ASK 2025 논문집

Annual Symposium of KIPS 2025

튜토리얼

On-device Sora: Enabling Training-Free Diffusion-based Text-to-Video Generation for Mobile Devices

이슬기 조교수(UNIST)

On-device Sora:
Enabling Training-Free Diffusion-based Text-to-Video Generation for Mobile Devices

Seulki Lee (이슬기)
Ulsan National Institute of Science and Technology (UNIST)
Department of Computer Science and Engineering
AI Graduate School

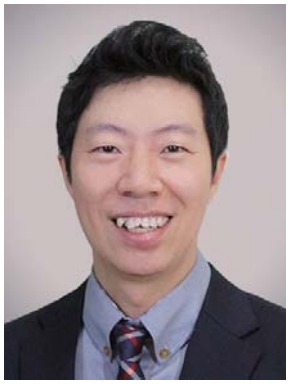
CONTACT

Ulsan National Institute of Science and Technology
Address 50 UNIST-gil, Ulju-gun, Ulsan, 44919, Korea
Tel. +82 52 217 0114 **Web.** www.unist.ac.kr

Computer Science and AI Graduate School
106 3rd Engineering Bldg
Tel. +82 52 217 6333 **Web.** https://cse.unist.ac.kr/

Introduction

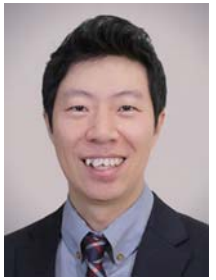
Seulki Lee (이슬기)



- **Ulsan National Institute of Science and Technology**
  - Assistant Professor [2021~]
    - Computer Science and Engineering
    - AI Graduate School
- **Research Area**
  - Embedded Systems, On-device AI, Real-Time Computing, AIoT
  - Mobile Computing, Cyber-Physical Systems
- **Experience**
  - Nokia Bell Labs [2018]
  - Mercedes-Benz Research & Development North America [2017]
  - Samsung Electronics [2009~2016]
- **Education**
  - Ph.D. in Computer Science (University of North Carolina at Chapel Hill) [2021]

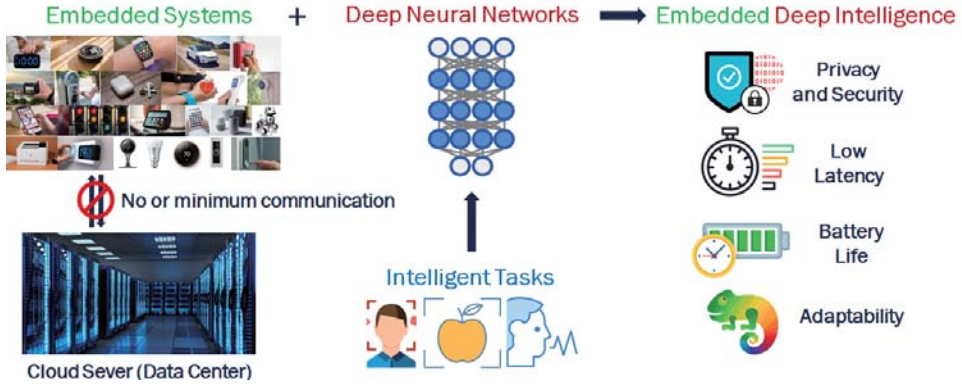
Embedded Artificial Intelligence Lab (EAI Lab)

Seulki Lee
CSE/AIGS
UNIST



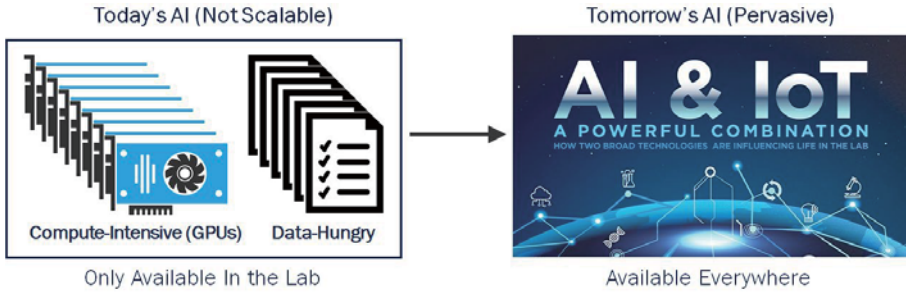
Homepage: https://sites.google.com/view/embeddedai/home

Embedded AI = Embedded Systems + Artificial Intelligence (AI)

