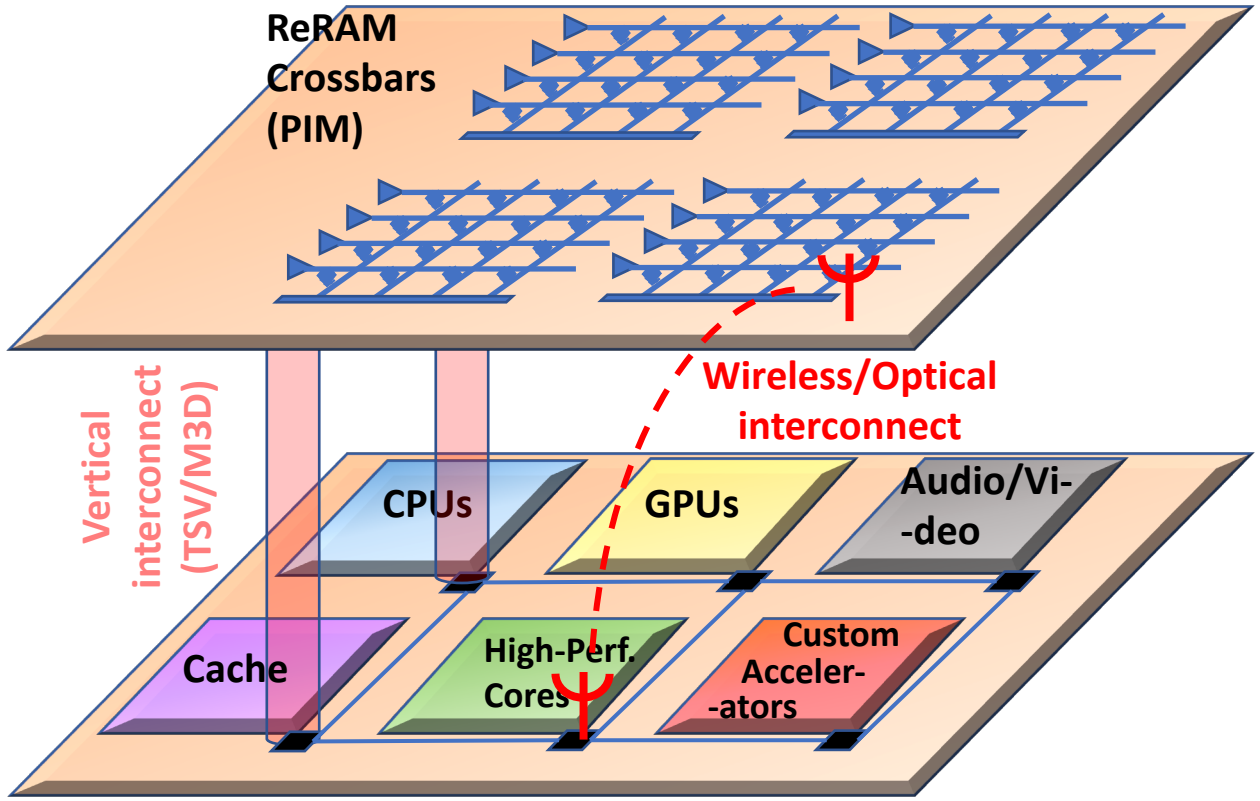
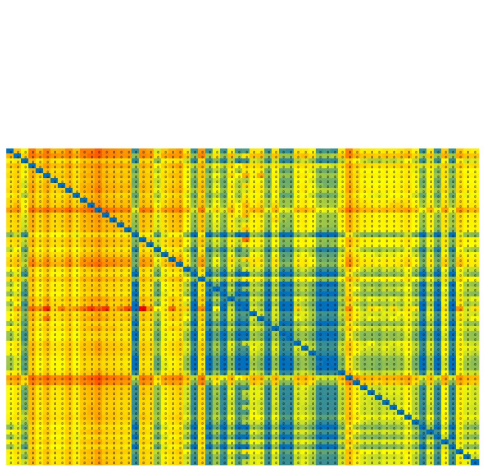
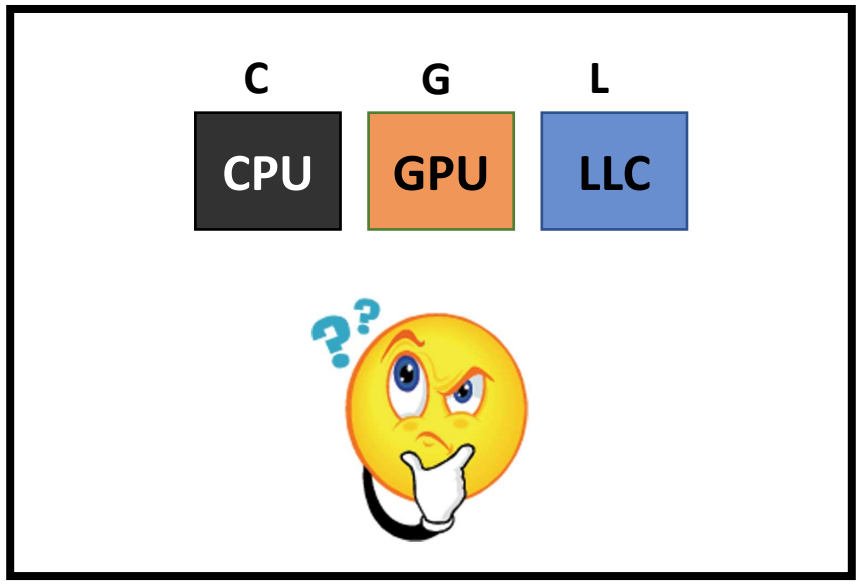
Heterogeneous manycore design

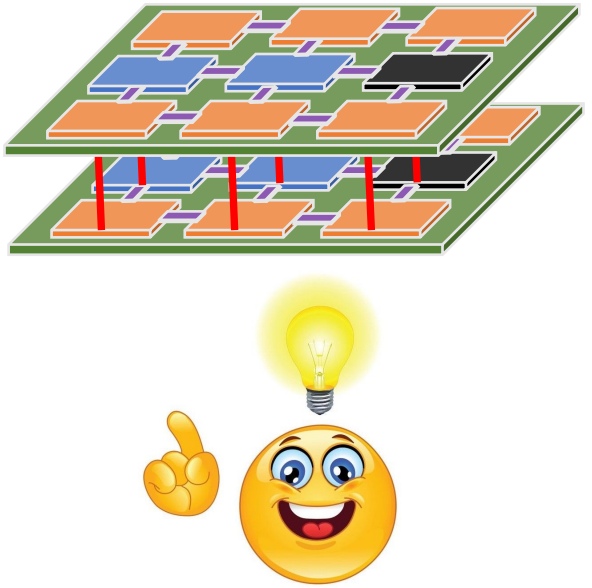
- Move CPUs/GPUs/Accelerators on to the chip
  - Avoid off-chip data transfers
- High Performance, Low Power, More flexibility
- Examples: Apple ‘A’ series, Qualcomm Snapdragon



Application-specific heterogeneous 3D manycore design



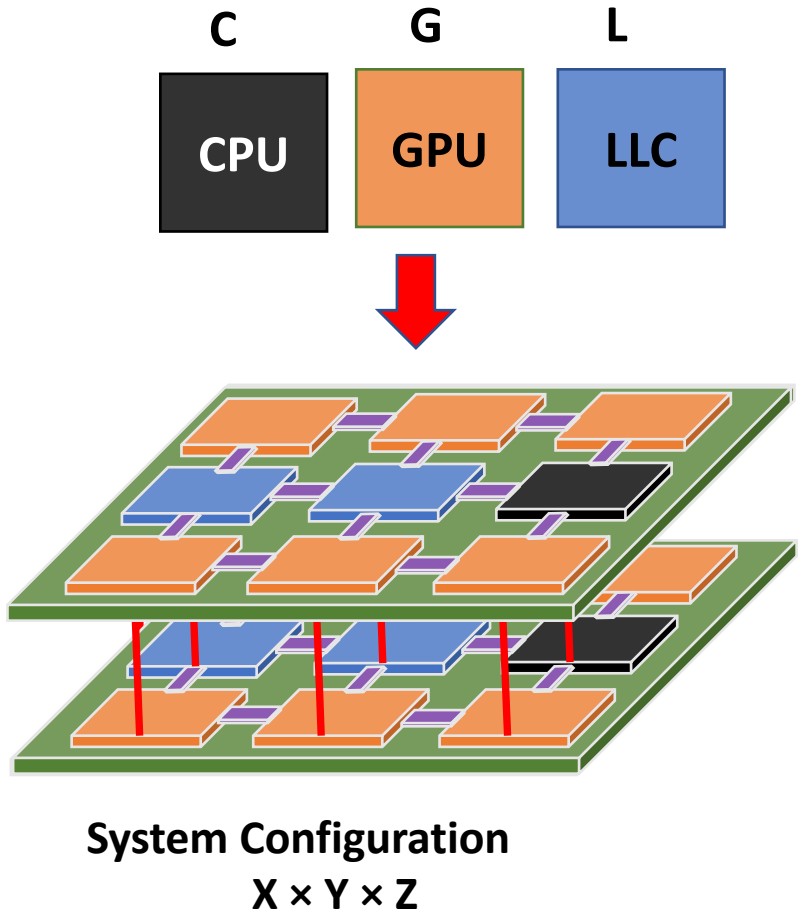




- Given an application behavior, what is the best architecture that we can design?
- Application behavior represented using traffic patterns/Power numbers
- Floor-planning problem

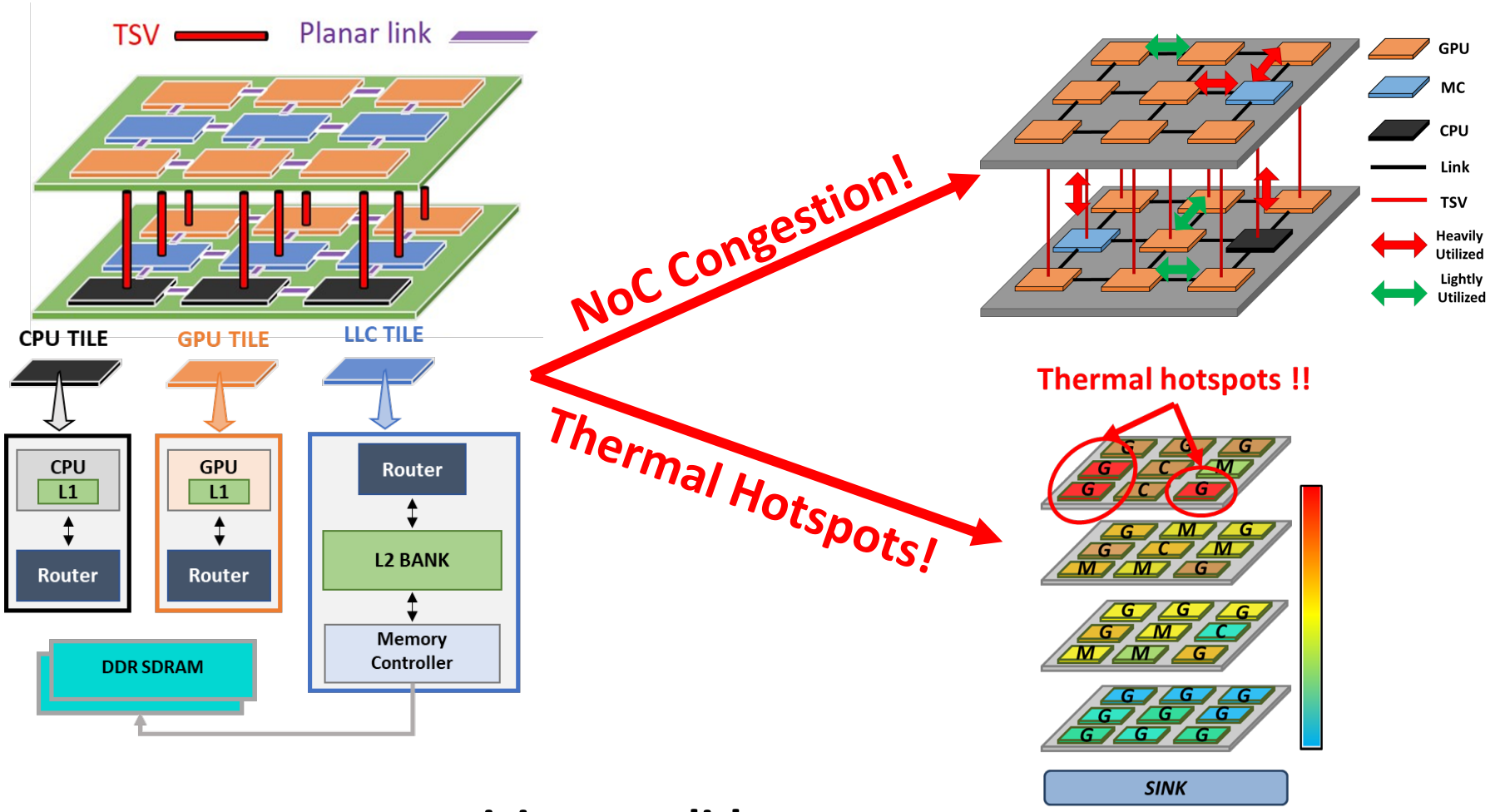
3D Manycore design

- **Given:**
  - ***C* CPUs, *G* GPUs, *L LLC*s and *mesh NoC***
  - **System configuration (X × Y × Z)**
- **Goal:**
  - **Place CPU, GPU, LLCs and links**
  - **Satisfy different objectives simultaneously**
    - ***GPU throughput***
    - ***CPU latency***
    - ***Thermal*, *etc.***



