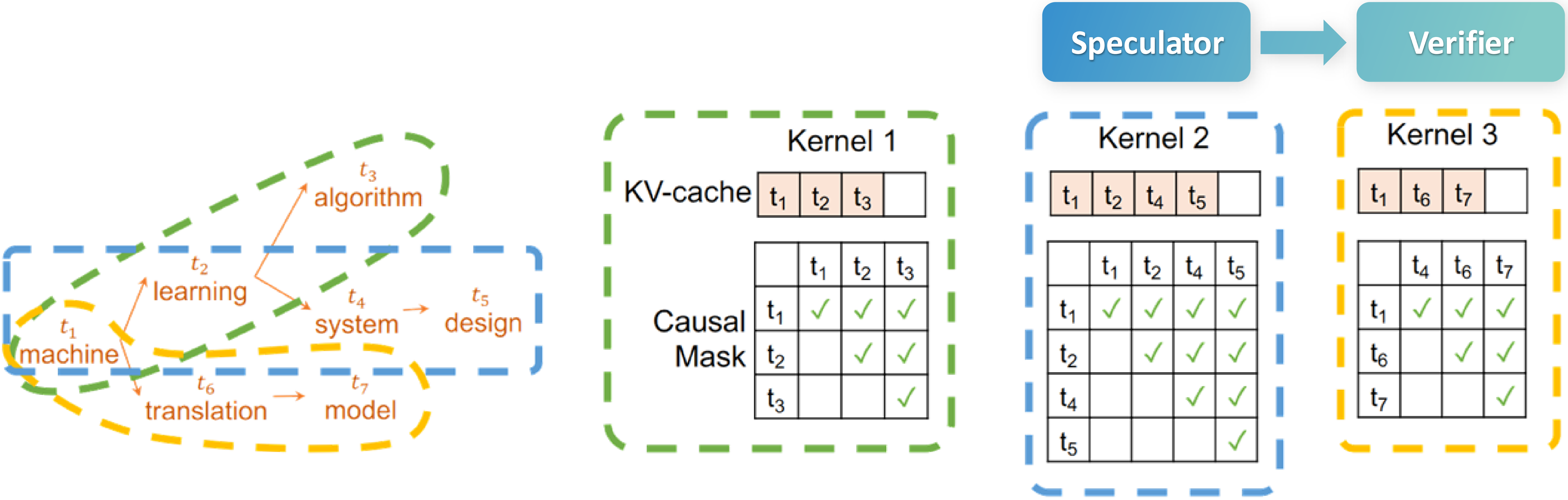
Token Tree

Token Tree Verifier

Speculator → Verifier



Sequence-based Decoding

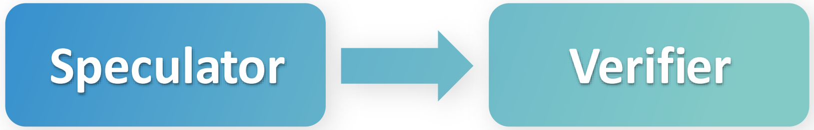


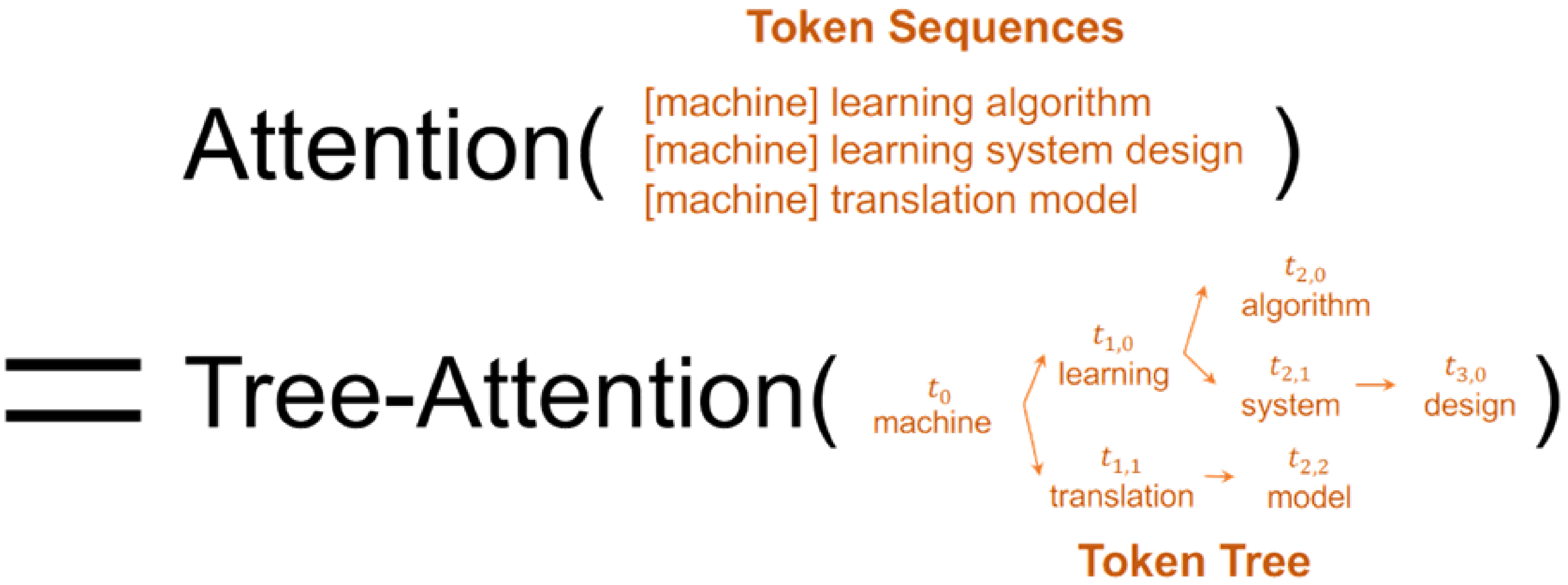
Issues:

- Redundant decoding computation
- More requests → more GPU memory for key/value cache

Tree Attention

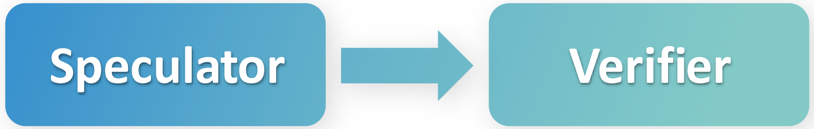
- same output as sequence attention for each token; no redundancy

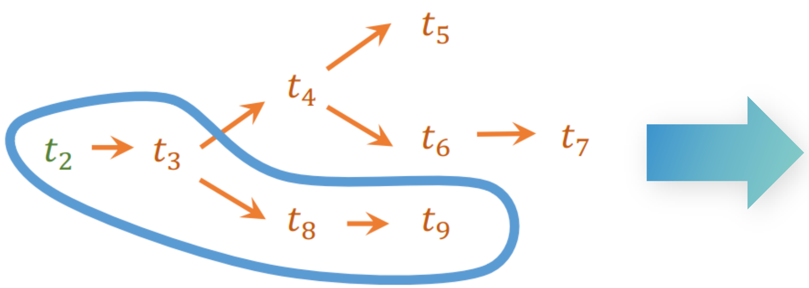




Tree-based Parallel Decoding

上海创智学院
Shanghai Innovation Institute





KV-cache

t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	…

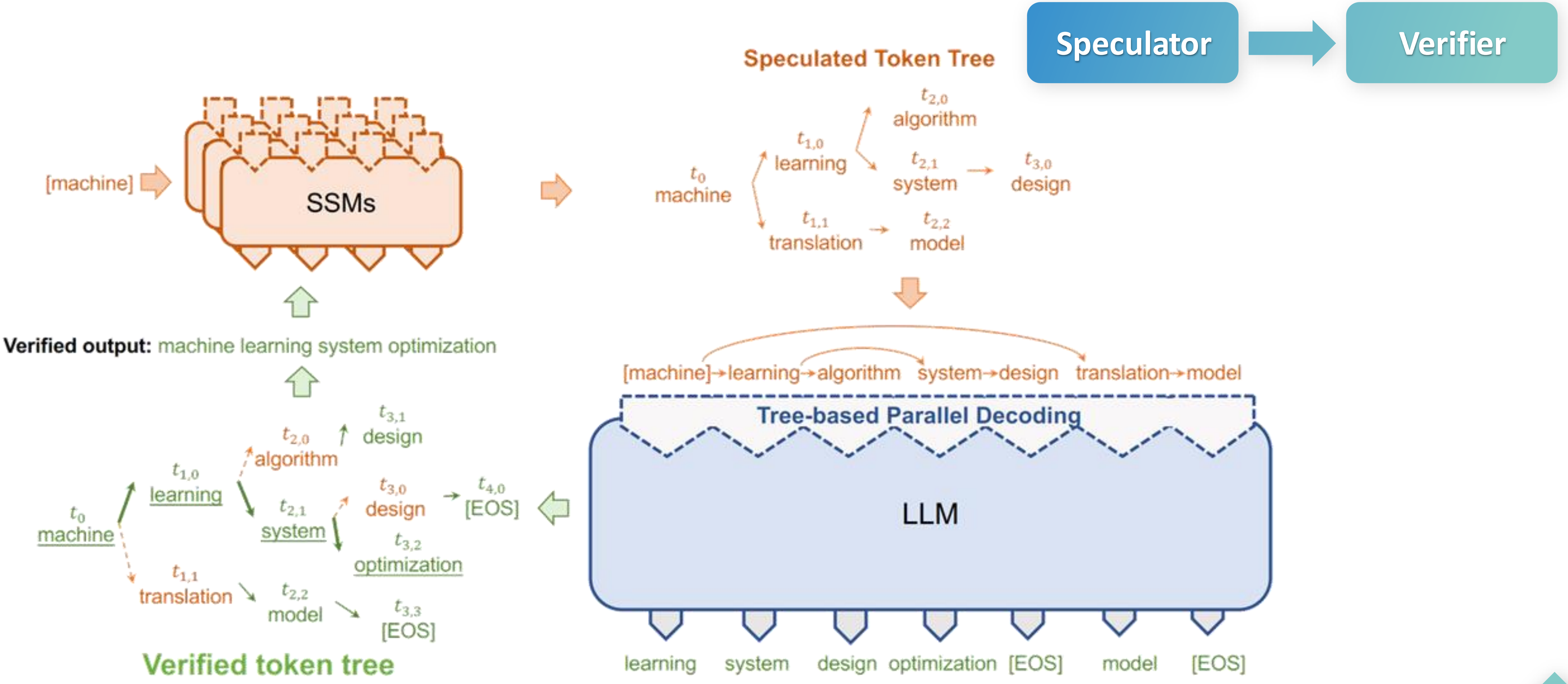
Topology-aware causal mask

	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
t_2	✓	✓	✓	✓	✓	✓	✓	✓
t_3		✓	✓	✓	✓	✓	✓	✓
t_4			✓	✓	✓	✓		
t_5				✓				
t_6					✓	✓		
t_7						✓		
t_8							✓	✓
t_9								✓

Key optimizations:

- A DFS-based approach to linearizing a token tree
- Tree topology-aware causal mask
- Decoding all tokens in a single GPU kernel

Verification Workflow



Stochastic Decoding

上海创智学院 Shanghai Innovation Institute

- **Challenge:** verifying stochastic equivalence

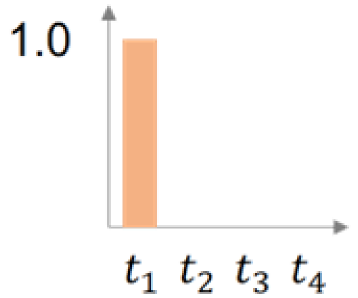
$$P_{\text{IncrDecode}}(\cdot \mid x_{<i}, \text{LLM}) = P_{\text{SpecInfer}}(\cdot \mid x_{<i}, \text{LLM}, \{\text{SSM}_i\})$$

- A strawman approach: **naïve sampling**
- Use LLM to sample $x_i \sim P_{\text{IncrDecode}}(\cdot \mid x_{<i}, \text{LLM})$
- Verify if x_i is in the token tree

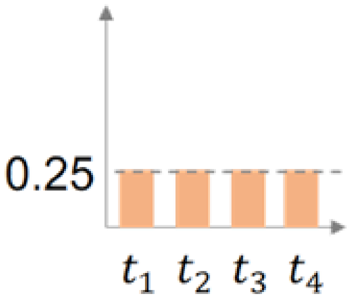
Naïve Sampling can be Suboptimal

上海创智学院 Shanghai Innovation Institute

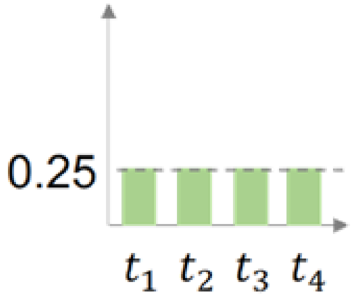
- Assume one LLM, two SSMs, and four possible tokens: t_1, t_2, t_3, t_4



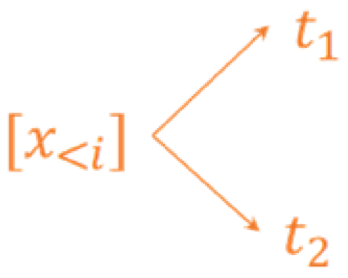
SSM 1: $P(\cdot\,|x_{<i}, \mathrm{SSM}_1)$