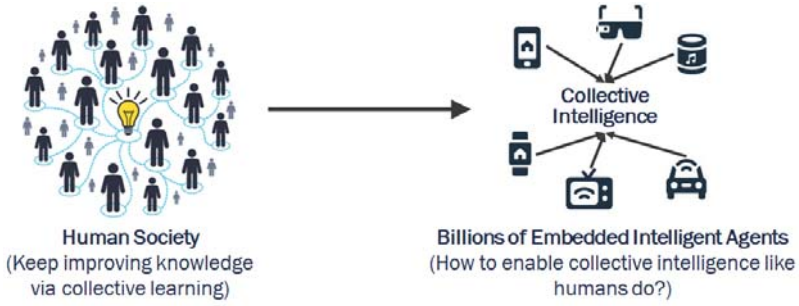


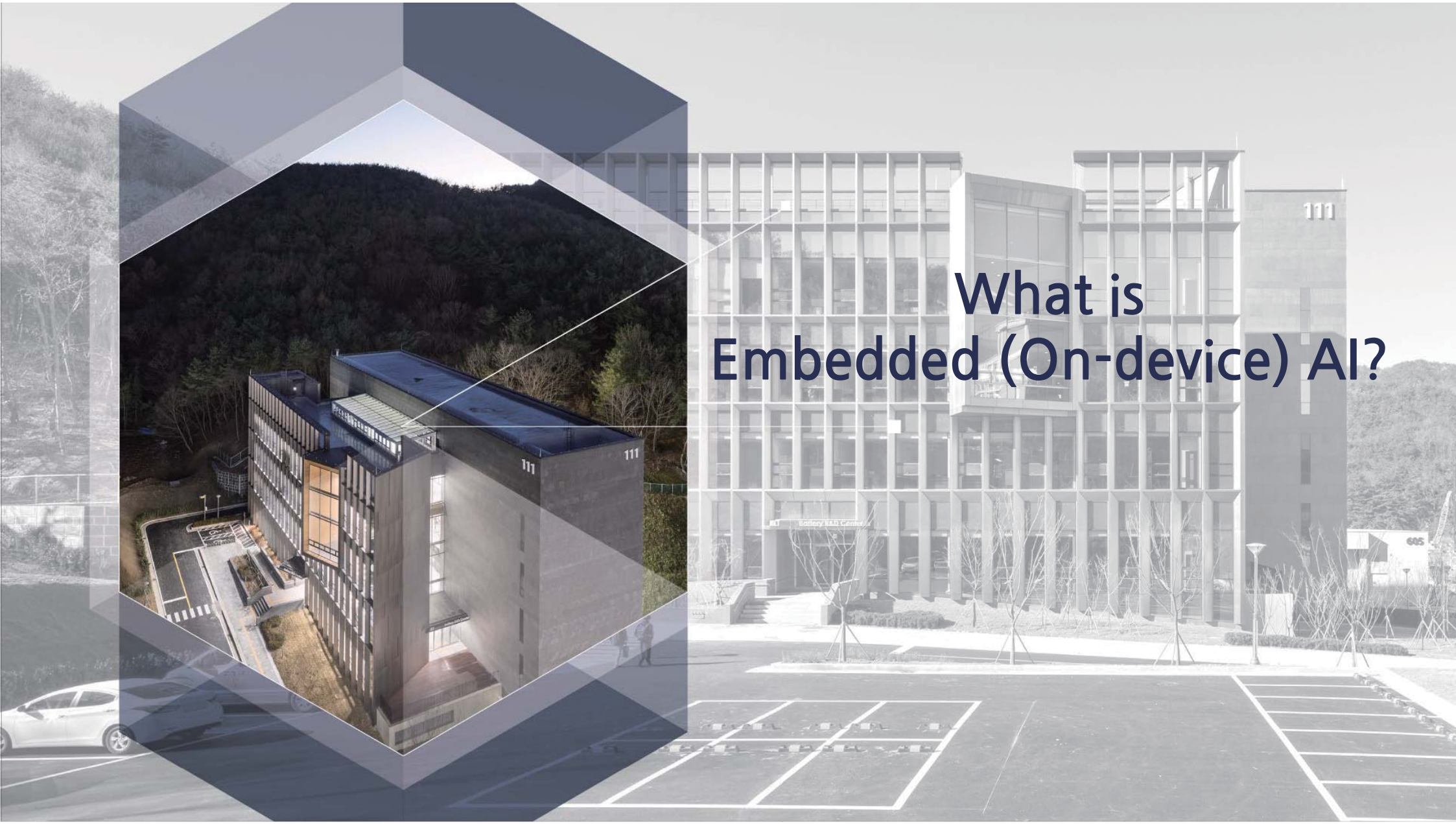
- **Embedded Artificial Intelligence (Embedded AI)**
  - On-device machine learning on embedded/mobile/IoT devices
  - Embedded computer vision, Embedded generative AI, Embedded NLP, Embedded reinforcement learning
  - Efficient model (deep learning) inference, training, and adaptation
  - Next-generation efficient learning model (beyond deep learning)
- **Collaborative and Collective Learning Systems**
  - Pervasive- and always-learning via connected systems
  - Scalable intelligence on networked systems (AIoT and intelligent edge)
  - Mobile computing and sensor data analytics

Pervasive, Scalable, and Everywhere AI



Collaborative and collective learning on connected systems





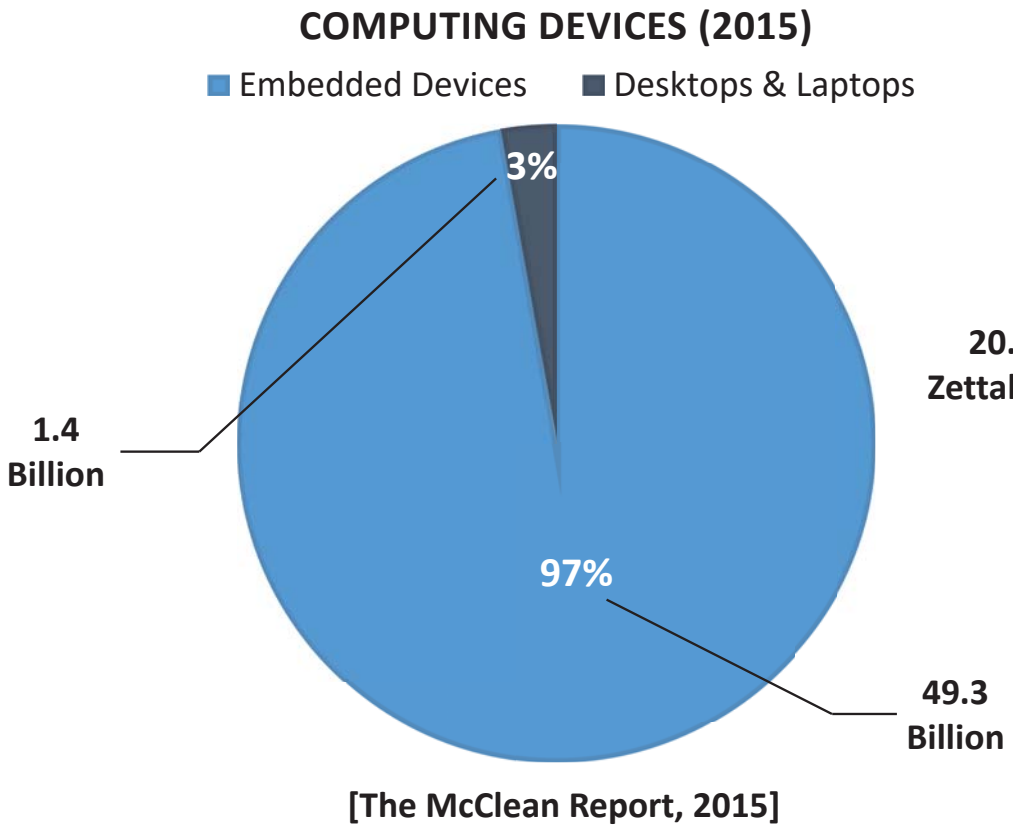
What is Embedded (On-device) AI?