System Configuration
X × Y × Z

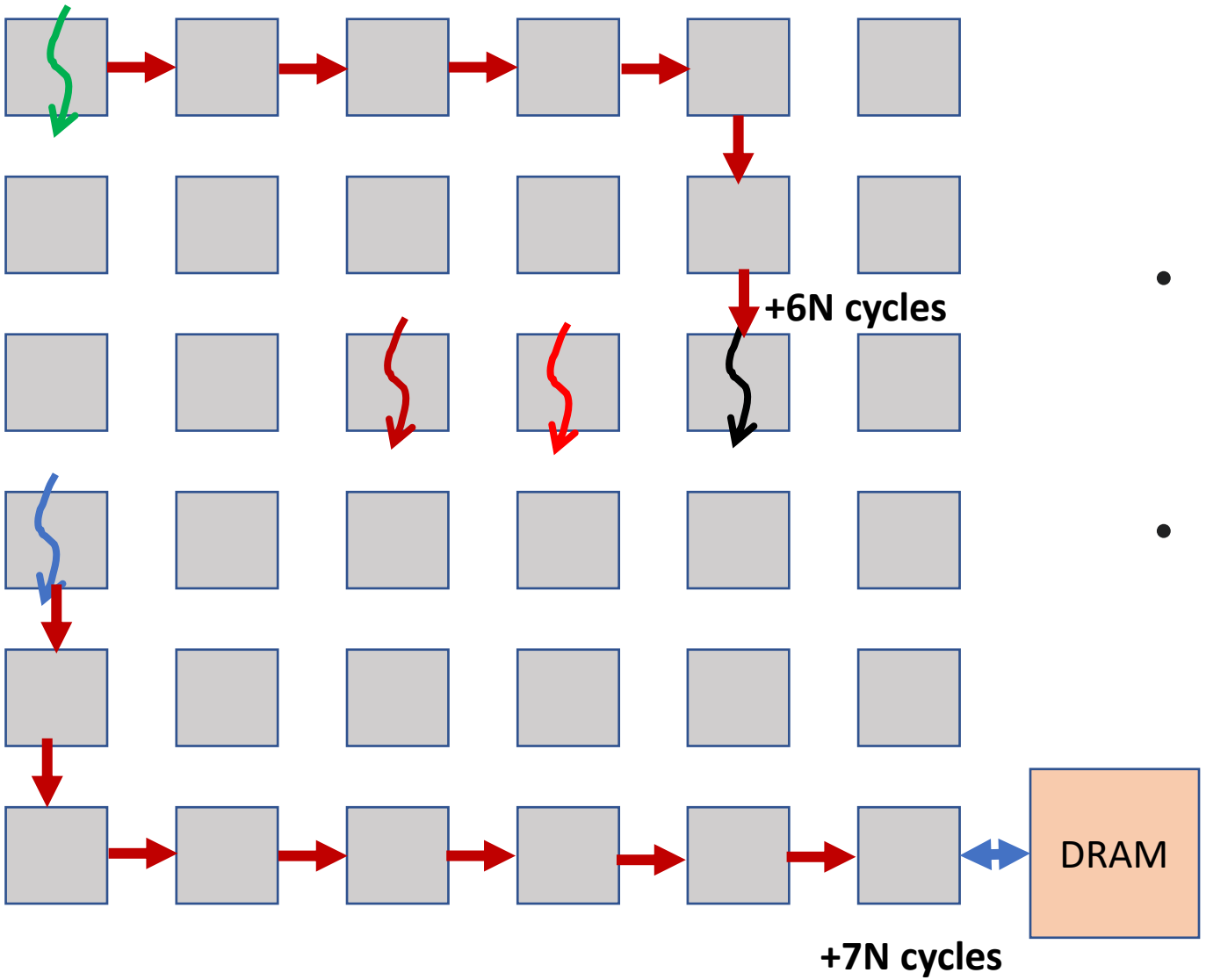
3D Manycore Design Challenges

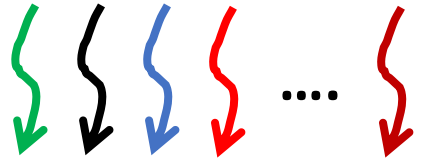


- **3D manycore are a promising candidate**

[1] **B. K. Joardar**, R. G. Kim, J. R. Doppa, P. P. Pande, D. Marculescu and R. Marculescu, "Learning-Based Application-Agnostic 3D NoC Design for Heterogeneous Manycore Systems," in IEEE Transactions on Computers, vol. 68, no. 6, pp. 852-866, 1 June 2019

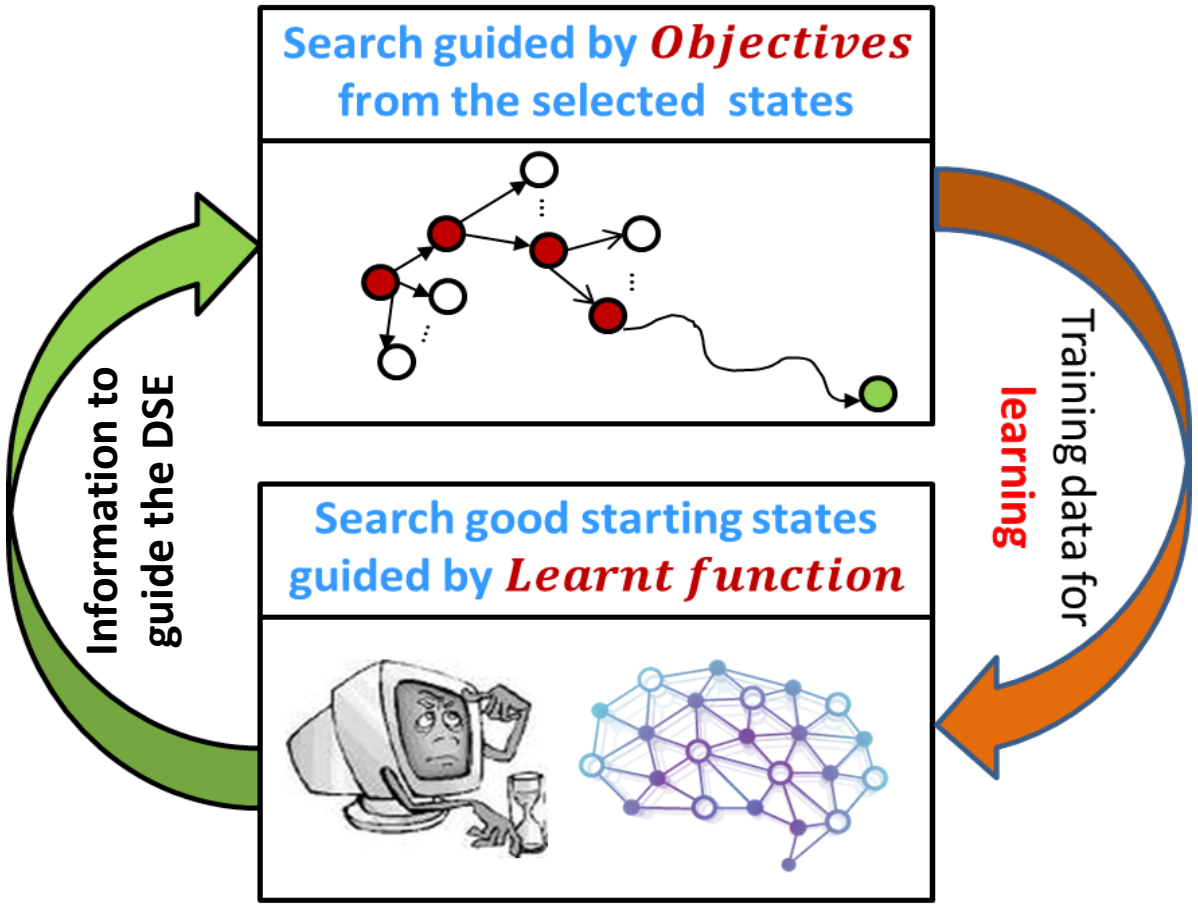
Similarity with task mapping





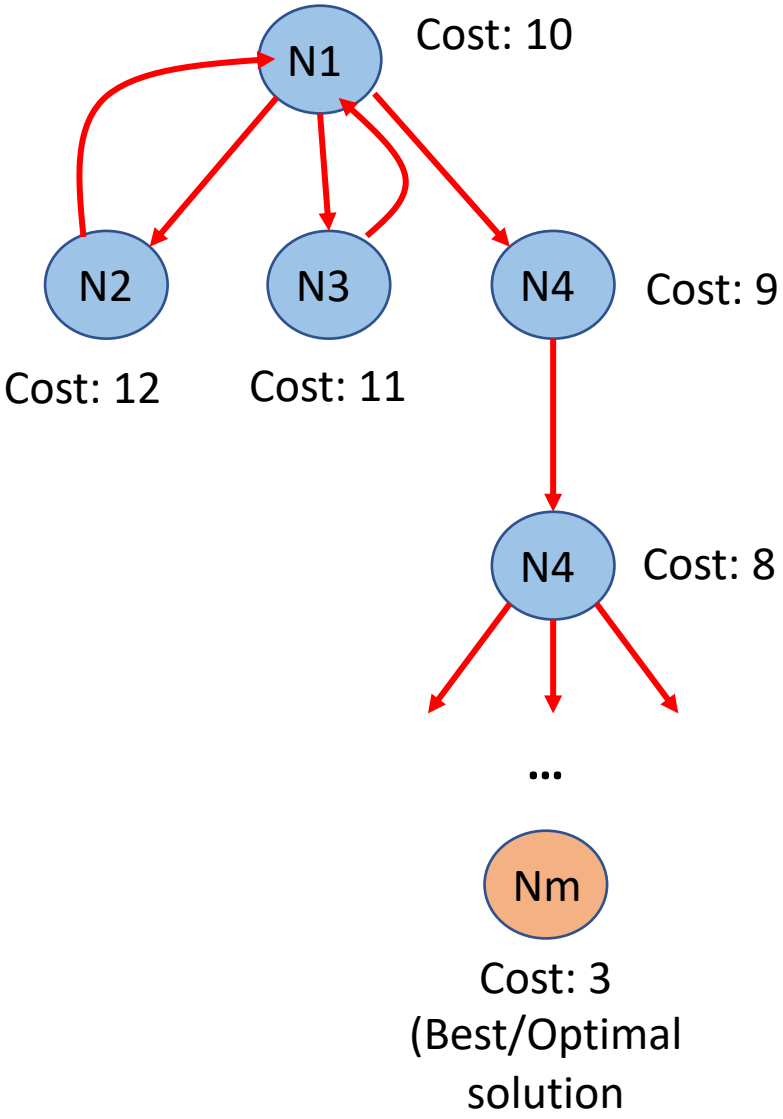
- N tasks → N cores
  - Communication
    - Performance/Execution time
  - Thermal
- N! (N factorial) cases
  - Impossible to explore all possible solutions
  - How to map these tasks?

