

EnergySmart: Toward Energy-Efficient Many-Cores for Near-Threshold Computing

Ulya R. Karpuzcu, Abhishek Sinkar*, Nam Sung Kim*, Josep Torrellas❖
University of Minnesota ❖University of Illinois *University of Wisconsin



UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN



WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON



UNIVERSITY OF MINNESOTA

Low Voltage Operation for Energy Efficiency

Power x Energy efficiency = Performance



Low Voltage Operation for Energy Efficiency

$$\text{Power} \times \text{Energy efficiency} = \text{Performance}$$
$$\propto f(\text{frequency}) \times \#\text{cores}$$



Low Voltage Operation for Energy Efficiency

$$\text{Power} \times \text{Energy efficiency} = \text{Performance}$$
$$\propto f(\text{frequency}) \times \#\text{cores}$$



Low Voltage Operation for Energy Efficiency

$$\text{Power} \times \text{Energy efficiency} = \text{Performance}$$
$$\propto f(\text{frequency}) \times \#\text{cores}$$



Low Voltage Operation for Energy Efficiency

$$\text{Power} \times \text{Energy efficiency} = \text{Performance}$$
$$\propto f(\text{frequency}) \times \#\text{cores}$$

- Vdd scaling increases energy-efficiency



Low Voltage Operation for Energy Efficiency

$$\text{Power} \times \text{Energy efficiency} = \text{Performance} \\ \propto f(\text{frequency}) \times \#\text{cores}$$

- Vdd scaling increases energy-efficiency
- Near-Threshold Voltage -- NTV Operation



Low Voltage Operation for Energy Efficiency

$$\text{Power} \times \text{Energy efficiency} = \text{Performance} \\ \propto f(\text{frequency}) \times \#\text{cores}$$

- Vdd scaling increases energy-efficiency
- Near-Threshold Voltage -- NTV Operation
 - Vdd reduces to slightly above Vth: ~0.5V



Low Voltage Operation for Energy Efficiency

$$\text{Power} \times \text{Energy efficiency} = \text{Performance} \\ \propto f(\text{frequency}) \times \#\text{cores}$$

- Vdd scaling increases energy-efficiency
- **Near-Threshold Voltage -- NTV Operation**
 - Vdd reduces to slightly above Vth: ~0.5V
 - vs. 1V for conventional or Super-threshold voltage -- STV



Low Voltage Operation for Energy Efficiency

$$\text{Power} \times \text{Energy efficiency} = \text{Performance} \\ \propto f(\text{frequency}) \times \#\text{cores}$$

- Vdd scaling increases energy-efficiency
- **Near-Threshold Voltage -- NTV Operation**
 - Vdd reduces to slightly above Vth: ~0.5V
 - vs. 1V for conventional or Super-threshold voltage -- STV
- Drawbacks



Low Voltage Operation for Energy Efficiency

$$\text{Power} \times \text{Energy efficiency} = \text{Performance} \\ \propto f(\text{frequency}) \times \#\text{cores}$$

- Vdd scaling increases energy-efficiency
- **Near-Threshold Voltage -- NTV Operation**
 - Vdd reduces to slightly above Vth: ~0.5V
 - vs. 1V for conventional or Super-threshold voltage -- STV
- Drawbacks
 - Lower f: Need more parallelism ➔ large chips



Low Voltage Operation for Energy Efficiency

$$\text{Power} \times \text{Energy efficiency} = \text{Performance} \\ \propto f(\text{frequency}) \times \#\text{cores}$$

- Vdd scaling increases energy-efficiency
- **Near-Threshold Voltage -- NTV Operation**
 - Vdd reduces to slightly above Vth: ~0.5V
 - vs. 1V for conventional or Super-threshold voltage -- STV
- Drawbacks
 - Lower f: Need more parallelism ➔ large chips
 - Higher impact of variation



Contribution



EnergySmart

Contribution

- Conventional techniques to tolerate variation are not cost-effective at NTV



Contribution

- Conventional techniques to tolerate variation are not cost-effective at NTV
 - Support for multiple on-chip Vdd domains hurts energy-efficiency



Contribution

- Conventional techniques to tolerate variation are not cost-effective at NTV
 - Support for multiple on-chip Vdd domains hurts energy-efficiency
- Propose **EnergySmart** organization to maximize energy-efficiency at NTV



Contribution

- Conventional techniques to tolerate variation are not cost-effective at NTV
 - Support for multiple on-chip Vdd domains hurts energy-efficiency
- Propose **EnergySmart** organization to maximize energy-efficiency at NTV
 - Single Vdd domain, multiple f domains



Contribution

- Conventional techniques to tolerate variation are not cost-effective at NTV
 - Support for multiple on-chip Vdd domains hurts energy-efficiency
- Propose **EnergySmart** organization to maximize energy-efficiency at NTV
 - Single Vdd domain, multiple f domains
 - Simple hardware



Contribution

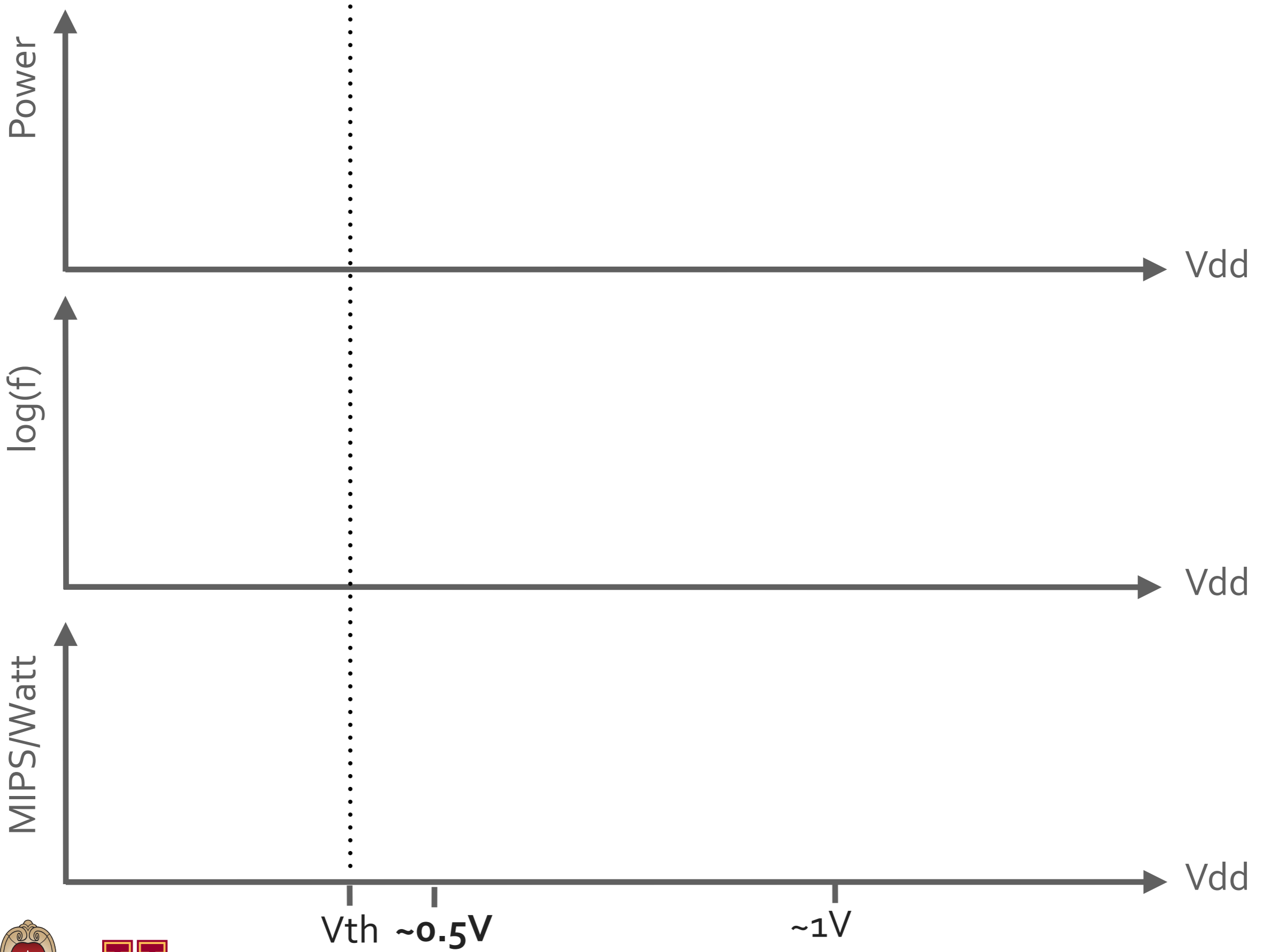
- Conventional techniques to tolerate variation are not cost-effective at NTV
 - Support for multiple on-chip Vdd domains hurts energy-efficiency
- Propose **EnergySmart** organization to maximize energy-efficiency at NTV
 - Single Vdd domain, multiple f domains
 - Simple hardware
 - Smart, variation-aware core assignment

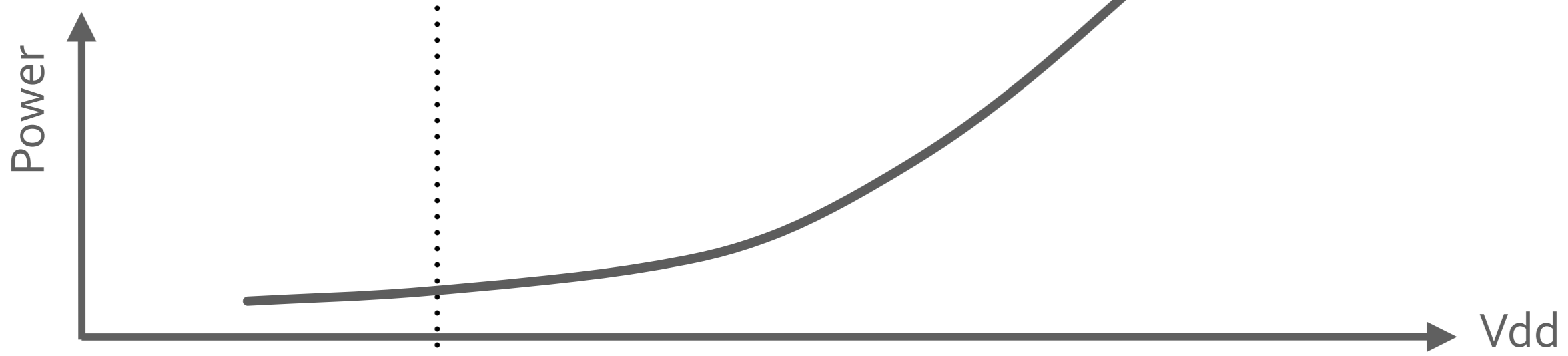


Contribution

- Conventional techniques to tolerate variation are not cost-effective at NTV
 - Support for multiple on-chip Vdd domains hurts energy-efficiency
- Propose **EnergySmart** organization to maximize energy-efficiency at NTV
 - Single Vdd domain, multiple f domains
 - Simple hardware
 - Smart, variation-aware core assignment
 - Energy-efficiency within 81% of perfect organization (as opposed to 69% for conventional)