Embedded/IoT/Mobile Systems are Everywhere

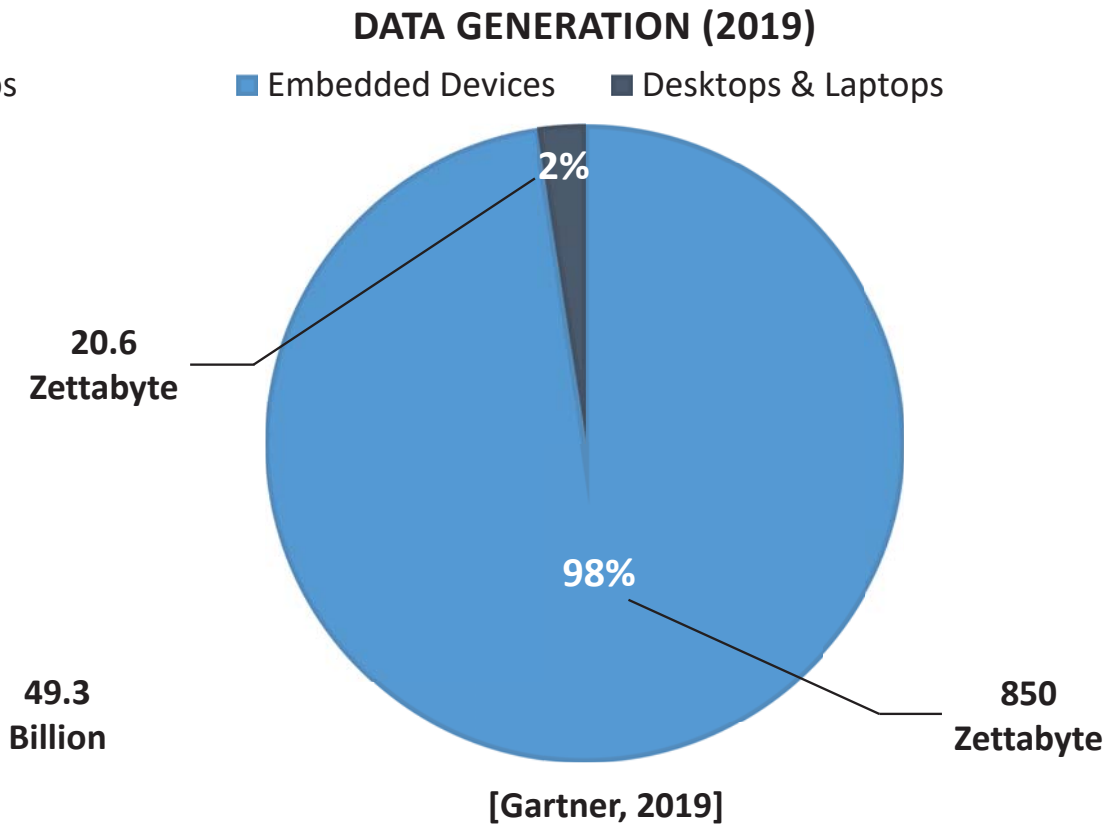
Smartphones, robots, self-driving cars, smart watches, IoT devices …

Embedded Systems Have Become the Major Computing Platform

They far outnumber the traditional computing platforms

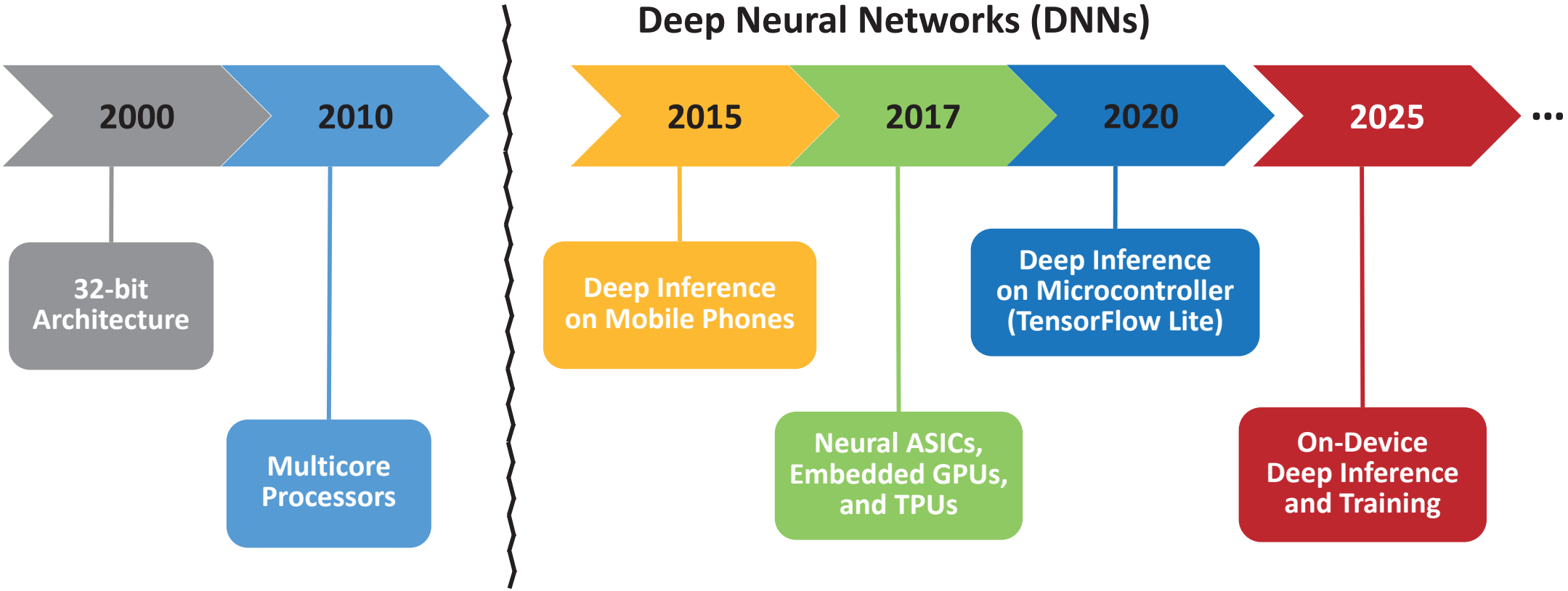


[The McClean Report, 2015]



[Gartner, 2019]

Embedded Systems - Embracing AI



Embedded AI Can Enable a Truly AI-Pervasive World

- AI can be democratized and widespread when AI can run on commodity embedded computing devices, e.g., AIoT (AI + IoT).

Today's AI (Not Scalable)

Compute-Intensive (GPUs)

Data-Hungry

Limited Availability

Needs To evolve!