ML based DSE



- Use ML to guide the DSE
- More scalable
- One instance: Use ML to choose better starting points
  - MOO-STAGE [1]
  - Replace “Random restart” by “Guided Restart”

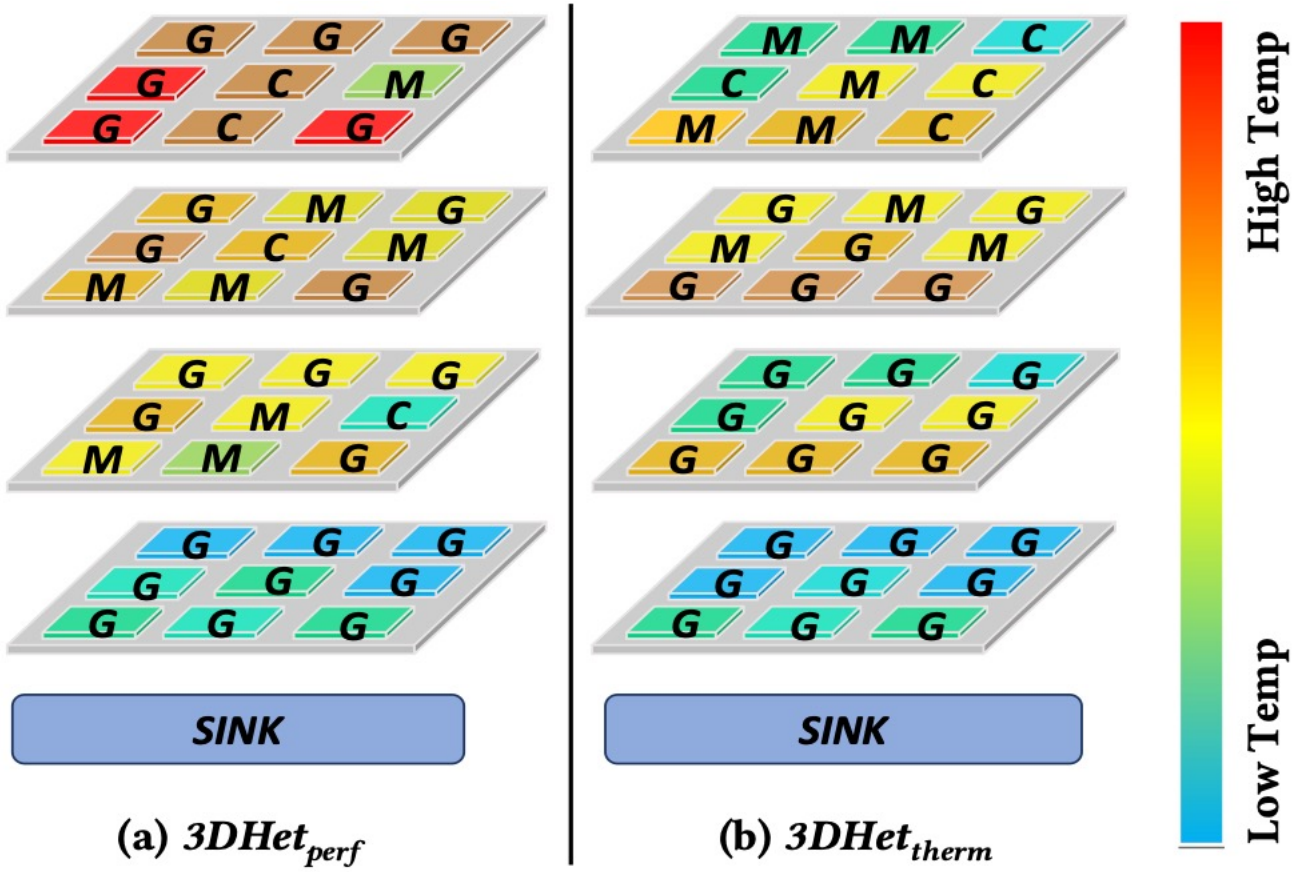
Manycore Optimization



- **Similar to MOO-STAGE based NoC design problem**
- **Consider "Thermal only" and "Performance-thermal joint" optimization**

Effect of core position on temperature

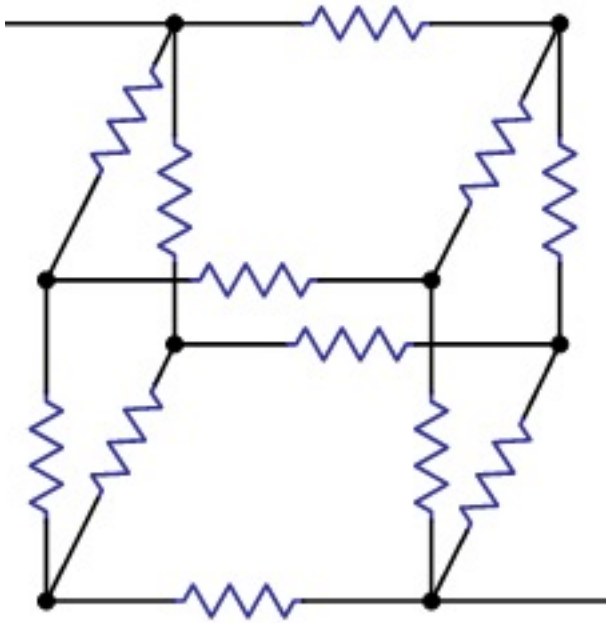
- By suitably mapping tasks to cores, we can reduce temperature
- High-power consuming tasks should be mapped near the sink
- Up to 15C cooler



B. K. Joardar et al., "3D NoC-enabled heterogeneous manycore architectures for accelerating CNN training: Performance and thermal trade-offs," 2017 Eleventh IEEE/ACM International Symposium on Networks-on-Chip (NOCS), 2017, pp. 1-8.

Thermal-aware DSE

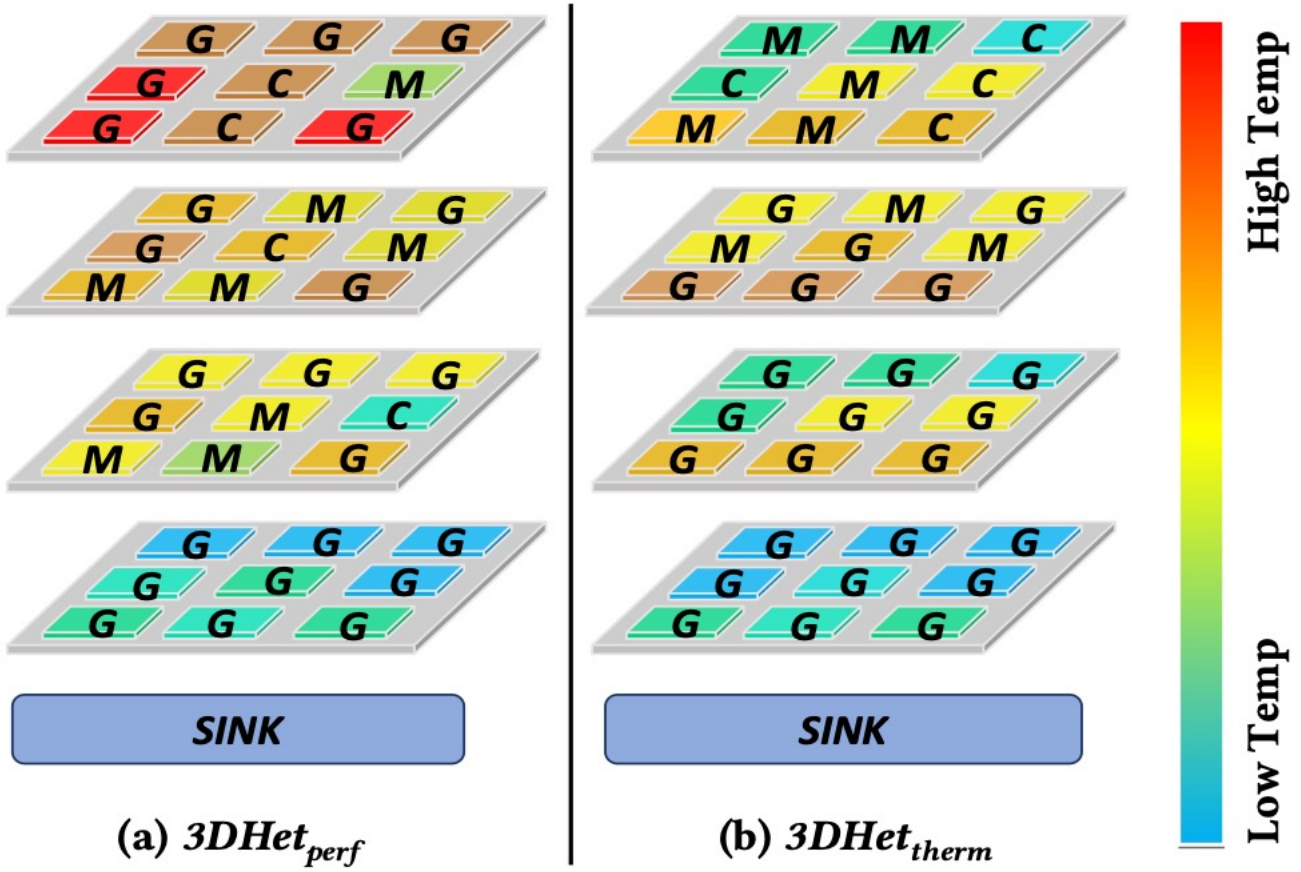
- **3D-specific objective: Temperature**
  - **Max. on-chip temperature**
    - $T = \max_{n,k}\{\sum_{i=1}^{k}(P_{n,i}(t)\sum_{j=1}^{i}R_j) + R_b\sum_{i=1}^{k}P_{n,i}(t)\} * T_H$



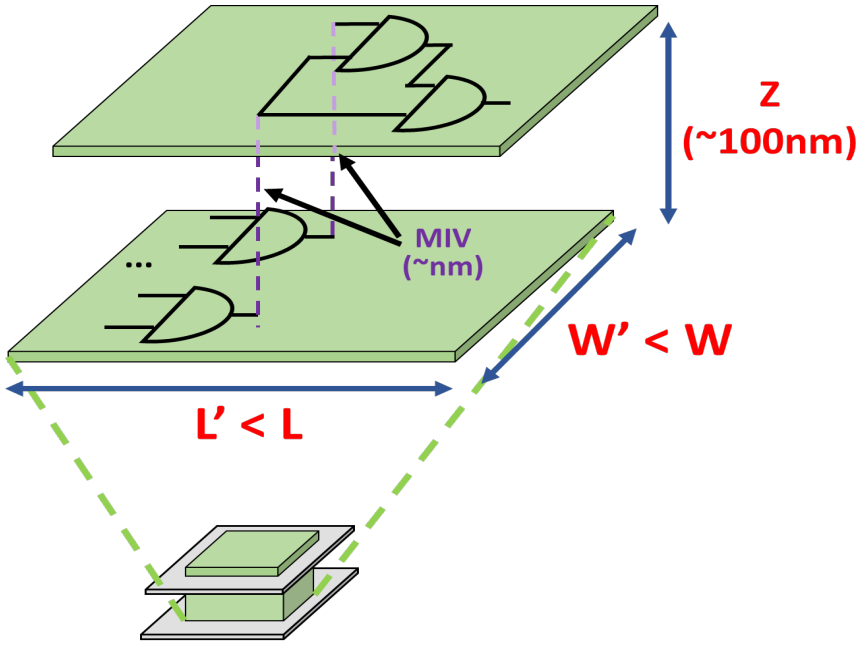
J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *Proc. of the ICCAD*, San Jose, CA, pp. 306-313, 2004