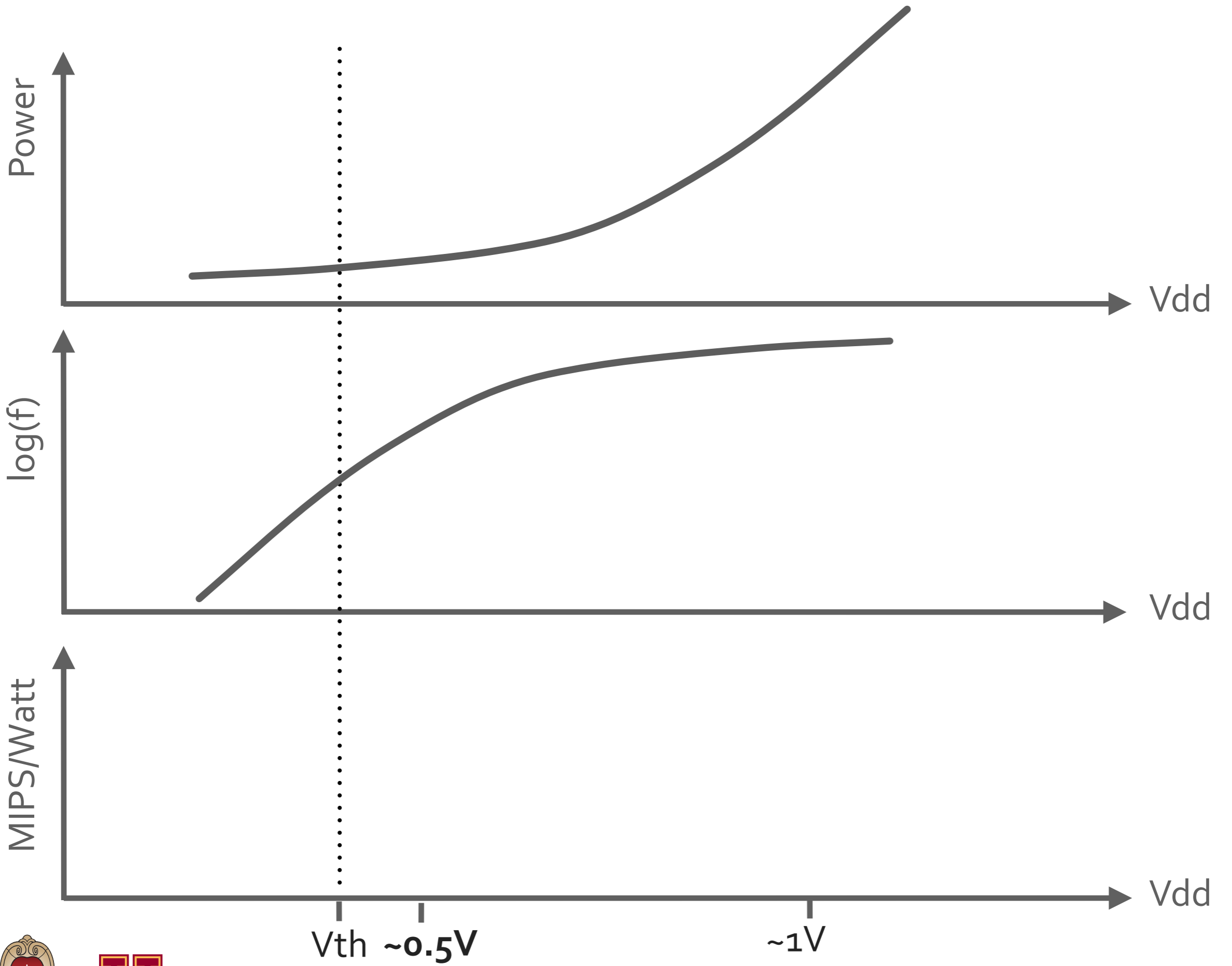


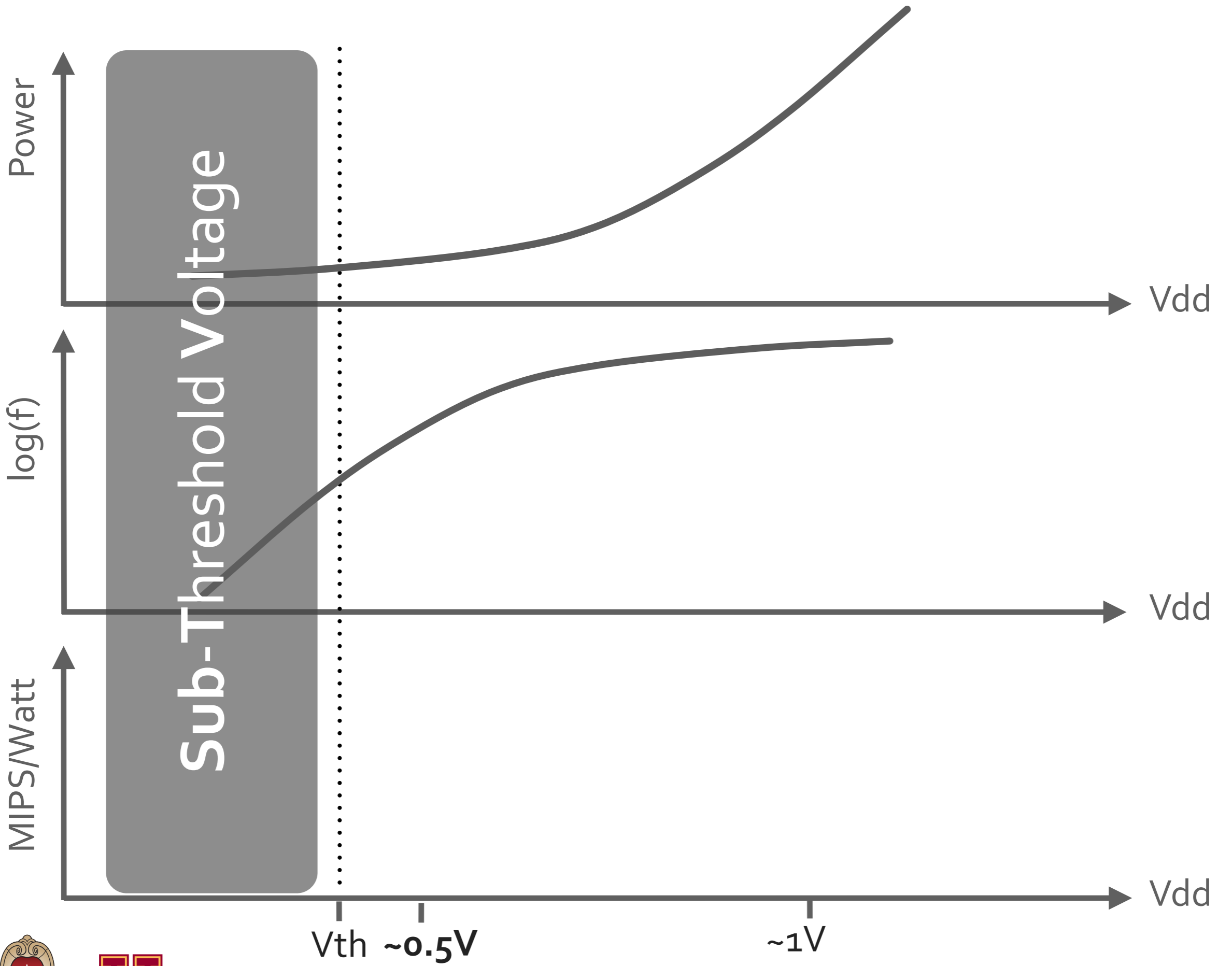
Vth ~0.5V

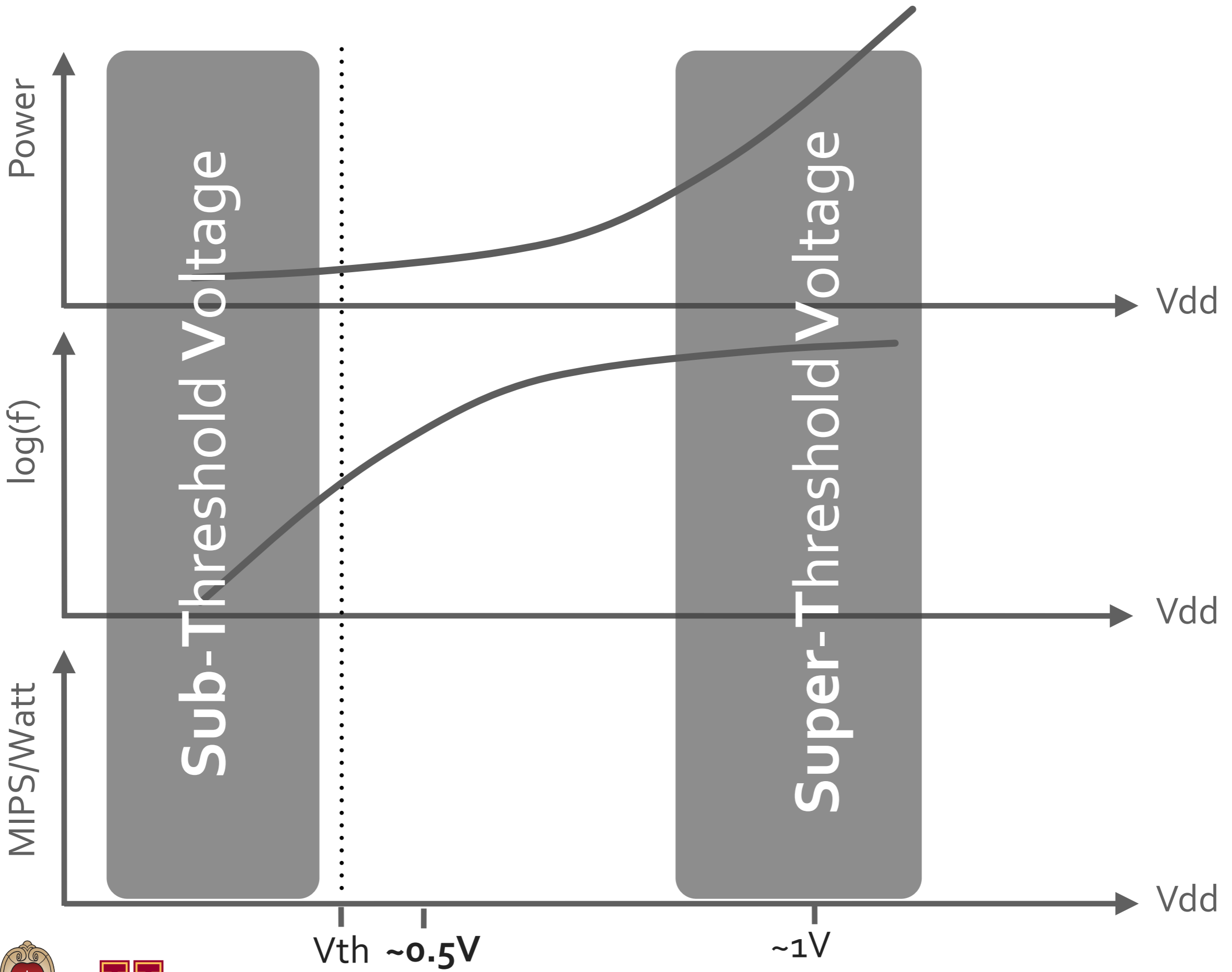
~1V

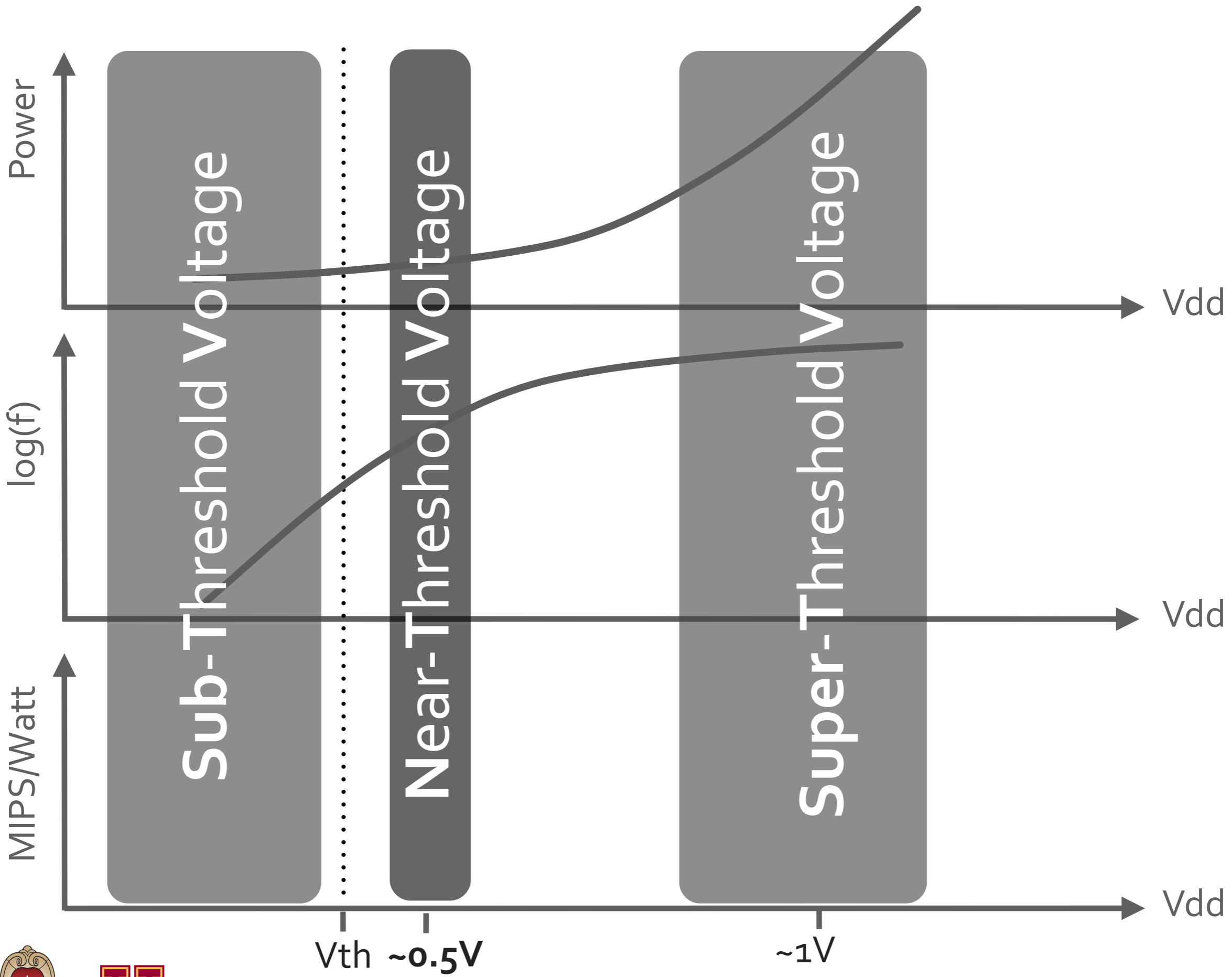
EnergySmart

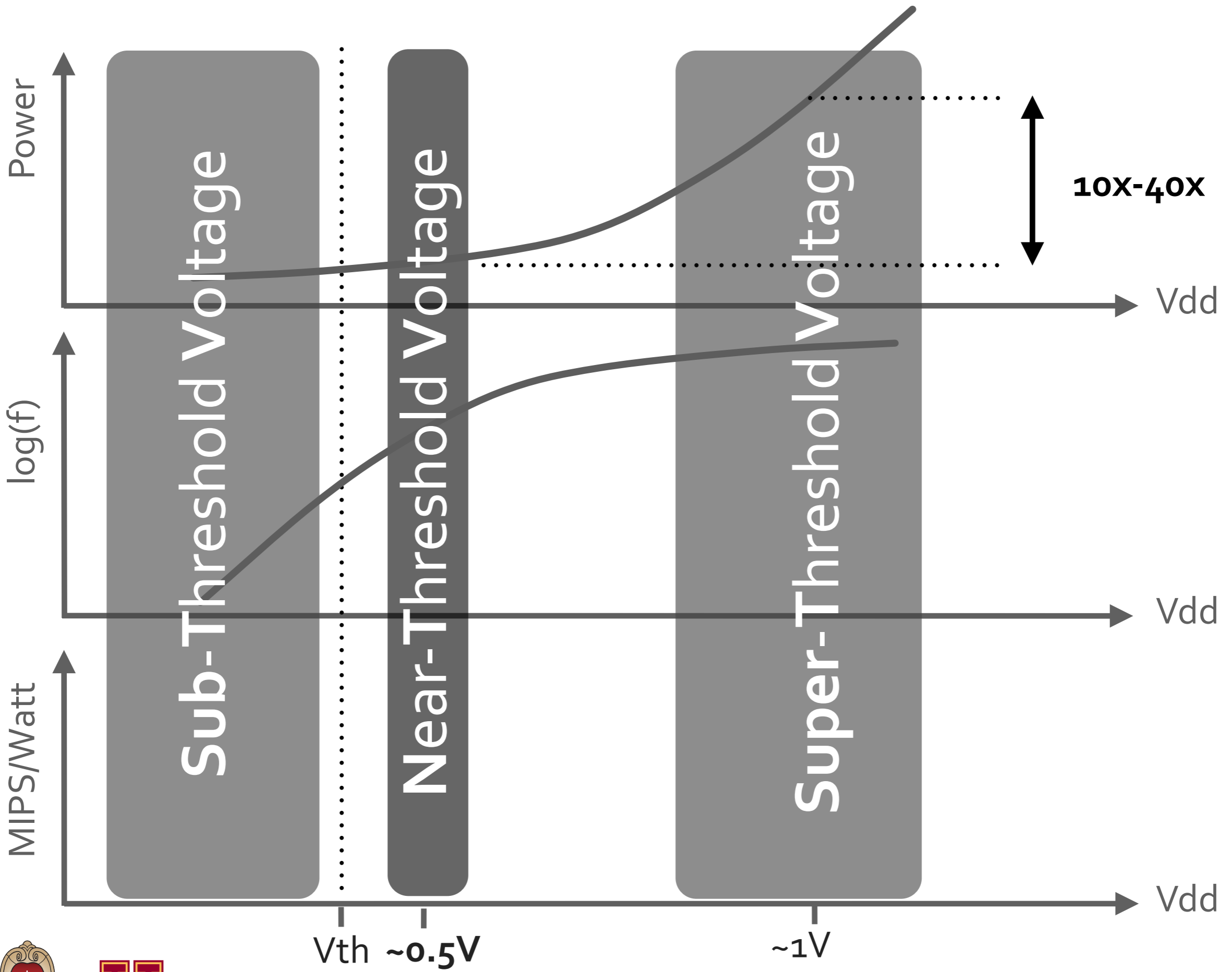


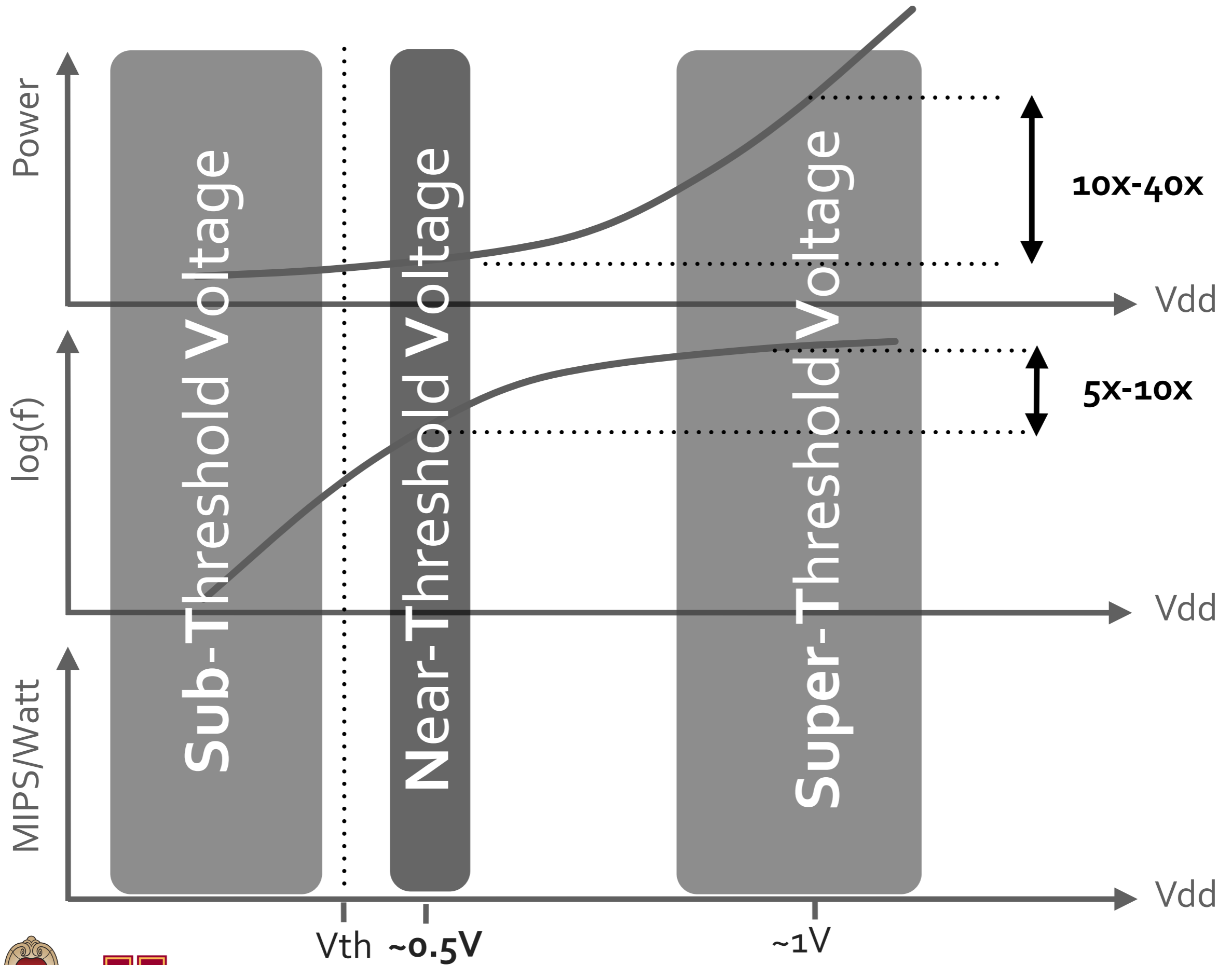


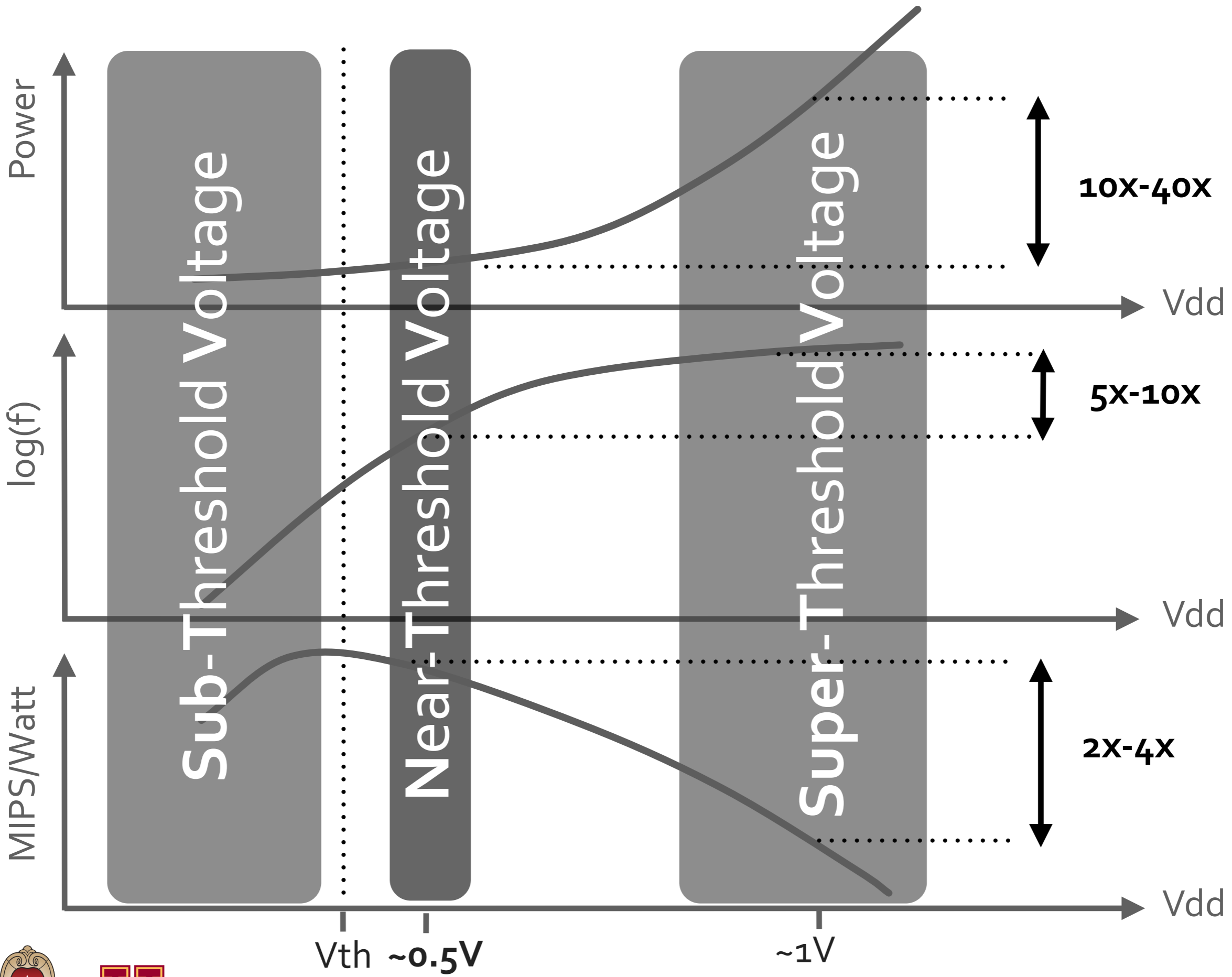


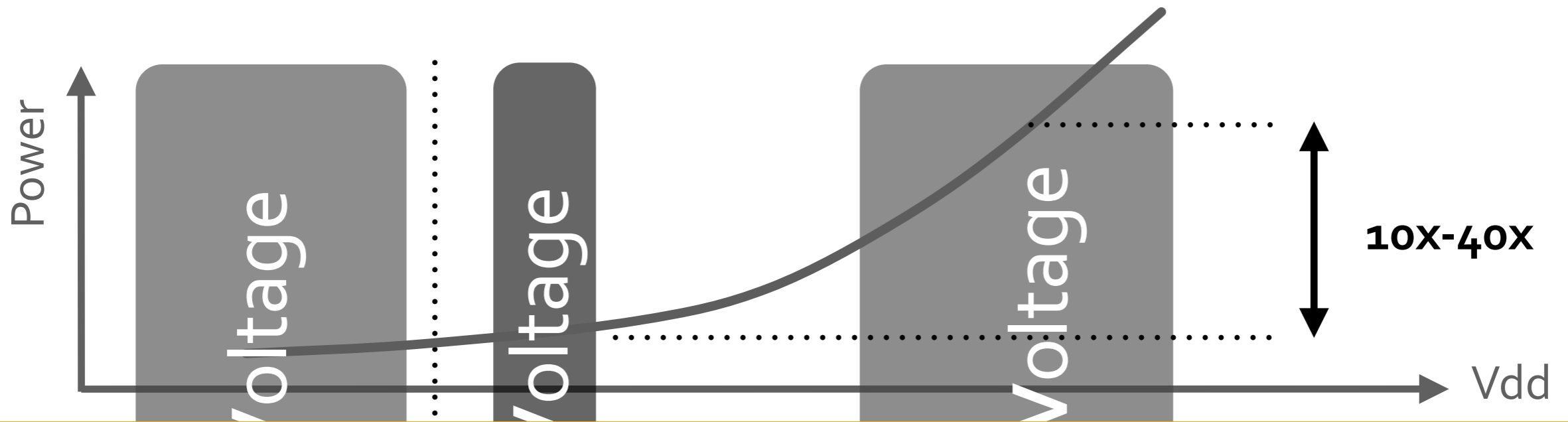




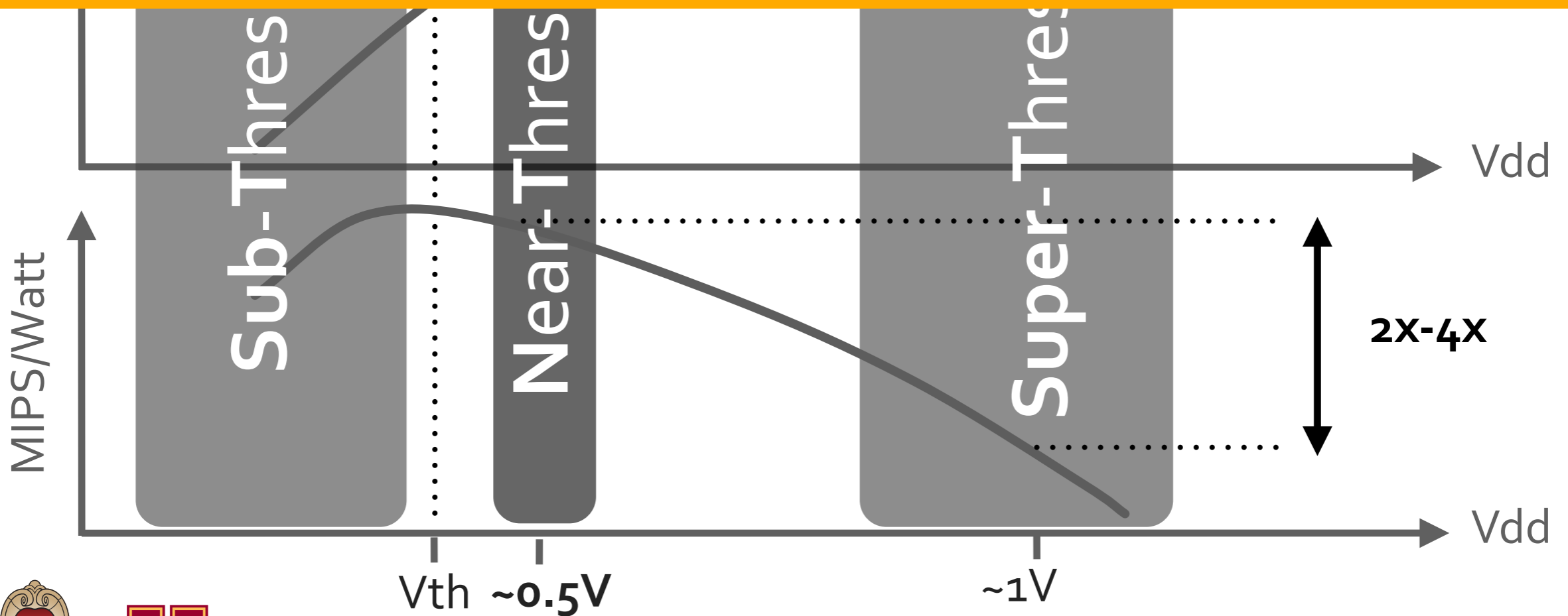








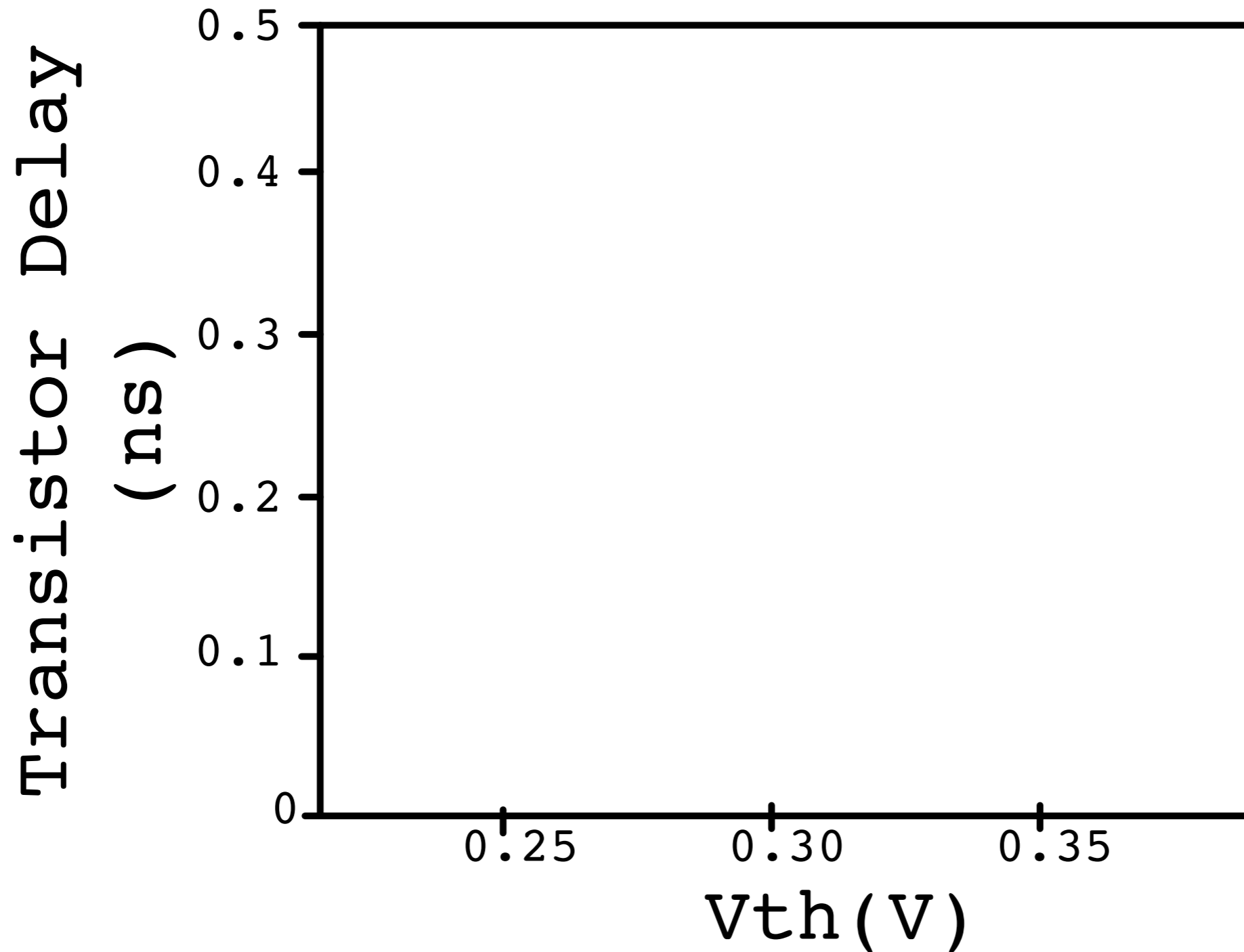
At NTV, more cores can be active than at STV



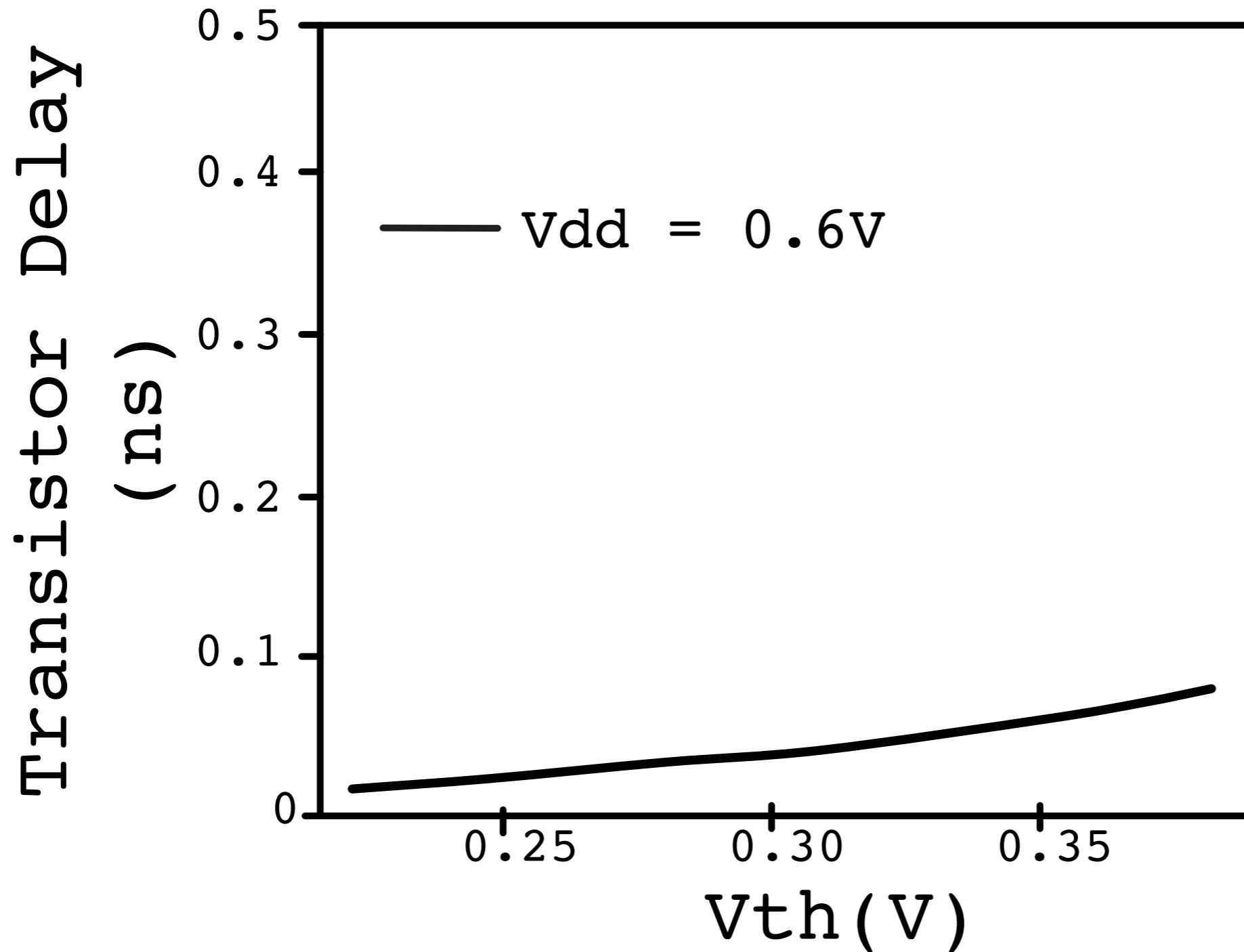
NTV Drawback: Parametric Variation



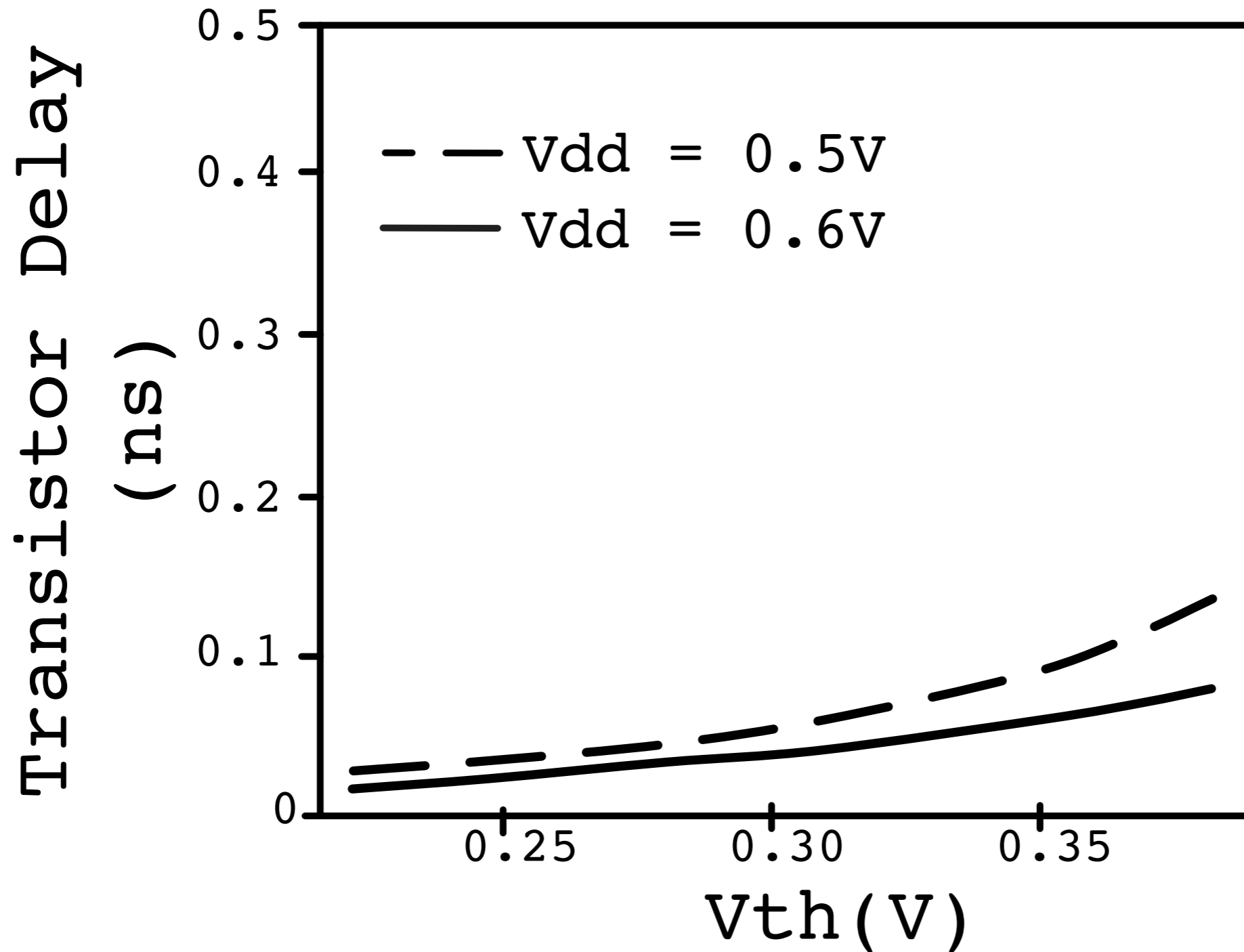
NTV Drawback: Parametric Variation



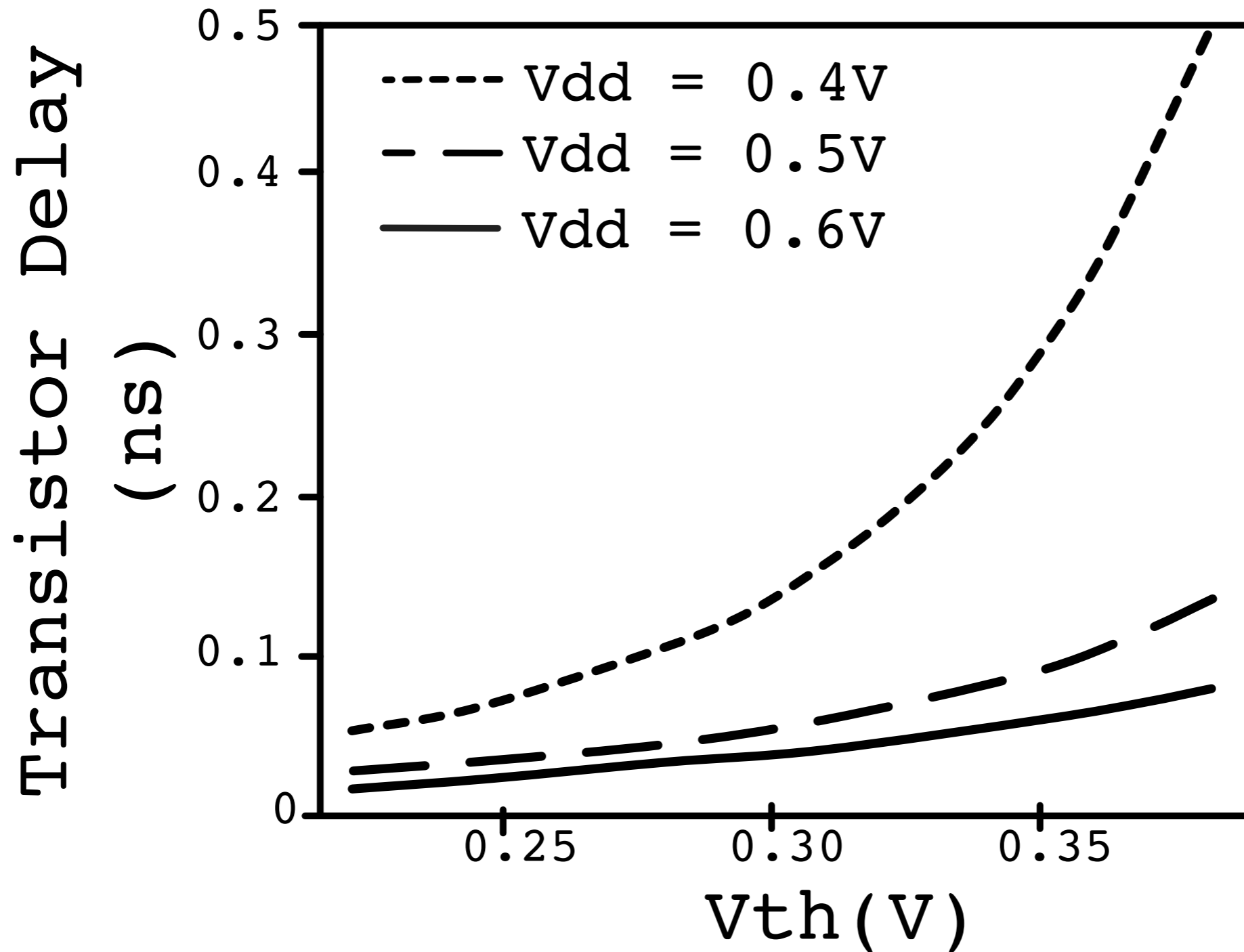
NTV Drawback: Parametric Variation



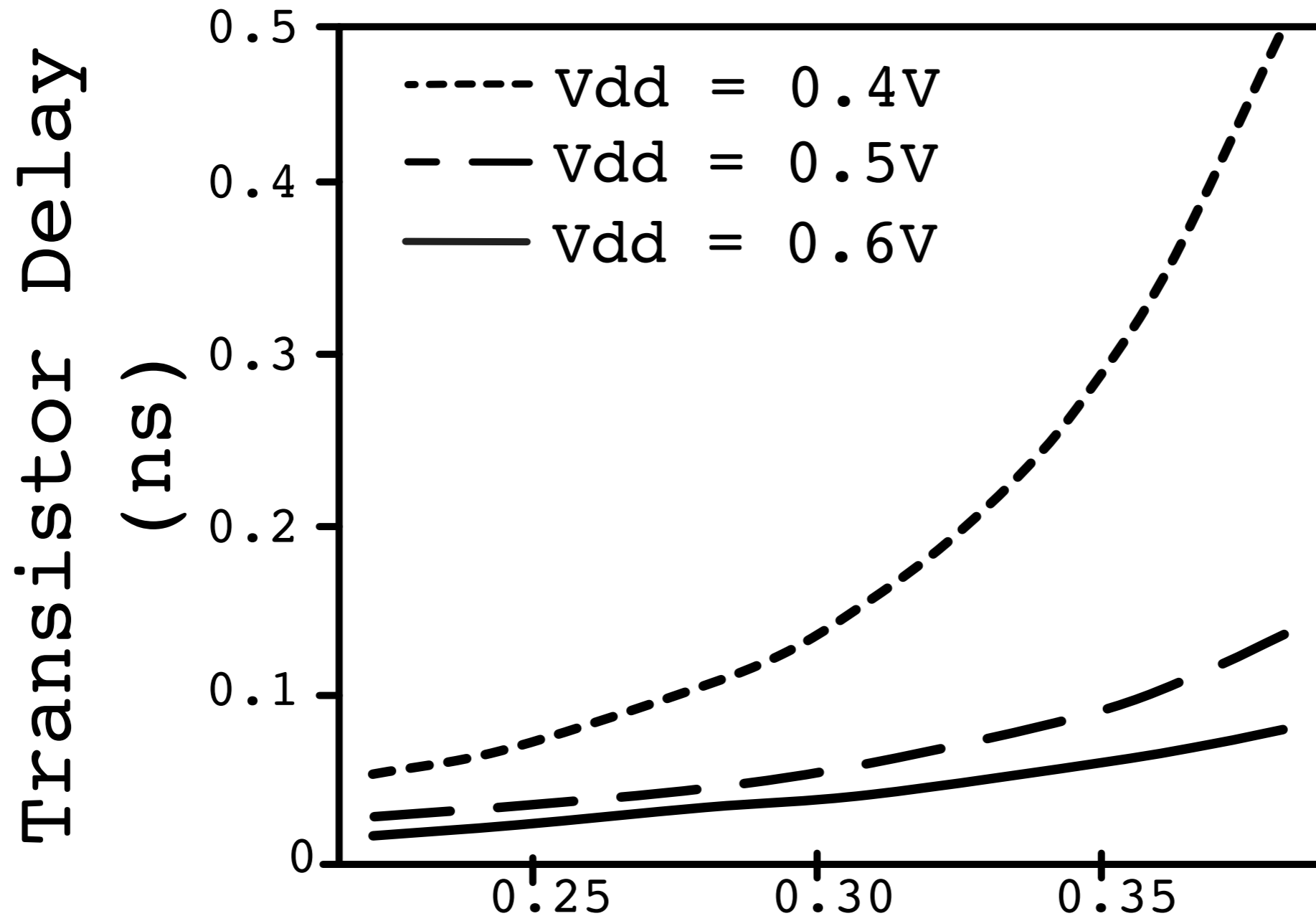
NTV Drawback: Parametric Variation



NTV Drawback: Parametric Variation



NTV Drawback: Parametric Variation



Same ΔV_{th} causes higher f variation at NTV than at STV



Traditional Ways of Handling Variation



Traditional Ways of Handling Variation

- Attack source of variation ➔ adapt V_{th}



Traditional Ways of Handling Variation

- Attack source of variation ➔ adapt V_{th}
 - Adaptive Body Biasing



Traditional Ways of Handling Variation

