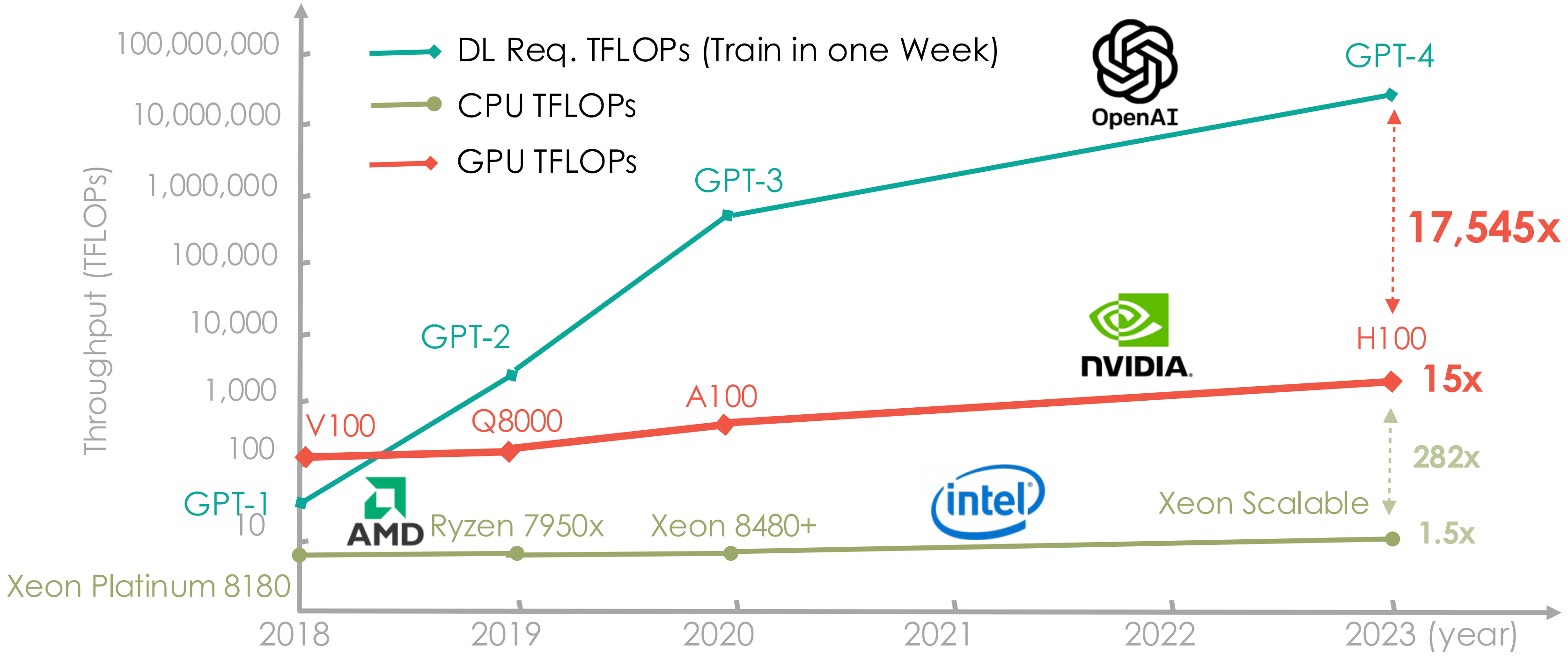
Systematic Approaches for Efficient and Scalable Deep Learning

Yuke Wang
Computer Science
Rice University

The Trend of DL Algorithm and Hardware

❖ Recap of DL algorithms and hardware performance scaling.



Huge Potential with GPUs! But it still has a **Large Gap!**

Deep Learning Drives Computing Innovations

❖ Overview of my prior Ph.D. Research.

Algorithm Innovation

ICML'20, CIKM'21, AAAI'21

Model Design, Model Pruning (2018-19)

Hardware Innovation

TCAD'21a, TCAD'21b, CCGrid'21

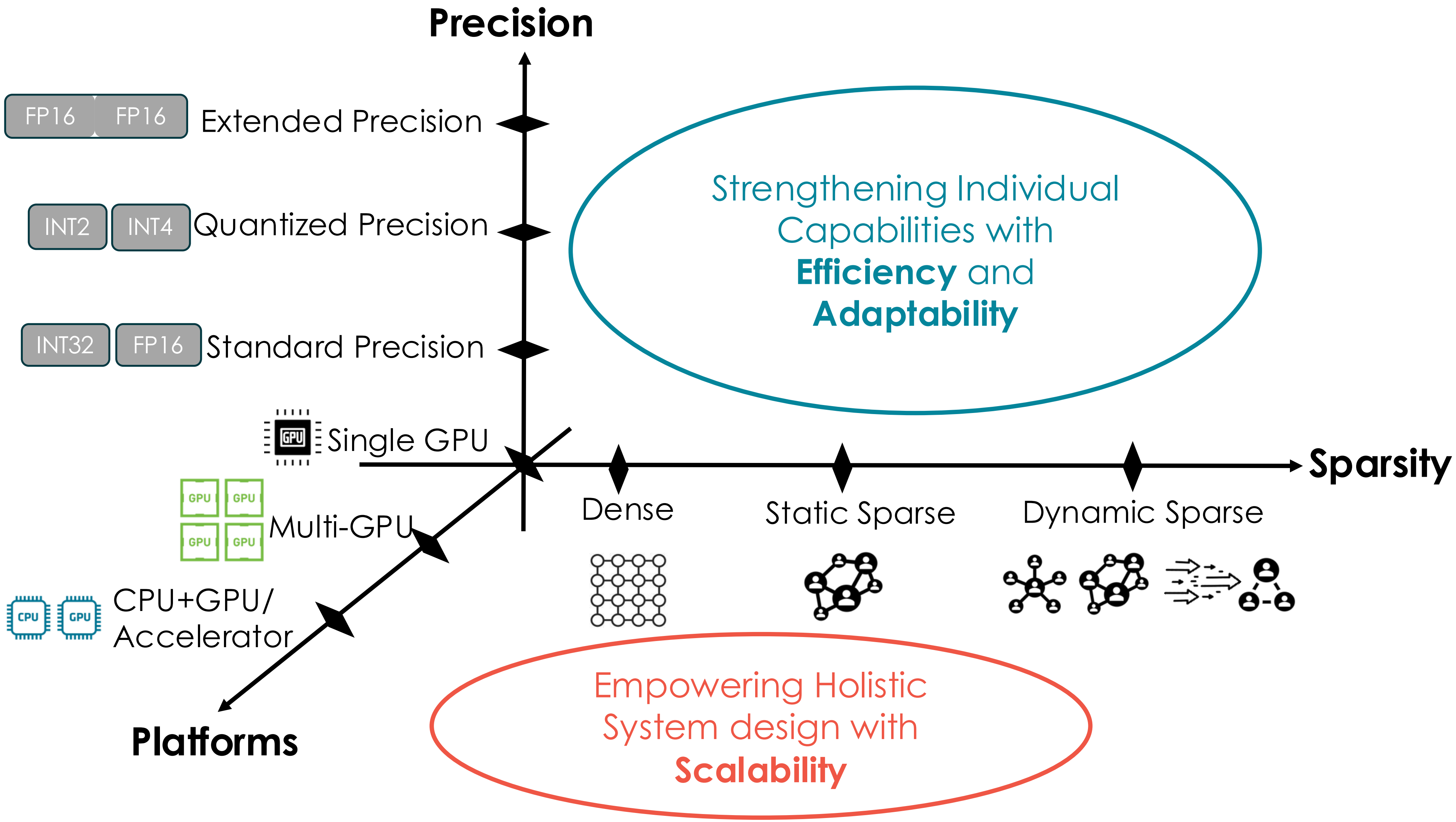
Hardware Accelerators (2020-21)

Software Innovation

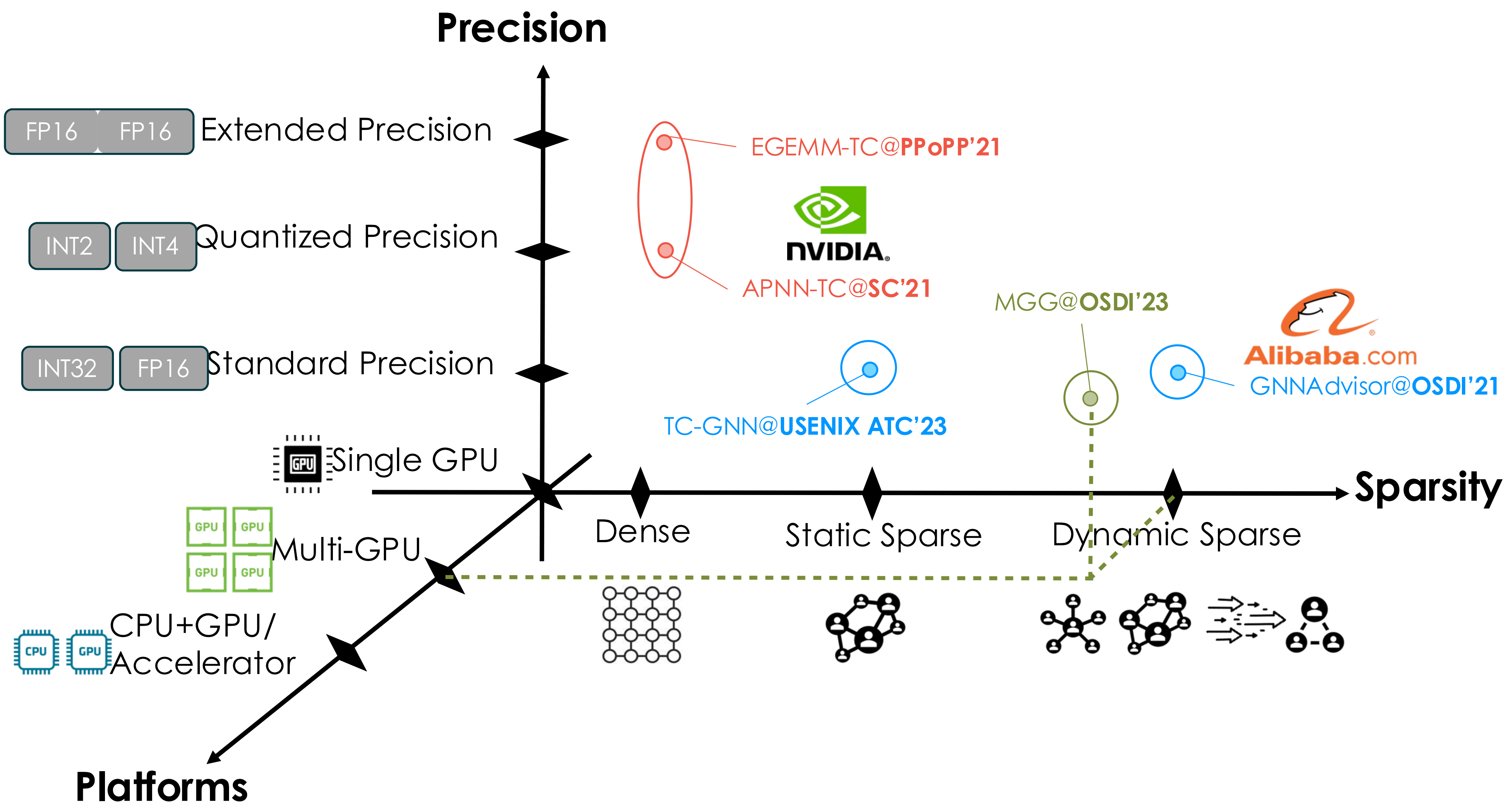
OSDI'21, SC'21, PPoPP'22, OSDI'23, USENIX ATC'23

