# T-Fusion: Thermal Prediction of 3D ICs with Multi-fidelity Fusion

Bingrui Zhang, **Wei W. Xing***, Xin Zhao, and Yuquan Sun*

Beihang University

& The University of Sheffield

**ASP-DAC 2025**

# Outlines

- ***BACKGROUND AND PREVIOUS WORK***
- ***METHODOLOGY***
- ***EXPERIMENT RESULTS***
- ***SUMMARY***

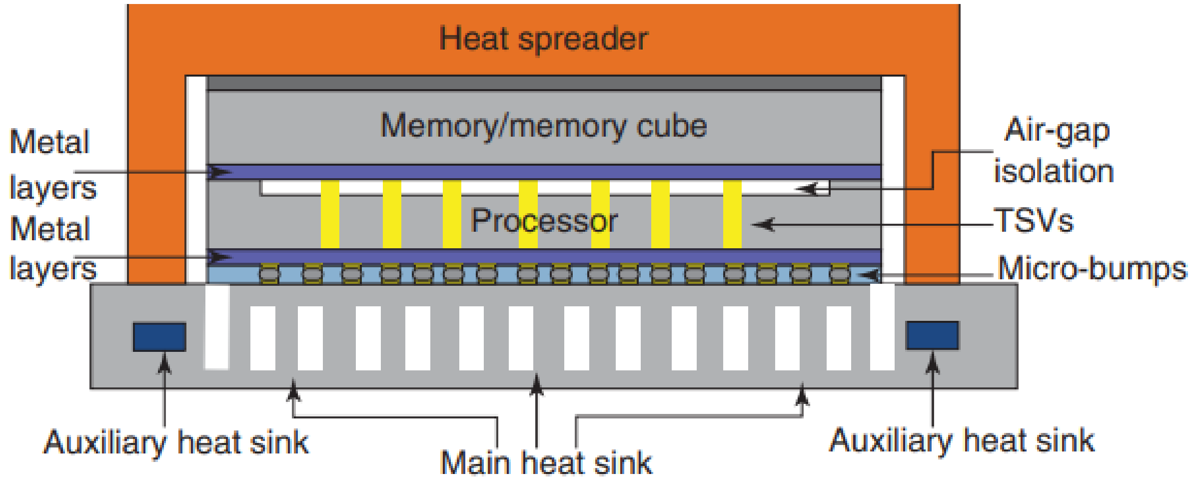
# Motivation: Why care about Thermal Modelling

Moore's Law is slowing down

3D Integration offers a solution

But brings thermal challenges:

- Higher power density
- Complex heat paths
- Performance degradation

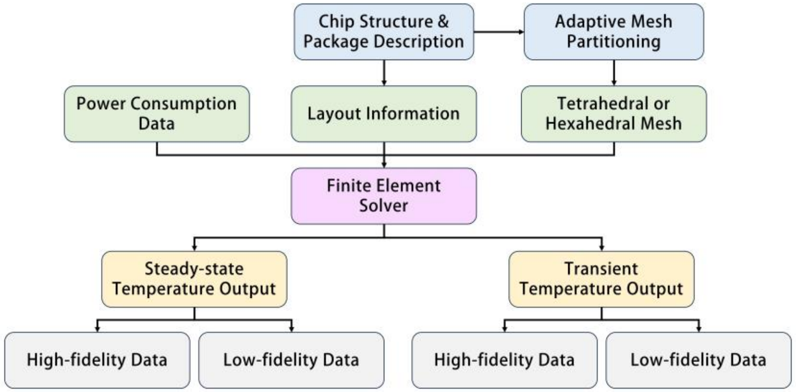



**Need fast and accurate thermal analysis**

# SOTA and limitation

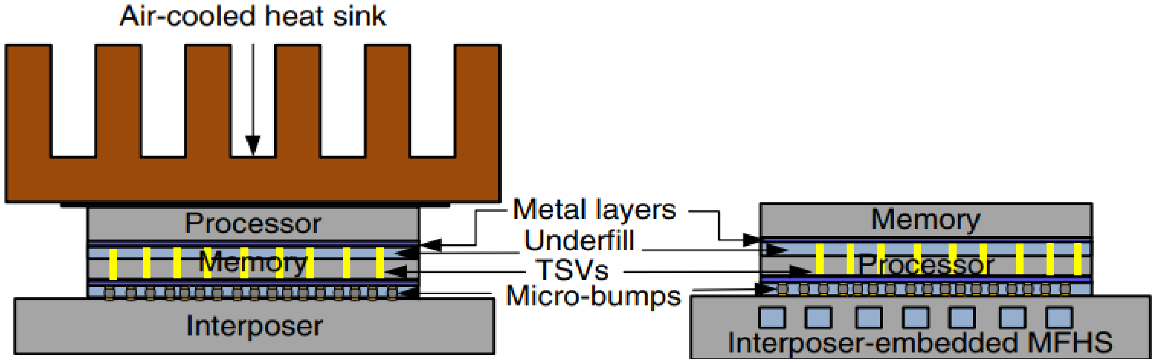
## Traditional Methods: numerical solvers

- COMSOL ( 1000 + seconds )
- Hotspot ( 1450 + seconds )
- MTA ( 200 + seconds )




## ML-based Methods: learn the mapping

- Deep OHeat ( 100 + hours training)
- ThermPINN ( 2D only)
- Therm-Transformer ( large data needed )



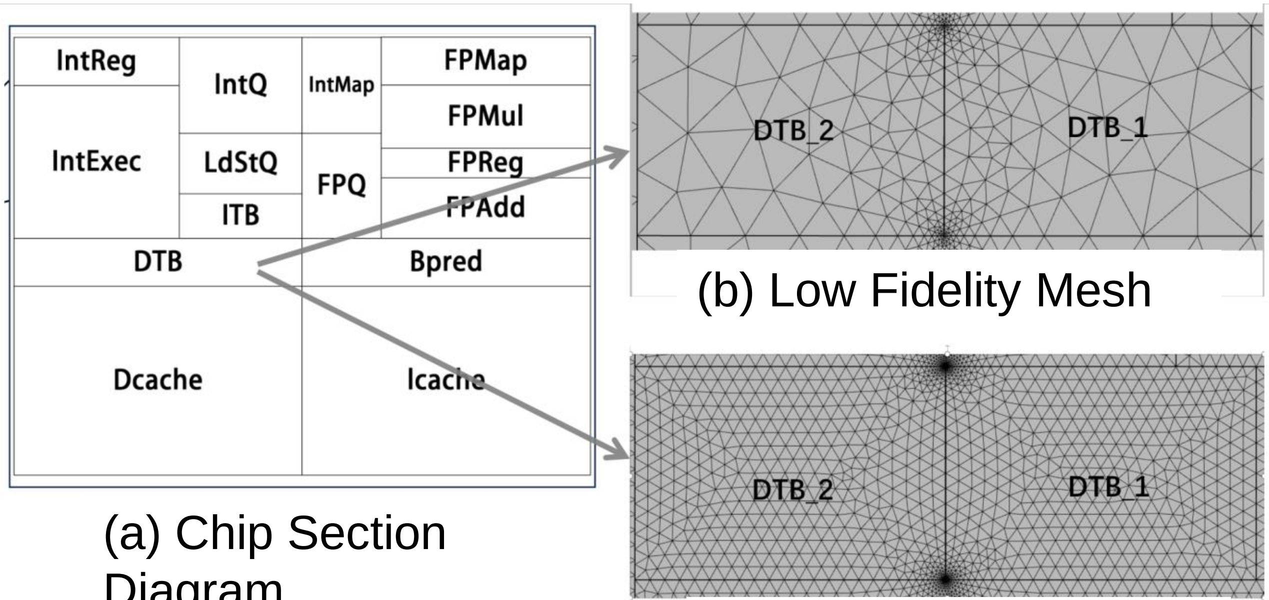

(a) Air-cooled heat dissipation (b)liquid-cooled heat dissipation

**Need fast and accurate thermal prediction with limited data**

# Insights

## Multi-fidelity Approach

- High-fidelity: Accurate but expensive
- Low-fidelity: Fast but less accurate
- Fusion: Best of both worlds