SSM 2: $P(\cdot\,|x_{<i}, \mathrm{SSM}_2)$



LLM: $P(\cdot\,|x_{<i}, \mathrm{LLM})$



Token Tree

Naïve sampling's verification prob. = **50%**

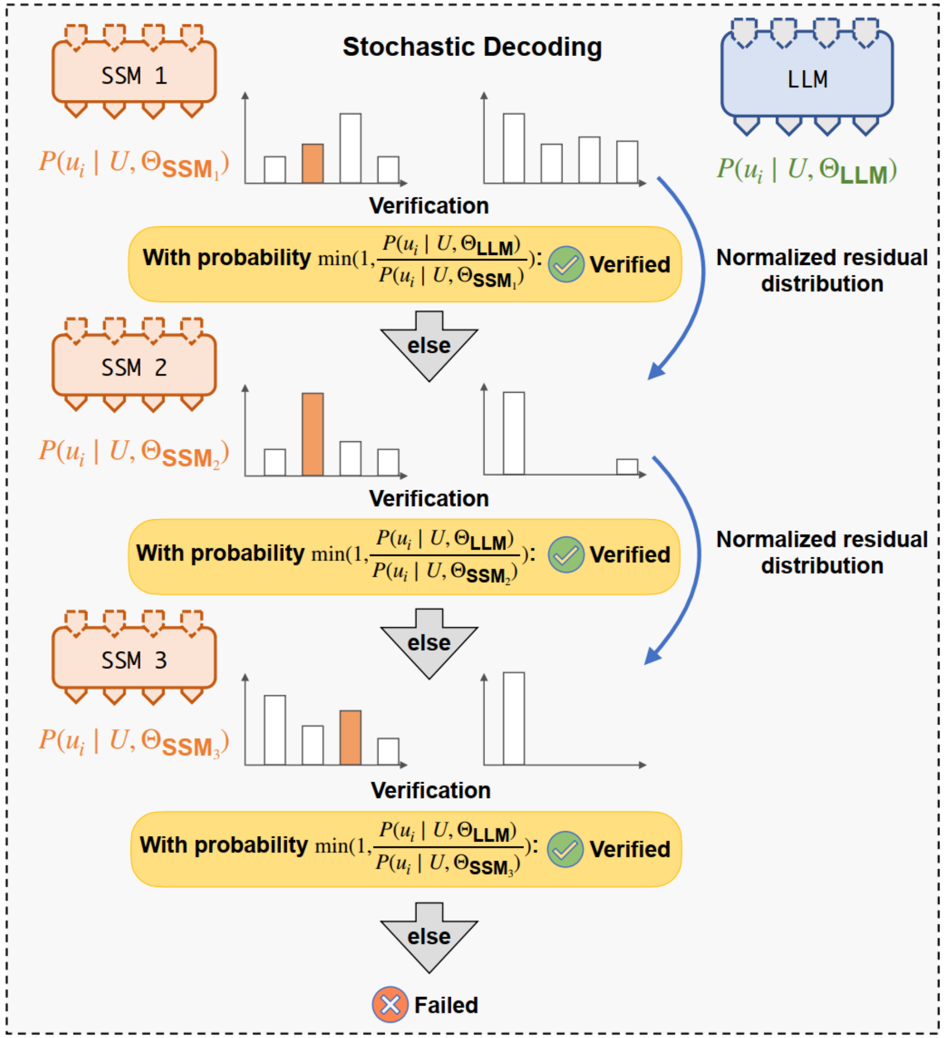
But we can do better by directly accepting SSM 2; verification prob. = 100%

Key issue: naïve sampling ignores correlation between $P(\cdot\,|x_{<i}, \mathrm{SSM})$ and $P(\cdot\,|x_{<i}, \mathrm{LLM})$

Speculative Sampling

1. Sample a token $x \sim P(u_i|U,\Theta_{SSM})$ using SSM
2. If $P(x|U,\Theta_{SSM}) \le P(x|U,\Theta_{LLM})$, directly accept x
3. If $P(x|U,\Theta_{SSM}) > P(x|U,\Theta_{LLM})$, accept x with prob. $\frac{P(x|U,\Theta_{LLM})}{P(x|U,\Theta_{SSM})}$
4. If reject x, normalize residual distribution

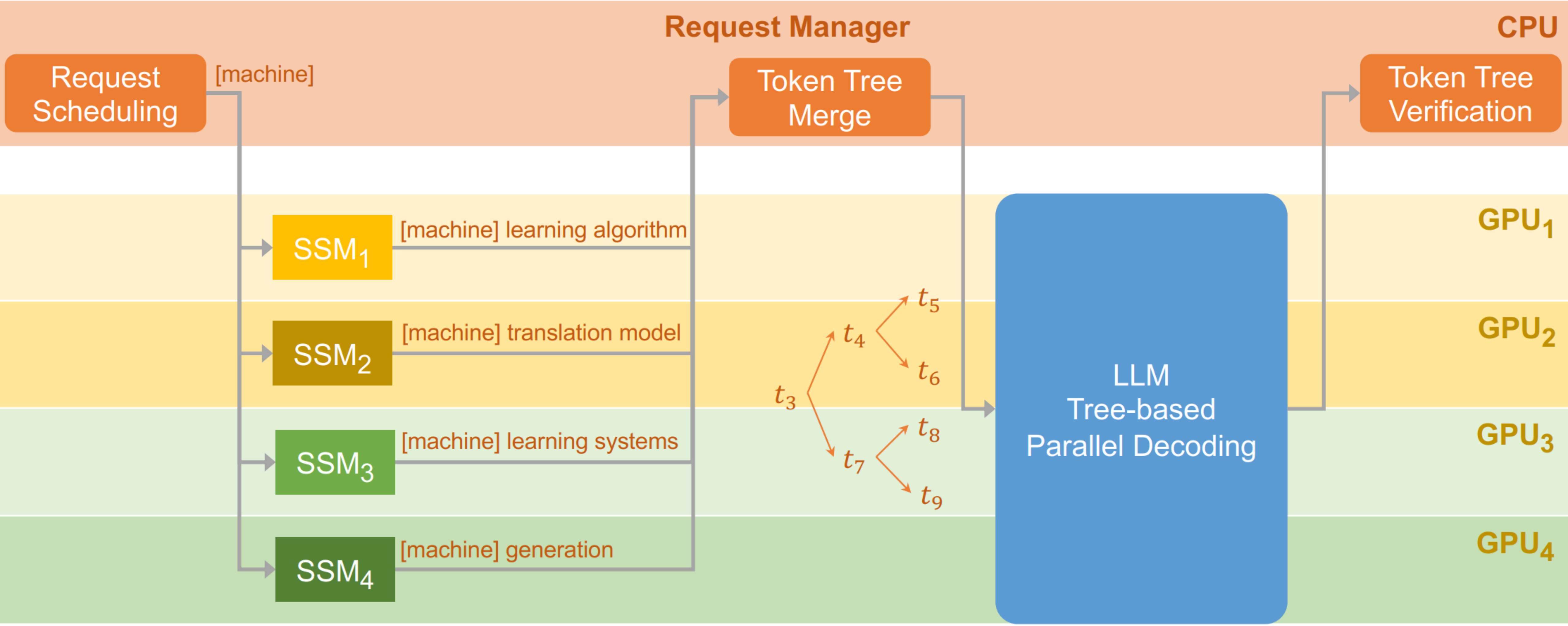
$$P'(x|U,\Theta_{LLM}) = norm(\max(0, P(x|U,\Theta_{LLM}) - P(x|U,\Theta_{SSM})))$$