- Attack source of variation ➔ adapt V_{th}
 - Adaptive Body Biasing
 - Questionable for new device architectures



Traditional Ways of Handling Variation

- Attack source of variation ➔ adapt V_{th}
 - Adaptive Body Biasing
 - Questionable for new device architectures
- Attack effect of variation ➔ adapt V_{dd}



Traditional Ways of Handling Variation

- Attack source of variation ➔ adapt V_{th}
 - Adaptive Body Biasing
 - Questionable for new device architectures
- Attack effect of variation ➔ adapt V_{dd}
 - Adaptive Supply Voltage, Dynamic Voltage Scaling



Traditional Ways of Handling Variation

- Attack source of variation ➔ adapt V_{th}
 - Adaptive Body Biasing
 - Questionable for new device architectures
- Attack effect of variation ➔ adapt V_{dd}
 - Adaptive Supply Voltage, Dynamic Voltage Scaling
 - Effectiveness increases with multiple voltage domains



Traditional Ways of Handling Variation

- Attack source of variation ➔ adapt V_{th}
 - Adaptive Body Biasing
 - Questionable for new device architectures
- Attack effect of variation ➔ adapt V_{dd}
 - Adaptive Supply Voltage, Dynamic Voltage Scaling
 - Effectiveness increases with multiple voltage domains
 - Cost-effective?