Runtime Systems, Compiler (2021-now)

My Prior Ph.D. Research Recap



My Prior PhD Research Recap



Precision

Sparsity

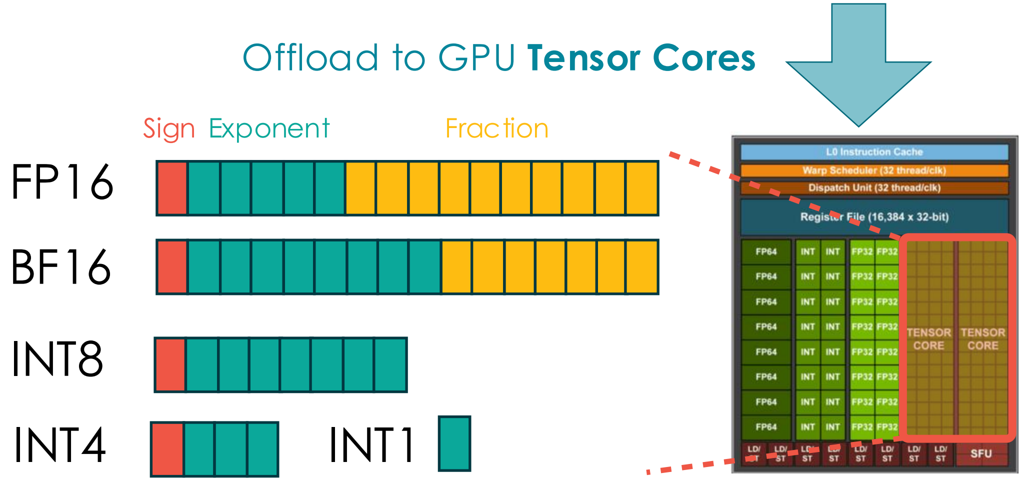
Scalability

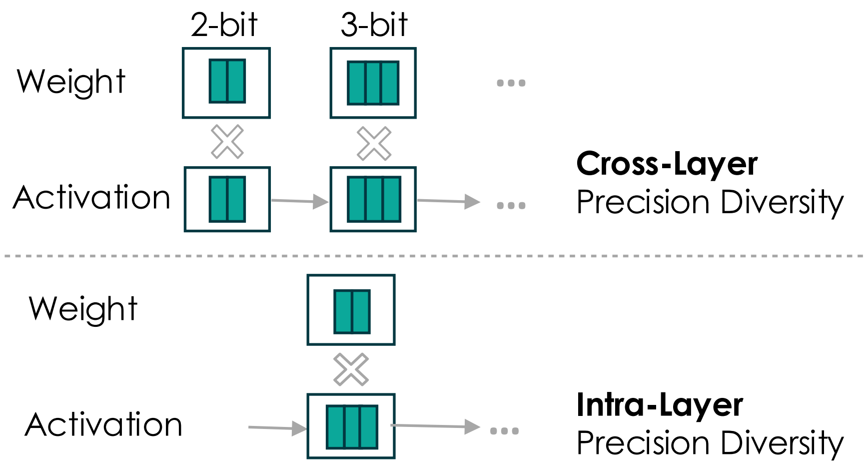
Diverse Precision Demands for DL Applications

❖**Low-precision** quantized deep-learning applications.

Quantized Deep Learning	Precision Requirements
QNNs [JMLR'18]	**1-bit Weight, 2-bit Activation** for Vision Model, **3-bit Weight, 4-bit Activation** for Language Model.
SGQuant [ICTAI'20]	Graph Attention Model: **2-bit** Neighbor **Attention**, **4-bit** Neighbor **Aggregation.**
LLM.int8() [NeurIPS'22]	**8-bit** Quantization for Transformers.
...	...

Offload to GPU **Tensor Cores**



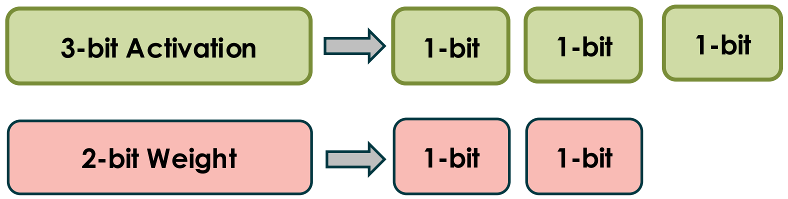


GPU Tensor cores would suffer from **low efficiency**.

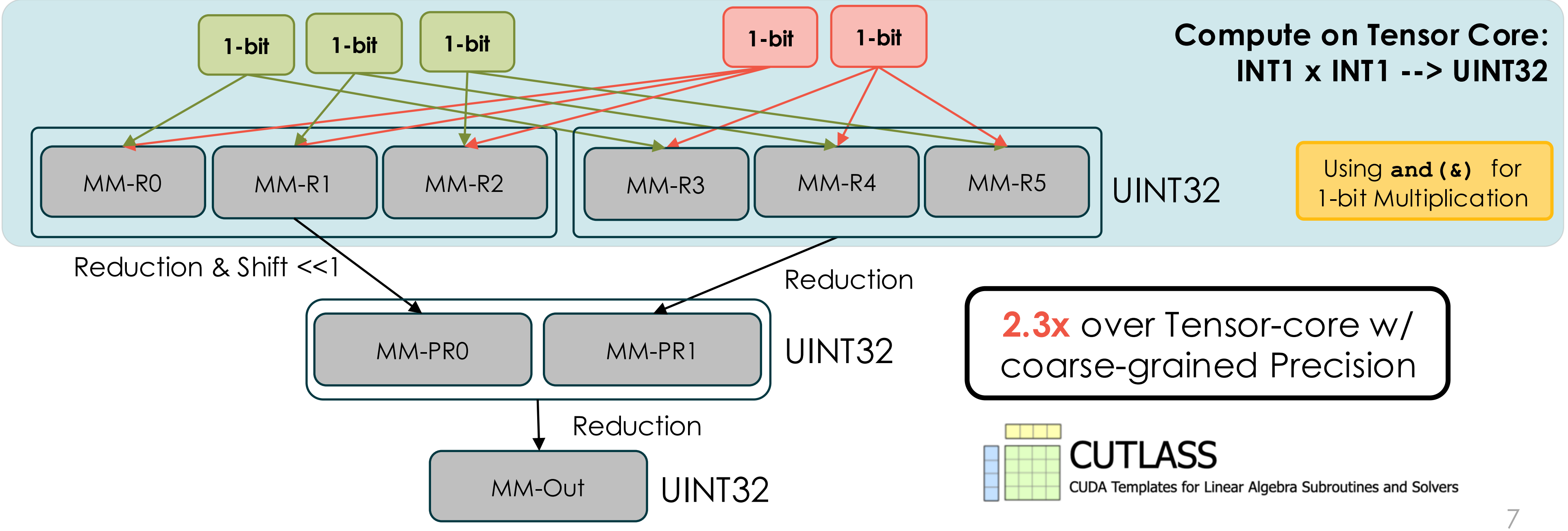
Bit Composition for Quantized Deep Learning [SC'21]

❖ Insight: **Quantized deep learning** can be **composed** with the **binary (1-bit) precision**.

Example of **2-bit** and **3-bit** Precision in Quantized DNN Computing.



Multiplication between 3-bit Activation and 2-bit Weight

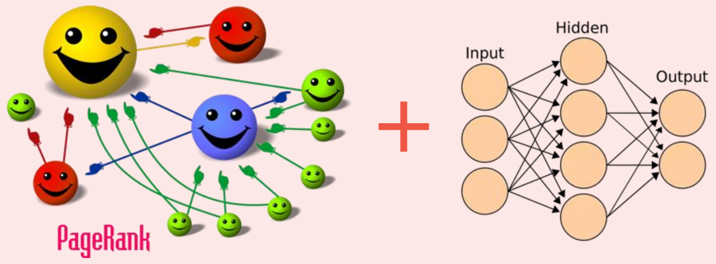


2.3x over Tensor-core w/ coarse-grained Precision

CUTLASS
CUDA Templates for Linear Algebra Subroutines and Solvers

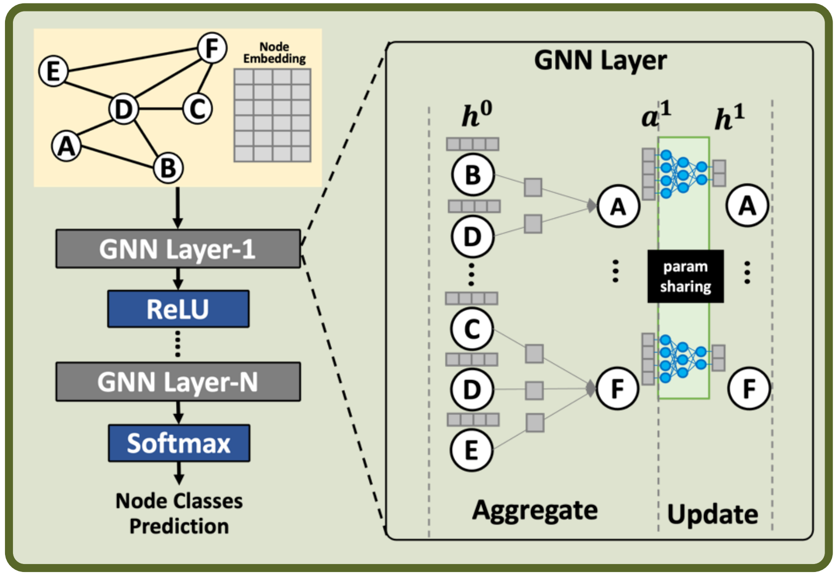
A Typical Paradigm of Graph Deep Learning