(a) Chip Section Diagram

(b) Low Fidelity Mesh

(c) High Fidelity Mesh

Leverage both Low- and High-Fidelity as data source

# Challenge:

## Key requirements

- Large-scale temperature fields
- Different resolutions across different fidelities
- Spatial correlation
- Temporal consistency
- Small training dataset
- Scalability

# Methodology

## T-Fusion Model

$$\underbrace{F^h(\mathbf{x}) = F^l(\mathbf{x})}_{\text{Low-fidelity model}} \underbrace{\times_1 \mathbf{W}_1 \times_2 \mathbf{W}_2 \times_3 \mathbf{W}_3}_{\text{Tensor multiplication}} + \underbrace{F^r(\mathbf{x})}_{\text{Residual model}}$$

**Key innovation:**

**Tensor based cross fidelity transformation**

- Preserve spatial relationship
- Resolution alignment
- Reducing computation from $O(d^3)$ to $O(d^2)$

**Efficient Training**

- $L = L^l + L^r$
- Reduce memory

# Experimental Results: Benchmark

## Test Cases of 3D ICs

- single-core
- quad-core
- eight-core chips

## Input:

Power consumption for each function block

## Output:

Spatial-temporal temperature profile

## Experiment platform

Intel Xeon CPU @ 2.40GHz NVIDIA A100 PCIe 40GB GPU 128 DDR-4 memory

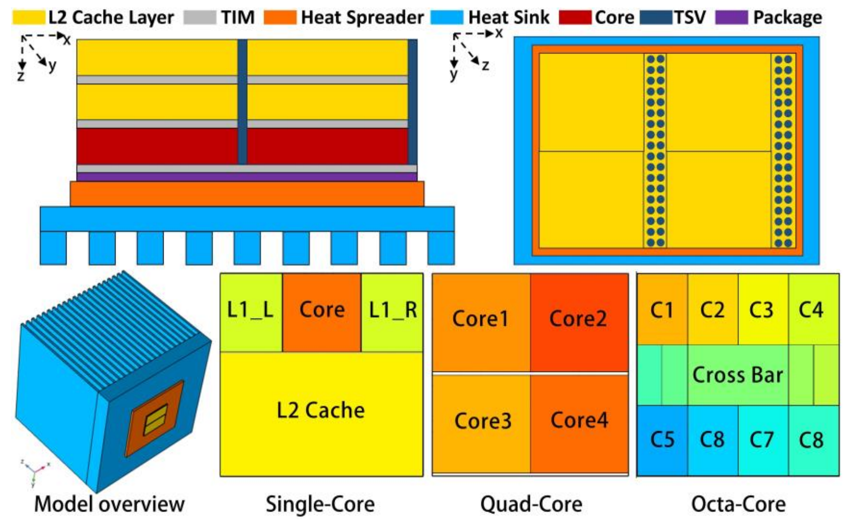
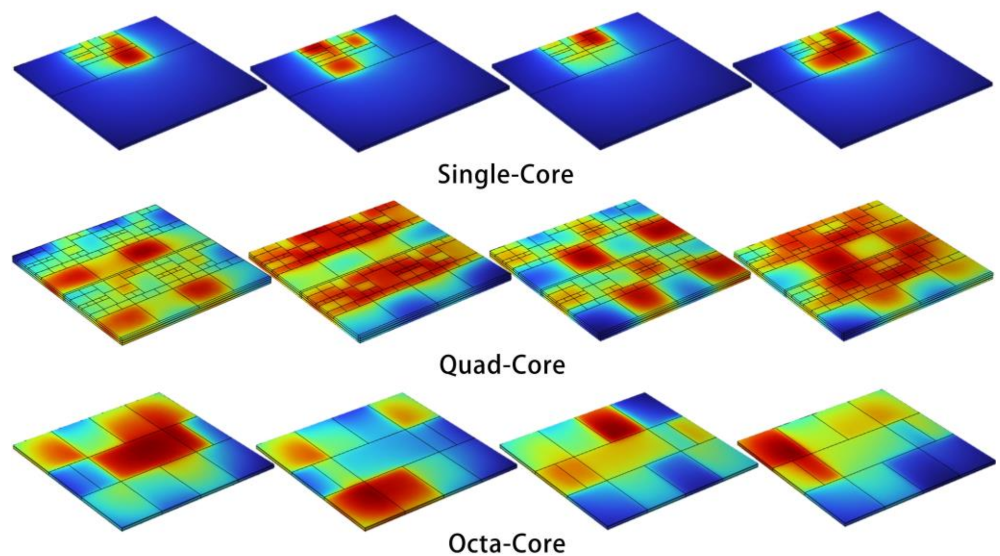