Details in SpecInfer: Accelerating Large Language Model Serving with Tree-based Speculative Inference and Verification. ASPLOS'24

Distributed LLM Serving

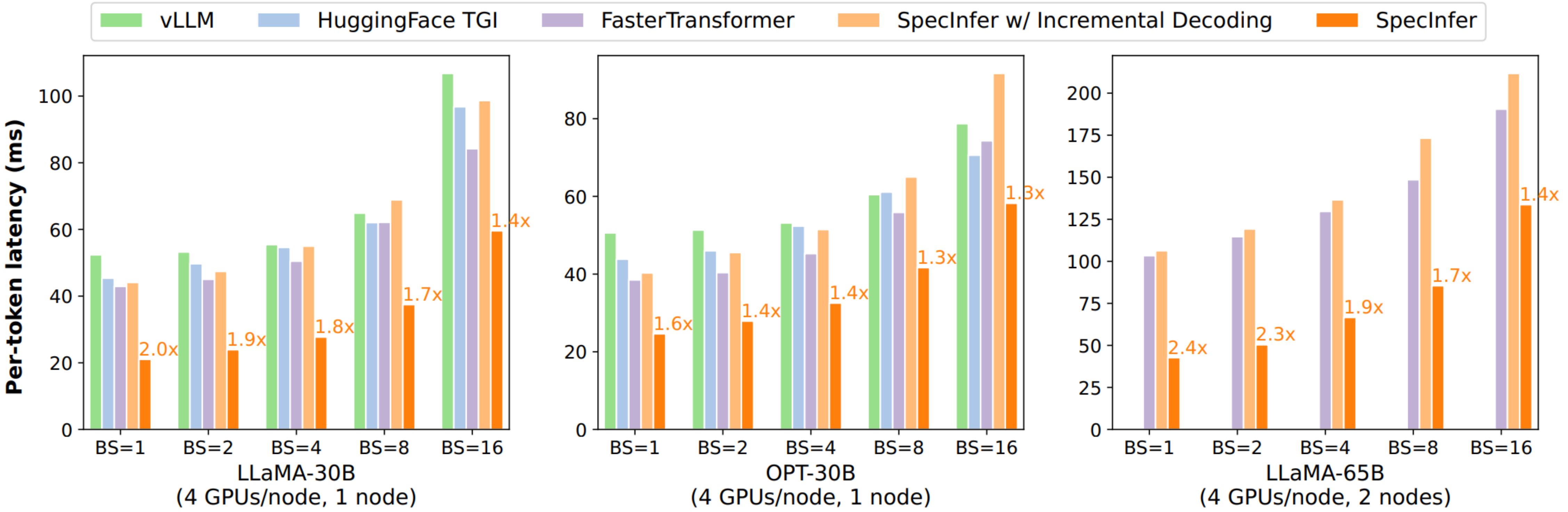
上 海 创 智 学 院
Shanghai Innovation Institute