Operation view

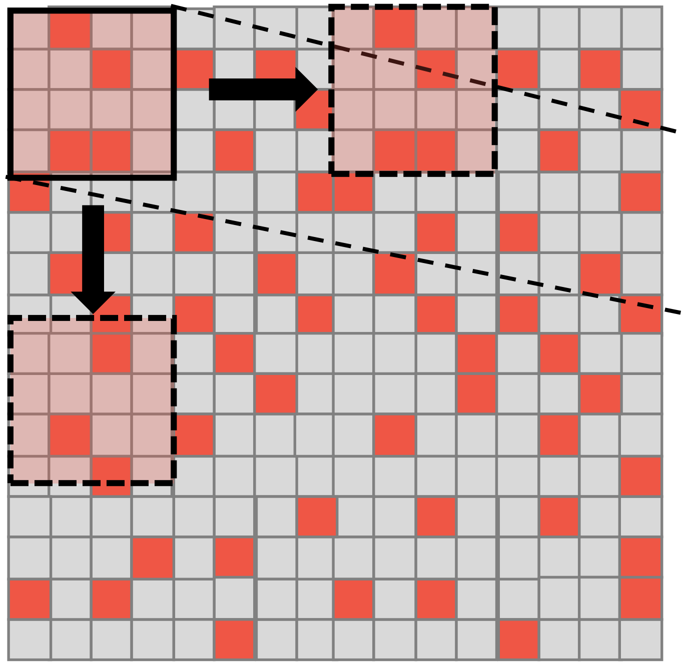
GNN Layer = 

Graph Ops + DNN Ops

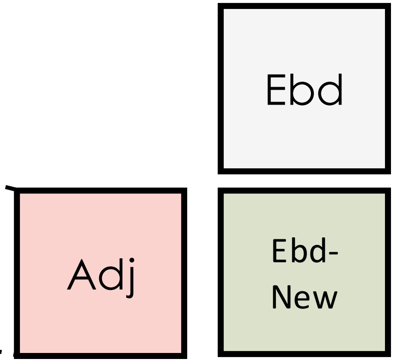
Model view



Challenge of Mapping Sparse Computing to Dense Units



Sparse Adjacent Matrix of Graph



>> A100/H100 (80GB)

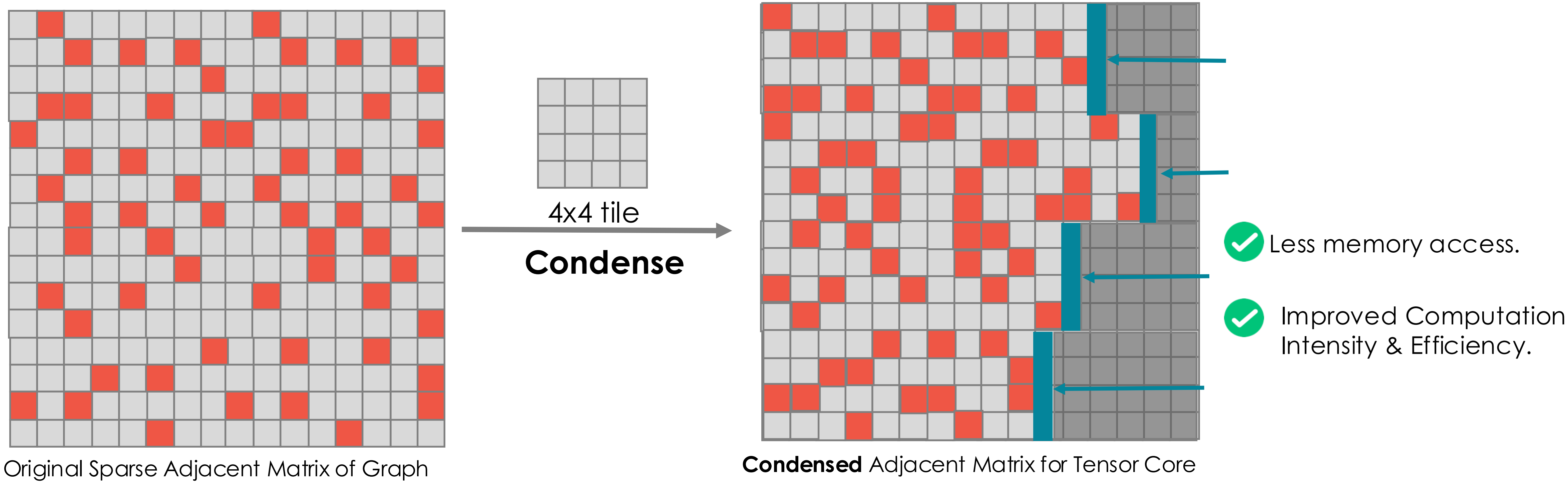
Dataset	# Nodes	# Edges	Memory	Eff.Comp
OVCR-8H	1,890,931	3,946,402	14302.48 GB	0.36%
Yeast	1,714,644	3,636,546	11760.02 GB	0.32%
DD	334,925	1,686,092	448.70 GB	0.03%

Largely **Wasted** Computation and Memory Access

Direct mapping: **high memory** consumption and **low computing** efficiency.

TC-GNN: Order-Invariant Transformation [ATC'23]

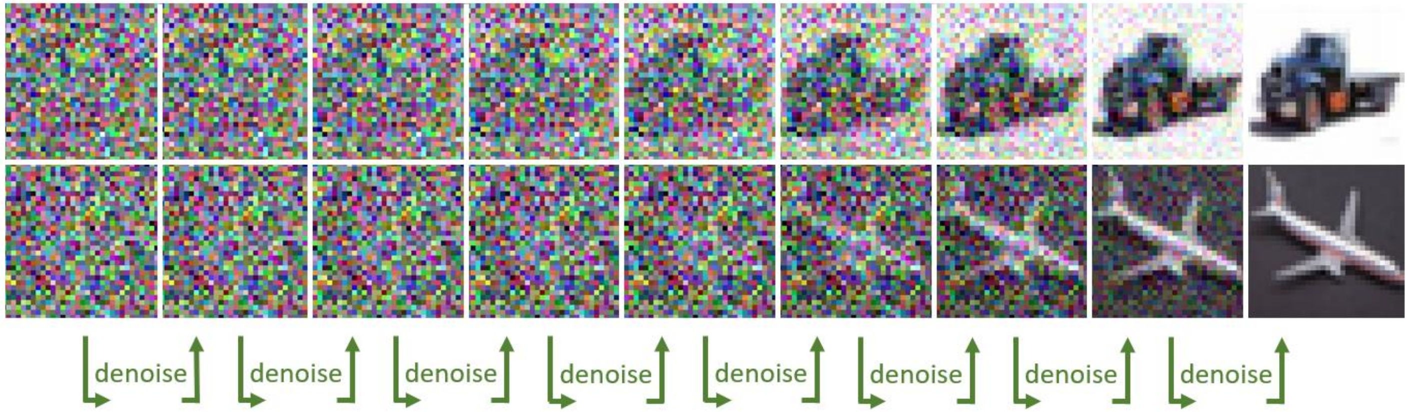
- Irregularly-scattered elements can be **condensed** to benefit high-performance dense GPU units.



1.50x ~ 6.70x over DGL operators (cuSPARSE).

Incorporated by SparseTIR in **TVM** Project.