**Illustration of three 3D-IC structure**




**Example Power Differences in Three 3D-ICs**

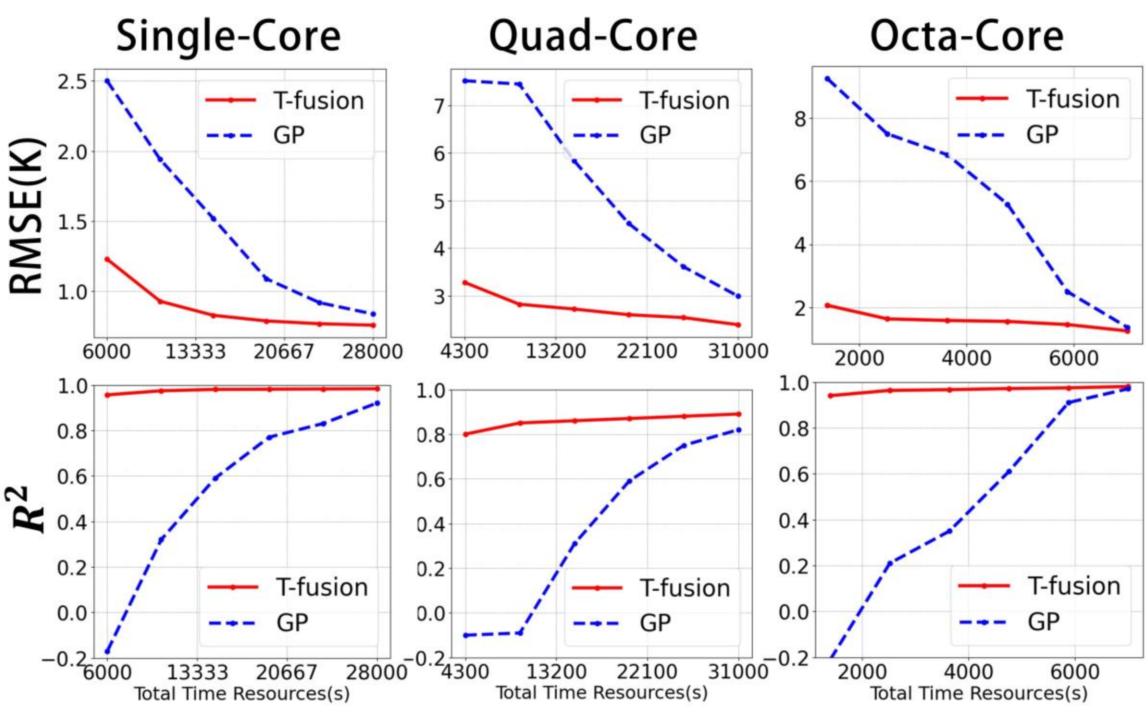
# Vs. Single Fidelity Steady-State Prediction

Steady-state temperature prediction against SOTA STGP with only high-fidelity data

based on a given certain computational time (for generating training data)