Tensor / Pipeline Model Parallelism

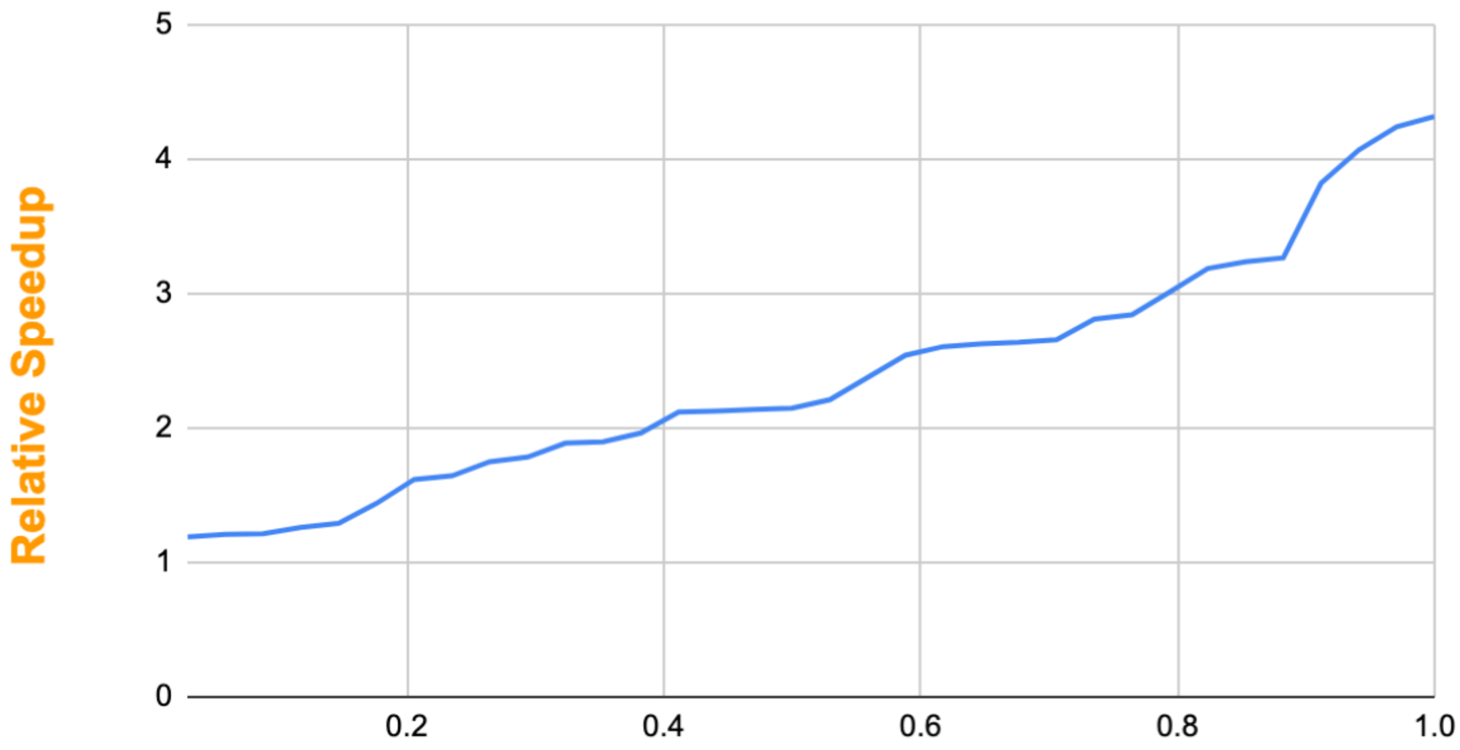
SpecInfer Accelerates LLM Inference by 1.3-2.4x