In The Era of Generative AI



Gemini | Nano Banana

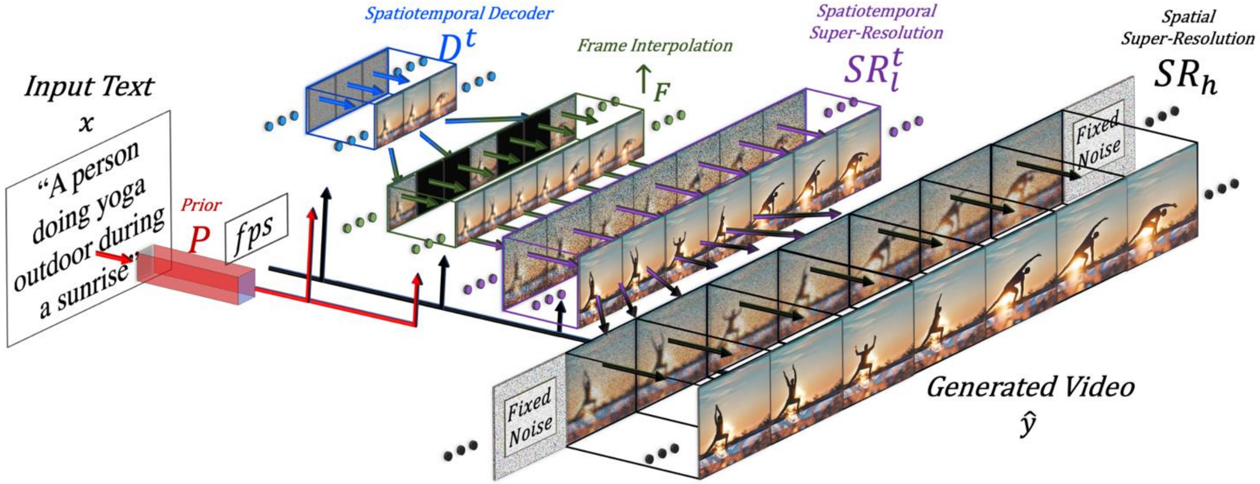
Adobe Firefly

Qwen

OpenAI SORA

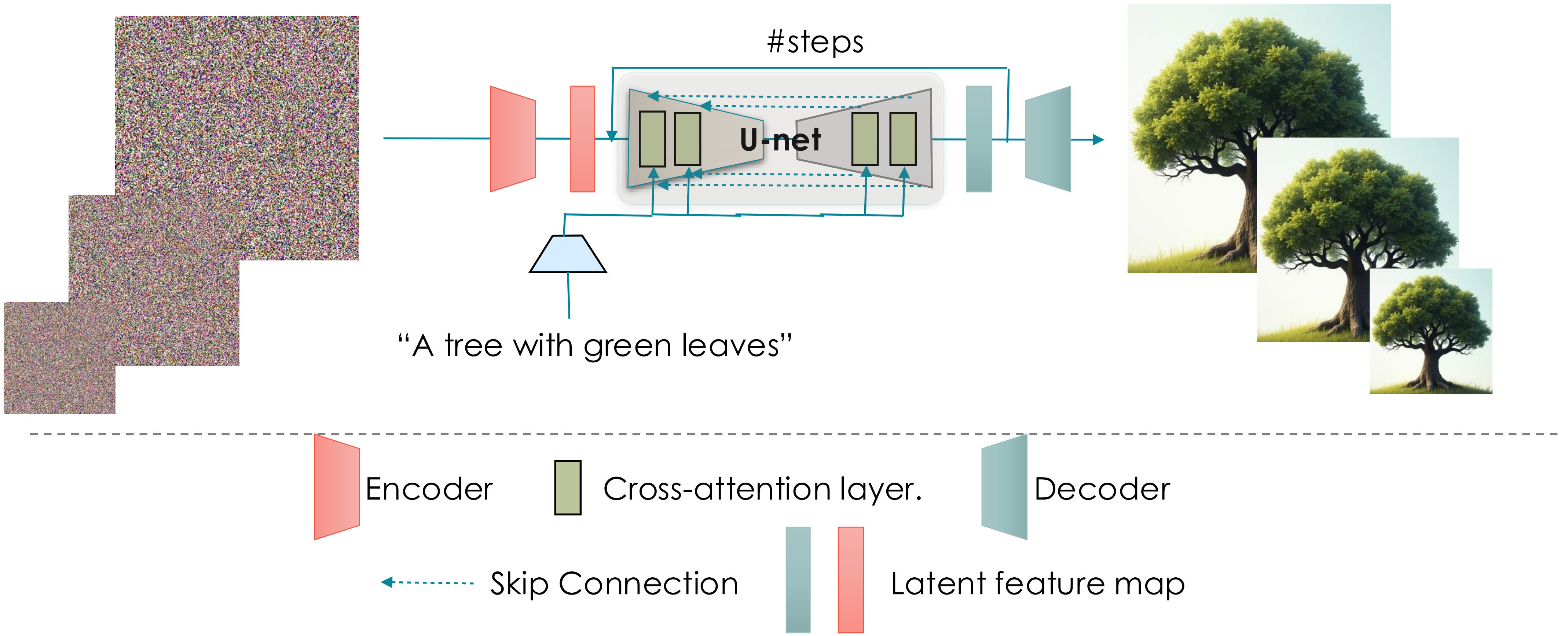
Pika

Tencent Hunyuan



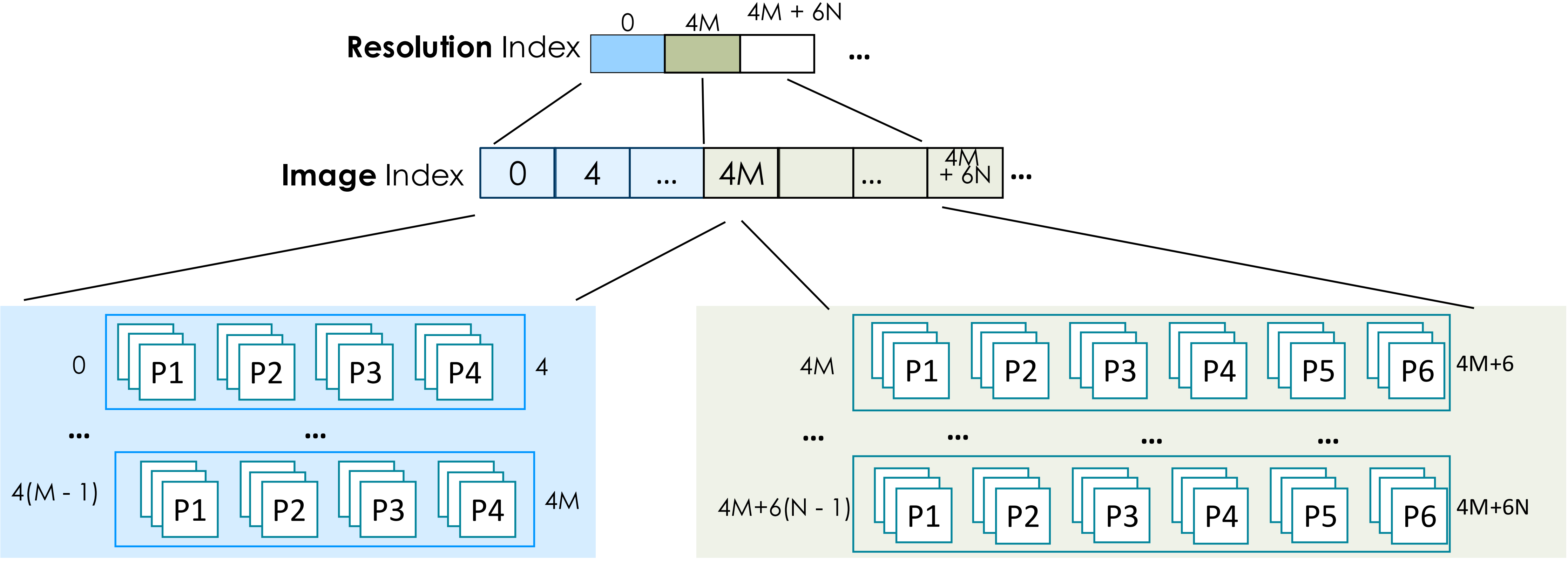
Patch-based Diffusion Serving [PPoPP'26]

- Architecture of **Image Diffusion** Model.

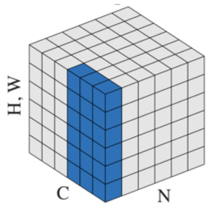


Note that Conv are omitted in Unit for simplicity.

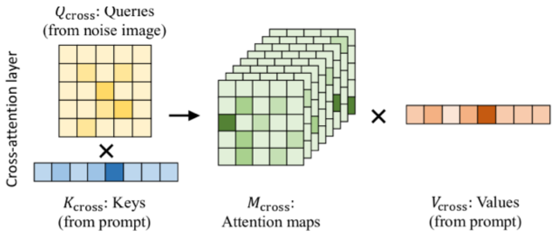
Hierarchical Tile Storage and Indexing



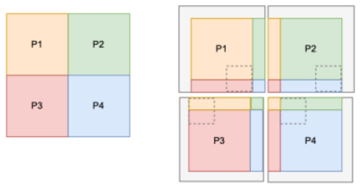
Group Norm



Cross Attention



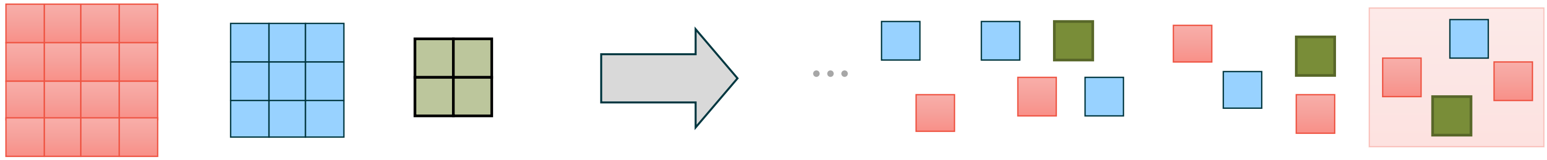
Convolution



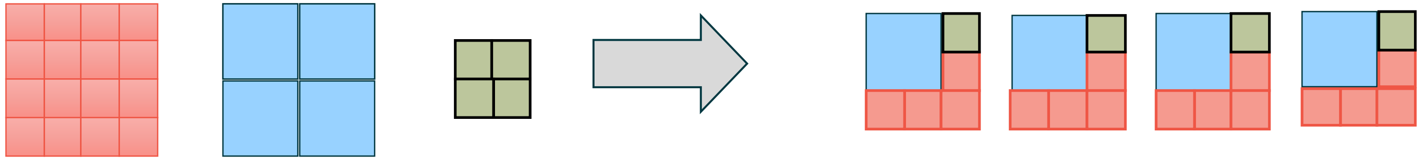
Note that the common factor of C*H*W is omitted for simplicity.

Tile-based Workload Scheduling

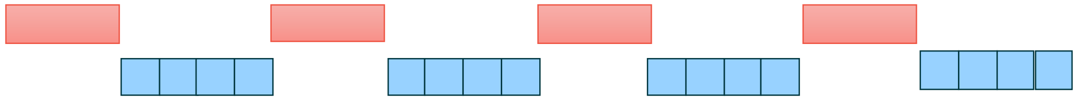
Unify the processing of different resolutions



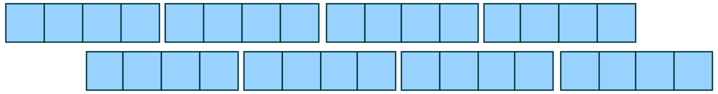
Control granularity for mixed workload composition



Unlock more fine-grained pipelining (FCFS)



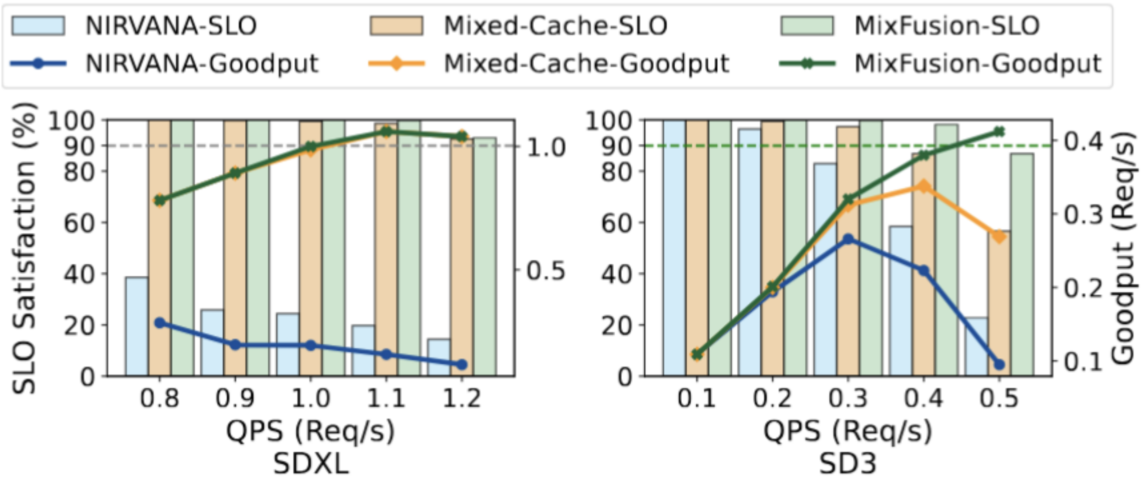
a. Resolution-mismatch for largely sequent processing.



b. Decomposed Resolution for batched processing.