Tomorrow's AI (Pervasive)

Available Everywhere

Embedded Artificial Intelligence (Embedded AI)

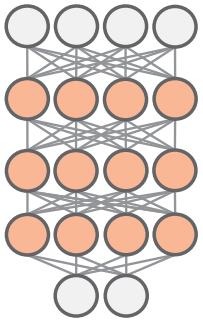
Embedded Deep Intelligence

- On-Device Deep Learning (inference and training) on Embedded Systems

Embedded Systems + **Deep Neural Networks** ⟶ **Embedded Deep Intelligence**

No or minimum communication

Cloud Sever



Intelligent Tasks

Privacy and Security

Low Latency

Battery Life

Adaptability

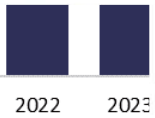
Emb

Globa

Big Tech Pushing On-Device AI as Privacy, Performance Booster

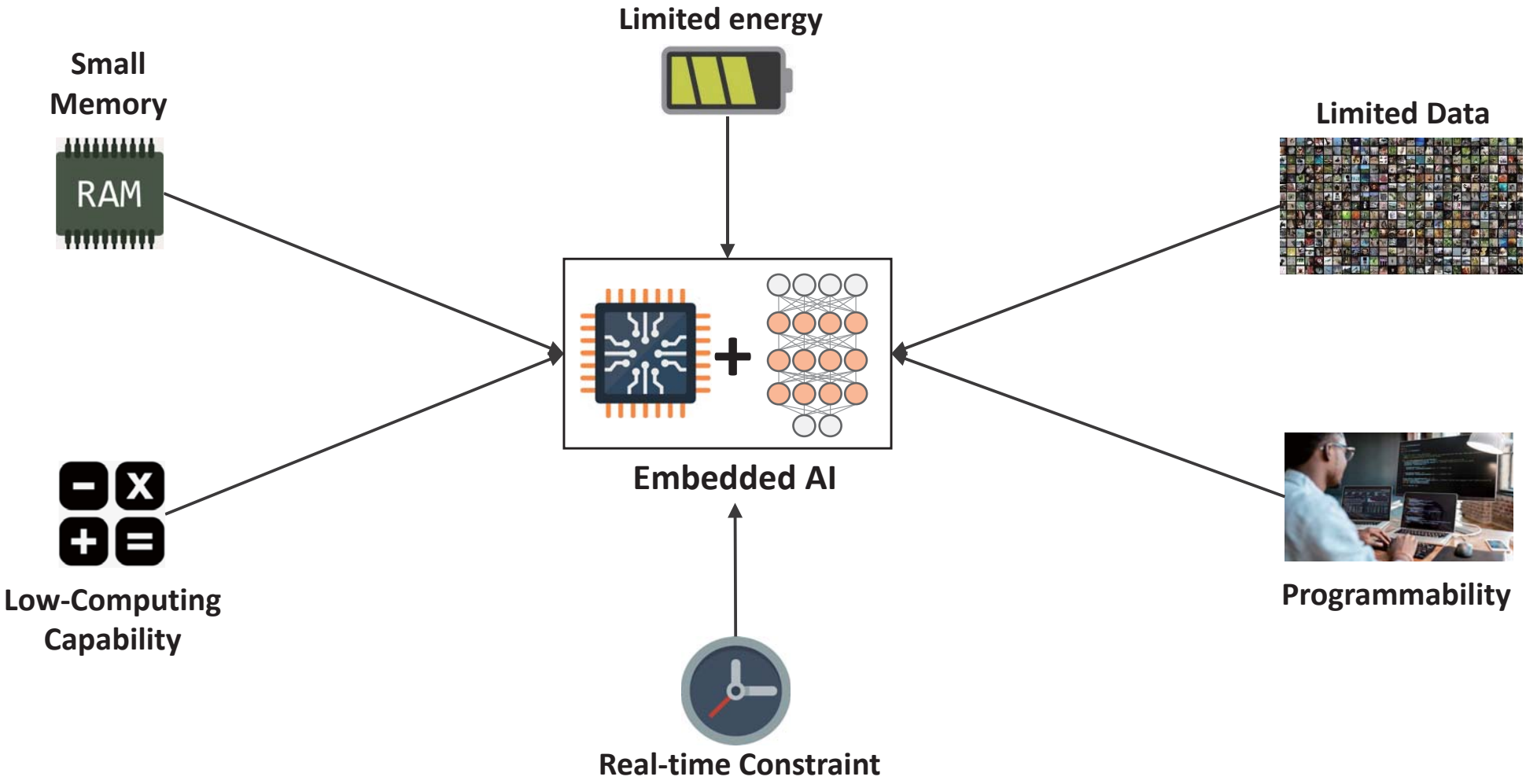
Artificial intelligence and other technology companies are pushing their large language models out of the cloud and onto users' personal devices in a move they say will enhance privacy and security.

By **Priya Khosla** June 27, 2024



Challenges to Embedded Artificial Intelligence

Memory, computing capability, real-time constraint, energy, limited data, programmability, interpretability, …

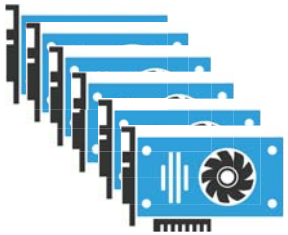


On-device Sora: Enabling Training-Free Diffusion-based Text-to-Video Generation for Mobile Devices

Sora

Currently, video generative technology remains neither widely accessible nor commonly available

- The high complexity of diffusion processes for video generation, combined with the gigantic size of DiT models, imposes significant computation and memory demands.
  - Sora is estimated to take an hour to produce five minutes of video using H100 GPUs.



A bunch of
H100 GPUs

Sora running on a datacenter → OK

iPhone 15 Pro
(**143x slower processor**
14x smaller memory)

Sora running in your palm → ??