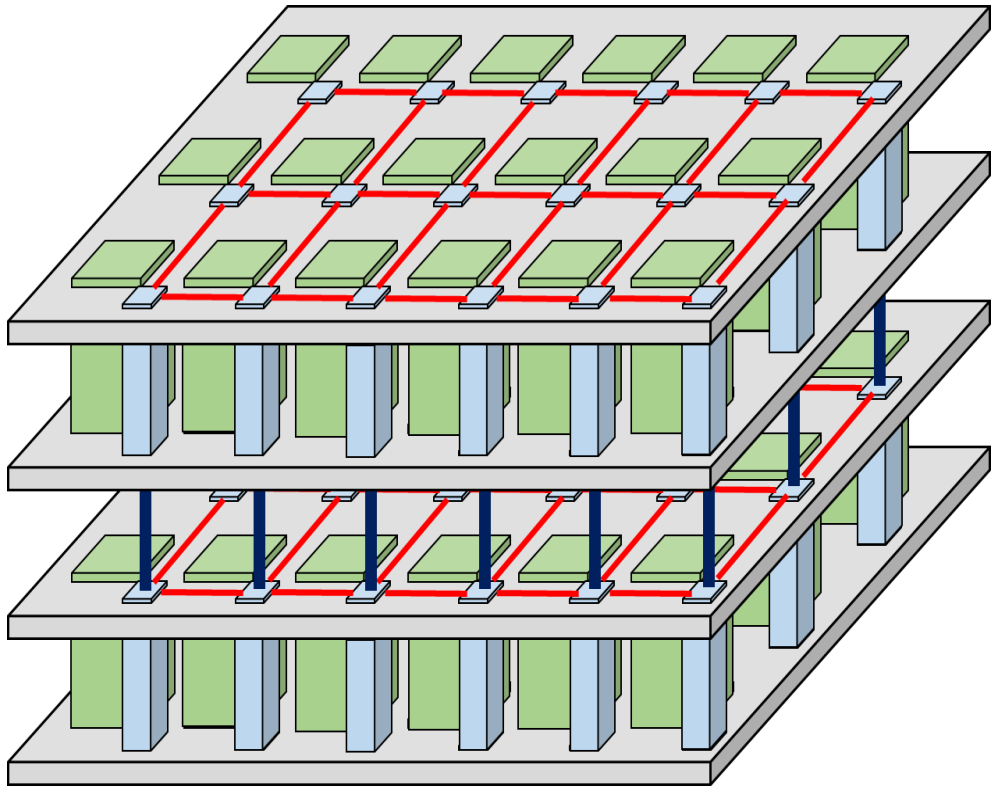
Thermal-aware manycore design with TSV

- 3D architecture by varying core placement
- Reduce temperature
  - Place higher power cores near the sink
- Mesh NoC



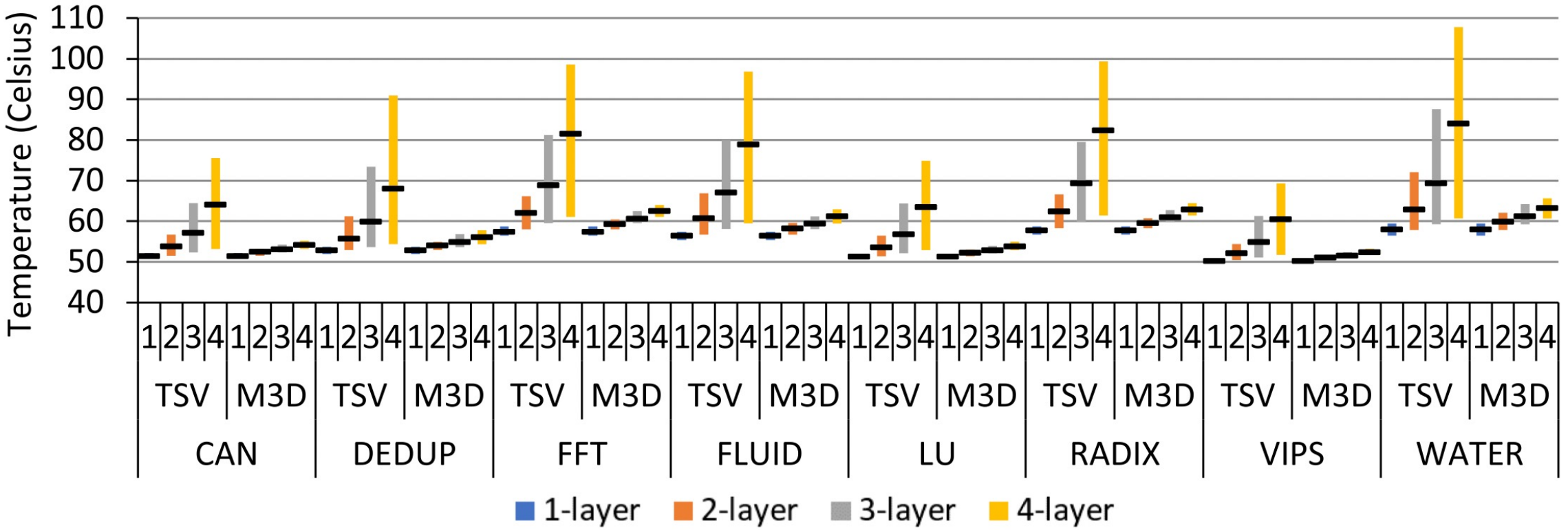
Fully manycore design using M3D





- Routers and cores can extend over 2 tiers
- Faster logic blocks
- Better performance

Temperature in 3D



- **Layer farthest from the sink has highest temperature**
- **M3D has better thermal profile than TSV**

Dongjin Lee, Sourav Das, Janardhan Rao Doppa, Partha Pratim Pande, and Krishnendu Chakrabarty. 2018. Performance and Thermal Tradeoffs for Energy-Efficient Monolithic 3D Network-on-Chip. ACM Trans. Des. Autom. Electron. Syst. 23, 5, Article 60 (October 2018), 25 pages

How to get power & temperature (2)?

- Cycle-accurate simulators for power
  - CPU: Gem5 + McPat
  - GPU: GPGPU-Sim + GPUWattch
- Allows us to run C/Cuda code
- Simulators for temperature
  - HotSpot, 3D-ICE

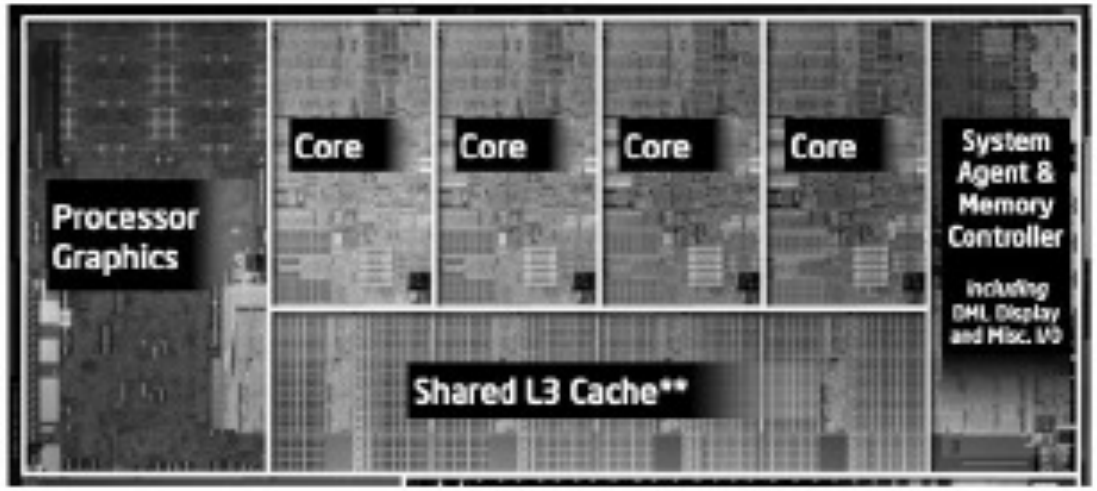
https://www.gem5.org

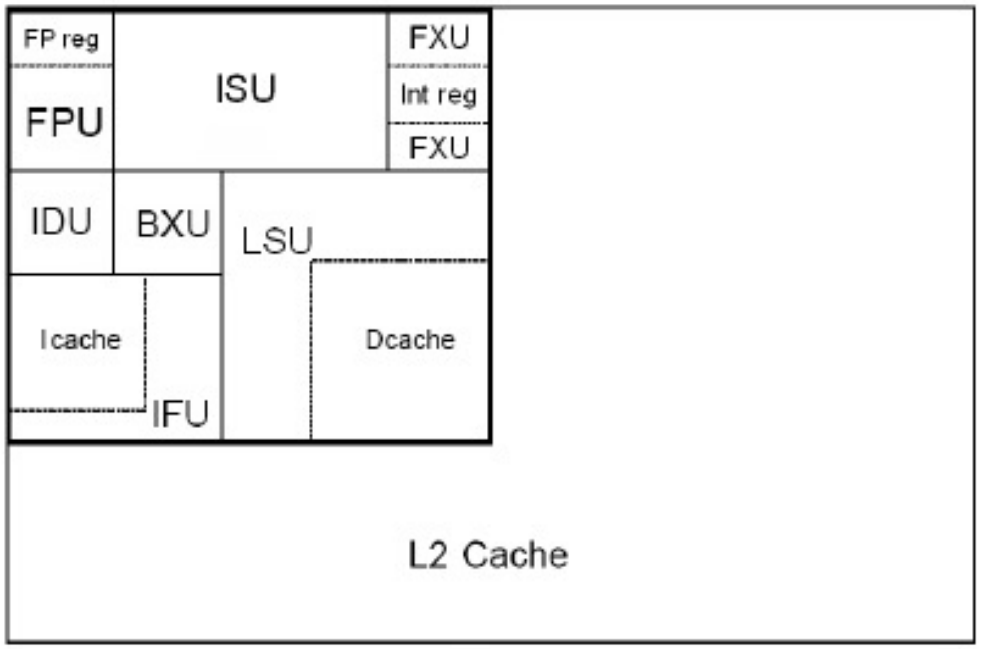
3D-ICE thermal simulator

- https://github.com/esl-epfl/3d-ice
- Build custom floorplan (.flp)
- Provide technology file (.stk)
- Run 3D-ICE assuming chip dimension of 10mm x 10mm
- Get temperature for various configurations

```
Core0 :

  position       0,    0 ;
  dimension   2500, 3500 ;

  power values 12.5, 14.0, 10.5, 13.5, 12.5 ;

Core1 :

  position    2500,    0 ;
  dimension   2500, 3500 ;

  power values 12.5, 14.0, 10.5, 13.5, 12.5 ;

Core2 :

  position    5000,    0 ;
  dimension   2500, 3500 ;

  power values 12.5, 14.0, 10.5, 13.5, 12.5 ;
```