Evaluation

- End-to-End SLO satisfaction Ratio



- Quality Score Comparison

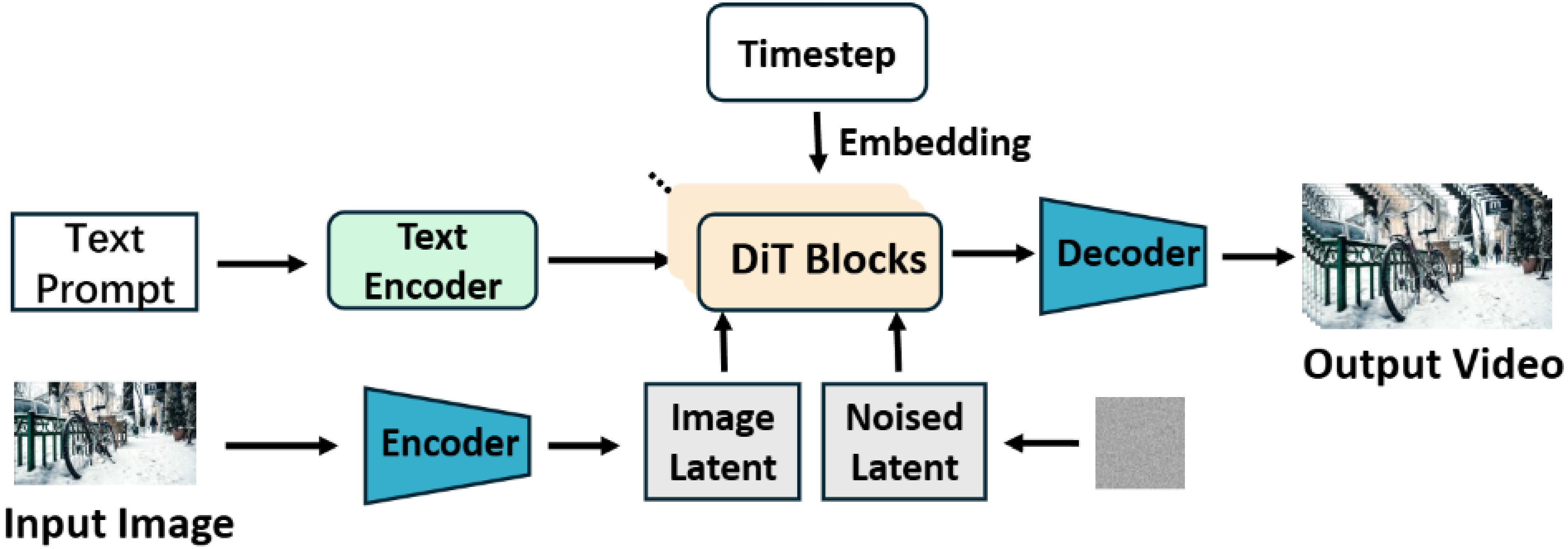
Model	Method	SDXL		SD3	
		COCO	diffusiondb	COCO	diffusiondb
CLIP (↑)	Original	14.92	16.24	14.79	16.65
	MixFusion	**15.43**	**16.62**	**15.13**	**17.06**
FID (↓)	Original	31.92	35.56	28.94	**32.38**
	MixFusion	**28.85**	**33.42**	**26.56**	38.01

5.33× higher goodput when achieving **90 % SLO** while over SOTA NIRVANA [1]

[1] Agarwal, Shubham, et al. "Approximate caching for efficiently serving Text-to-Image diffusion models." *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. 2024.

Ultra-Resolution Video Generation

- Architecture of Diffusion Transformer Model.



Observation of Ultra-high-resolution Video Generation

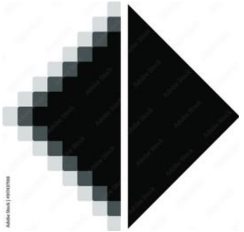
- Maximum supported resolution and running time

Model	Max Resolution	Frames	VRAM	Latency
CogVideoX1.5 5B [15]	1360 × 768	80	40GB	400s
HunyuanVideo 13B [20]	1280 × 720	128	70GB	1,800s

$$T_{4K} \approx 1800\text{s} \times \left(\frac{3840\times2160}{1280\times720}\right)^2 / 3600 \approx 40\ \text{h},$$

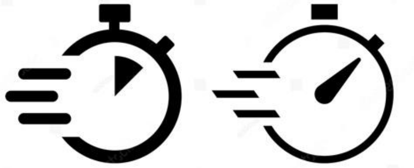
Why only supports 720p

Not enough high-resolution training datasets. (Billions)

Generation inefficiency

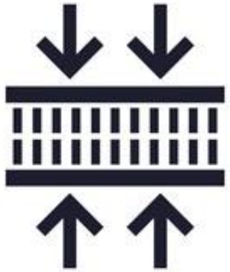
5-second 4K video directly could take a day and intensive memory cost.



How to training-free and efficiently to generate ultra-high-resolution video?

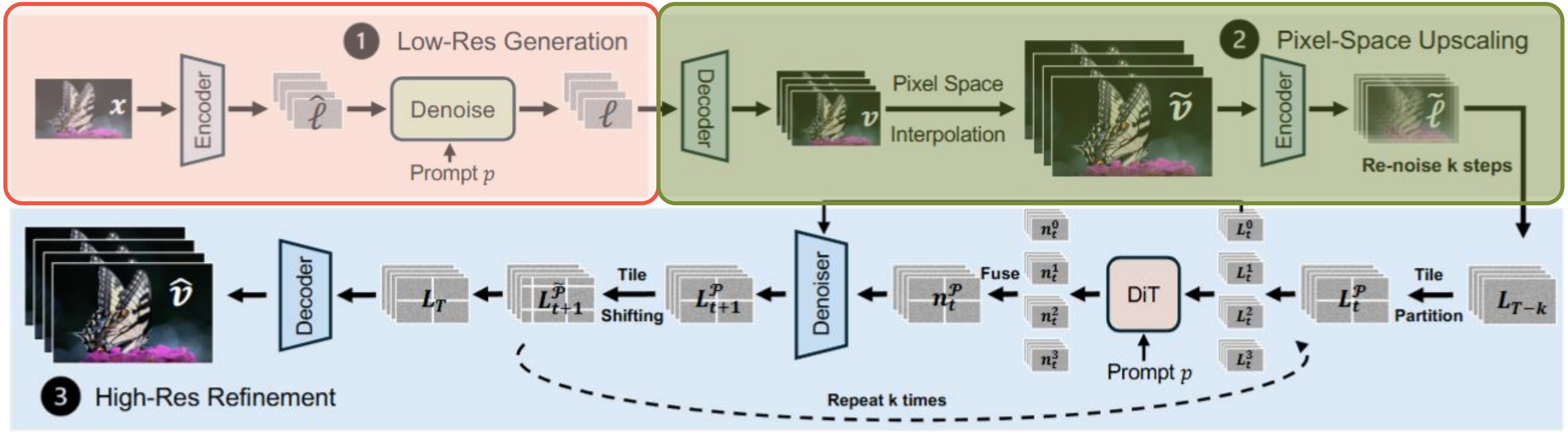
Current super-resolution video generation?