Multiple Vdd Domains: Cost-effective?



Multiple Vdd Domains: Cost-effective?

(I) On chip regulators' high power loss



Multiple Vdd Domains: Cost-effective?

(I) On chip regulators' high power loss

- 10% power loss or more



Multiple Vdd Domains: Cost-effective?

(I) On chip regulators' high power loss

- 10% power loss or more
- Barely any loss is acceptable at energy-conscious NTV



Multiple Vdd Domains: Cost-effective?

(I) On chip regulators' high power loss

- 10% power loss or more
- Barely any loss is acceptable at energy-conscious NTV

(II) Vdd domain granularity (# of cores per domain)



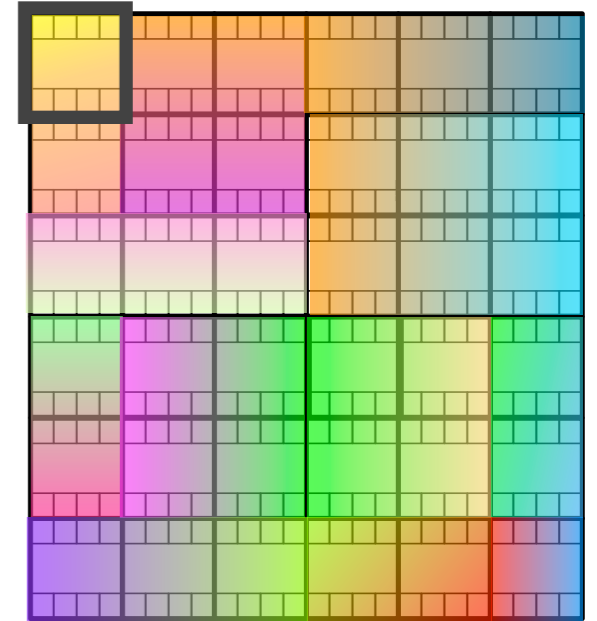
Multiple Vdd Domains: Cost-effective?

(I) On chip regulators' high power loss

- 10% power loss or more
- Barely any loss is acceptable at energy-conscious NTV

(II) Vdd domain granularity (# of cores per domain)

- Fine: Effective tuning but expensive due to large #cores



Multiple Vdd Domains: Cost-effective?

(I) On chip regulators' high power loss

- 10% power loss or more
- Barely any loss is acceptable at energy-conscious NTV

(II) Vdd domain granularity (# of cores per domain)

- Fine: Effective tuning but expensive due to large #cores
- Coarse: More practical, yet suffers from variation inside



Multiple Vdd Domains: Cost-effective?

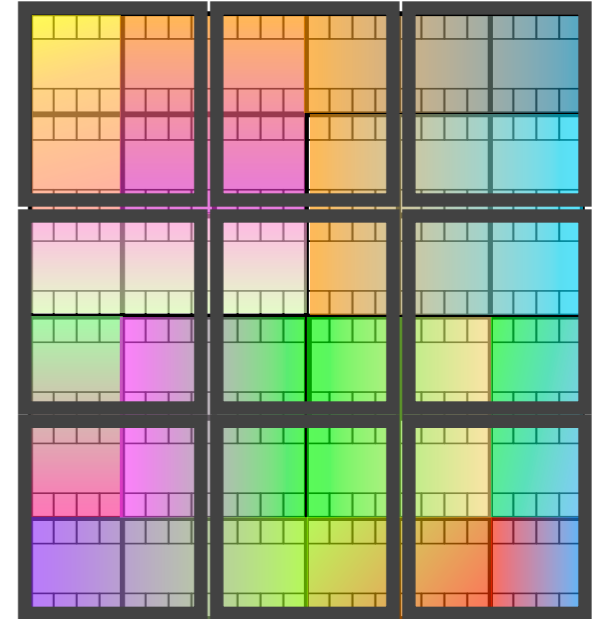
(I) On chip regulators' high power loss

- 10% power loss or more
- Barely any loss is acceptable at energy-conscious NTV

(II) Vdd domain granularity (# of cores per domain)

- Fine: Effective tuning but expensive due to large #cores
- Coarse: More practical, yet suffers from variation inside

(III) Vdd noise



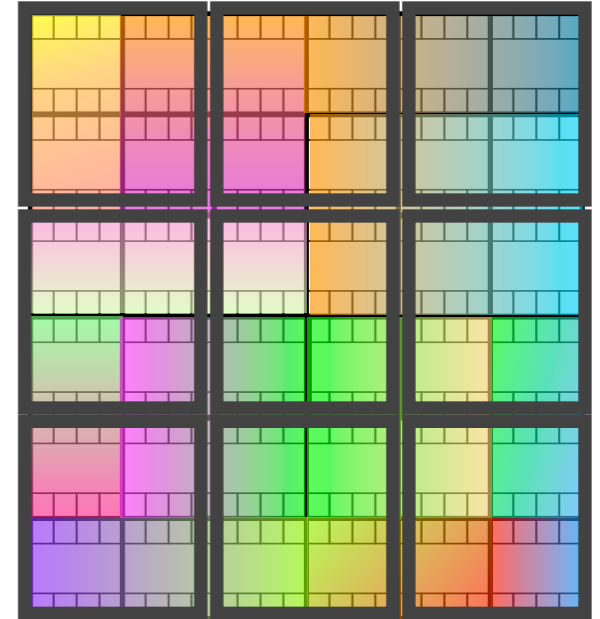
Multiple Vdd Domains: Cost-effective?

(I) On chip regulators' high power loss

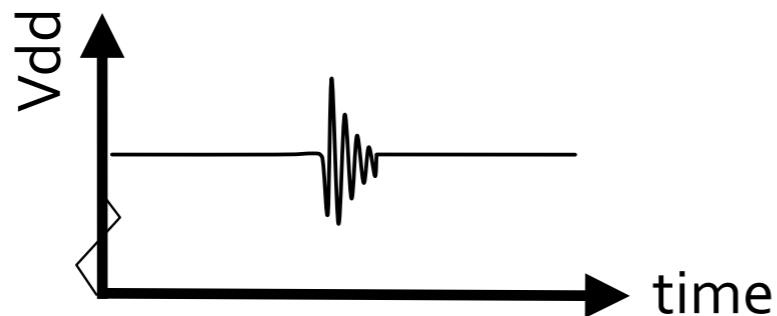
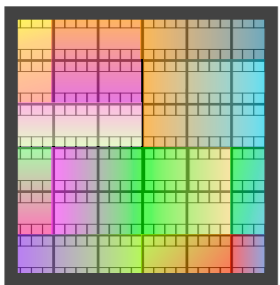
- 10% power loss or more
- Barely any loss is acceptable at energy-conscious NTV

(II) Vdd domain granularity (# of cores per domain)

- Fine: Effective tuning but expensive due to large #cores
- Coarse: More practical, yet suffers from variation inside



(III) Vdd noise



Large domain

- Averaging effects in the current drawn

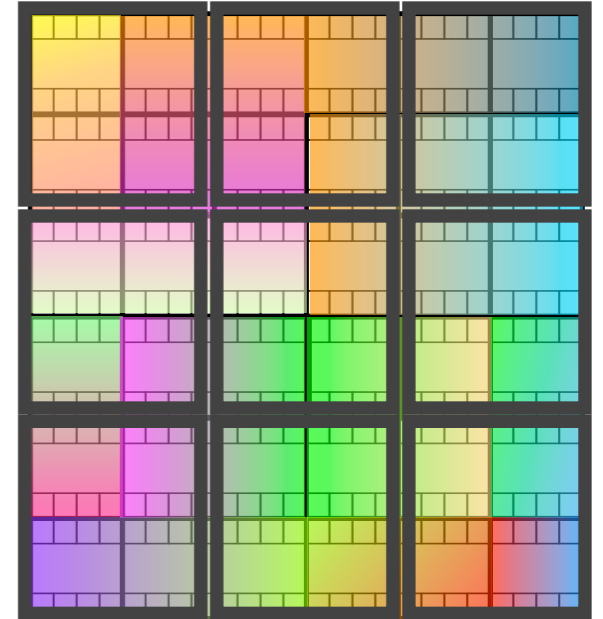
Multiple Vdd Domains: Cost-effective?

(I) On chip regulators' high power loss

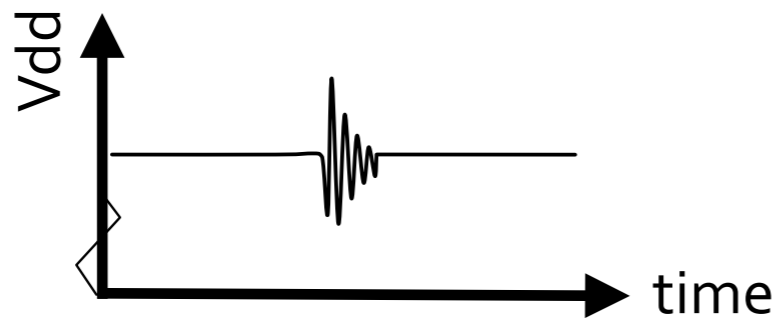
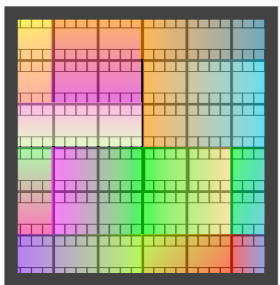
- 10% power loss or more
- Barely any loss is acceptable at energy-conscious NTV

(II) Vdd domain granularity (# of cores per domain)

- Fine: Effective tuning but expensive due to large #cores
- Coarse: More practical, yet suffers from variation inside

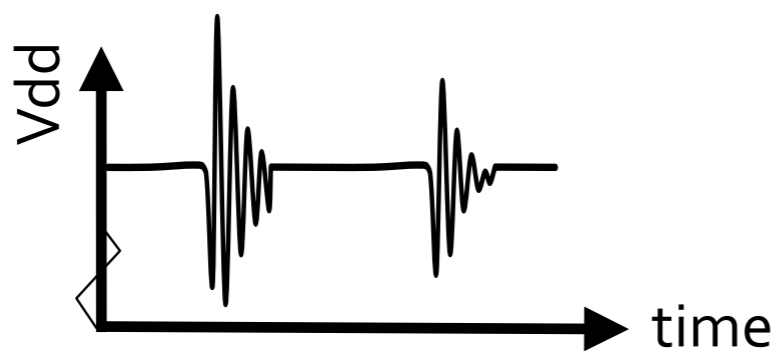
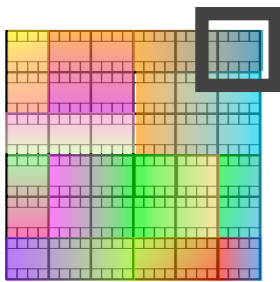


(III) Vdd noise



Large domain

- Averaging effects in the current drawn



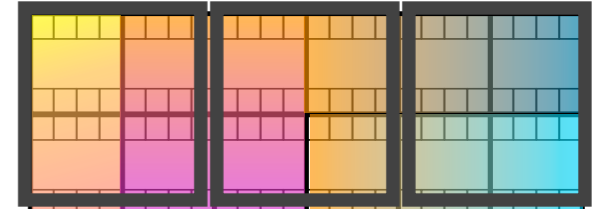
Small domain

- Less averaging \Rightarrow Higher Vdd droops
- Need to **increase Vdd margin**

Multiple Vdd Domains: Cost-effective?

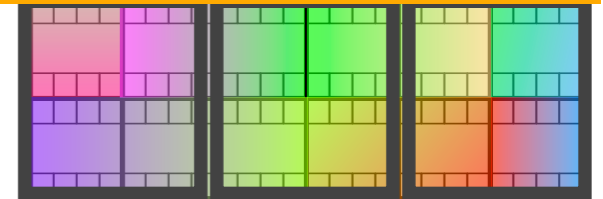
(I) On chip regulators' high power loss

- 10% power loss or more

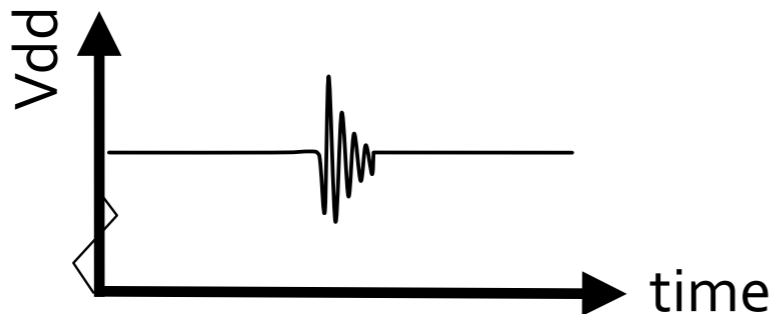
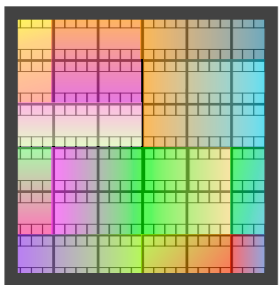


Let us keep one Vdd domain per chip. How to survive?

- Fine: Effective tuning but expensive due to large #cores
- Coarse: More practical, yet suffers from variation inside

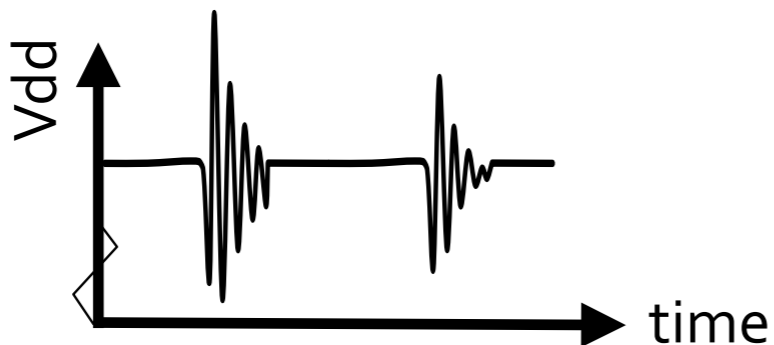
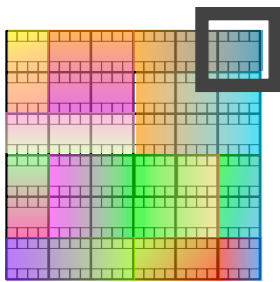


(III) Vdd noise



Large domain

- Averaging effects in the current drawn



Small domain

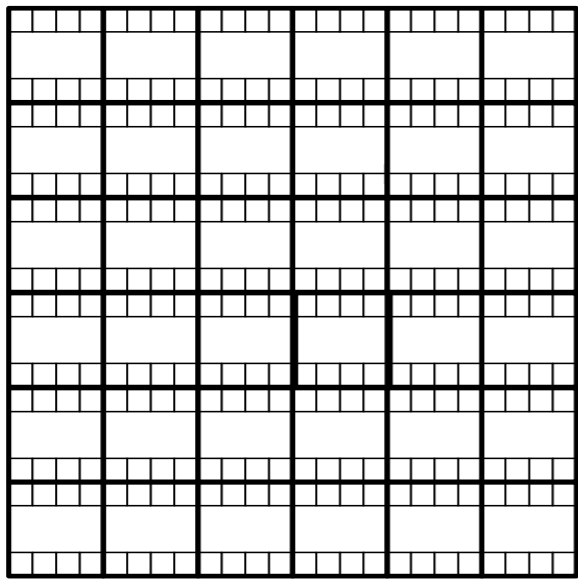
- Less averaging \Rightarrow Higher Vdd droops
- Need to **increase Vdd margin**

EnergySmart: Eschewing Multiple Vdd Domains



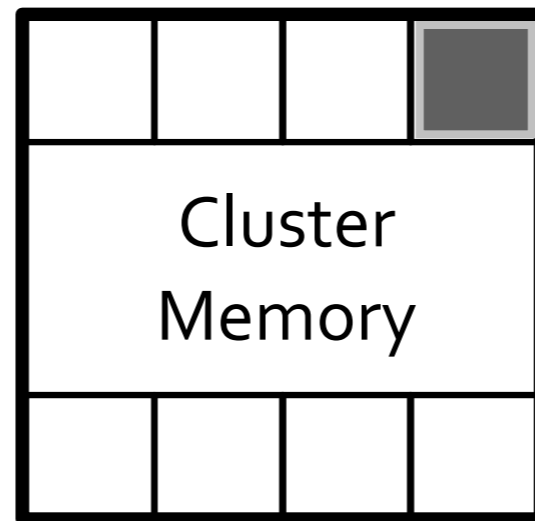
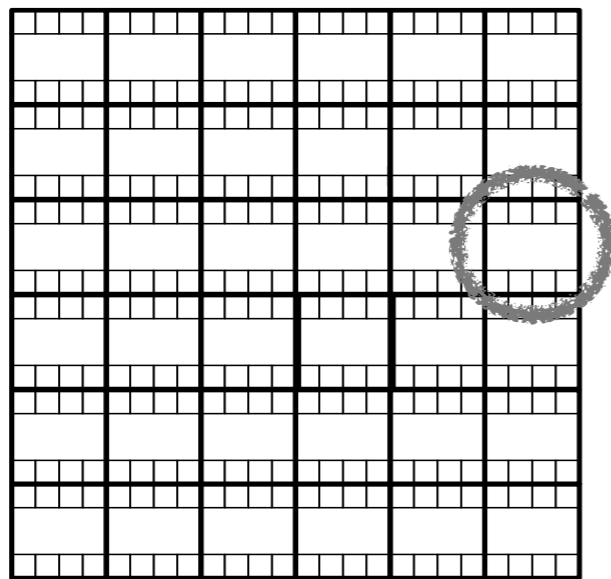
EnergySmart: Eschewing Multiple Vdd Domains

- Simple, clustered hardware to exploit within-die variation



EnergySmart: Eschewing Multiple Vdd Domains

- Simple, clustered hardware to exploit within-die variation
 - Each cluster supports a different max f at chip-wide, single Vdd