Floorplanning



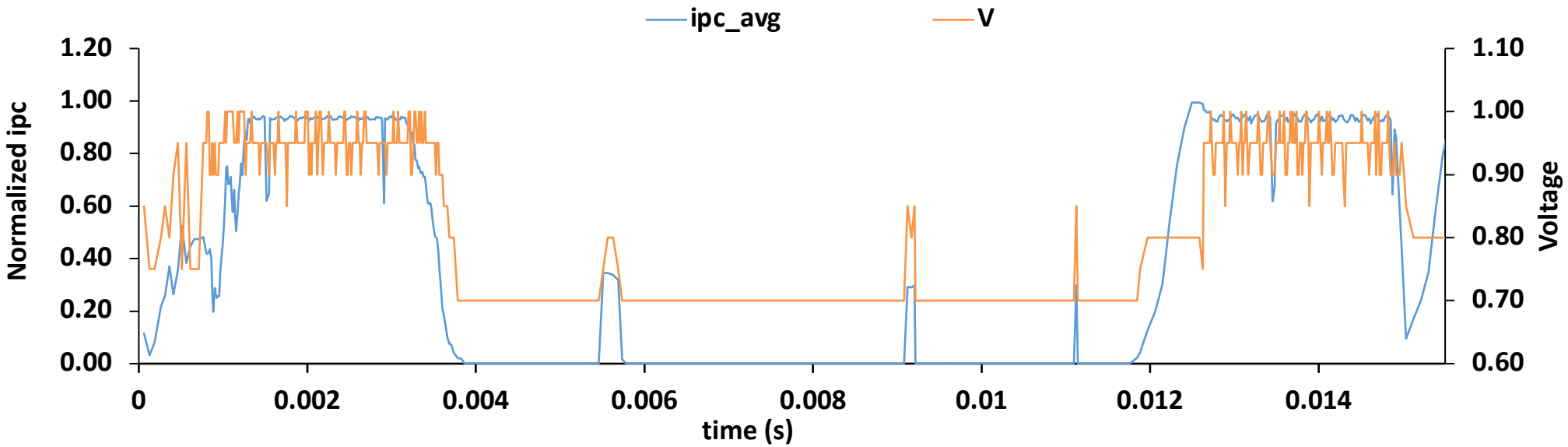


- **Physical placement of cores/logic blocks**
  - **Necessary for thermal simulations**
  - **HotSpot, 3D-ICE**

Power management recap

- Power/Energy is important, especially for embedded systems
  - Mobile and portable devices: Smartphones, Laptops
  - Other devices: Wearables, Medical devices, etc.
- Lower power usage = Better battery life
- More power = More heat

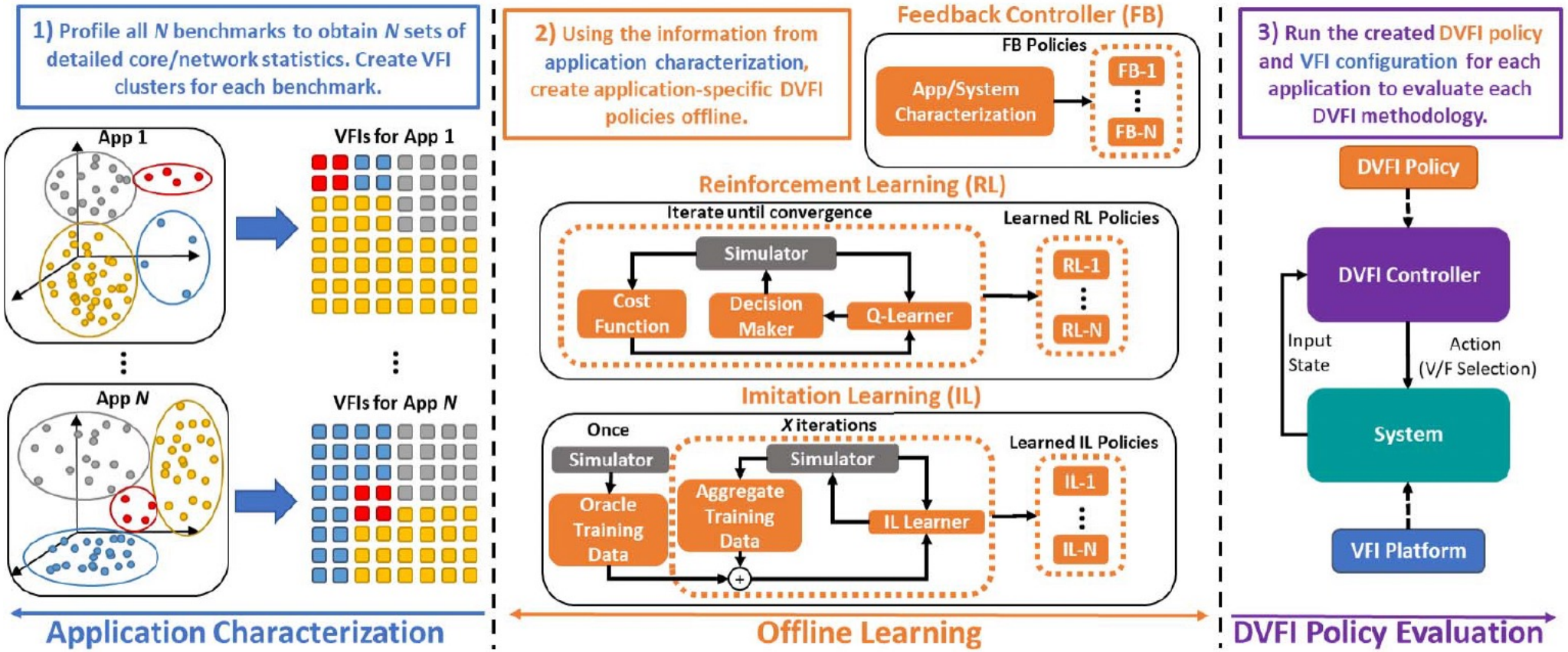
DVFS recap



- IPC for FFT with respect to time
- Voltage closely matches the behavior
- ML policy used for prediction

A. Chatterjee, S. Musavvir, R. G. Kim, J. R. Doppa, and P. P. Pande. 2021. Power Management of Monolithic 3D Manycore Chips with Inter-tier Process Variations. J. Emerg. Technol. Comput. Syst. 17, 2, Article 13

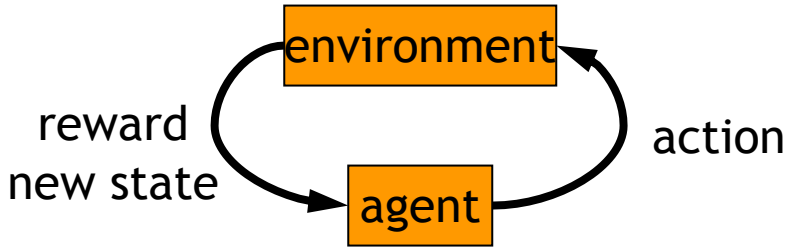
Overall methodology



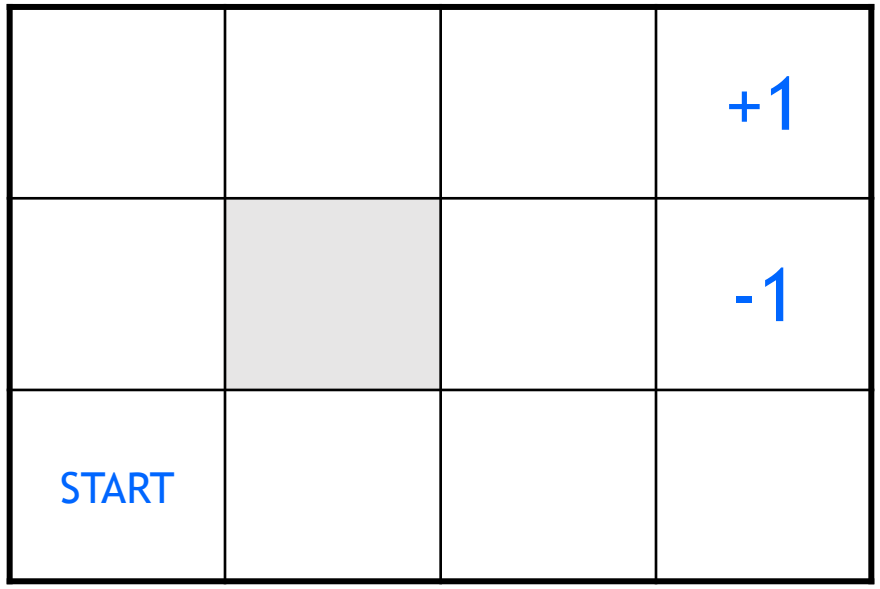
R. G. Kim et al., "Imitation Learning for Dynamic VFI Control in Large-Scale Manycore Systems," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 9, pp. 2458-2471, Sept. 2017, doi: 10.1109/TVLSI.2017.2700726.

Reinforcement learning

- Supervised learning
  - classification, regression
- Unsupervised learning
  - clustering
- Reinforcement learning
  - more general than supervised/unsupervised learning
  - learn from interaction w/ environment to achieve a goal



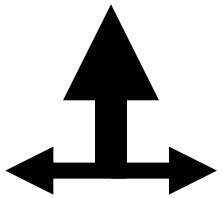
RL example



actions: UP, DOWN, LEFT, RIGHT

UP

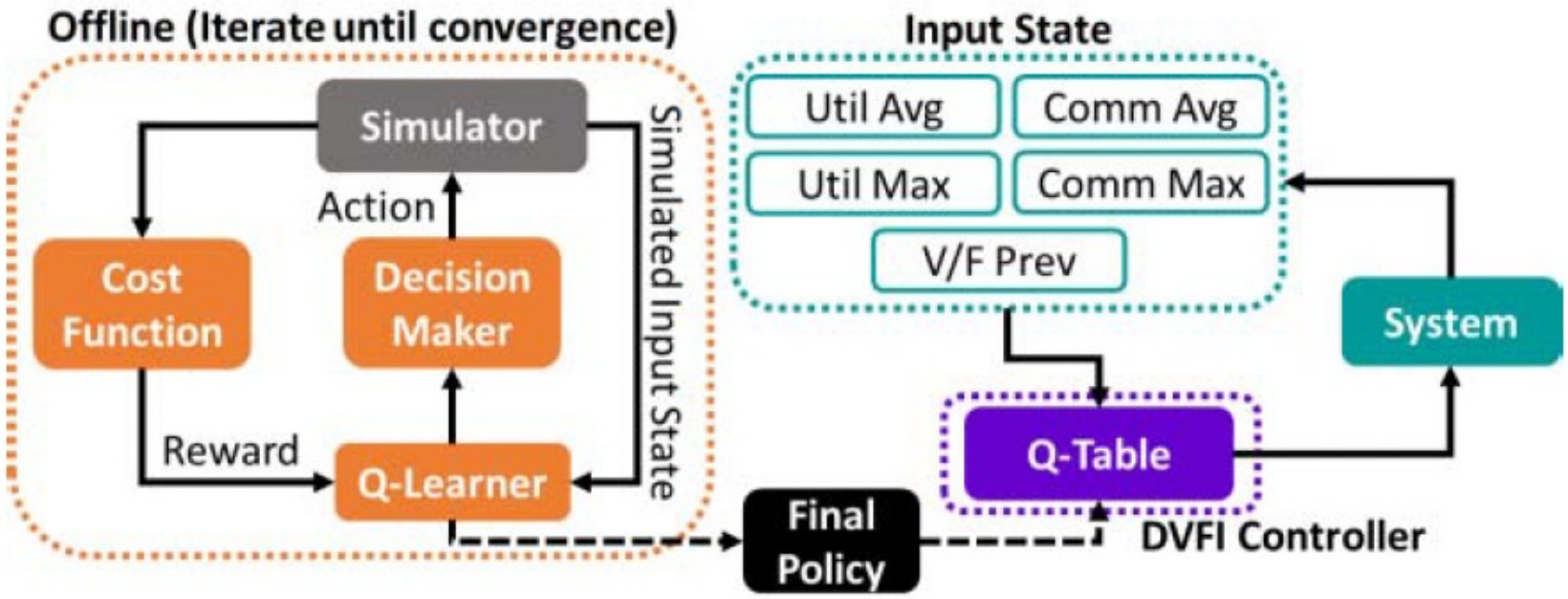
80% move UP
10% move LEFT
10% move RIGHT



- Maze solving problems
- reward +1 at [4,3], -1 at [4,2]
- reward -0.04 for each step
- what's the strategy to achieve max reward?

https://www.youtube.com/watch?v=pc-H4vyg2L4
https://deeplizard.com/learn/playlist/PLZbbT5o_s2xoWNVdDudn51XM8lOuZ_Njv

Reinforcement learning for power management



- **Which <V,F> pair for a certain time window?**
- **RL to learn the best descision**

Z. Chen and D. Marculescu, "Distributed reinforcement learning for power limited many-core system performance optimization," 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015, pp. 1521-1526.