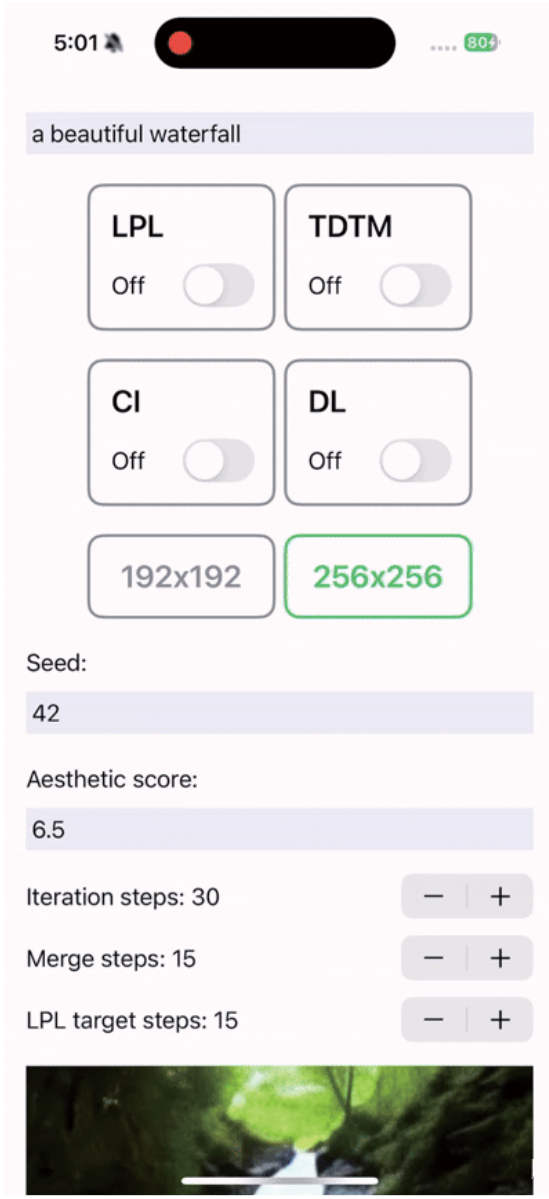
On-device Sora

Enabling Diffusion-based Text-to-Video Generation for Mobile Devices

- **On-device Sora**
  - The first model-training-free diffusion-based on-device text-to-video generation on smartphone-grade devices.
  - Working without cloud (server) connections.

Prompt:
“A beautiful waterfall”

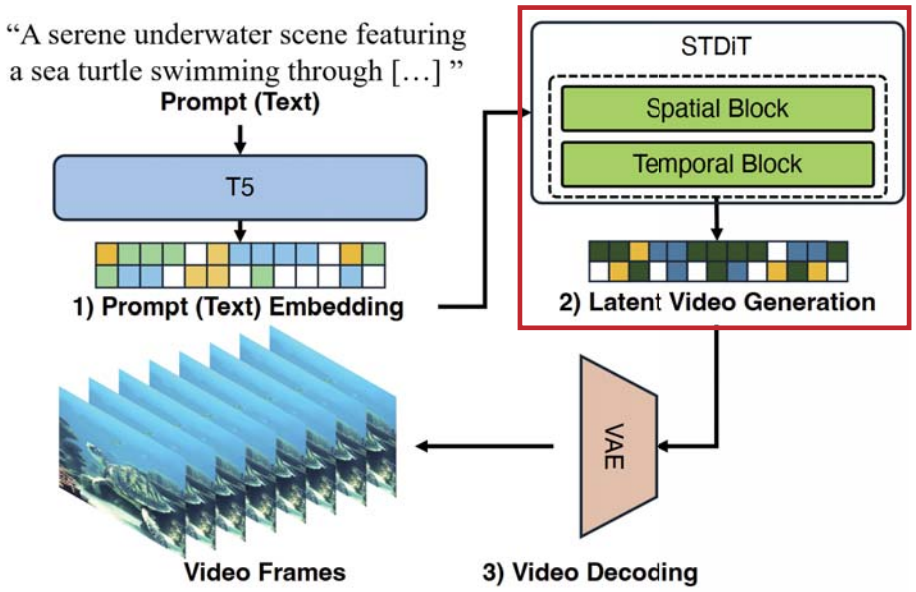
iPhone 15 Pro

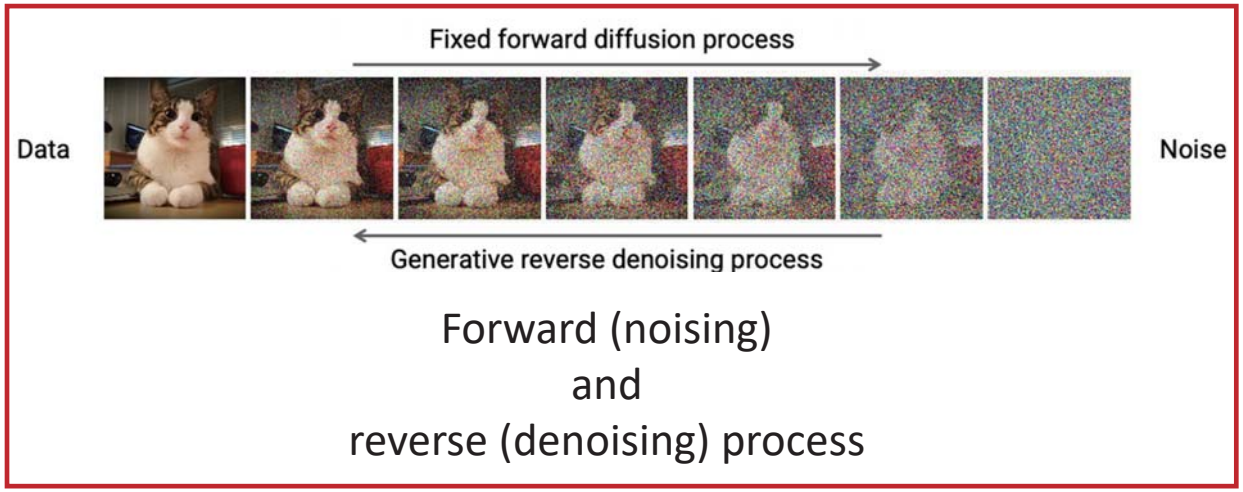


Background: Diffusion-based Text-to-Video Generation

Prompt (text) embedding → Generating the latent video representation conditioned on the prompt → Video decoding

- By using DiT (Diffusion Transformer), videos are generated from prompts (texts) through:
  1. Prompt (text) embedding
  2. Latent video generation
  3. Video decoding





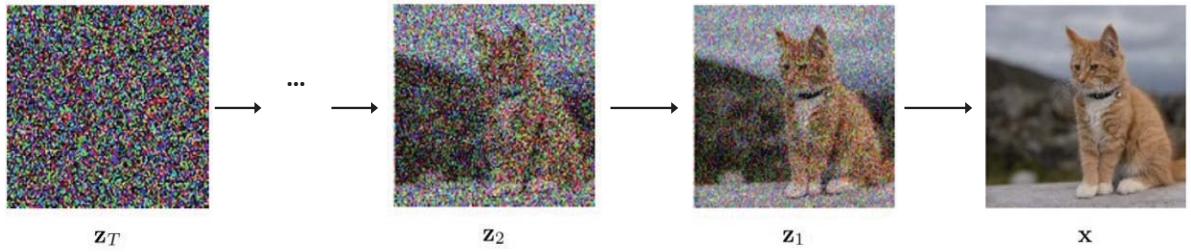
Challenges in On-device Video Generation

Adapting advanced generative capability (e.g., Sora) to mobile devices presents key challenges

- **Challenge 1: Excessive Denoising Steps**
  - The denoising process performed by STDiT is the most time-consuming.
  - A substantial number of denoising steps is required to remove the noise during latent video generation.