上海创智学院 Shanghai Innovation Institute



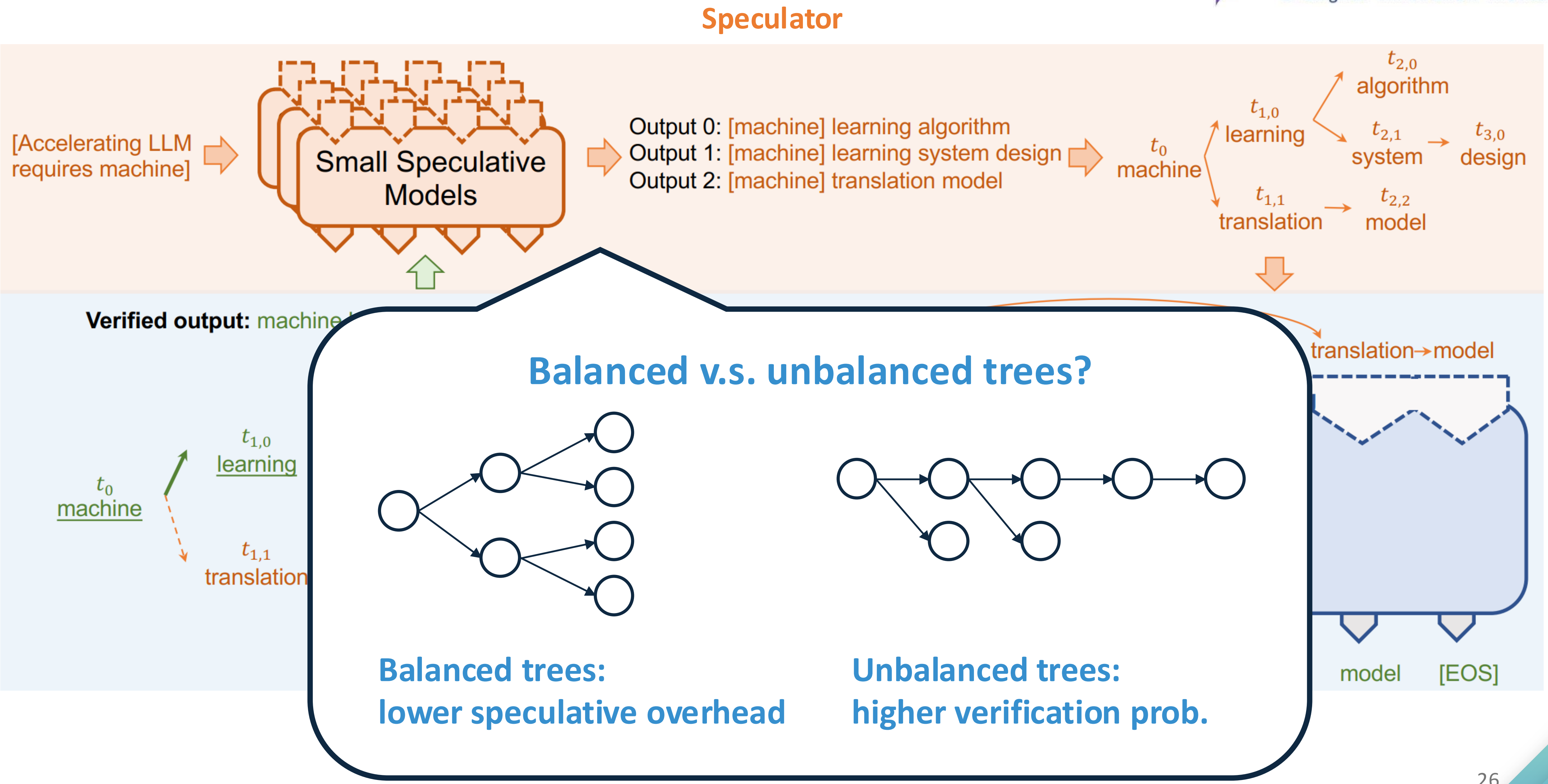
SpecInfer can Consistently Accelerate LLM Inference