Table 2: The Mesh Grid Configurations

| | fidelity | mesh scale(mm) | DOF | runtime |
|---|---|---|---|---|
| Single-Core | low | 3.3-24 | 151706 | 14s |
| | high | 0.12-1.2 | 11749552 | 1272s |
| Quad-Core | low | 3.3-24 | 142513 | 23s |
| | high | 0.12-1.2 | 13087276 | 1092s |
| Octa-Core | low | 3.3-24 | 105152 | 15s |
| | high | 0.1-1 | 15956118 | 1276s |




Improvement:

- Training cost: **3×** reduction
- Model Training Time: **5×** faster
- Model Inference Time: **5×** improvement

# Vs. SOTA ML Model and Numerical Tools

**Table 3: Steady Temperature Comparisons of T-Fusion, GP and DeepOHeat(SOTA)**

| | | High Fidelity Training Sets | $R^2$ | MAPE(%) | PAPE(%) | Training Time | Mean Predict Time |
|---|---|---|---|---|---|---|---|
| DeepOHeat[8] | \ | 50 in each iteration (10000 iterations) | \ | 0.16 | 1.00 | 100 h | 0.1s |
| **T-Fusion** | Single-Core | 15 | 0.98 | 0.21 | 0.26 | 132.05s | 0.0016s |
| | Quad-Core | 15 | 0.90 | 0.52 | 0.33 | 184.87s | 0.0015s |
| | Octa-Core | 15 | 0.97 | 0.31 | 0.54 | 66.91s | 0.0015s |
| GP | Single-Core | 30 | 0.74 | 0.40 | 0.19 | 40.26s | 0.0003s |
| | Quad-Core | 30 | 0.66 | 0.95 | 1.24 | 35.98s | 0.0004s |
| | Octa-Core | 30 | 0.89 | 0.62 | 0.63 | 15.23s | 0.0008s |

## Vs. DeepOHeat

- Training cost:
  **3×** reduction
- Model Training Time:
  **5×** faster
- Model Inference Time:
  **5×** improvement

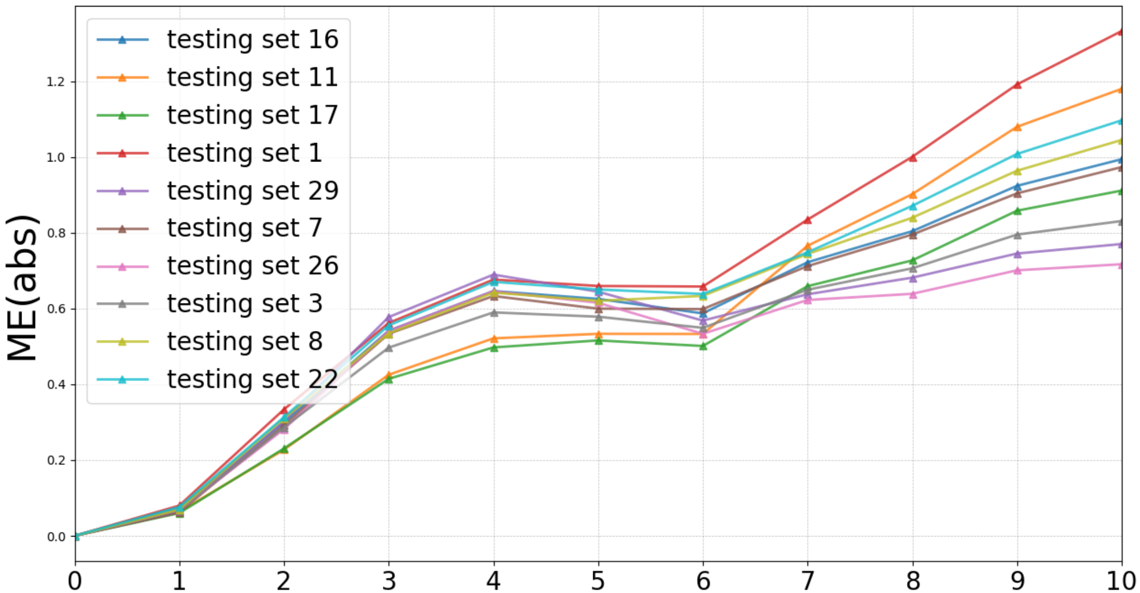
**Table 4: runtime and Maximum temperature comparisons among COMSOL, MTA, Hotspot and T-Fusion on Single-Core**