Component	Iterations	Inference Time (s)	Total Latency (s)
T5 [52]	1	110.505	110.505
STDiT [82]	50	35.366	1768.320
VAE [17]	1	135.047	135.047

The number of executions (iterations) of each model component (i.e., T5, STDiT, and VAE) in Open- Sora and their total latencies on iPhone 15 Pro.

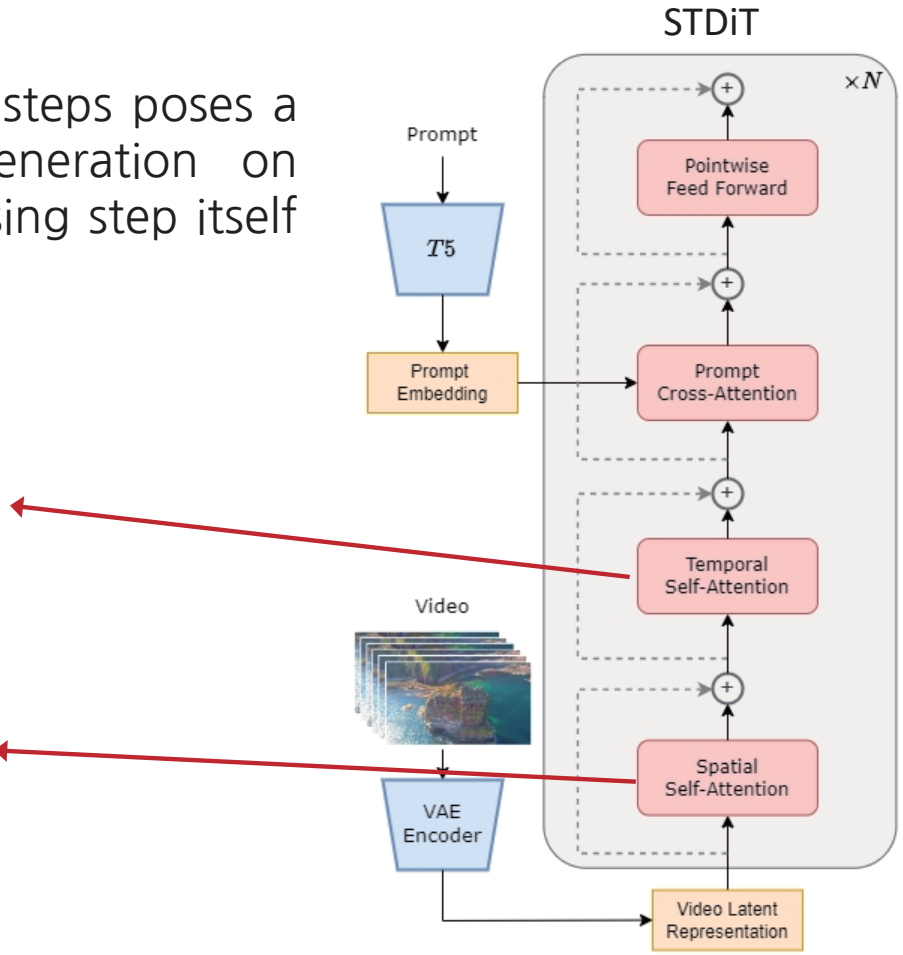


Several tens or hundreds of denoising steps are required to generate videos

Challenges in On-device Video Generation

Adapting advanced generative capability (e.g., Sora) to mobile devices presents key challenges

- **Challenge 2: Intensive Token Processing**
  - While a large number of denoising steps poses a significant challenge to video generation on mobile devices, even a single denoising step itself is computationally intensive.



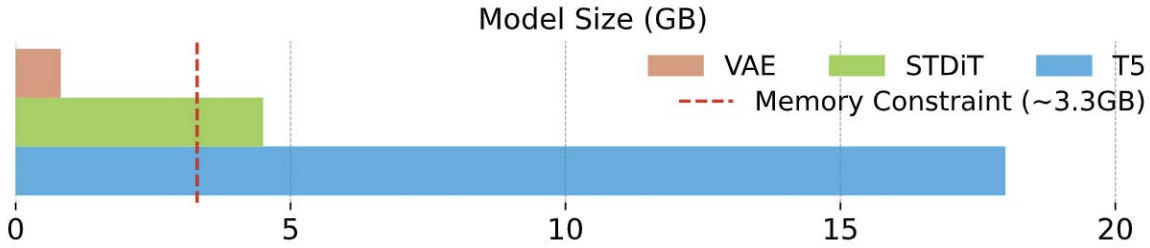
Computation grows quadratically, i.e. $O(n^2)$

Existing token merging methods only apply to self-attention, not cross-attention

Challenges in On-device Video Generation

Adapting advanced generative capability (e.g., Sora) to mobile devices presents key challenges

- **Challenge 3: High Memory Requirements**
  - The cumulative memory demand, i.e., 23 GB, surpasses the memory capacity of many mobile devices (3.3 GB of iPhone 15 Pro).
  - Even the memory requirements of T5 and STDiT exceed 3.3 GB.
  - Some memory must be reserved for model execution (inference).