上海创智学院 Shanghai Innovation Institute

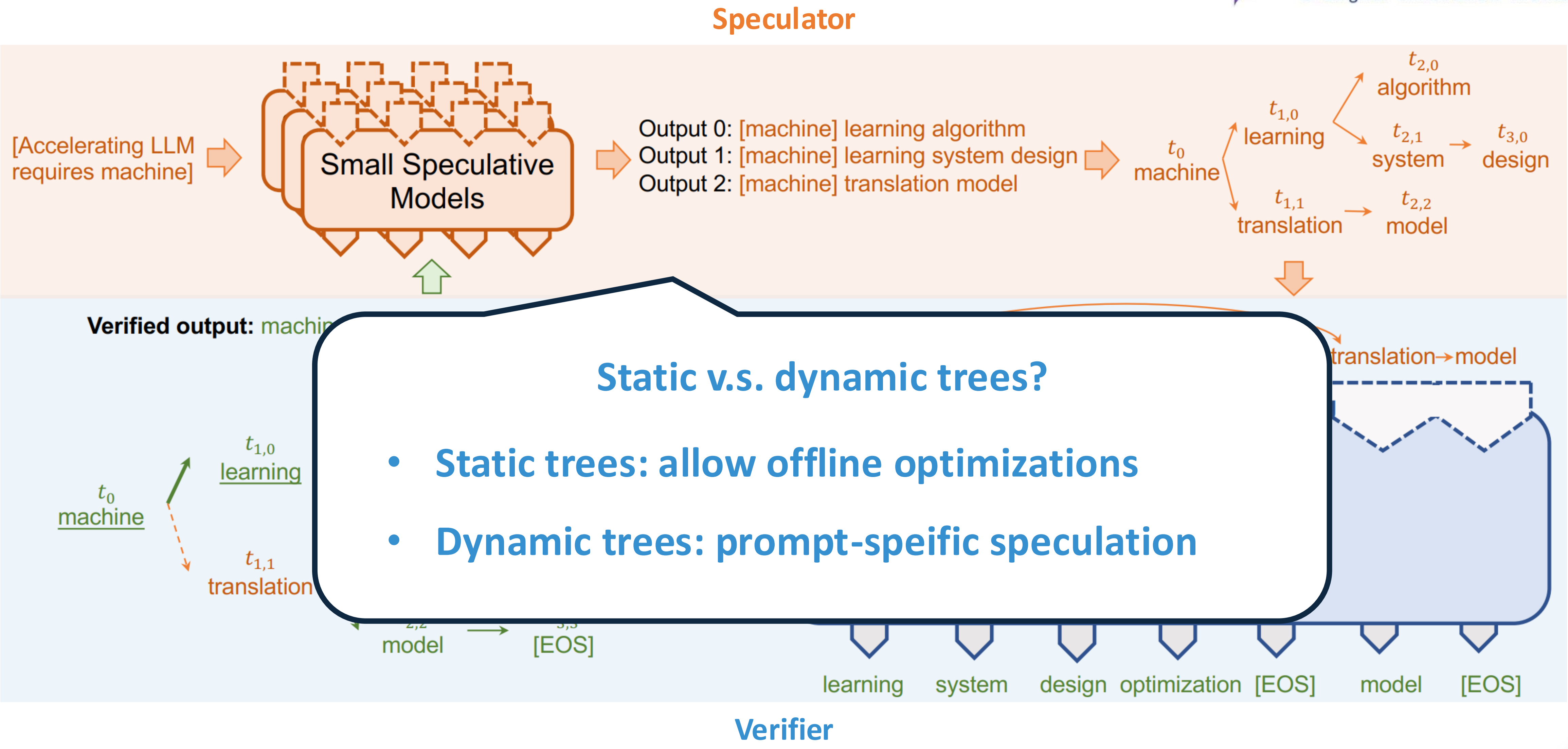


LLM: LLaMA-65B, SSMs: LLaMA-160M

Open Research Questions



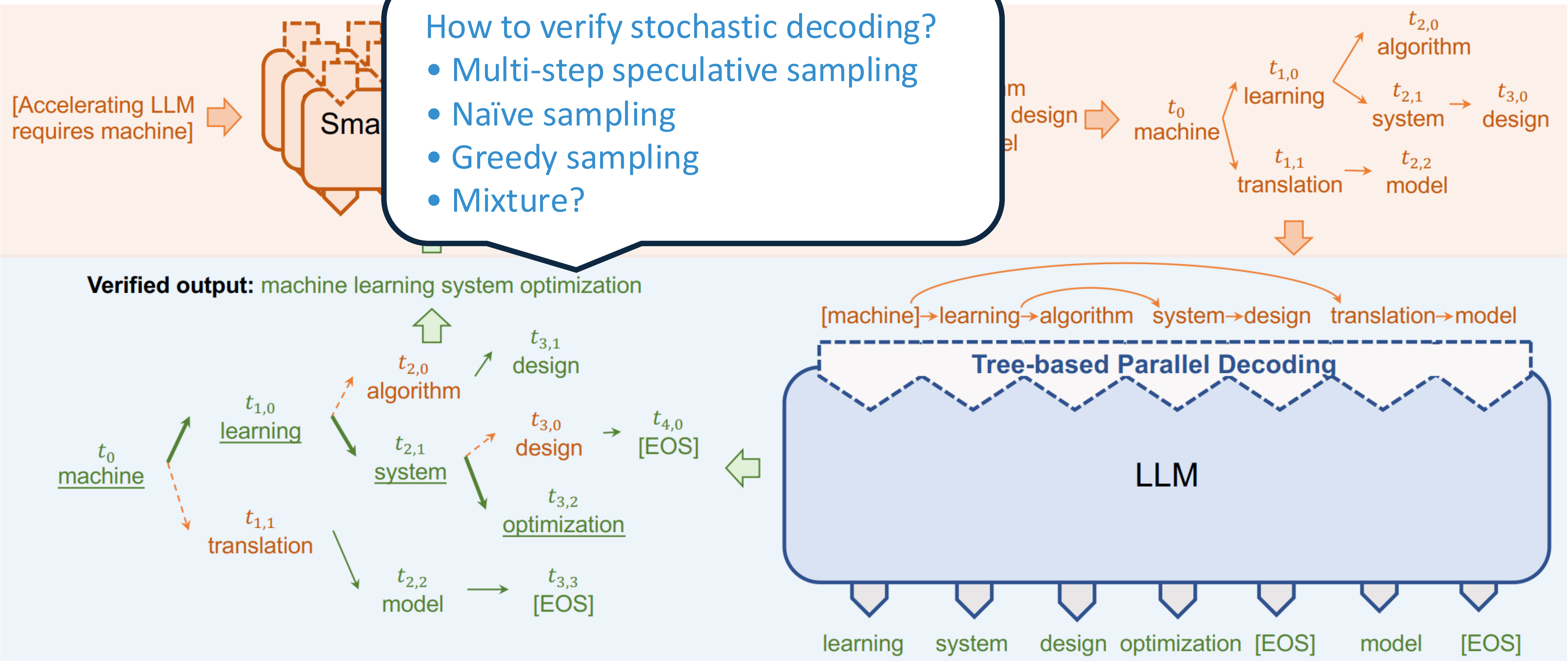
Open Research Questions



Static v.s. dynamic trees?

- Static trees: allow offline optimizations
- Dynamic trees: prompt-speific speculation

Open Research Questions



How to verify stochastic decoding?

- Multi-step speculative sampling
- Naïve sampling
- Greedy sampling
- Mixture?

Verifier

Acknowledgement

The development of this course, including its structure, content, and accompanying presentation slides, has been significantly influenced and inspired by the excellent work of instructors and institutions who have shared their materials openly. We wish to extend our sincere acknowledgement and gratitude to the following courses, which served as invaluable references and a source of pedagogical inspiration:

- Machine Learning Systems[15-442/15-642], by **Tianqi Chen** and **Zhihao Jia** at **CMU**.
- Advanced Topics in Machine Learning (Systems)[CS6216], by **Yao Lu** at **NUS**

While these materials provided a foundational blueprint and a wealth of insightful examples, all content herein has been adapted, modified, and curated to meet the specific learning objectives of our curriculum. Any errors, omissions, or shortcomings found in these course materials are entirely our own responsibility. We are profoundly grateful for the contributions of the educators listed above, whose dedication to teaching and knowledge-sharing has made the creation of this course possible.

System for Artificial Intelligence

Thanks

Siyuan Feng
Shanghai Innovation Institute