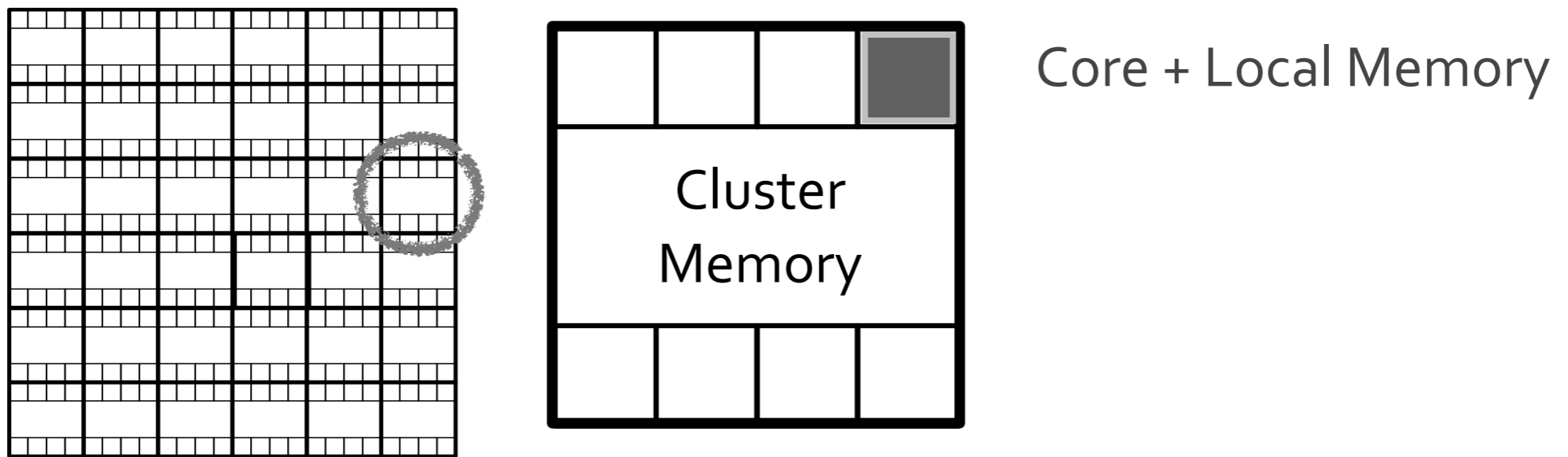


Core + Local Memory



EnergySmart: Eschewing Multiple Vdd Domains

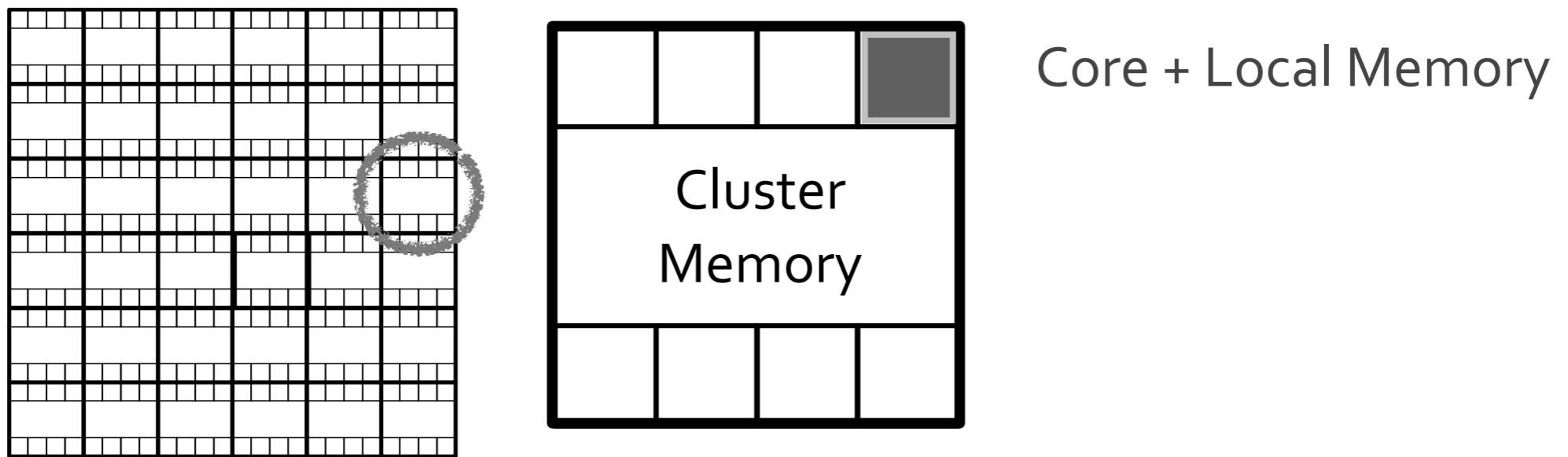
- Simple, clustered hardware to exploit within-die variation
 - Each cluster supports a different max f at chip-wide, single Vdd



- Cluster selection mimics multi-Vdd adaptation

EnergySmart: Eschewing Multiple Vdd Domains

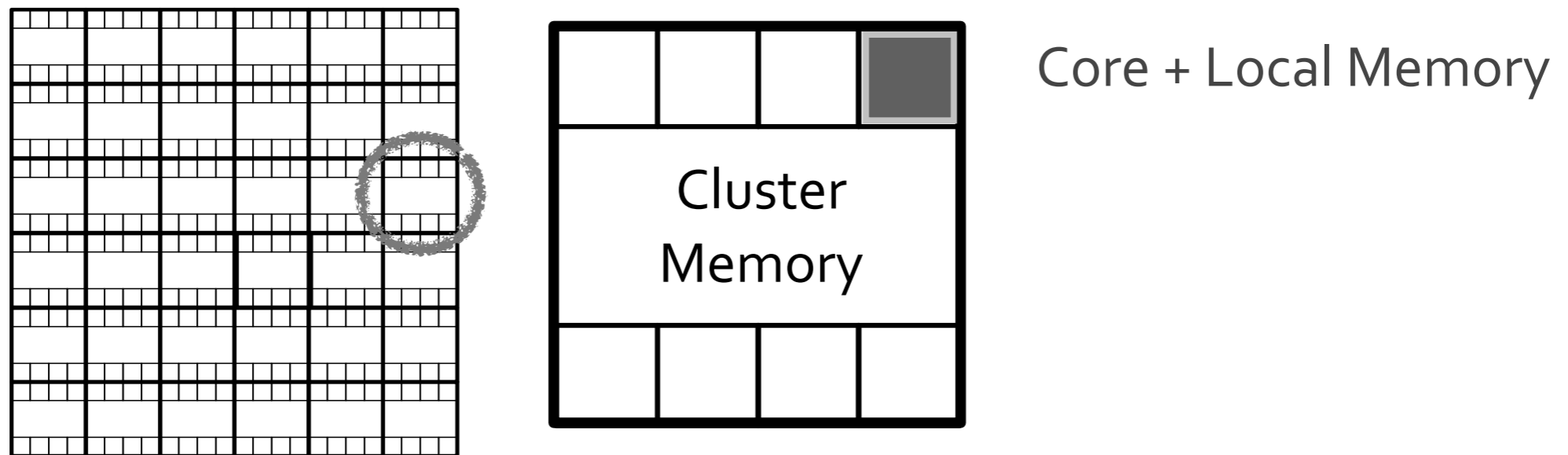
- Simple, clustered hardware to exploit within-die variation
 - Each cluster supports a different max f at chip-wide, single Vdd



- Cluster selection mimics multi-Vdd adaptation
 - Assignment in units of multiples of clusters called **Ensembles**

EnergySmart: Eschewing Multiple Vdd Domains

- Simple, clustered hardware to exploit within-die variation
 - Each cluster supports a different max f at chip-wide, single Vdd

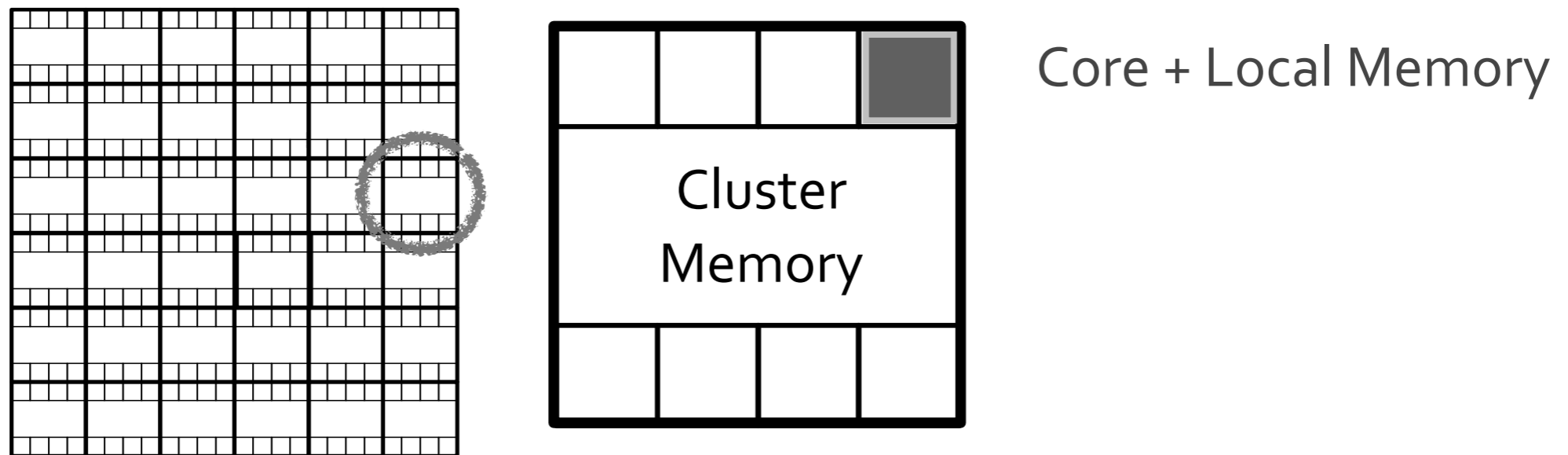


- Cluster selection mimics multi-Vdd adaptation
 - Assignment in units of multiples of clusters called **Ensembles**
 - Each ensemble constitutes a f domain



EnergySmart: Eschewing Multiple Vdd Domains

- Simple, clustered hardware to exploit within-die variation
 - Each cluster supports a different max f at chip-wide, single Vdd



- Cluster selection mimics multi-Vdd adaptation
 - Assignment in units of multiples of clusters called **Ensembles**
 - Each ensemble constitutes a f domain
 - Each ensemble cycles at the f of slowest component cluster



Cluster Assignment

Cluster
Assignment



Cluster Assignment

- Goal: Maximize MIPS/Watt subject to

Cluster
Assignment



Cluster Assignment

- Goal: **Maximize MIPS/Watt** subject to
 - Power budget and maximum temperature

Cluster
Assignment



Cluster Assignment

- Goal: **Maximize MIPS/Watt** subject to
 - Power budget and maximum temperature
 - Distance between allocated clusters

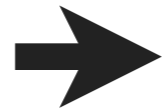
Cluster
Assignment



Cluster Assignment

- Goal: **Maximize MIPS/Watt** subject to
 - Power budget and maximum temperature
 - Distance between allocated clusters

Per Cluster Variation Profile
(P_{STA} , f_{MAX})

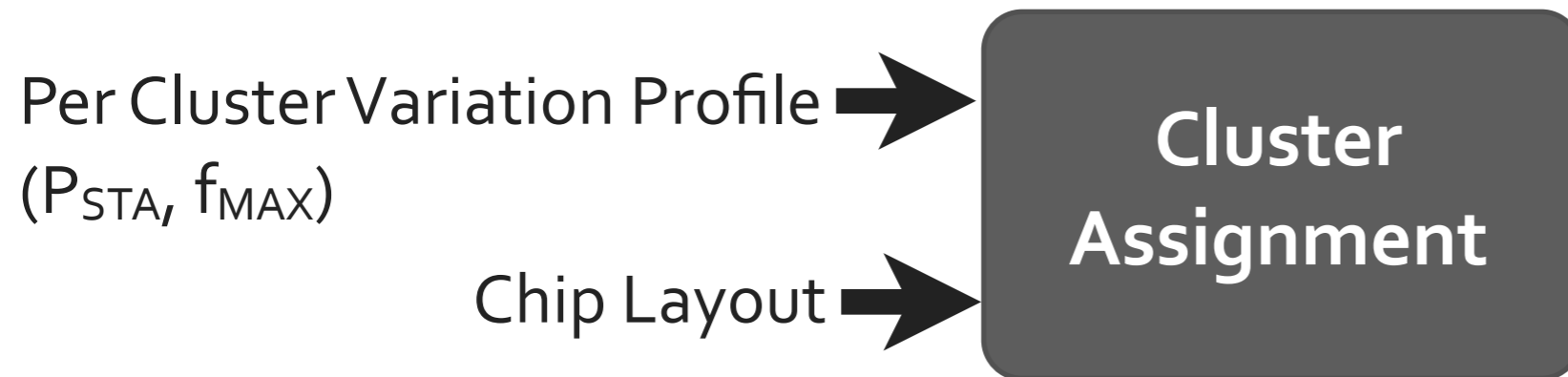


**Cluster
Assignment**



Cluster Assignment

- Goal: **Maximize MIPS/Watt** subject to
 - Power budget and maximum temperature
 - Distance between allocated clusters



Cluster Assignment

- Goal: **Maximize MIPS/Watt** subject to
 - Power budget and maximum temperature
 - Distance between allocated clusters



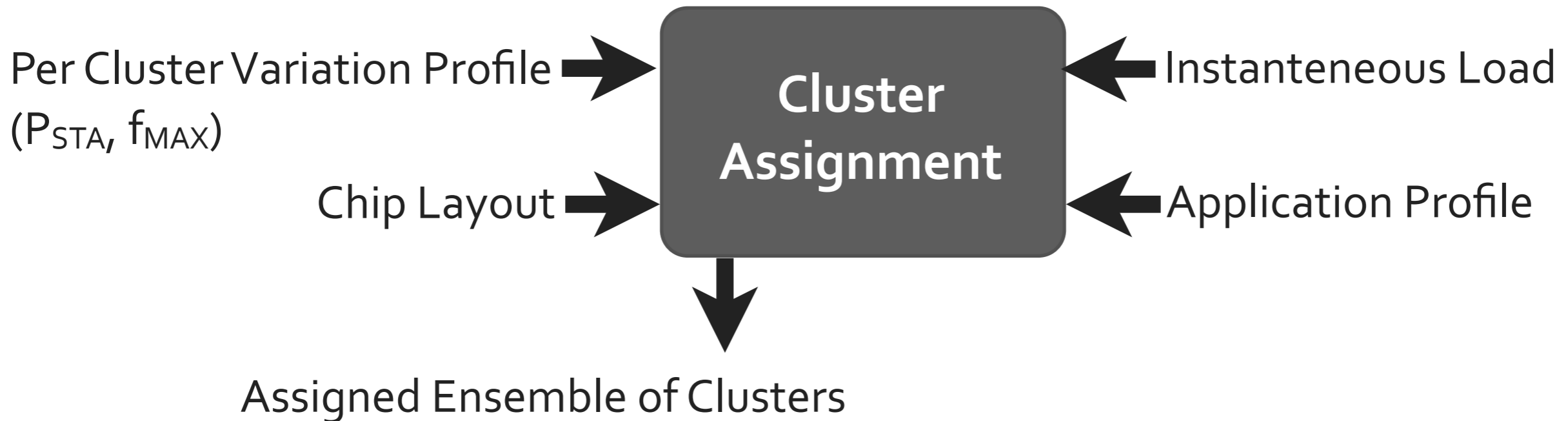
Cluster Assignment

- Goal: **Maximize MIPS/Watt** subject to
 - Power budget and maximum temperature
 - Distance between allocated clusters



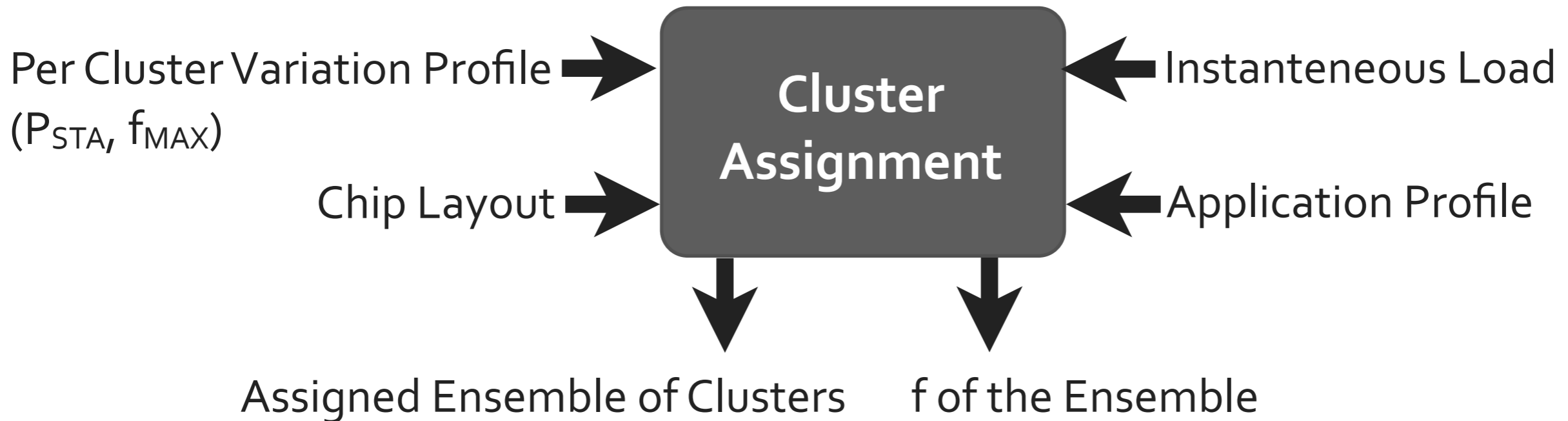
Cluster Assignment

- Goal: **Maximize MIPS/Watt** subject to
 - Power budget and maximum temperature
 - Distance between allocated clusters



Cluster Assignment

- Goal: **Maximize MIPS/Watt** subject to
 - Power budget and maximum temperature
 - Distance between allocated clusters



Cluster Assignment

- Goal: **Maximize MIPS/Watt** subject to
 - Power budget and maximum temperature
 - Distance between allocated clusters