| | Max(K) | Error(K) | Predict Time(s) | Speedup |
|---|---|---|---|---|
| **T-Fusion** | 349.91 | -1.03 | 0.0015 | \ |
| COMSOL | 350.94 | \ | 1249 | 1,000,000x |
| Hotspot | 353.37 | +2.43 | 1450.6 | 1,000,000x |
| MTA | 350.944 | +0.04 | 217 | 100,000x |

## Vs. commercial numerical simulation

Acceleration: **100,000x -1,000,00x**; Error: **<1K**

# Transient Temperature Field Prediction




(a) mean error (ME) vs. time step

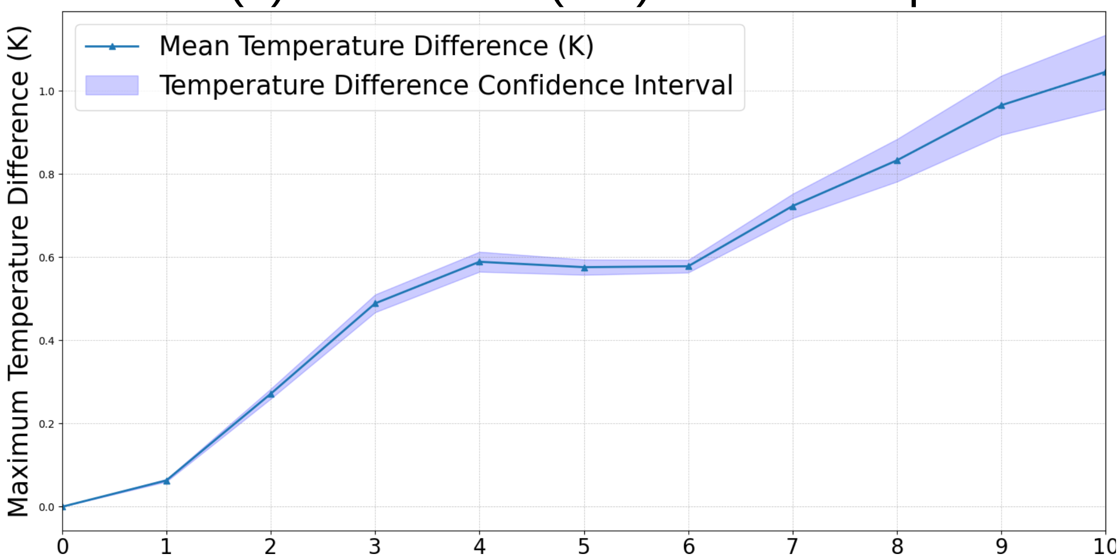
Prediction with 20 high-fidelity samples and 64 low-fidelity samples