Only fixed scale factor, not training-free



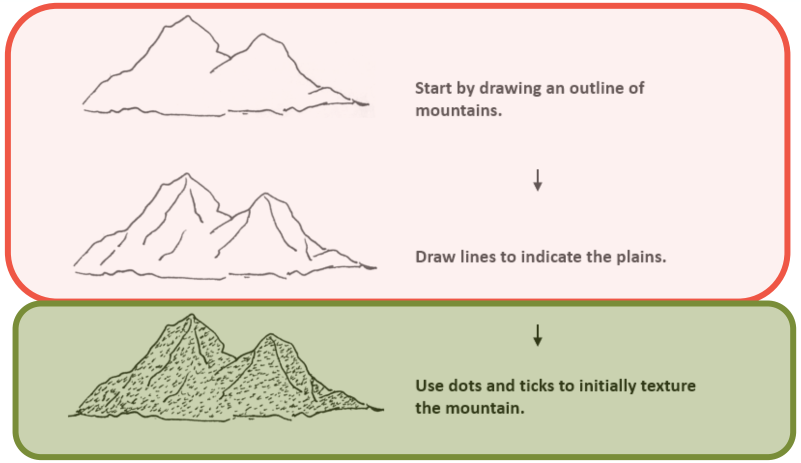
Opportunity: Sketching and Tiling

- Training-free Two-stage Generation



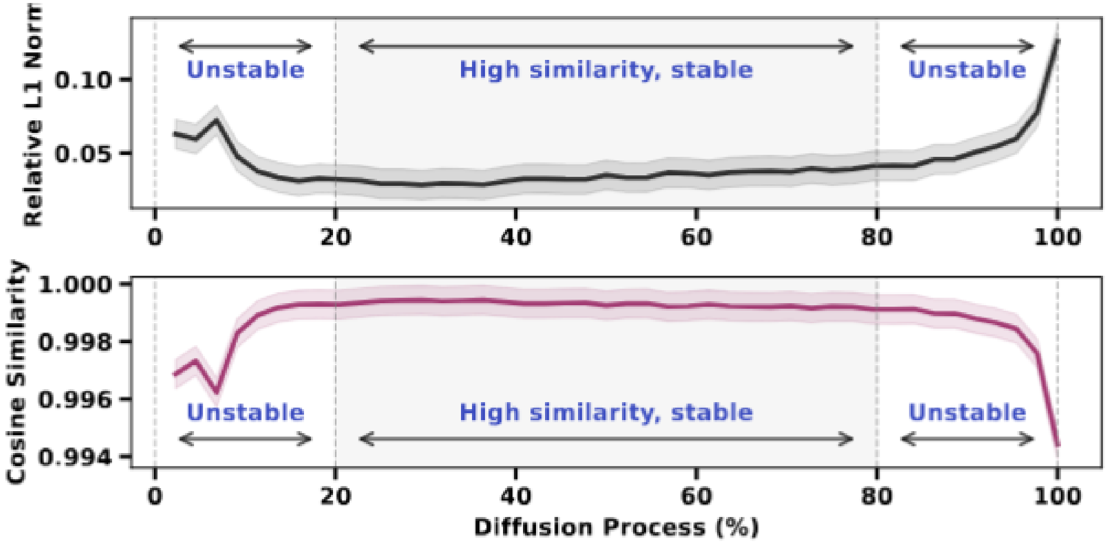
First stage: Global Semantic Guidance

Second stage: Local Details Refinement



System Support: Fine-grained Cache

- Observation: High similarity of predicted noise across timestep



$$\mathrm{L1}_{\mathrm{rel}}(\mathbf{O},t) = \frac{\|\mathbf{O}_t - \mathbf{O}_{t+1}\|_1}{\|\mathbf{O}_{t+1}\|_1}, \mathrm{CosSim}(\mathbf{O},t) = \frac{\langle \mathbf{O}_t, \mathbf{O}_{t+1} \rangle}{\|\mathbf{O}_t\|_2 \, \|\mathbf{O}_{t+1}\|_2}$$

How to Utilize this Similarity?

System Support: Fine-grained Caching Strategy

- Fine-grained Region-aware Cache

Introduce cache residual $\delta_t \triangleq O_t - I_t$

Approximate $O_t \approx I_t + \delta_c.$

Error Accumulation $E_{c\to t} = \sum_{k=c+1}^{t} \|O_k - O_{k-1}\|.$

Involve transformation rate $k_t = \frac{\|\mathbf{O}_t - \mathbf{O}_{t-1}\|}{\|\mathbf{I}_t - \mathbf{I}_{t-1}\|}.$

$$E_{c\to t} \approx \sum_{k=c+1}^{t} k_c \|I_k - I_{k-1}\| = k_c \sum_{k=c+1}^{t} \|I_k - I_{k-1}\| = k_c L_{c\to t},$$

Cache Decision

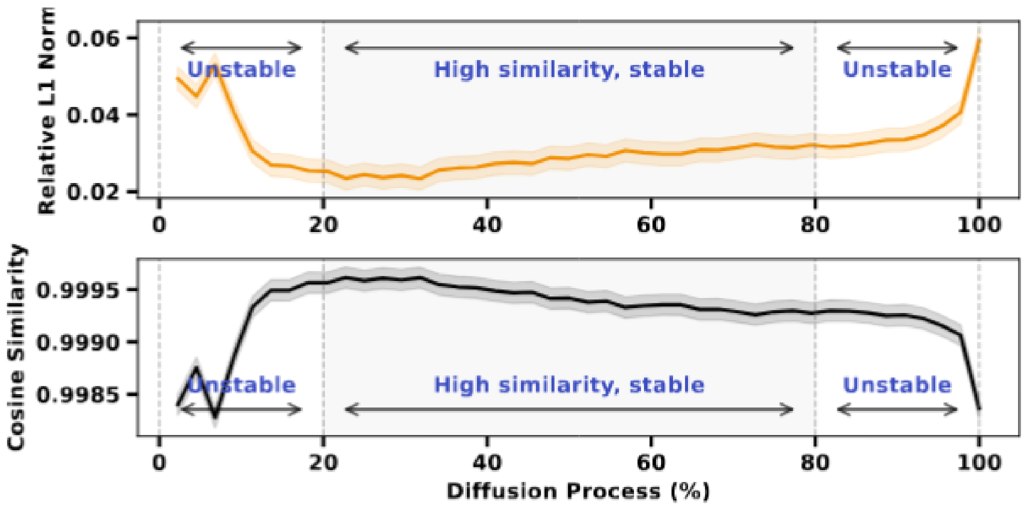
if $k_c L_{c\to t} < \tau$ $\Rightarrow$ reuse cache at step t,

else $k_c L_{c\to t} \geq \tau$ $\Rightarrow$ recompute O_t and set $c \leftarrow t$.

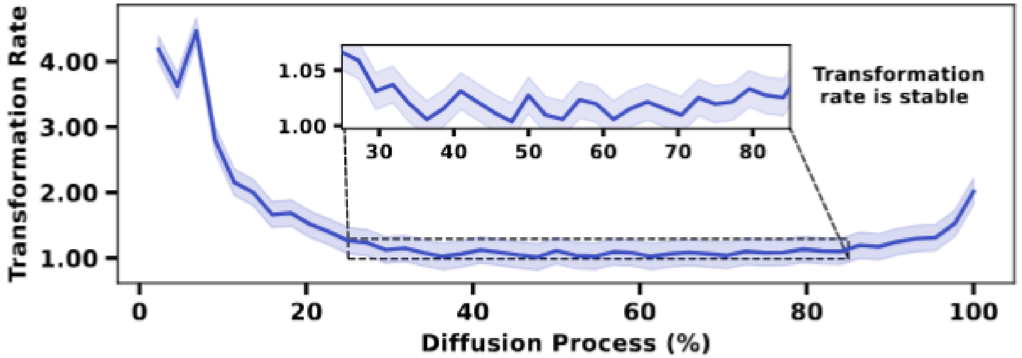
1) **Predict** next diffusion step **outcome** based on current outcome.

2) **Accumulate** the prediction **errors** to determine caching.

High similarity of Cache Residual

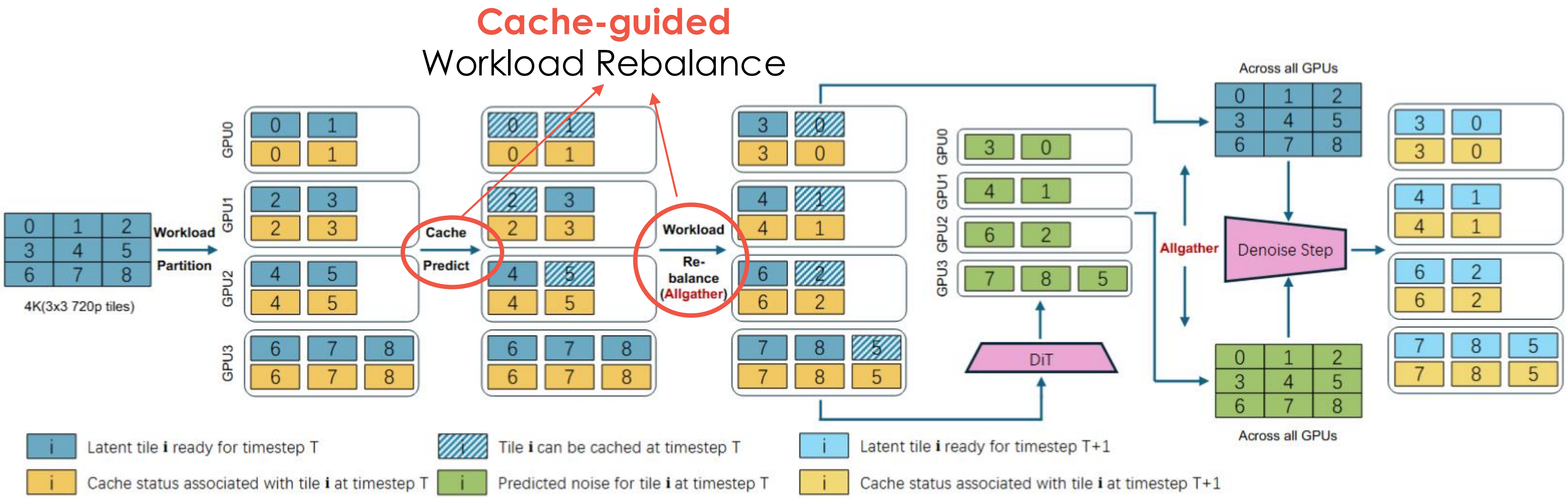


Transformation Rate K is Stable



System Support: Intelligent Cost-efficient parallelism

Workload imbalance: e.g., 9 tiles on 4 GPUs



QKV in different tile are **independent**,
Minimum communication.

1) Reuse the computed tiles in **nearby GPUs**.

2) Reuse the computed tiles in from **prior iteration**.

Evaluation: Video Quality

Table 2: Quality results of SUPERGEN on VBench benchmark. V1–V5 denote the five evaluation metrics: **V1**: Subject Consistency, **V2**: Background Consistency, **V3**: Motion Smoothness, **V4**: Aesthetic Quality, and **V5**: Imaging Quality.

Model	Setting	V1(%)	V2(%)	V3(%)	V4(%)	V5(%)	Avg.
Cogvideo	720p	96.29	96.23	98.41	61.88	70.20	**84.60**
	2K w/o Cache	95.66	96.06	97.22	63.86	70.38	**84.64**
	2K w/ Cache	95.45	95.91	97.21	62.75	69.75	**84.21**
	4K w/o Cache	92.94	94.11	98.10	57.92	67.38	**82.09**
	4K w/ Cache	93.22	94.32	98.04	57.95	67.56	**82.22**
Hunyuanvideo	720p	98.55	97.86	99.53	64.54	70.83	**86.26**
	2K w/o Cache	98.02	97.31	99.42	66.47	69.62	**86.17**
	2K w/ Cache	98.30	97.48	99.44	66.16	70.26	**86.33**
	4K w/o Cache	97.76	97.24	99.43	63.07	69.68	**85.44**
	4K w/ Cache	98.12	97.58	99.51	62.57	70.31	**85.62**

Both 2K and 4K can achieve **high quality**.