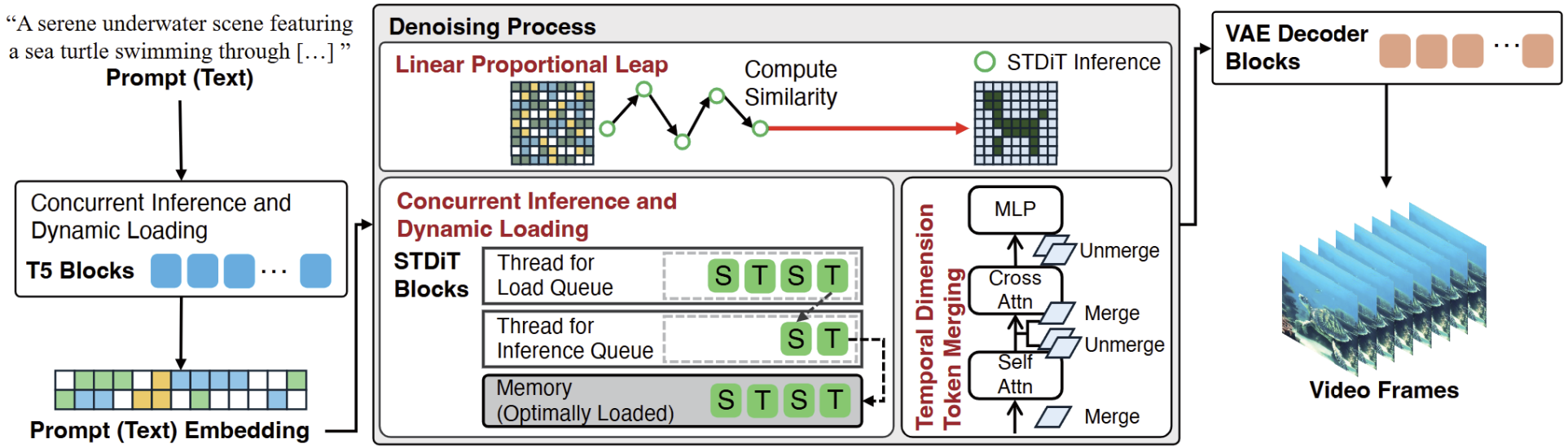
The memory requirements of model components required for video generation: T5 (**18.00 GB**), STDiT (**4.50 GB**), and VAE (**0.82 GB**), which exceeds the available memory capacity of iPhone 15 Pro (**3.3 GB**).

iPhone 15 Pro (**3.3 GB**)

Overview of On-device Sora

Linear Proportional Leap, Temporal Dimension Token Merging, and Concurrent Inference and Dynamic Loading

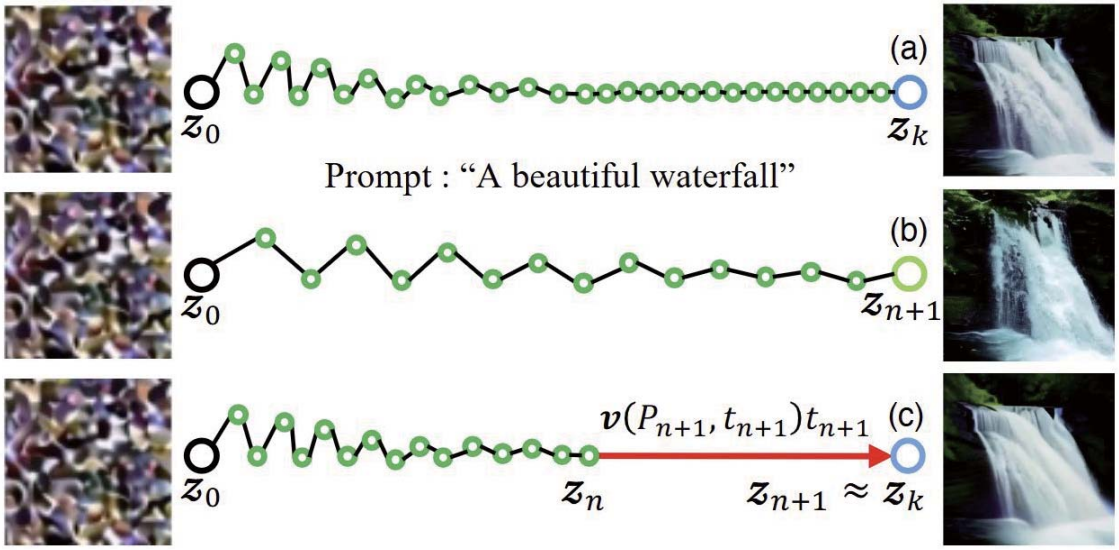
- On-device Sora applies three techniques to address the challenges of video generation on computation- and memory-limited mobile devices without model compression.
  1. **Linear Proportional Leap (LPL)**: reducing the excessive denoising steps
  2. **Temporal Dimension Token Merging (TDTM)**: minimizing token-processing compute
  3. **Concurrent Inference with Dynamic Loading (CI-DL)**: addressing the challenges of limited device memory



1. Linear Proportional Leap (LPL)

Reducing the excessive number of denoising steps without model training, architecture modifications, or data calibration

- Enabling the generation of high-quality videos with half of the required full denoising steps
  - Instead of performing full denoising steps, **it makes a direct leap along the linear trajectory toward the target** by utilizing the pre-trained flow fields.



$$z_k = z_{k-1} + v(P_k, t_k)dt_k$$

$$= z_0 + \sum_{i=1}^{k-1} v(P_i, t_i)(t_i - t_{i+1}) + v(P_k, t_k)t_k$$

apply the identical drift $v(P_{n+1}, t_{n+1})dt_{n+1}$ to the remaining $n+1 \le i \le k$ steps

$$z_k = z_n + v(P_{n+1}, t_{n+1}) \sum_{i=n+1}^{k-1} (t_i - t_{i+1}) + v(P_{n+1}, t_{n+1})t_k$$

$$= z_n + v(P_{n+1}, t_{n+1})(t_{n+1} - t_{n+2} + \cdots + t_{k-1} - t_k + t_k)$$

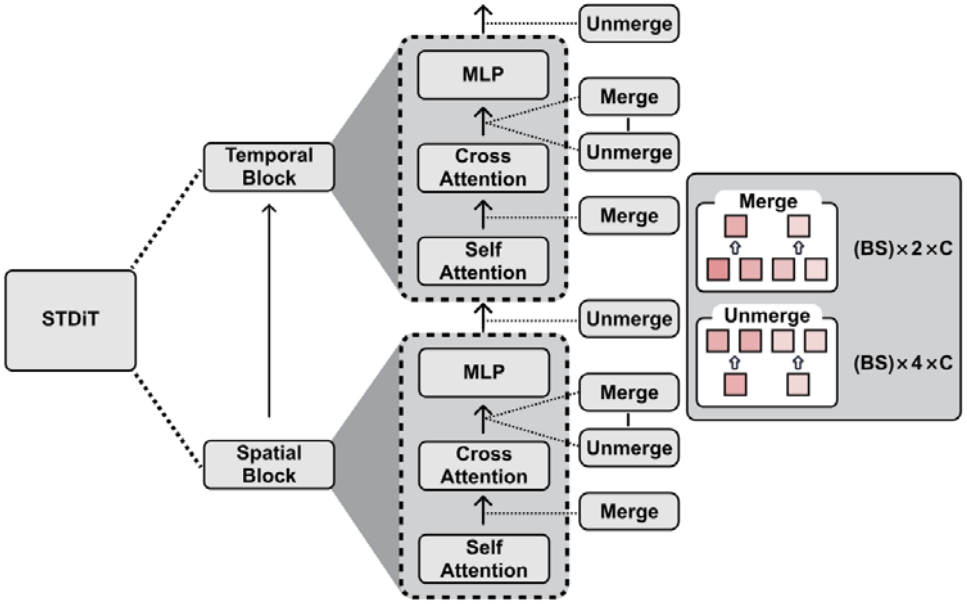
$$= z_n + v(P_{n+1}, t_{n+1})t_{n+1}$$

(a) Rectified Flow with full k=30 denoising steps, generating intact and complete video data
(b) Rectified Flow with n+1=16 denoising steps without applying Linear Proportional Leap, generating low-quality video data
(c) Linear Proportional Leap with n+1=15+1 denoising steps, producing video data nearly equivalent to (a).