Prediction of 10 time steps

**Observation:**

Error < 1K

The error does not explode



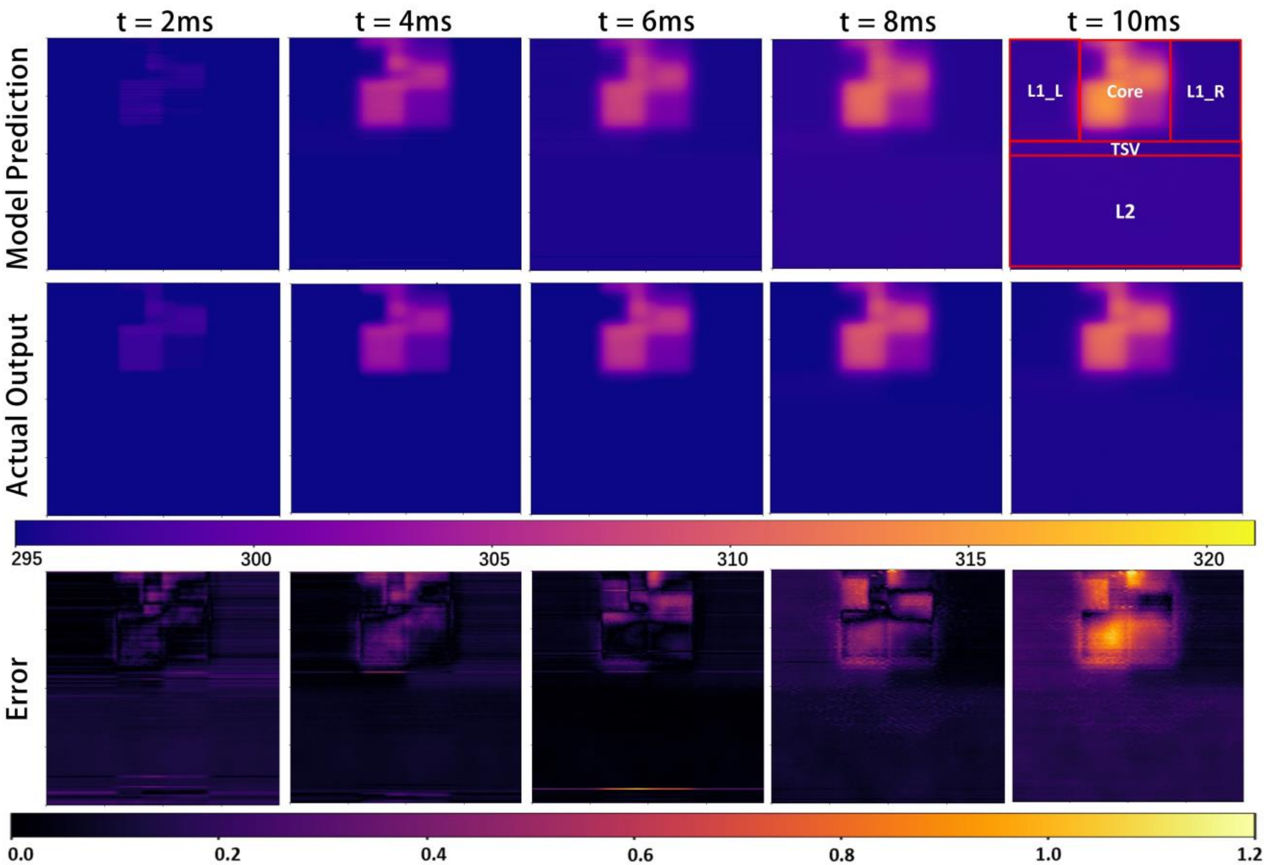

(b)Mean maximum error (confidence interval 95%)

**Observation:**

Average prediction error over 32 random power profiles

# Transient Temperature Field Prediction

Visualization of the cross-section with the largest thermal gradient in the entire 3D chip structure.




The chip temperature diffusion trends from time step 1 to time step 10 and a comparison between T-Fusion model predictions and the ground truth.

## Observation:

with drastic temperature changes, low-fidelity data introduces greater errors compared to high-fidelity data

# Summary and Future work

**Key Contributions:**

- Novel multi-fidelity fusion approach for thermal field prediction
- Up to **10,000×** reduction in training data (vs. single fidelity ML model)
- Up to **1,000,000×** speedup (vs. numerical solvers)
- Consistently **<1K** error (Vs. Comsol)
- Effective for both **steady-state** and **transient** analysis

**Future Directions:**

- Extension to more complex 3D architectures
- Integration with design optimization tools
- Real-time thermal monitoring applications

# *Thank you for your attention and engagement*

Contact: w.xing@sheffield.ac.uk, sunyq @buaa.edu.cn

**Acknowledgment:**

Multi-fidelity methods supported by IceLab Sheffield.

Codes and more multi-fidelity models and optimization can be downloaded at:

https://github.com/IceLab-X/FidelityFusion/