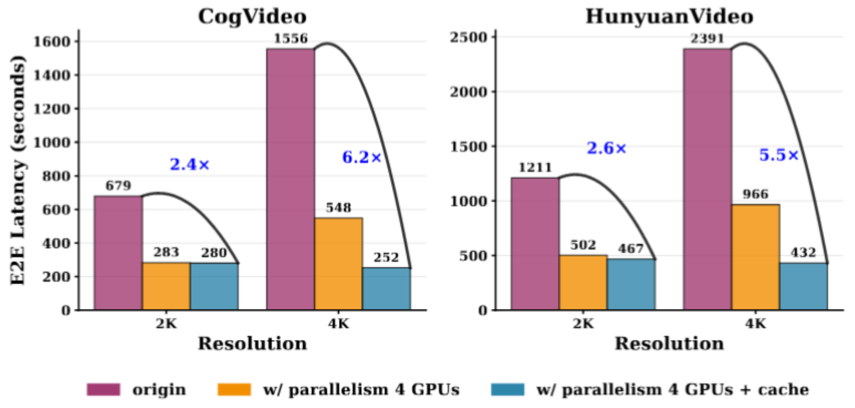
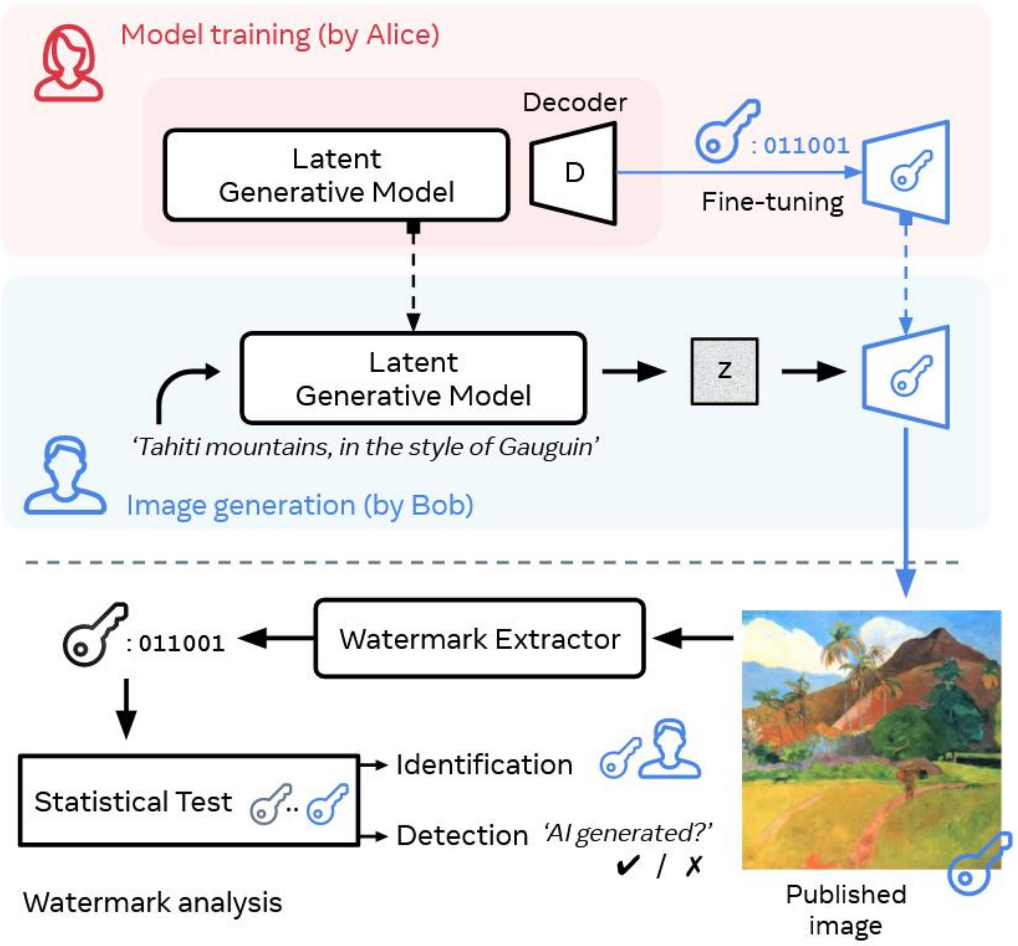


Figure 12: End-to-end latency. Origin setting is evaluated with 1 GPU without cache. The other two settings with parallelism are measured with workload rebalance.

End-to-end can achieve up to **6.2x** speedup.

Efficient and Adaptive Watermark Detection

- Overview of diffusion image watermarking



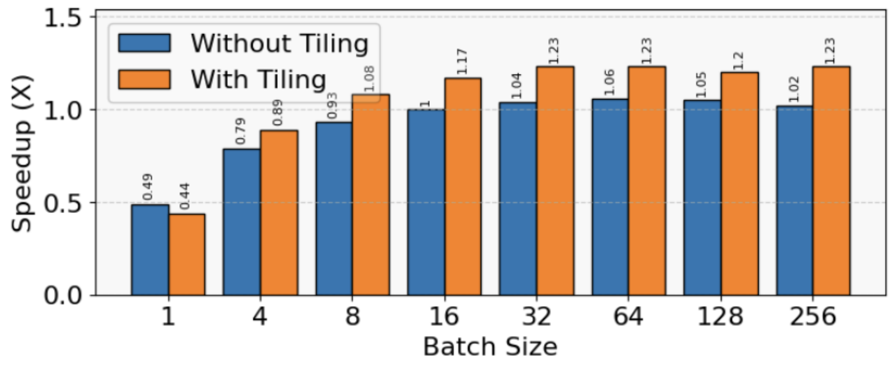
- Diffusion models now generate images nearly **indistinguishable** from real photos.
- Social platforms face **billions** of daily uploads (YouTube, TikTok, Facebook).
- Watermarking is critical for **verifying** AI-generated content and copyright attribution.

Challenges of Diffusion Watermarking Detection

- Validation accuracy of naive tile-based Stable Signature on different tile size.

None	128	96	80	64	48	32	16
0.997	0.960	0.933	0.897	0.875	0.804	0.714	0.624

- Naive tile-based design brings only limited improvement.



Challenge-1: How to balance the trade-off between accuracy and efficiency?

Challenge-2: How to optimize the tile-based watermark detection pipeline for higher throughput?

Motivation from Real-world Observations

Observation1: **One tile** of an entire image is **sufficient** for watermark detection.

None

Crop 10%

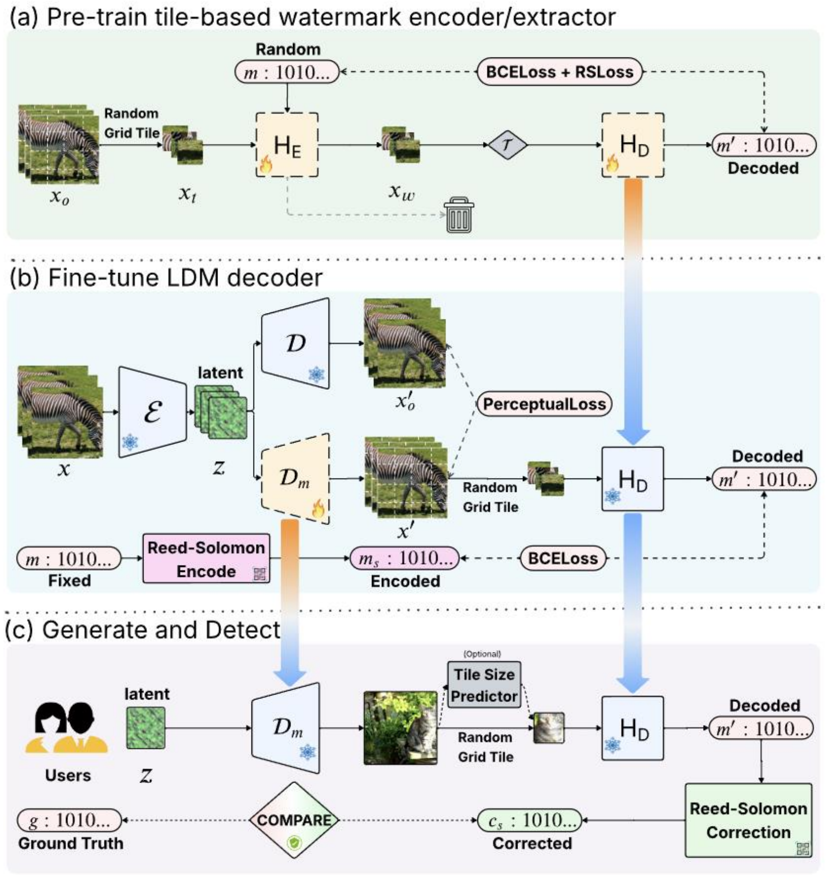
Accuracy reduction: 99% ->95%
(even without pretraining on tiles)

Observation2: A QR code can still be **accurately** and **quickly** even when part of it is **missing**.

https://en.m.wikipedia.org/wiki/Main_Page

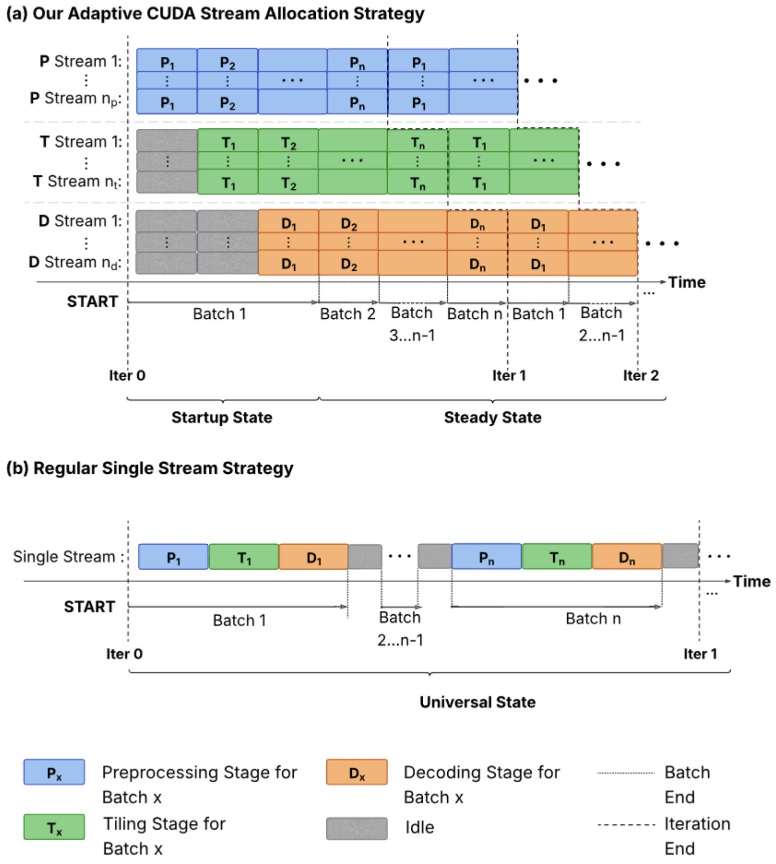
Algorithmic-System Co-Design Solution

- Tile-based detection with RS Correction



Leverage **RS correction** to offset the accuracy degradation caused by **tiling**