Simple

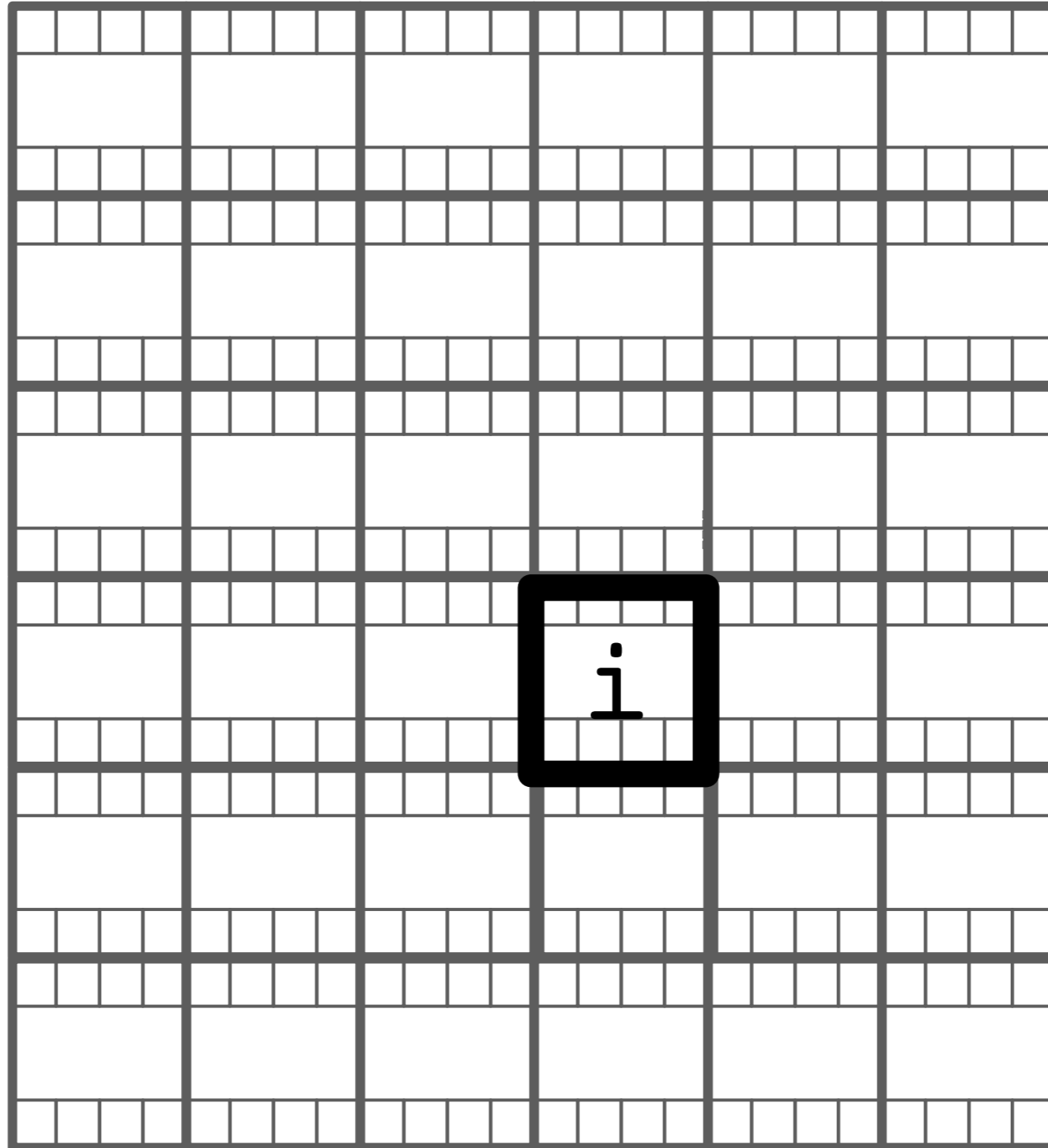
Single Vdd value

Assignment at cluster granularity

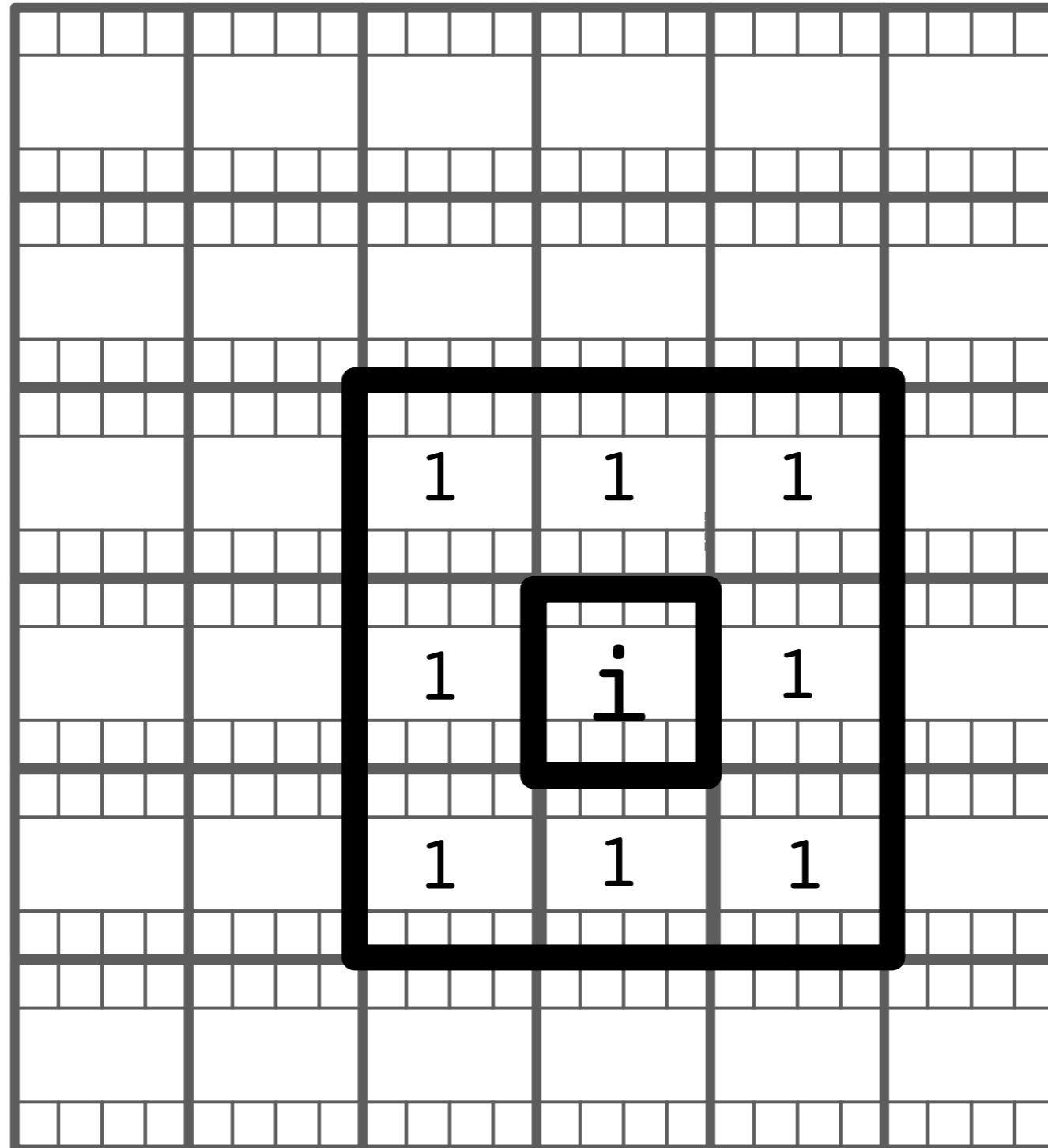
Single f per ensemble



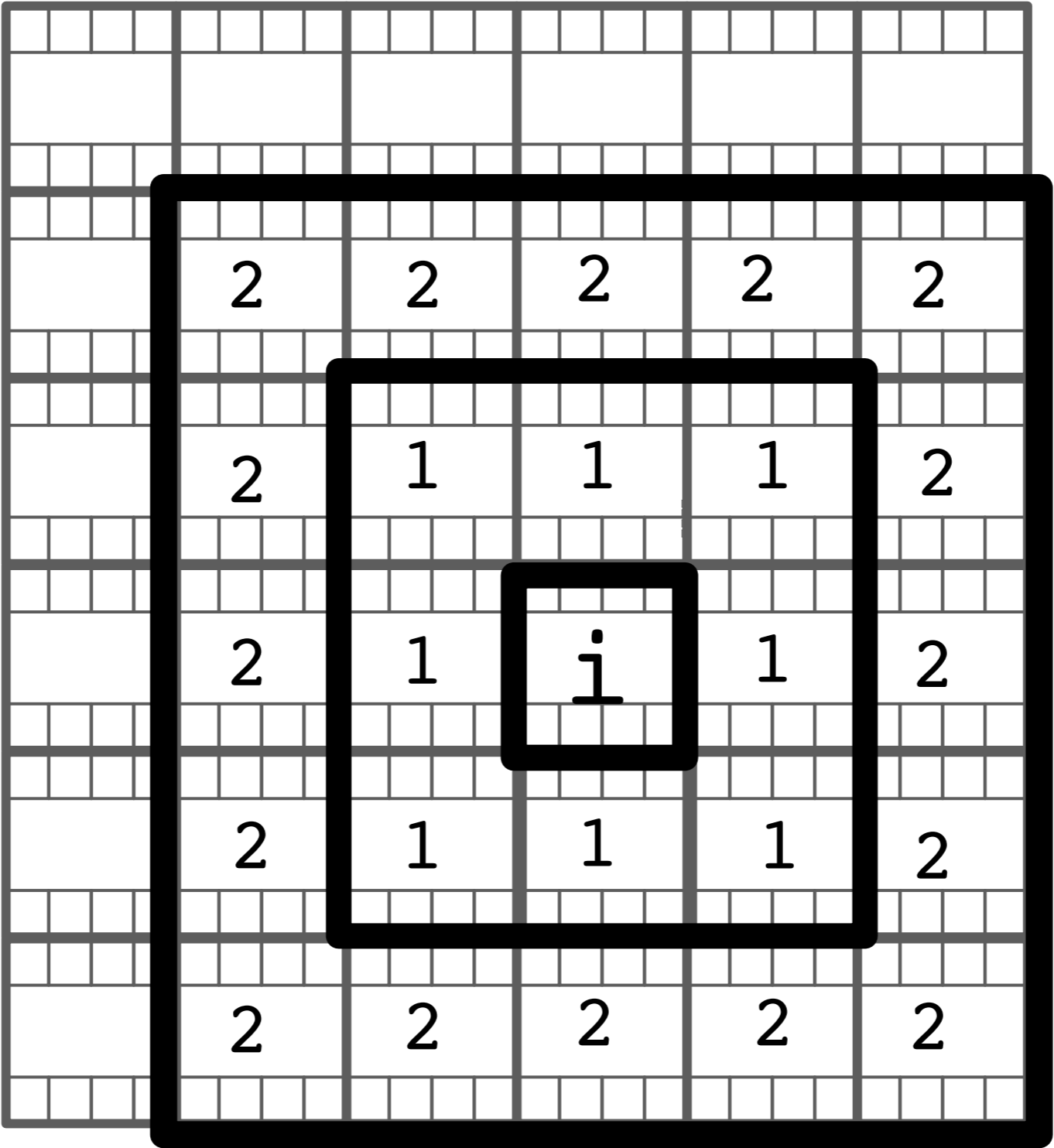
Distance as Wavefronts



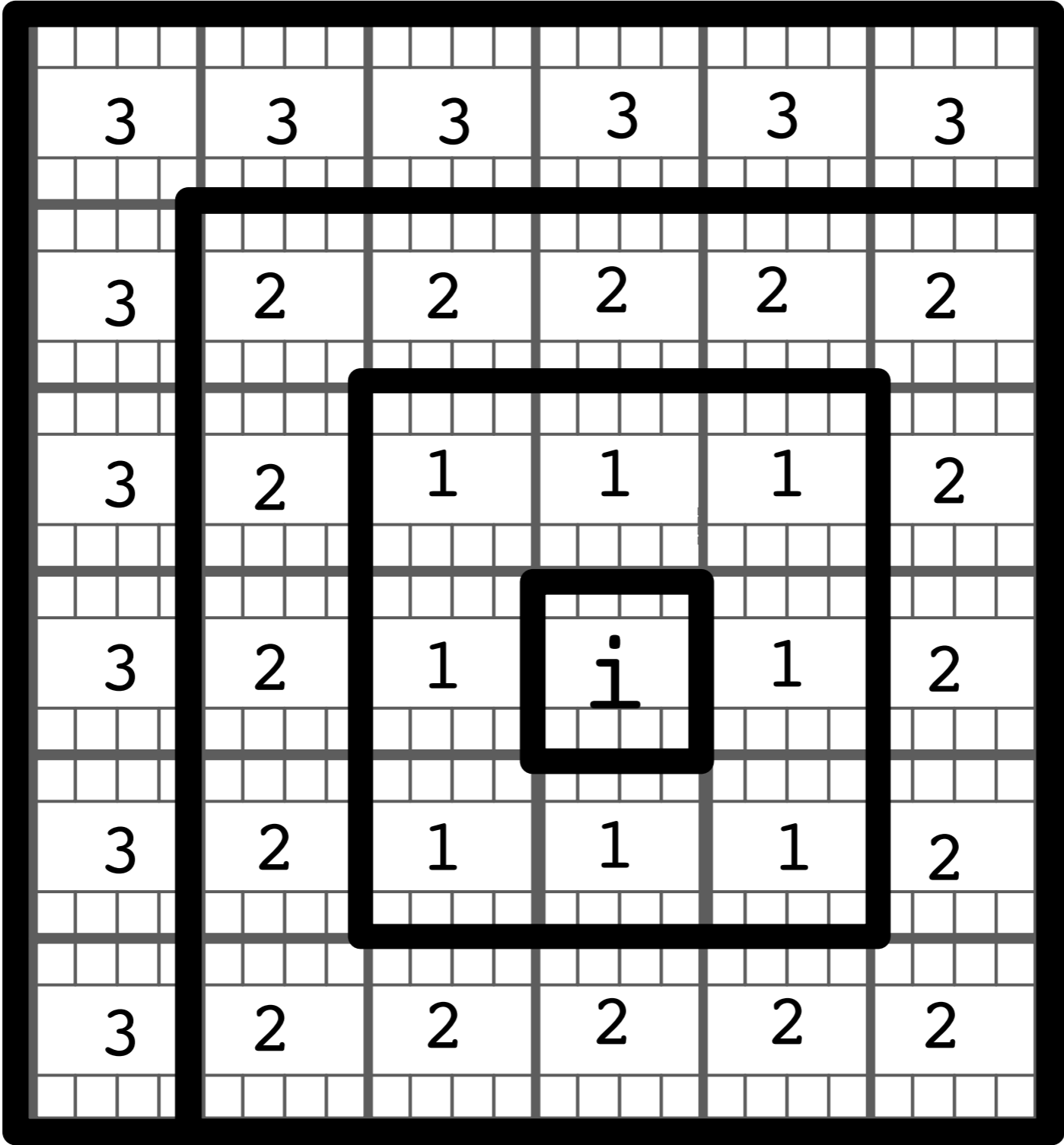
Distance as Wavefronts



Distance as Wavefronts



Distance as Wavefronts



Example Operation

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36



Example Operation

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

$$f_A = \min(f_8, f_9, f_{14}, f_{15})$$



Example Operation

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

$$f_A = \min(f_8, f_9, f_{14}, f_{15})$$

$$f_B = \min(f_5, f_6, f_{11}, f_{12}, f_{17}, f_{18}, f_{23}, f_{24})$$



Example Operation

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

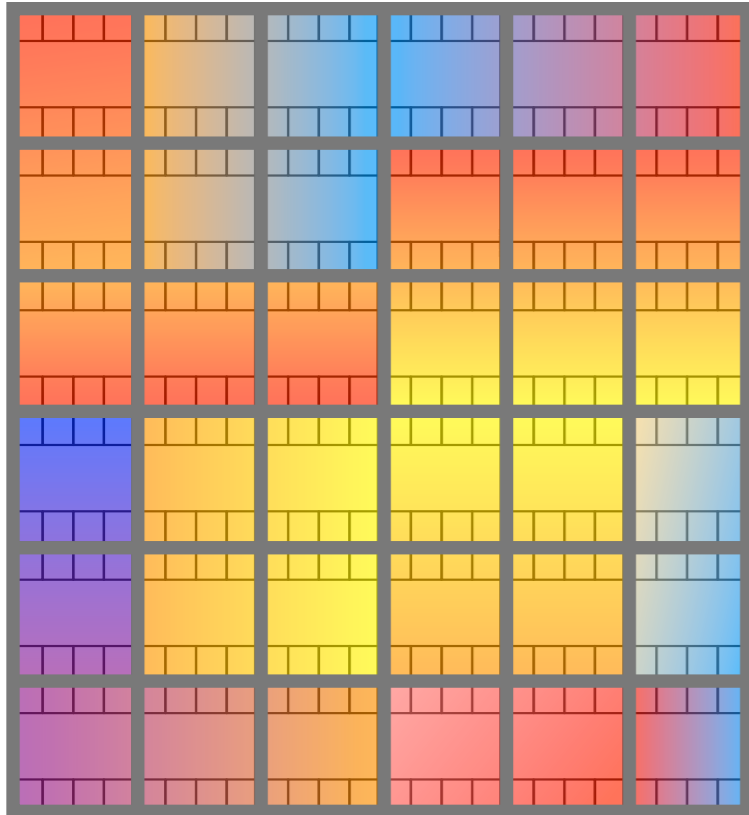
$$f_A = \min(f_8, f_9, f_{14}, f_{15})$$

$$f_B = \min(f_5, f_6, f_{11}, f_{12}, f_{17}, f_{18}, f_{23}, f_{24})$$

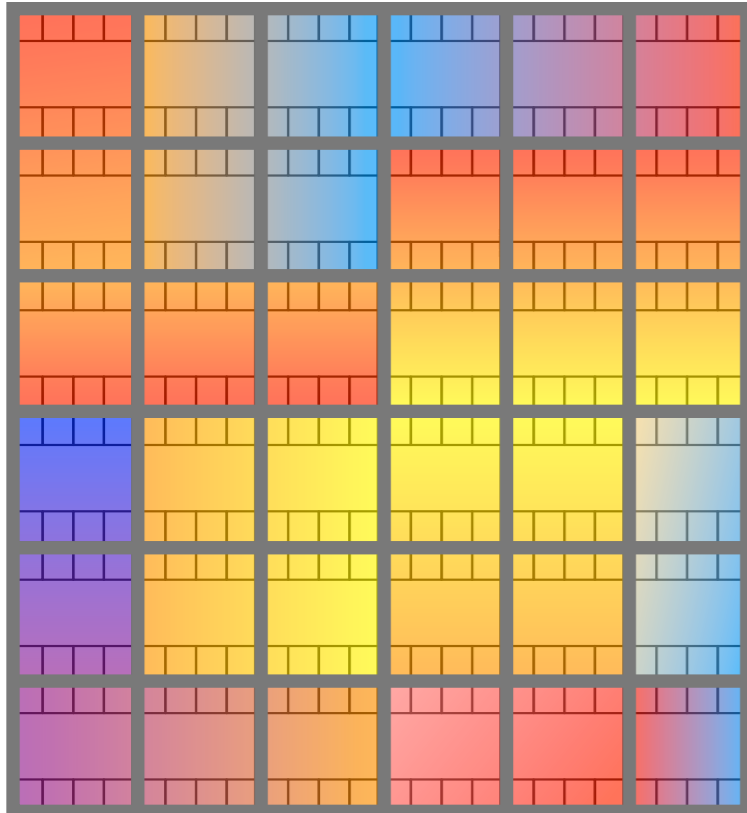
$$f_C = \min(f_{20}, f_{26}, f_{27}, f_{28}, f_{29}, f_{30}, f_{32}, f_{33})$$



Cluster Assignment: Efficient Exhaustive Search

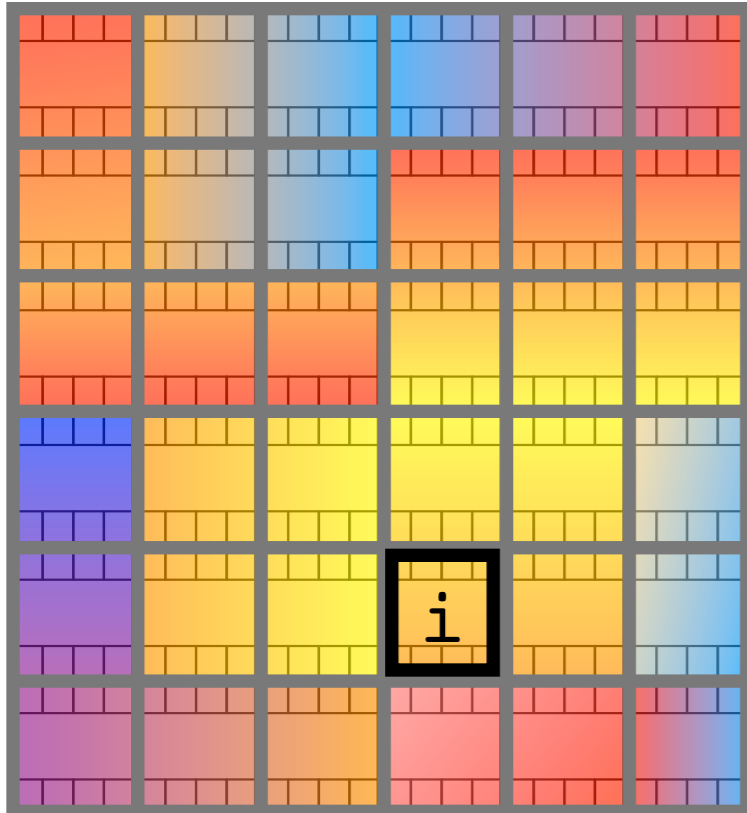


Cluster Assignment: Efficient Exhaustive Search



A task demands $N = |E|$ clusters

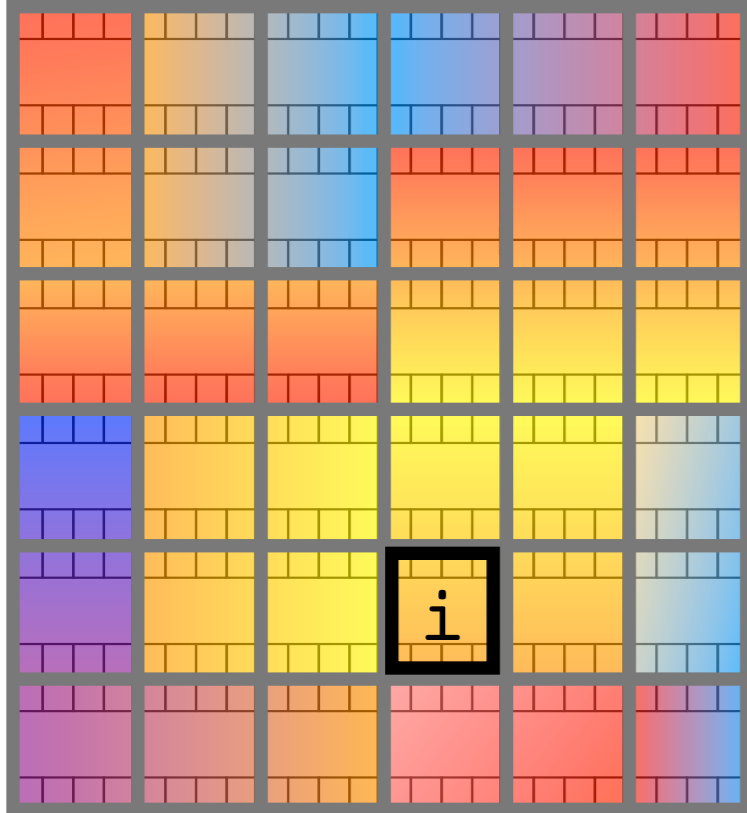
Cluster Assignment: Efficient Exhaustive Search