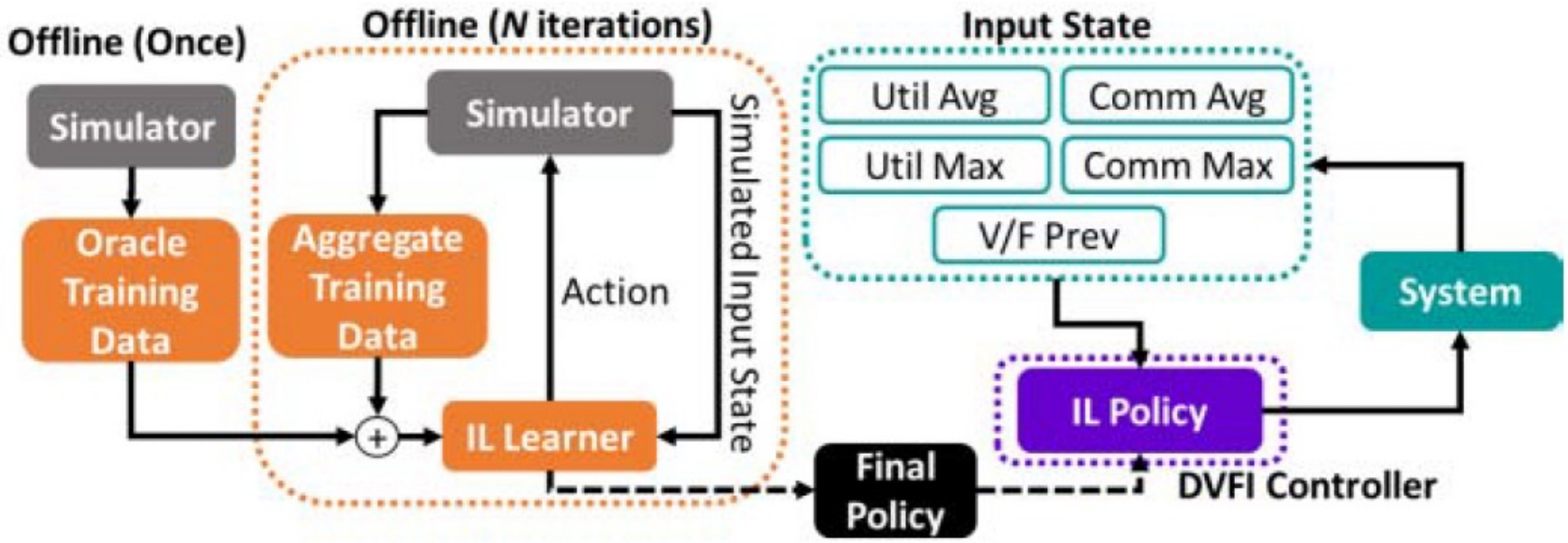
Imitation learning (1)

- **Imitate the expert**
- **Simpler and faster than RL**

Imitation learning vs Reinforcement learning

- **Reinforcement Learning: Learning policies guided by sparse rewards, e.g., win the game.**
  - **Good: simple, cheap form of supervision**
  - **Bad: High sample complexity**
- **In simulation, where we can afford a lot of trials, easy to parallelize**
- **Not in robotic systems:**
  - **Action/reward takes long**
  - **we cannot afford to fail (e.g., Offroad navigation!)**
  - **safety concerns**
- **Imitation learning is useful when it is easier for the expert to demonstrate (or oracle creation can be done easily)**

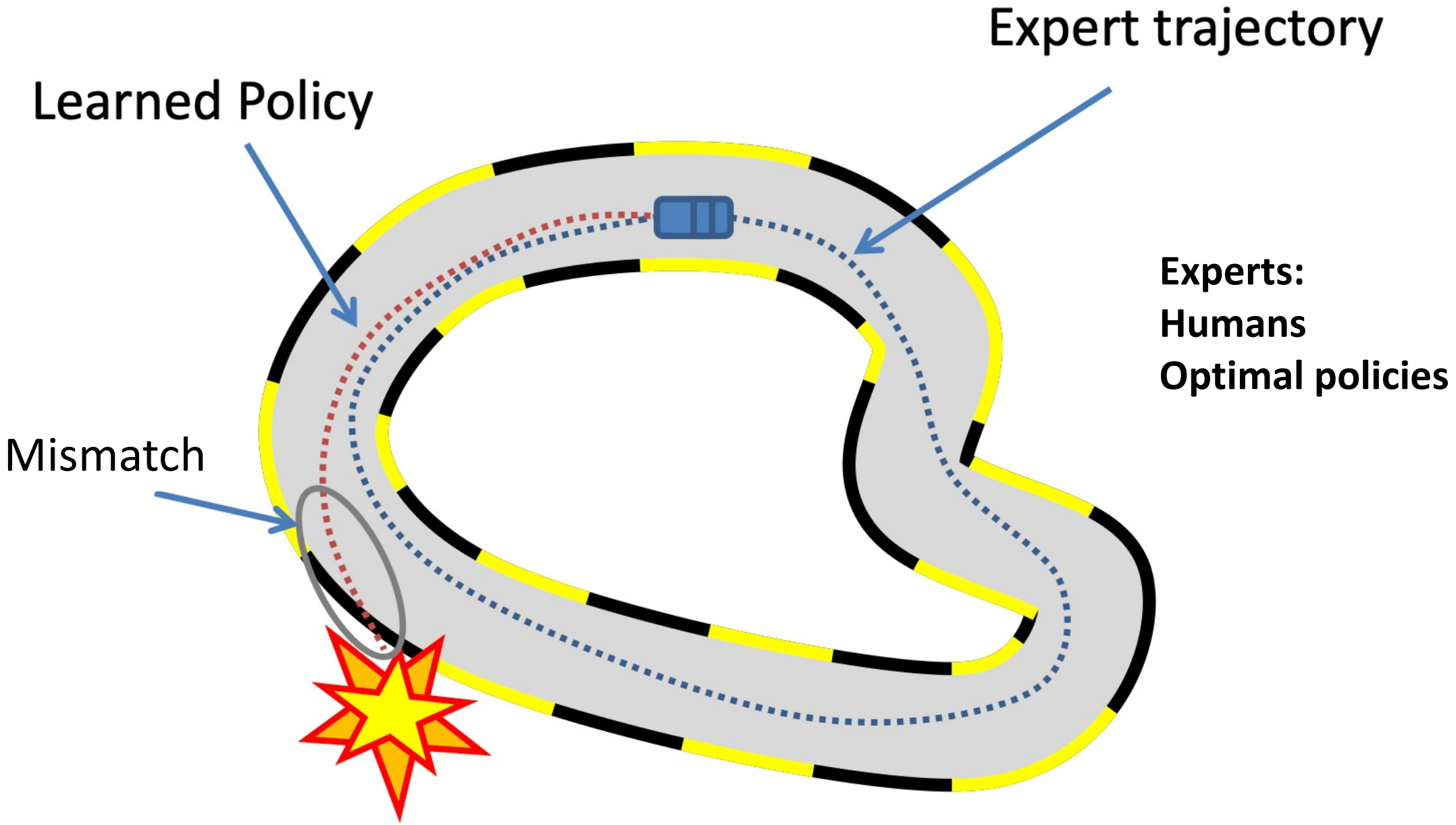
Imitation learning for VFI



- Which <V,F> pair for a certain time window?
- IL to learn the best decision
- Where is the expert?

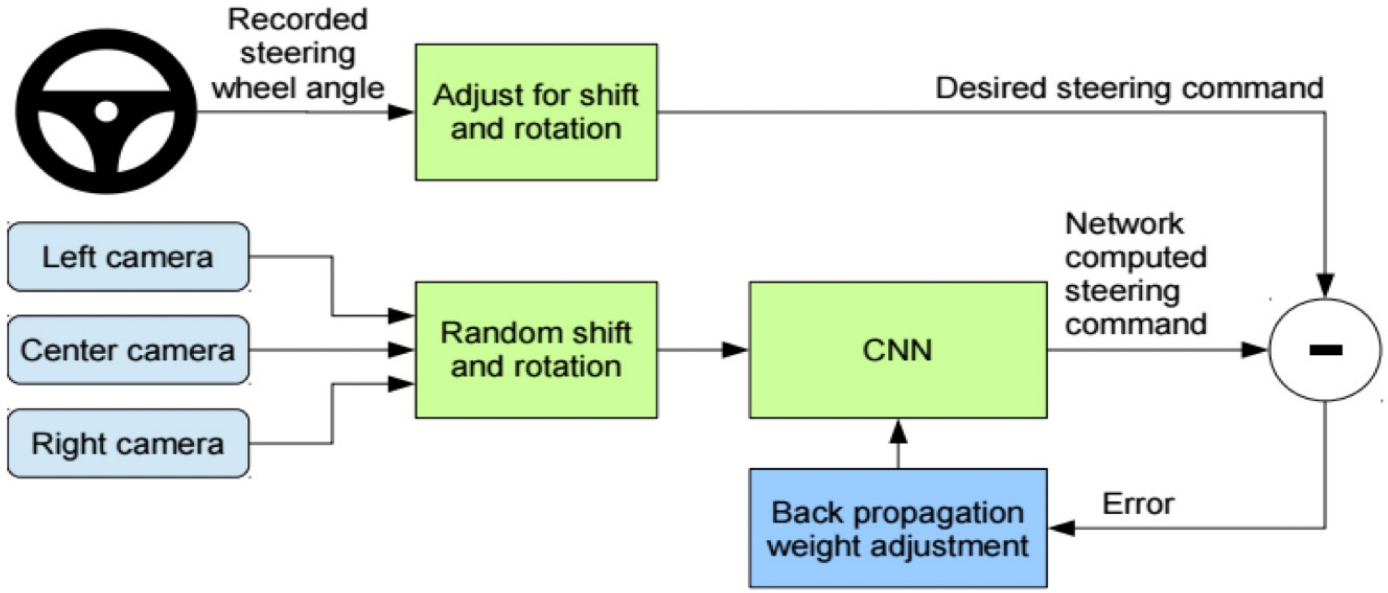
R. G. Kim et al., "Imitation Learning for Dynamic VFI Control in Large-Scale Manycore Systems," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 9, pp. 2458-2471, Sept. 2017, doi: 10.1109/TVLSI.2017.2700726.