2. Temporal Dimension Token Merging (TDTM)

Lightening the intensive computation required for token processing

- Merging video frames in the form of la- tent representations at attention layers of STDiT
  - Reducing the amount of tokens to be processed by half and lowers the computational complexity of attention modules up to one-quarter.



Two consecutive tokens are merged along the temporal di mension and subsequently unmerged after processing.

$$T_{merged} = TDTM_{merge}(T_{in})$$

$$TDTM_{merge}(T)[i] = \frac{1}{2}(T[:, i, :] + T[:, i+1, :])$$

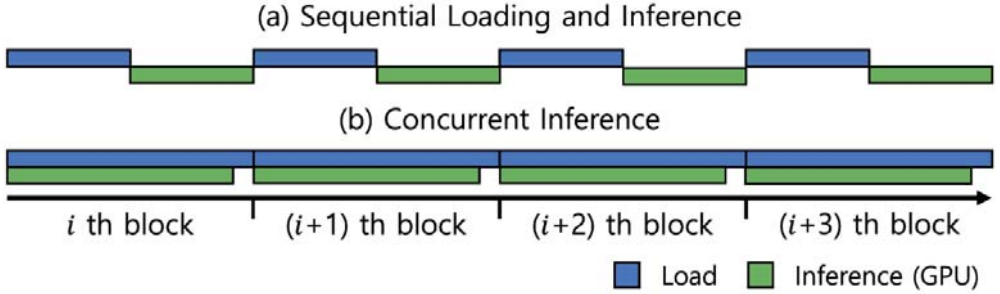
$$T_{unmerged} = TDTM_{unmerge}(\text{Attention}(T_{merged}))$$

$$TDTM_{unmerge}(T)[2i] = T[:, i, :]$$

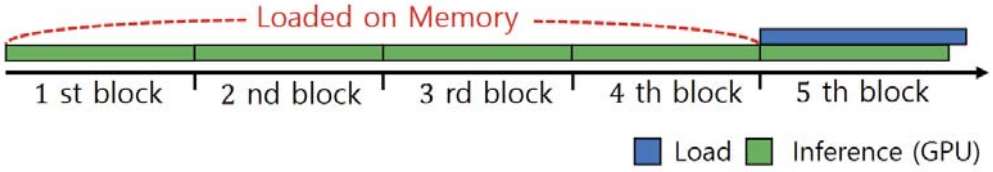
3. Concurrent Inference with Dynamic Loading (CI-DL)

Executing large video generative model components (i.e., T5 and STDiT) with the limited device memory

- Partitioning the models into smaller blocks that can be **loaded into the memory and executed concurrently**.
  - By parallelizing model execution and block loading, it effectively accelerates iterative model inference, e.g., multiple denoising steps.
  - Improving memory utilization while minimizing the block loading overhead by retaining specific model blocks in memory dynamically based on the available runtime memory.



Concurrent Inference (CI):
To minimize the increase in model inference latency caused by sequential block loading and execution, it leverages both the CPU and GPU for parallel block loading and execution.



Dynamic Loading (DL):
Maintains a subset of model blocks in memory without unloading them, dynamically determined based on the device's available memory at runtime.

Results: Accelerating Video Generation Latency up to 10x

iPhone 15 Pro's GPU is 143 times slower and has 14 times less memory compared to the NVIDIA A6000

- iPhone 15 Pro: **2.15 TFLOPS** with **3.3 GB** of available memory
- NVIDIA A6000: **309 TFLOPS** and **48 GB** of memory

LPL Setting	SSIM↑	FVD↓	Temporal Quality↑					Frame-Wise Quality↑		Speedup↑
			Subject Consistency	Background Consistency	Temporal Flickering	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	
16/30 (53%)	0.805	527.27	0.97	0.97	0.99	0.99	0.18	0.50	0.55	1.94×
21/30 (70%)	0.832	344.40	0.97	0.97	0.99	0.99	0.20	0.50	0.56	1.49×
23/30 (76%)	0.840	305.47	0.97	0.97	0.99	0.99	0.21	0.50	0.56	1.34×
25/30 (80%)	0.848	276.68	0.97	0.97	0.99	0.99	0.21	0.50	0.56	1.24×
30/30 (100%)	-	-	0.97	0.97	0.99	0.99	0.21	0.50	0.57	1.00×
Dynamic (μ:17.73/30)	0.827	370.86	0.97	0.97	0.99	0.99	0.20	0.50	0.56	1.53×

The video quality and generation speedup under different settings of LPL (Linear Proportional Leap).

Merging Steps	SSIM↑	FVD↓	Temporal Quality↑					Frame-Wise Quality↑		Speedup↑
			Subject Consistency	Background Consistency	Temporal Flickering	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	
30/30 (100%)	0.595	1225.69	0.97	0.97	0.99	0.99	0.06	0.50	0.56	1.27×
25/30 (83%)	0.599	1168.85	0.97	0.97	0.99	0.99	0.06	0.50	0.56	1.21×
20/30 (66%)	0.604	1056.47	0.97	0.97	0.99	0.99	0.06	0.50	0.56	1.16×
15/30 (50%)	0.612	924.92	0.97	0.97	0.99	0.99	0.12	0.50	0.57	1.13×
10/30 (33%)	0.622	784.67	0.97	0.97	0.99	0.99	0.16	0.50	0.56	1.10×
0/30 (0%)	-	-	0.96	0.97	0.99	0.99	0.23	0.50	0.58	1.00×

The video quality and speedup under different merging steps of TDTM (Temporal Dimension Token Merging).

Results

Example prompts and video snapshots

- Achieving equivalent video generation performance

"a stack of dried leaves burning in a forest"

"close up of a lemur"

A visual comparison of videos generated by On-device Sora and Open-Sora, evaluated using VBench.

More Videos


UNIST
ULSAN NATIONAL INSTITUTE OF
SCIENCE AND TECHNOLOGY
2007

THANK YOU