A task demands $N = |E|$ clusters

- For each available cluster i

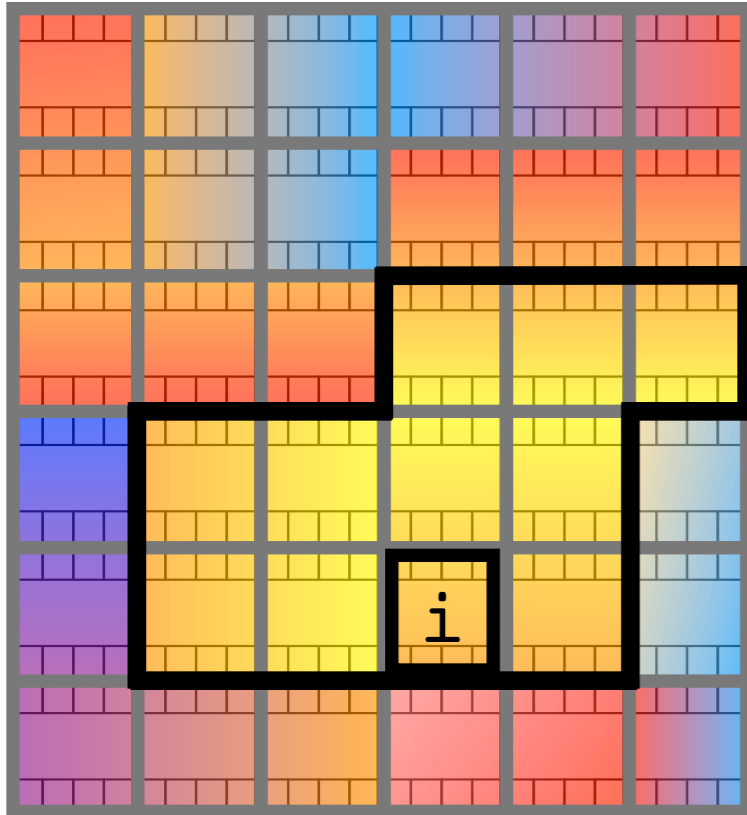
Cluster Assignment: Efficient Exhaustive Search



A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$

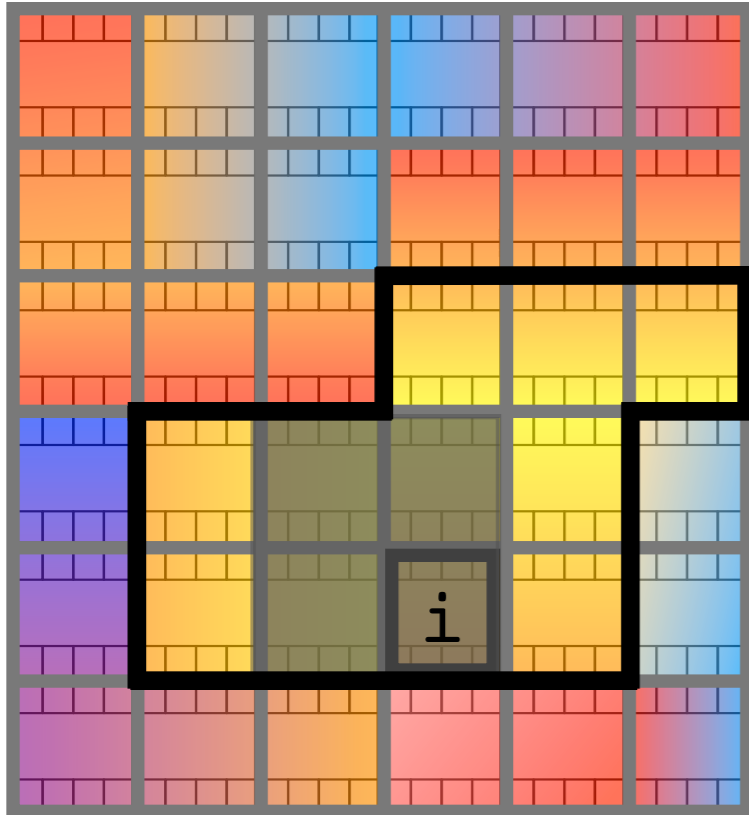
Cluster Assignment: Efficient Exhaustive Search



A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}

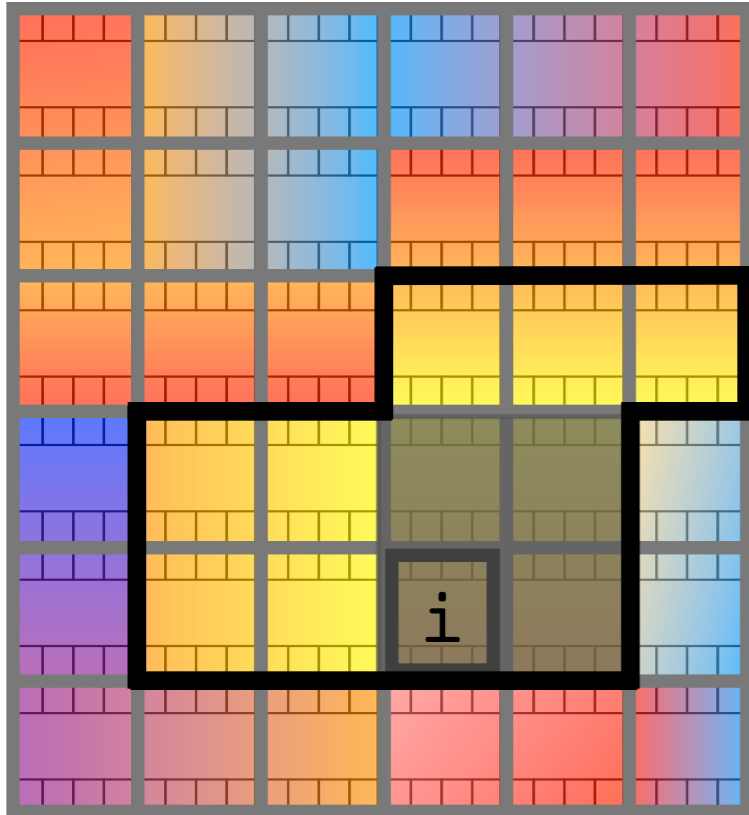
Cluster Assignment: Efficient Exhaustive Search



A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{MIPS/Watt}$**

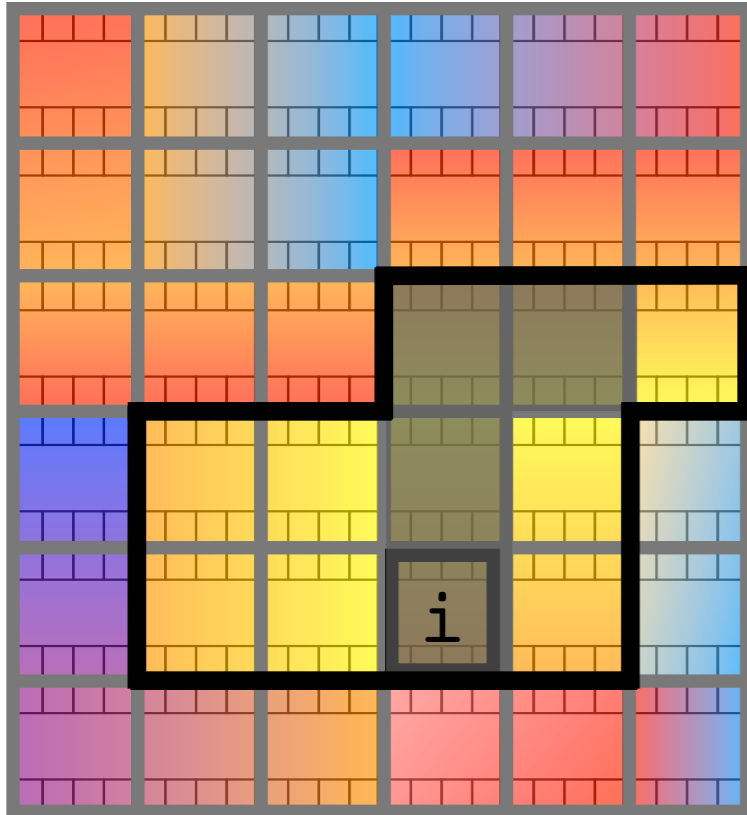
Cluster Assignment: Efficient Exhaustive Search



A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{ MIPS/Watt}$**

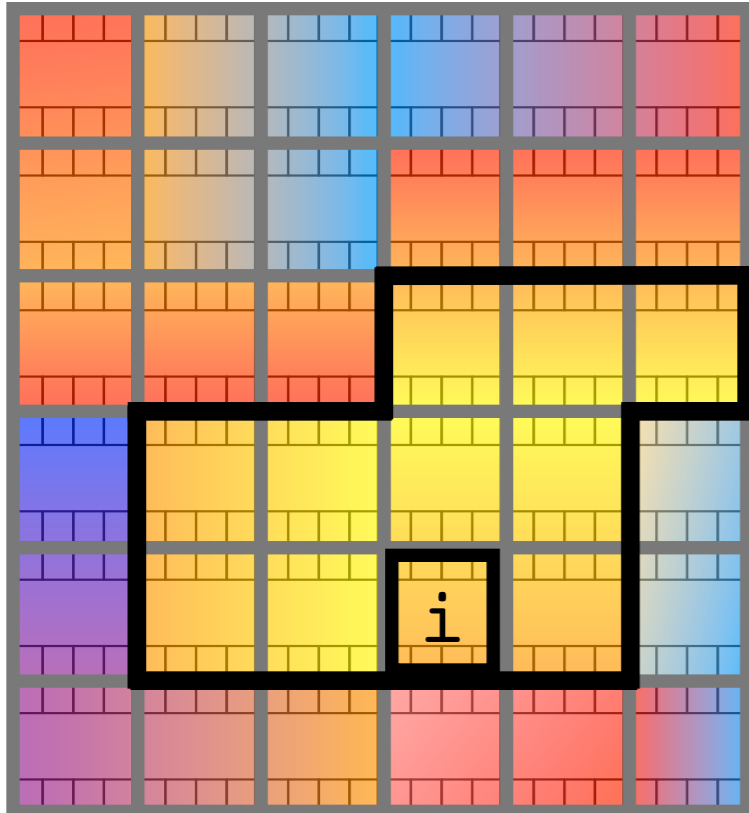
Cluster Assignment: Efficient Exhaustive Search



A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{ MIPS/Watt}$**

Cluster Assignment: Efficient Exhaustive Search



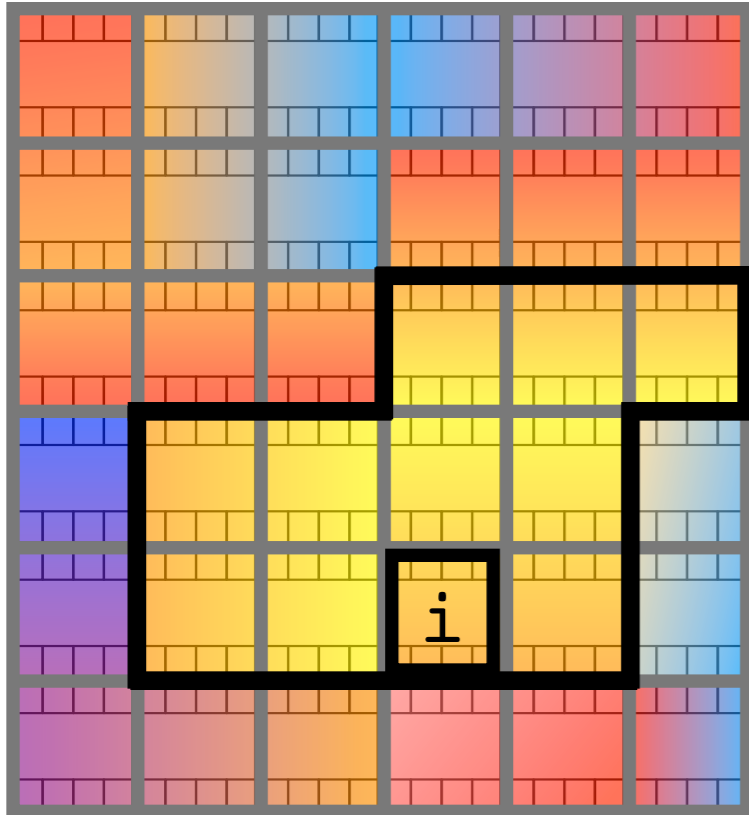
A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{ MIPS/Watt}$**

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

Cluster Assignment: Efficient Exhaustive Search



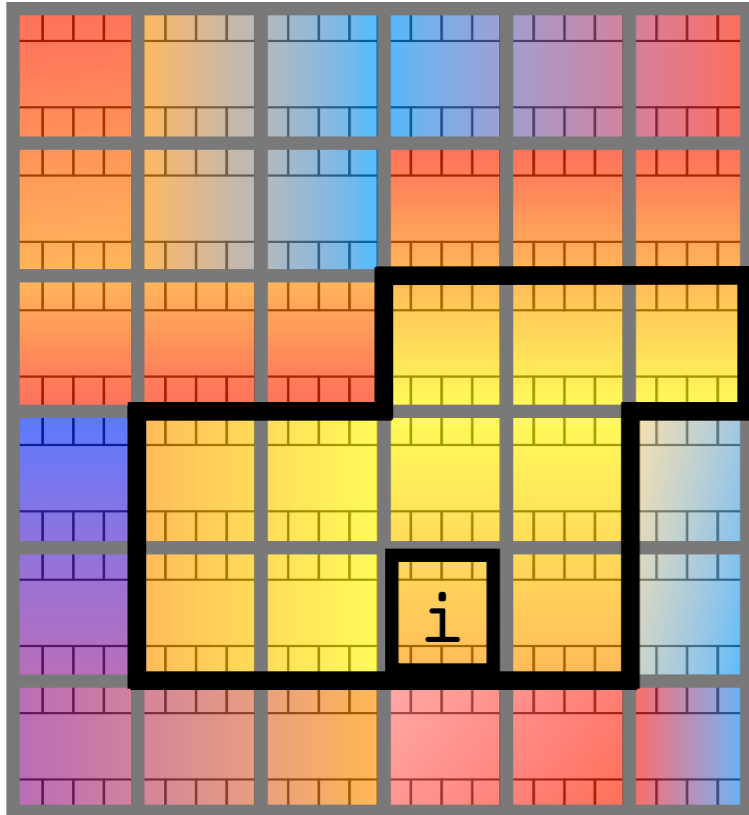
A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{MIPS/Watt}$**

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

$$\max_E \frac{\text{IPC}(f_E) \times |E| \times f_E}{\text{Watt}}$$

Cluster Assignment: Efficient Exhaustive Search



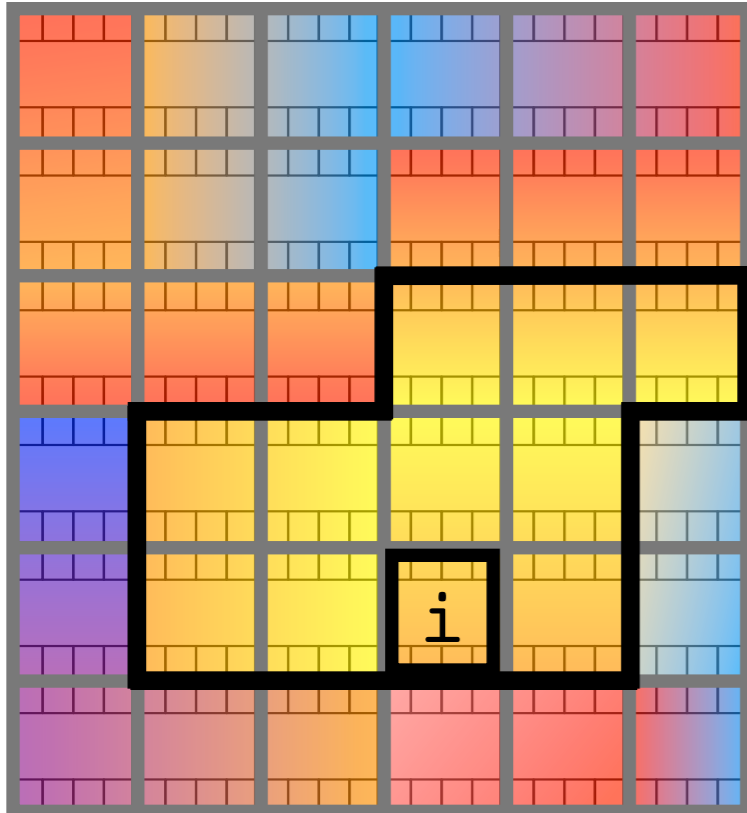
A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{MIPS/Watt}$**

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

$$\max_E \frac{\text{IPC}(f_E) \times |E| \times f_E}{\sum_E P_{STA} + \sum_E P_{DYN}}$$

Cluster Assignment: Efficient Exhaustive Search



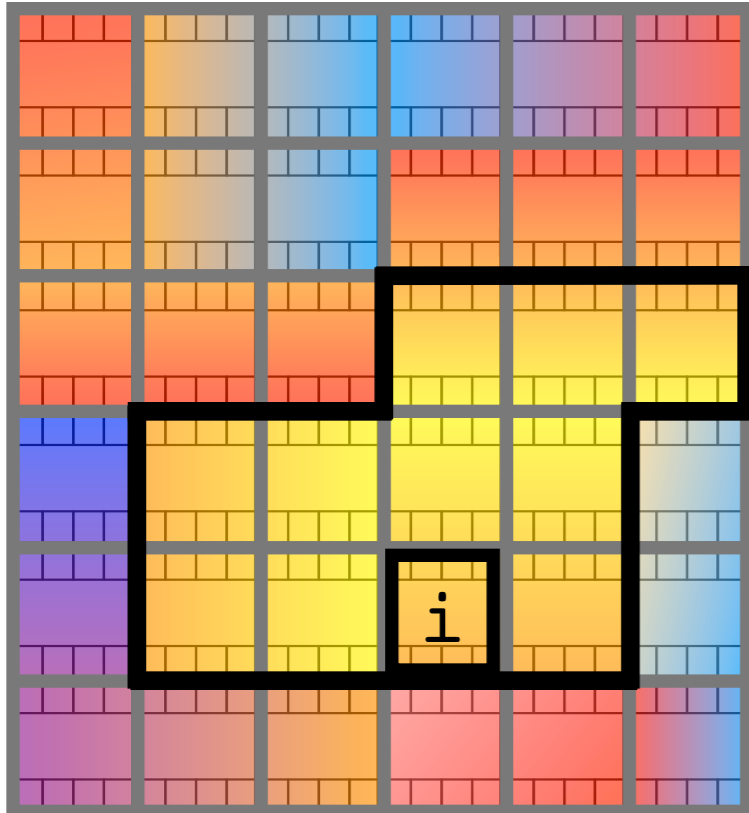
A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{MIPS/Watt}$**

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

$$\max_E \frac{\text{IPC}(f_E) \times |E| \times f_E}{\sum_E P_{STA} + C \times V_{dd}^2 \times |E| \times f_E}$$

Cluster Assignment: Efficient Exhaustive Search



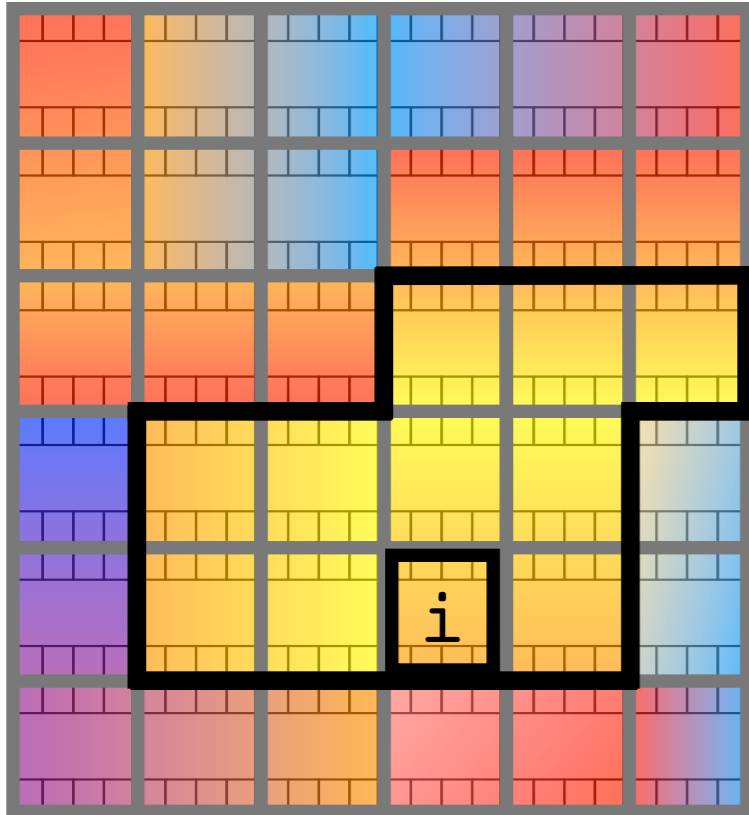
A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{MIPS/Watt}$**