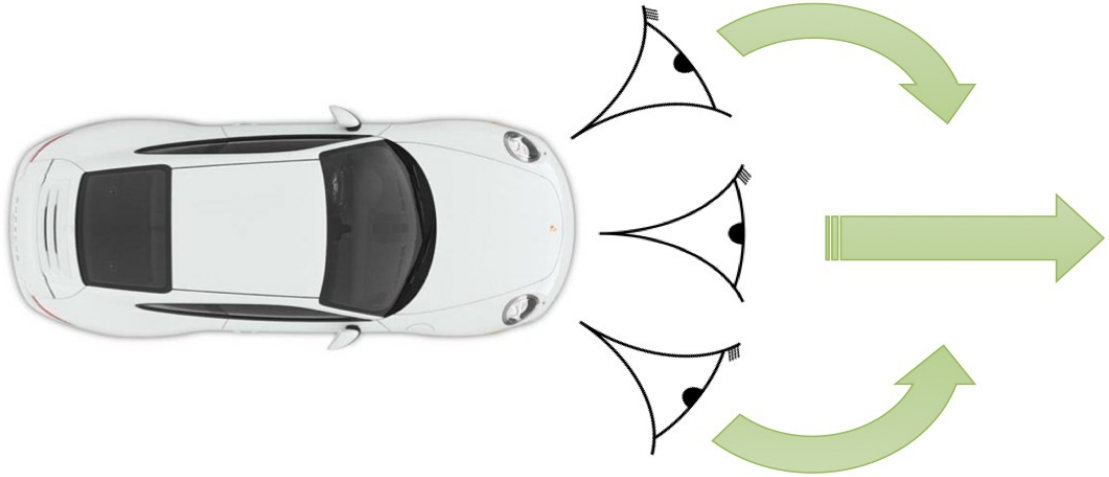
Imitation learning (2)



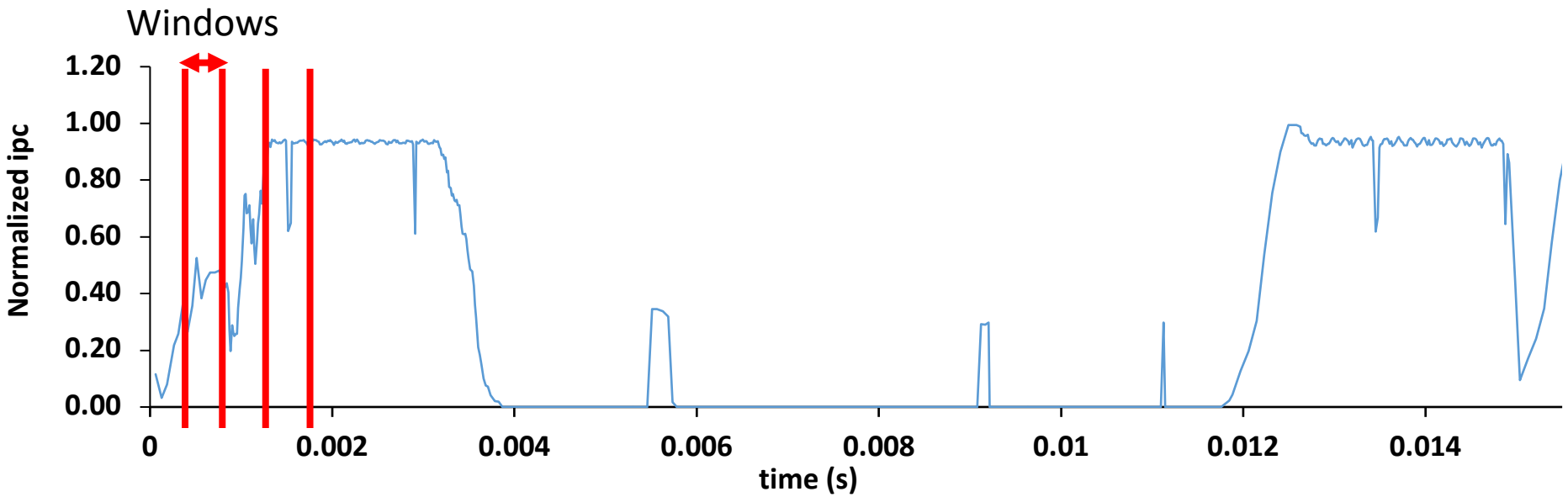
Experts:
Humans
Optimal policies

Imitation learning (3)



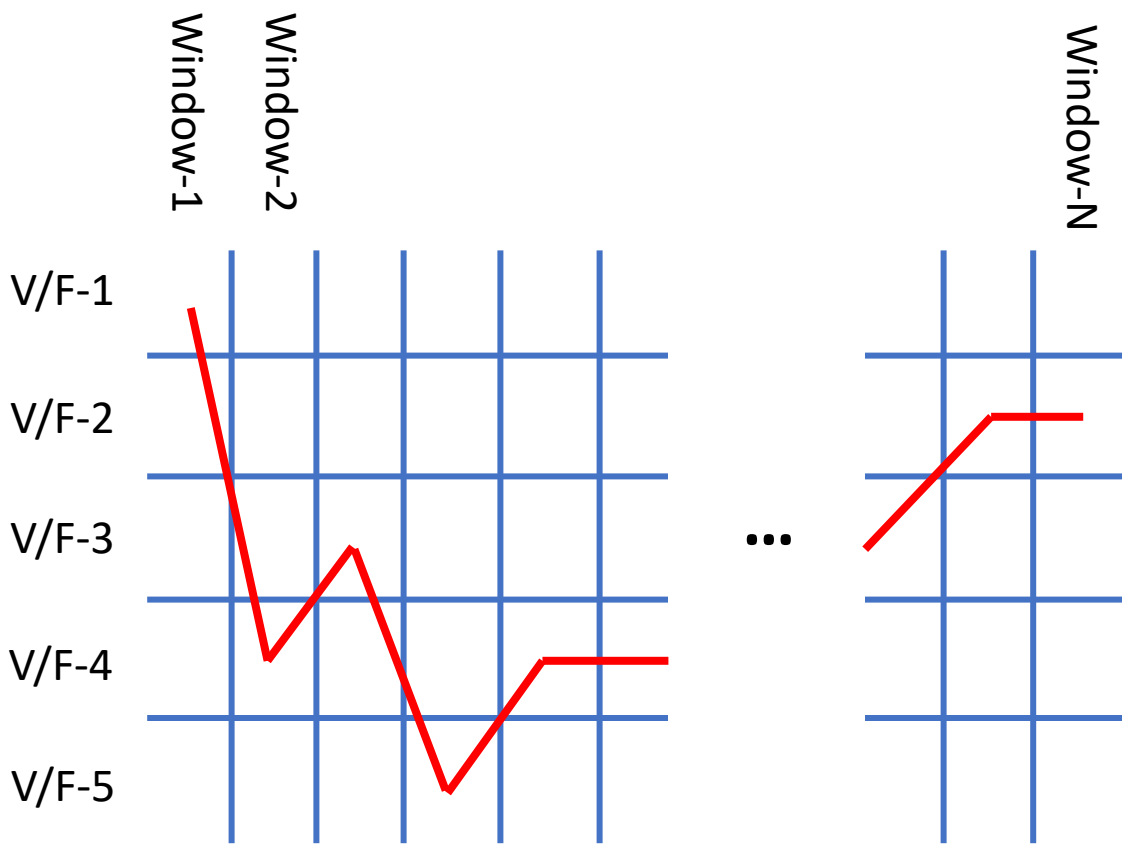


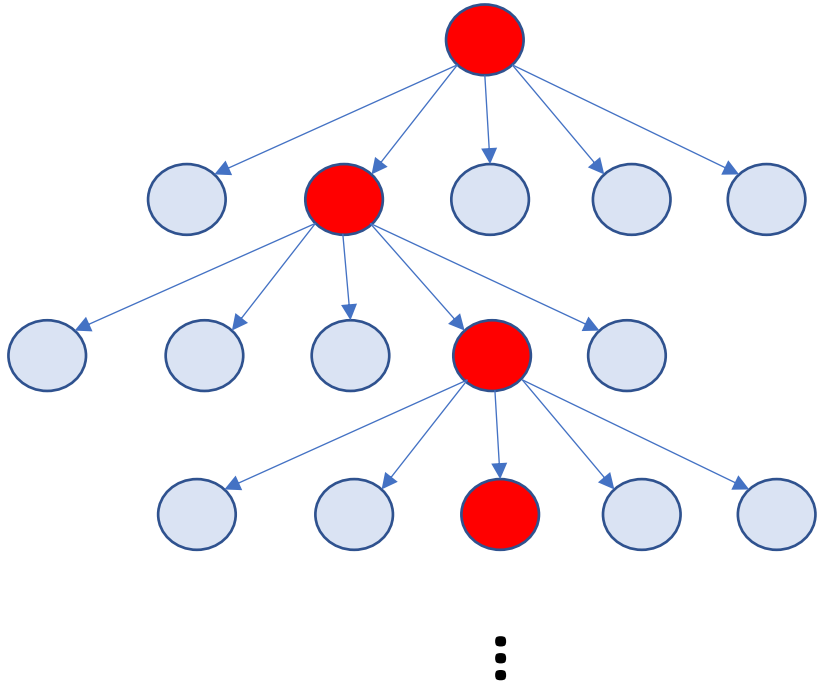
Creating the oracle for IL-based DVFI(1)



- Window-wise data for different V/F using Gem5
- Impossible to create oracle considering all windows
- Create the oracle using smaller segments

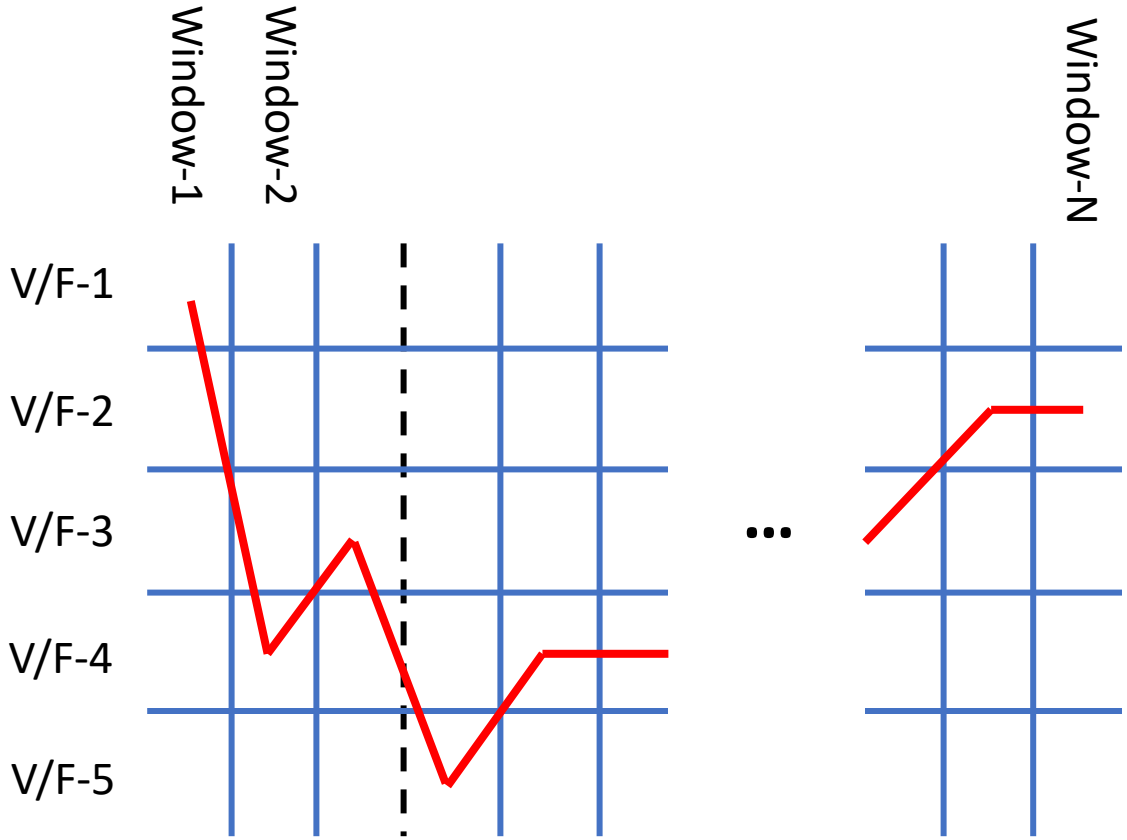
Creating the oracle for IL-based DVFI(2)