- Adaptively allocate CUDA stream for each stage



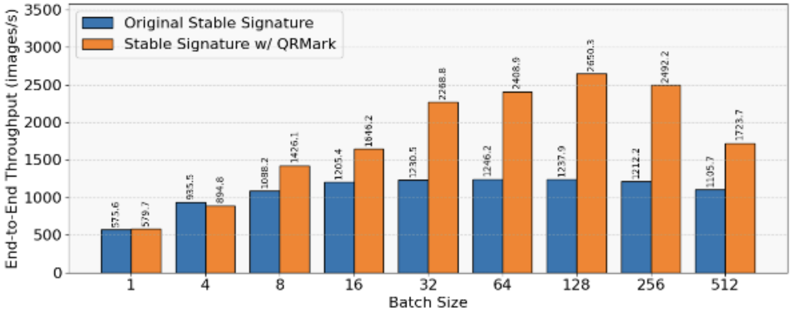
Equalize the **per-minibatch execution** time across stages

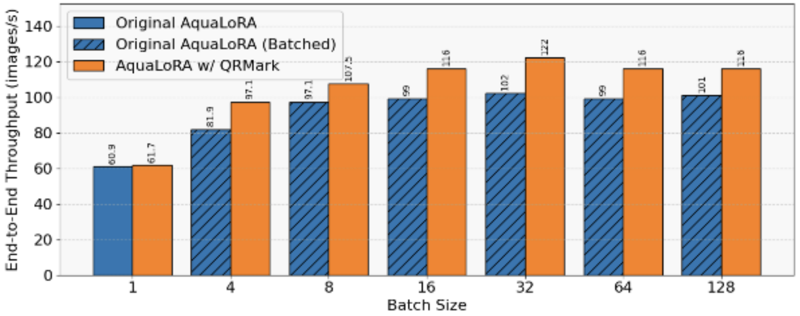
End-to-end Performance

- End-to-end accuracy and robustness

Model	TS	BitAcc. ↑	BitAcc. (ADV.) ↑	PSNR ↑	TPR ↑
Stable_{QR}	16	0.748	0.665	27.67	0.761
	32	0.989	0.907	29.47	0.993
	48	0.997	0.933	29.63	0.996
	64	0.999	0.945	30.35	0.998
	80	0.999	0.949	30.76	0.999
Stable_{BL}	–	0.999	0.974	30.05	0.993
AquaLoRA_{QR}	256	0.947	0.883	17.13	0.970
AquaLoRA_{BL}	–	0.958	0.912	17.65	0.985

- End-to-end Throughput

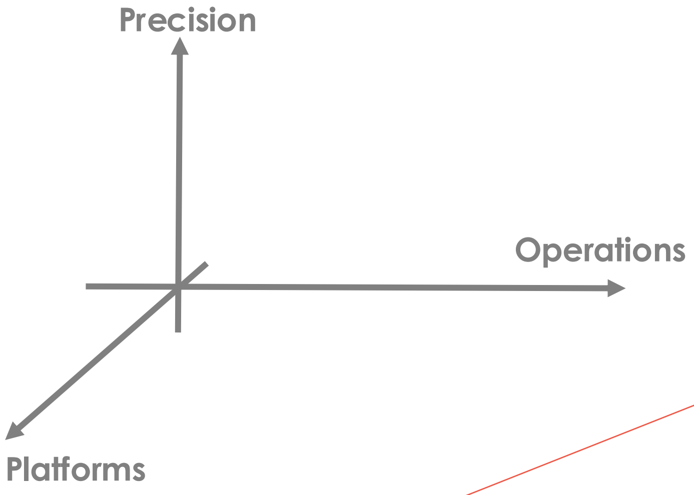




2.20x over Stable Signature [1] and **2X** over AquaLoRA [2].

[1] Fernandez, Pierre, et al. "The stable signature: Rooting watermarks in latent diffusion models." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.
[2] Feng, Weitao, et al. "Aqualora: Toward white-box protection for customized stable diffusion models via watermark lora." *arXiv preprint*

Future Research: New Hardware-System Optimization



Portability (Sustainable AI)

New **Hardware-System** Designs that adapt to diverse real-world application settings.

Efficient DLRM with **Disaggregated Memory (e.g., CXL)**

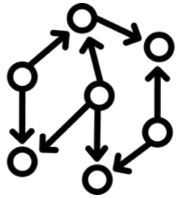
- Co-design/optimization with diverse accelerators.
- Co-scheduling Workloads with Near-data Processing.
- Multi-Tenant Support for co-locating Diverse DL applications.

Eco-friendly DL with **Energy/Carbon-aware HW-System** Co-Optimization

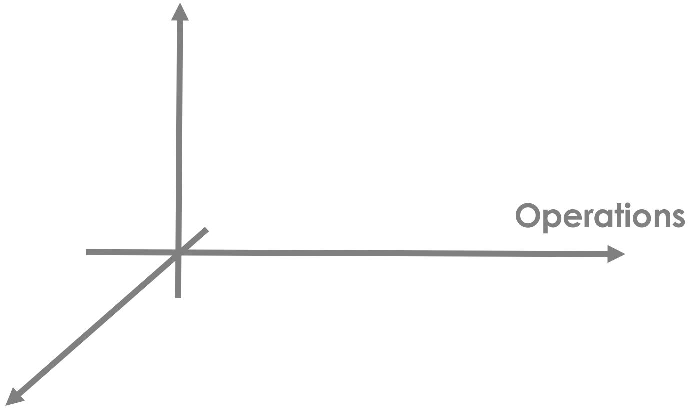
- Dynamic Energy Scaling to balance runtime performance and energy cost.
- Carbon-Aware Scheduling to balance energy availability and job requirements.

Accelerated DL with **Reconfigurable Dataflow Architecture** (e.g., SambaNova)

- Compiler optimization for dataflow design space search.
- Workload-aware Runtime Dataflow Reconfiguration.



Future Research: Exploring New DL Workloads



+

Portability (Sustainable AI)

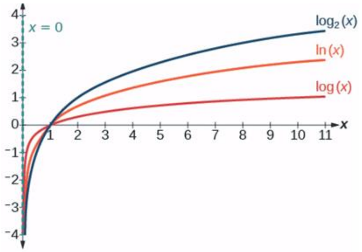
Satisfaction (Human-Centric AI)

New **Algorithm-System** designs that can intelligently suit various demands.

Patch-based
Diffusion Model Serving

- SLO-aware dynamic re-patching for efficient parallelism.
- Patch-based early exit for efficient diffusion.

Resource-Efficient
DL with Logarithmic Number System (LNS)

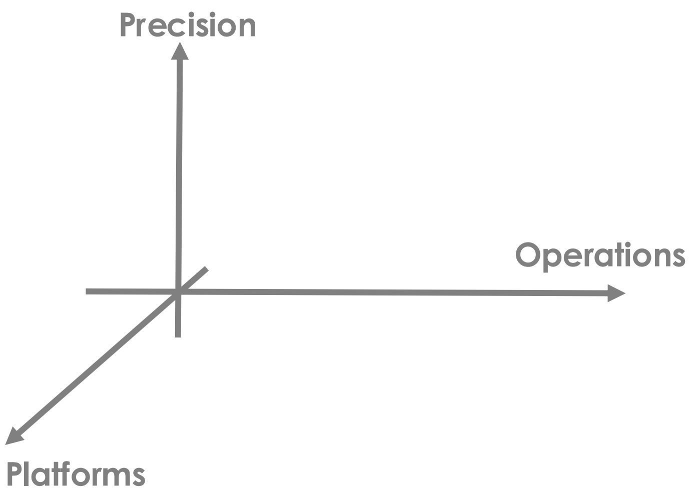


- Dynamic DL precision with multi-base LNS.
- Memory-efficient model quantization with selective LNS.

Locality-aware
Text-to-3D Scene Construction

- Cross-iteration spatial and temporal in 360 Panorama rendering and 3D scene lifting.
- Cross device (e.g., multi-GPU platforms) objects artifacts locality.

Future Future: Secure and Resilient DL System



+

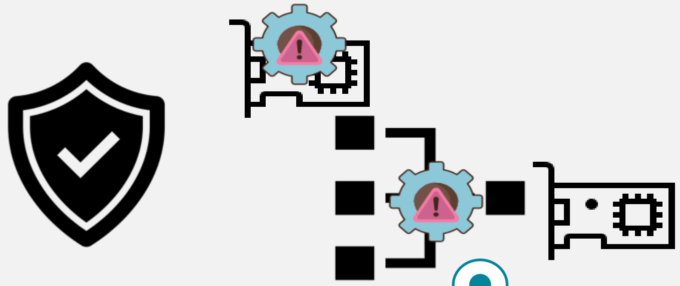
...

Portability (Sustainable AI)

Ne
tc

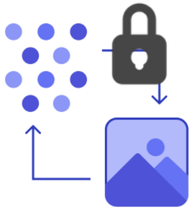
Satisfaction (Human-Centric AI)

Safety (Trustworthy AI)



System that are **Robust** and **Resilient.**

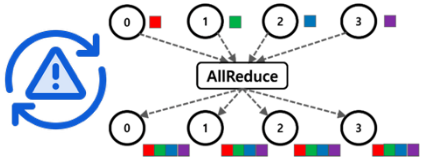
Confidential/ Encrypted DL Model Serving



- Holistic scheduling of encryption and regular DL ops for low-cost DL model training.

Failure-resilient Collective Communication

- Failure-resilient routine for heterogenous memory and compute devices.



Efficient **Watermarking** for IP-Protected DL.

- Robustness-aware Watermarking for efficient real-time processing.

RICE UNIVERSITY

Thank You

Q & A

yuke.wang@rice.edu

github.com/YukeWang96

wang-yuke.com