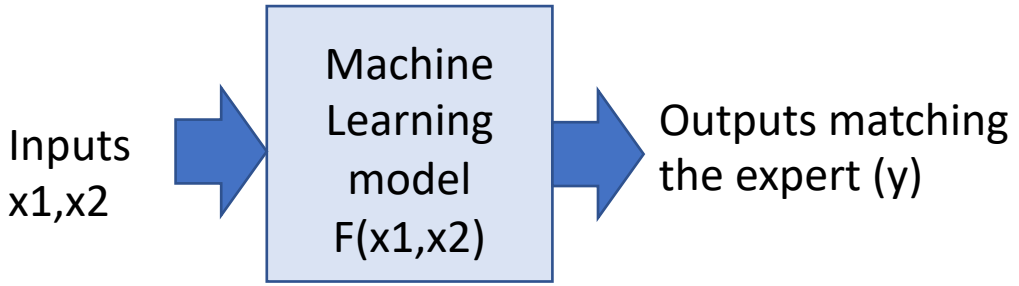




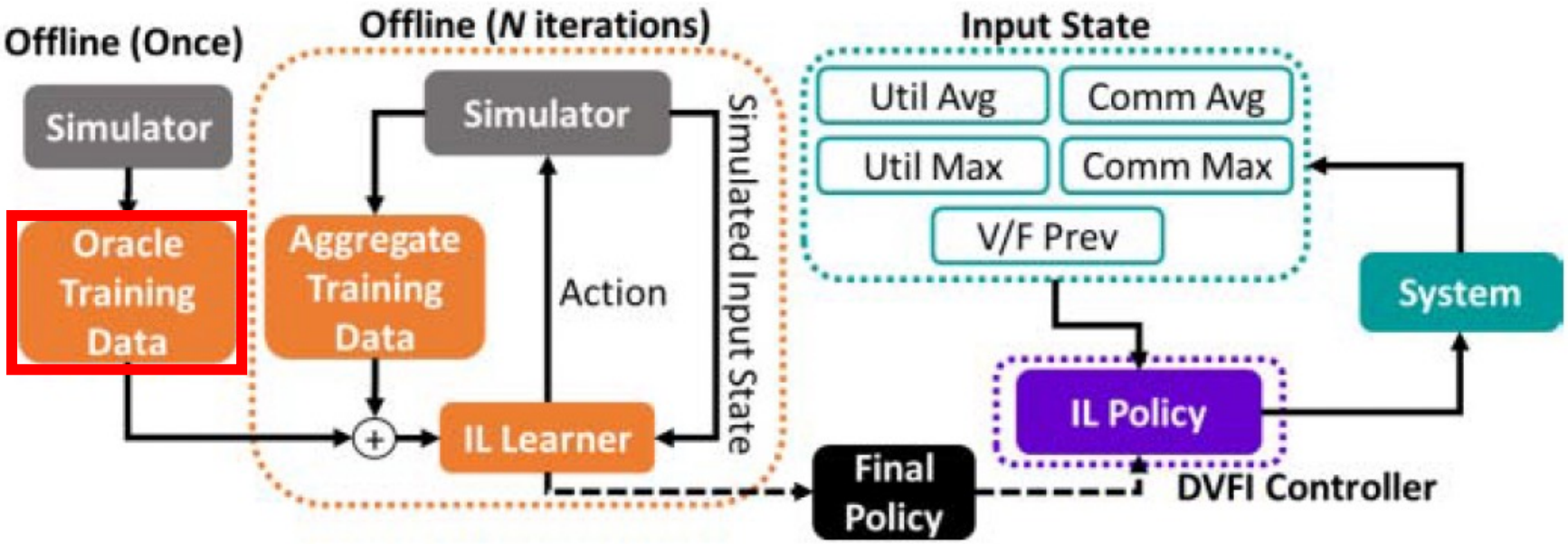
- **Find the right set of V/F for each window to reduce overall power**

Imitating the Expert



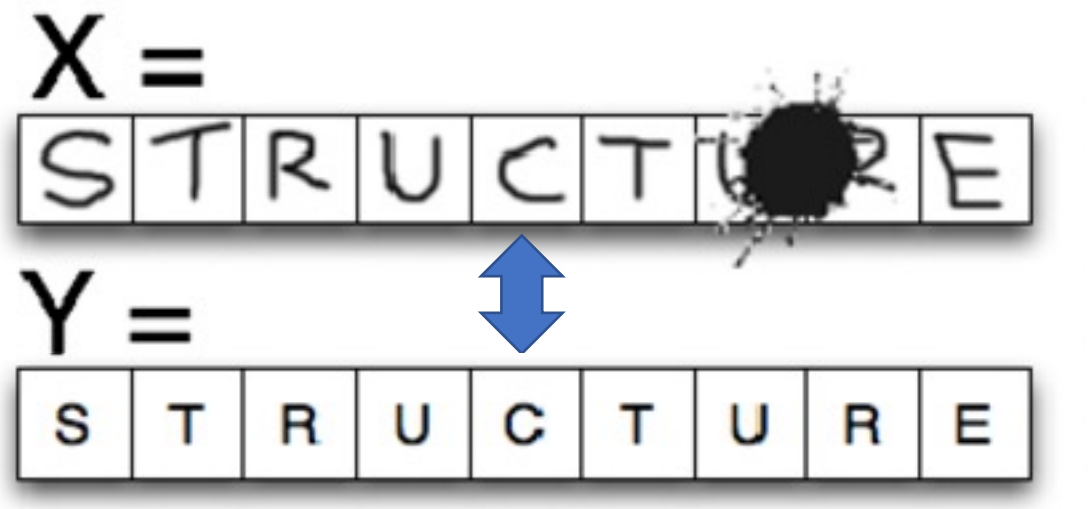


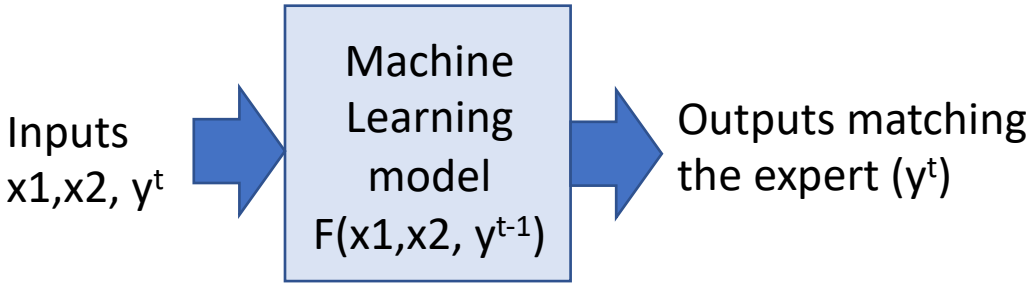
IL for calculating V/F



- We have the oracle now
- Use traditional ML to learn using oracle as ground truth
- IL learner can be linear regression or any other model

Structured Prediction



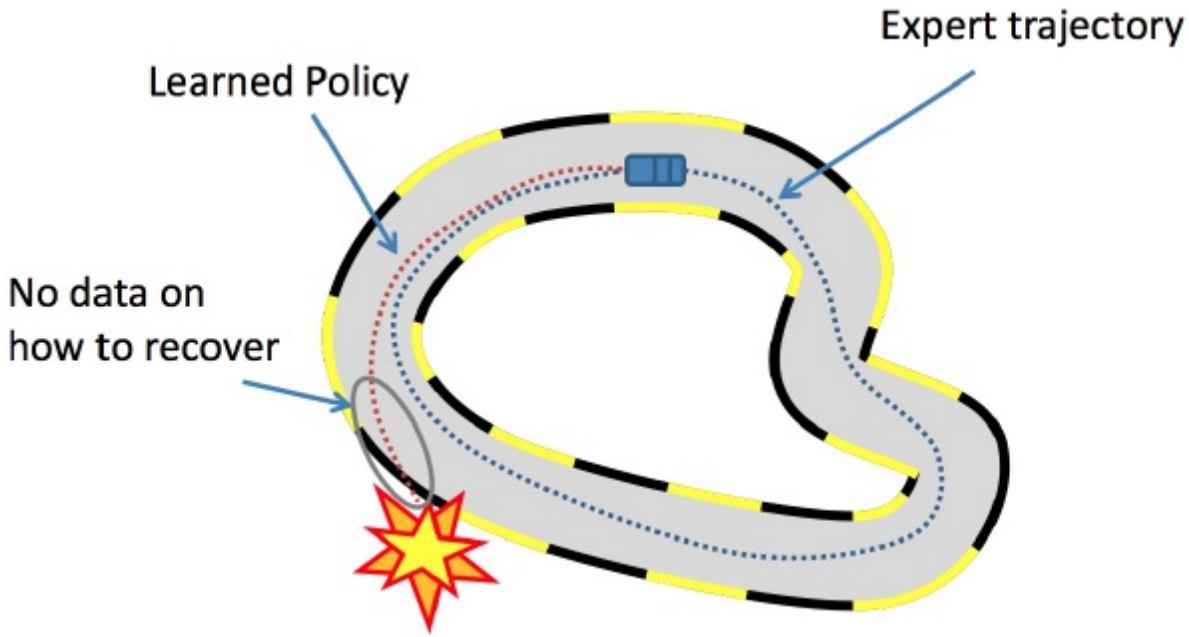


- **Does V/F of previous window affect V/F of current window?**

Dagger algorithm(1)

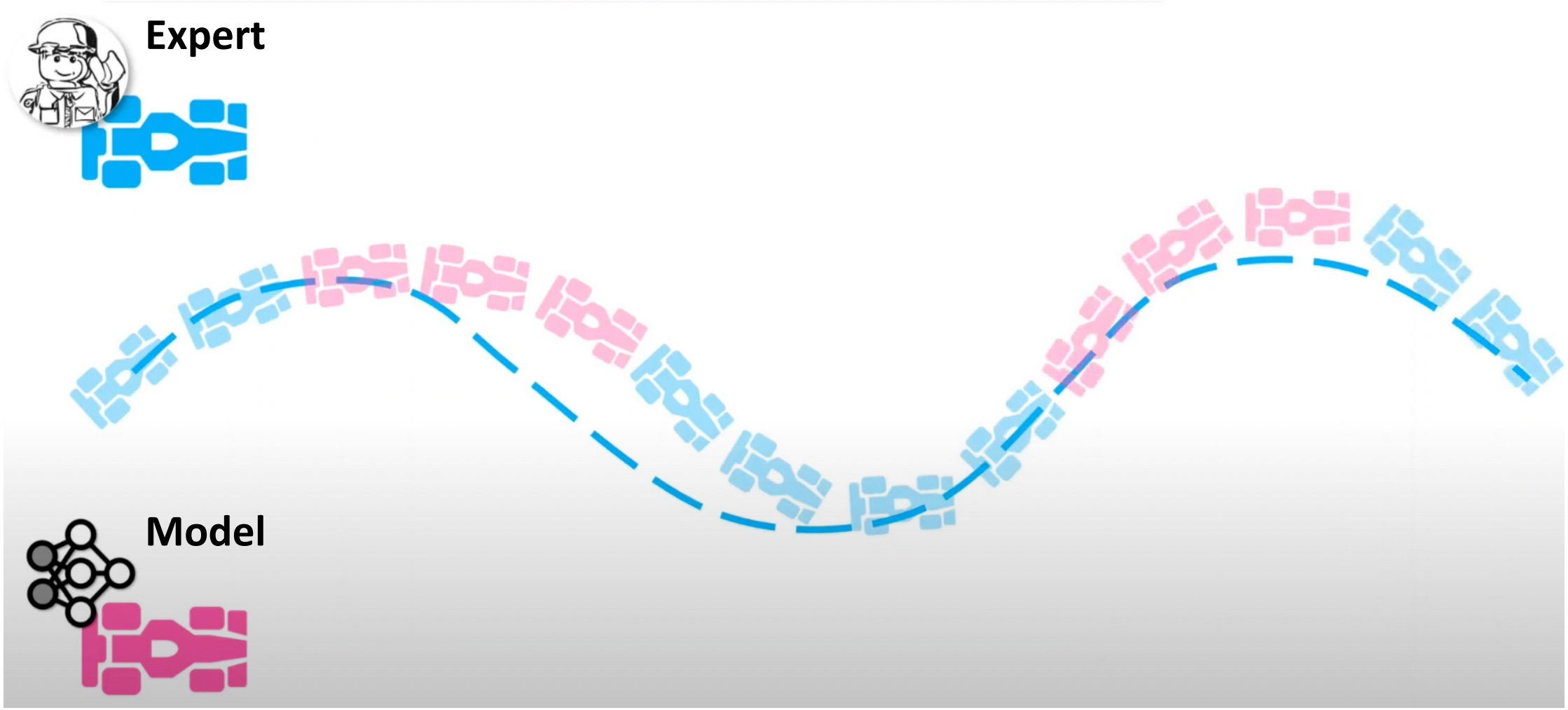
Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.

Algorithm 3.1: DAGGER Algorithm.



https://www.youtube.com/watch?v=gffZio5UmQc

Dagger algorithm(2)



- Sometimes let the model make mistakes
- Correct them using the expert