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

$$\max_E \frac{\text{IPC}(f_E) \times |E| \times f_E}{\sum_E P_{STA} + C \times V_{dd}^2 \times |E| \times f_E}$$

Cluster Assignment: Efficient Exhaustive Search



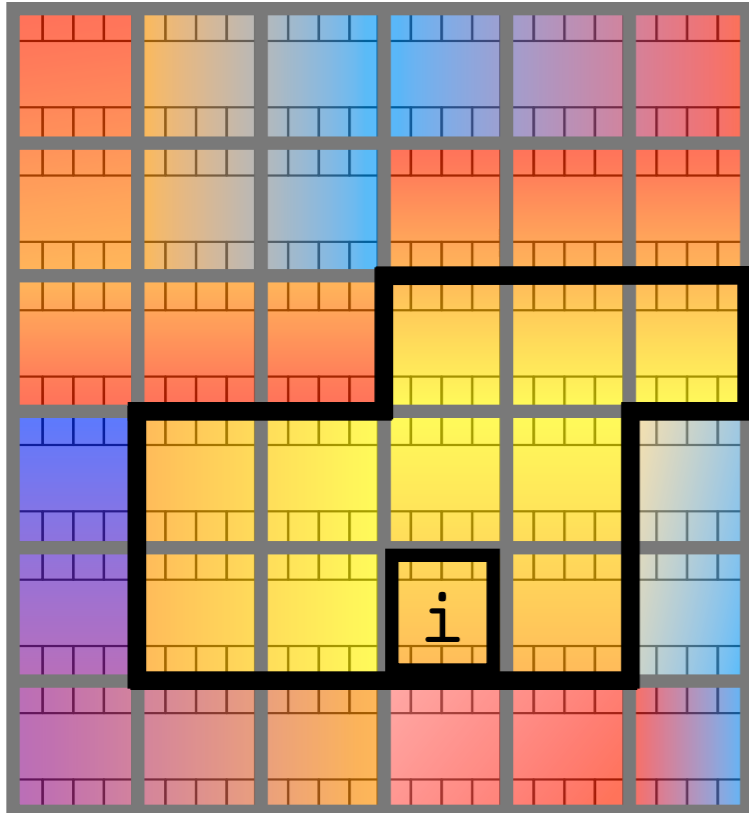
A task demands $N = |E|$ clusters

- For each available cluster i
 - Enumerate all ensembles E where the cluster i is slowest
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble $\max_E \text{MIPS/Watt}$

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

$$\max_E \frac{\text{IPC}(f_E) \times |E| \times f_E}{\sum_E P_{STA} + C \times V_{dd}^2 \times |E| \times f_E}$$

Cluster Assignment: Efficient Exhaustive Search



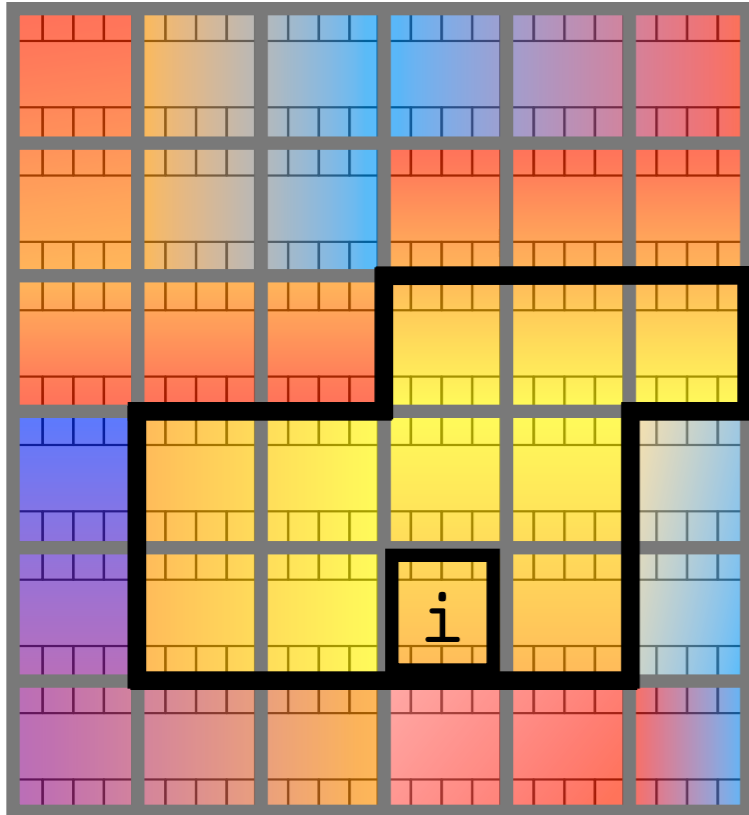
A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{ MIPS/Watt}$**

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

$$\max_E \frac{\text{IPC}(f_E) \times |E| \times f_E}{\sum_E P_{STA} + C \times V_{dd}^2 \times |E| \times f_E}$$

Cluster Assignment: Efficient Exhaustive Search



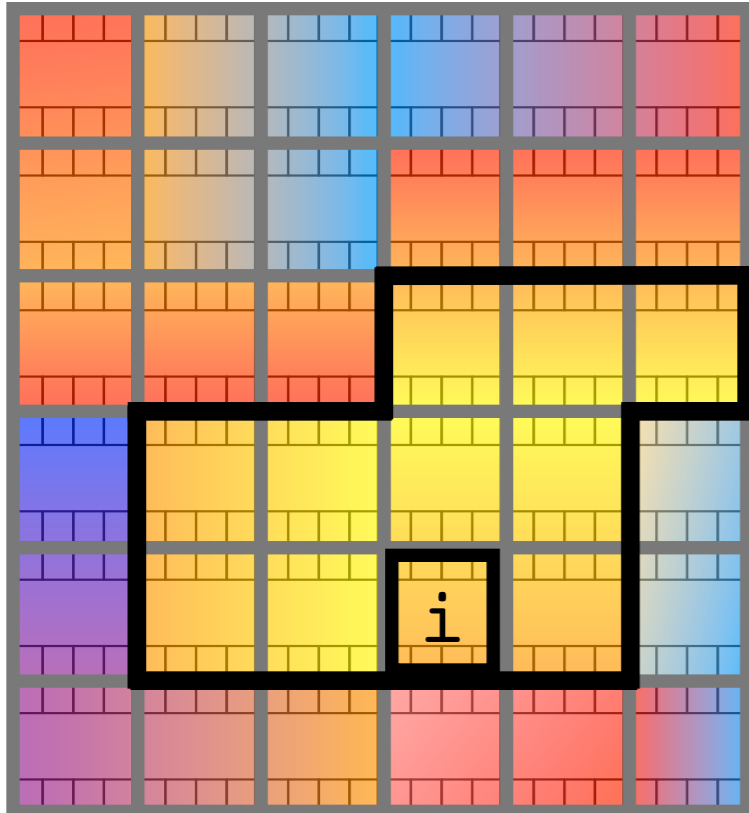
A task demands $N = |E|$ clusters

- For each available cluster i
 - Enumerate all ensembles E where the cluster i is slowest
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble $\max_E \text{MIPS/Watt}$

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

$$\max_E \frac{\text{IPC}(f_E) \times |E| \times f_E}{\sum_E P_{STA} + C \times V_{dd}^2 \times |E| \times f_E}$$

Cluster Assignment: Efficient Exhaustive Search



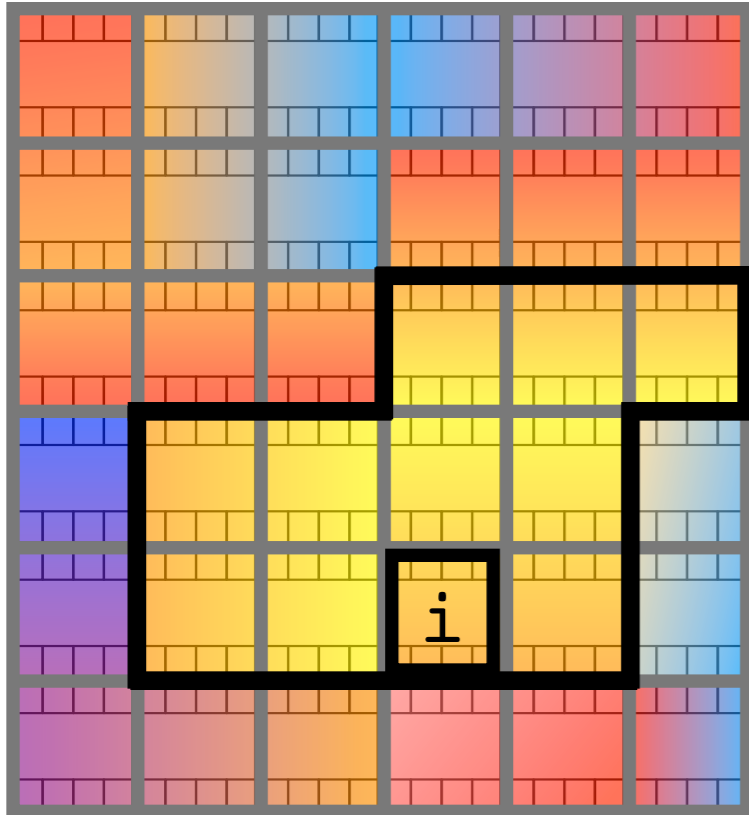
A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{MIPS/Watt}$**

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

$$\max_E \frac{\text{IPC}(f_E) \times |E| \times f_E}{\sum_E P_{STA} + C \times V_{dd}^2 \times |E| \times f_E}$$

Cluster Assignment: Efficient Exhaustive Search



A task demands $N = |E|$ clusters

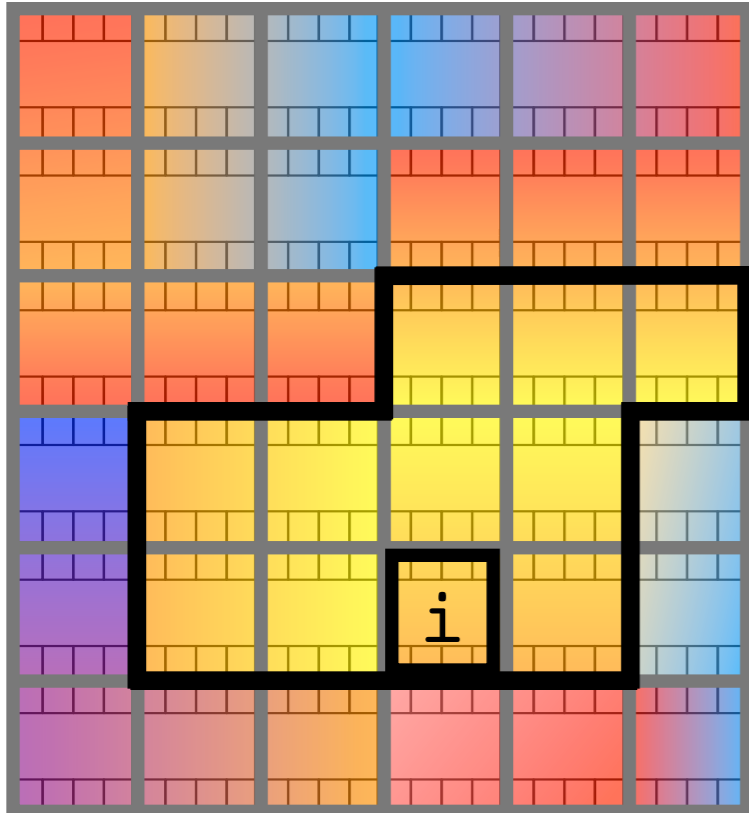
- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{MIPS/Watt}$**

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

$$\max_E \frac{\text{IPC}(f_E) \times |E| \times f_E}{\sum_E P_{STA} + C \times V_{dd}^2 \times |E| \times f_E}$$

$$\equiv \min_E \sum_E P_{STA}$$

Cluster Assignment: Efficient Exhaustive Search



A task demands $N = |E|$ clusters

- For each available cluster i
 - **Enumerate all ensembles E where the cluster i is slowest**
 - Ensemble frequency $f_E = f_i$
 - Consider only clusters of higher f_{MAX}
 - Pick the ensemble **$\max_E \text{MIPS/Watt}$**
- ▶ Linear-complexity operation if clusters are sorted offline

$$\max_E \frac{\text{MIPS}}{\text{Watt}}$$

$$\max_E \frac{\text{IPC}(f_E) \times |E| \times f_E}{\sum_E P_{STA} + C \times V_{dd}^2 \times |E| \times f_E}$$

$$\equiv \min_E \sum_E P_{STA}$$

EnergySmart: Pros and Cons



EnergySmart: Pros and Cons

- ✓ Simple clustered hardware architecture



EnergySmart: Pros and Cons

- ✓ Simple clustered hardware architecture
- ✓ Tolerates variation with a single physical Vdd domain



EnergySmart: Pros and Cons

- ✓ Simple clustered hardware architecture
- ✓ Tolerates variation with a single physical Vdd domain
 - Cluster selection replaces multi-Vdd adaptation



EnergySmart: Pros and Cons

- ✓ Simple clustered hardware architecture
- ✓ Tolerates variation with a single physical Vdd domain
 - Cluster selection replaces multi-Vdd adaptation
- ✓ Simple cluster assignment



EnergySmart: Pros and Cons

- ✓ Simple clustered hardware architecture
- ✓ Tolerates variation with a single physical Vdd domain
 - Cluster selection replaces multi-Vdd adaptation
- ✓ Simple cluster assignment
- Needs cluster profiling



Evaluation Setup



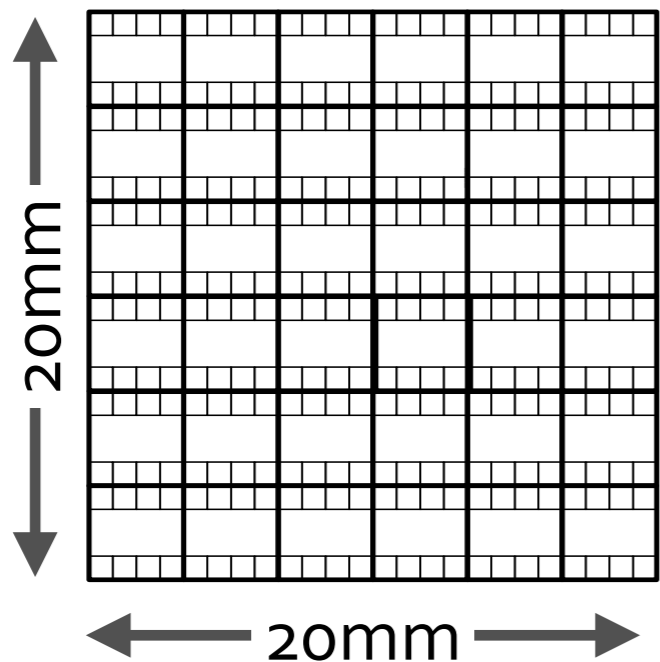
Evaluation Setup

- Simulated 288 core chip at 11nm:



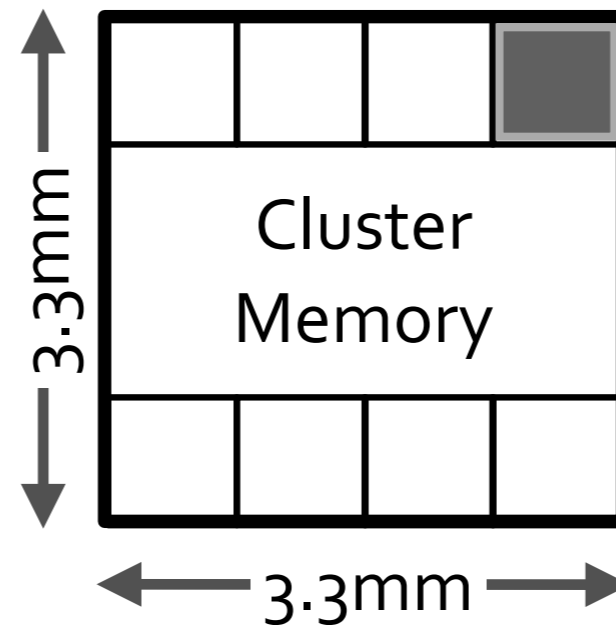
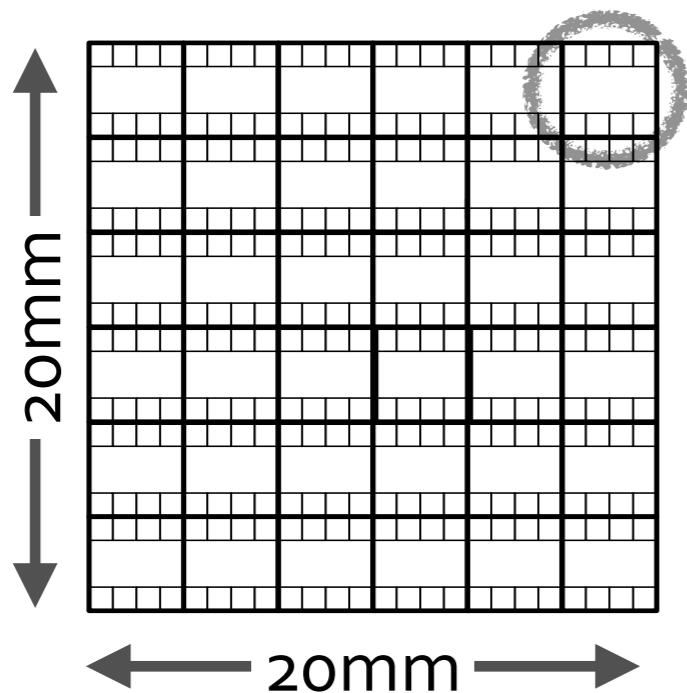
Evaluation Setup

- Simulated 288 core chip at 11nm:
 - 36 clusters, 8 cores per cluster



Evaluation Setup

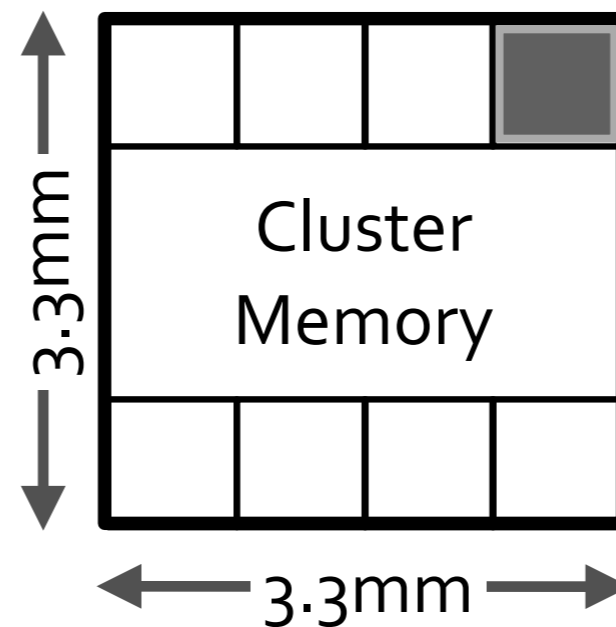
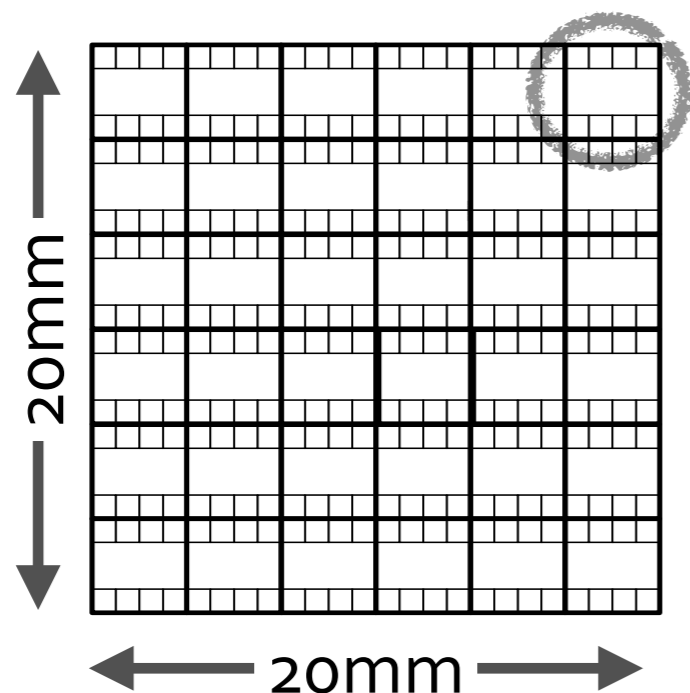
- Simulated 288 core chip at 11nm:
 - 36 clusters, 8 cores per cluster
 - Core: Single issue in-order



Core + Local Memory

Evaluation Setup

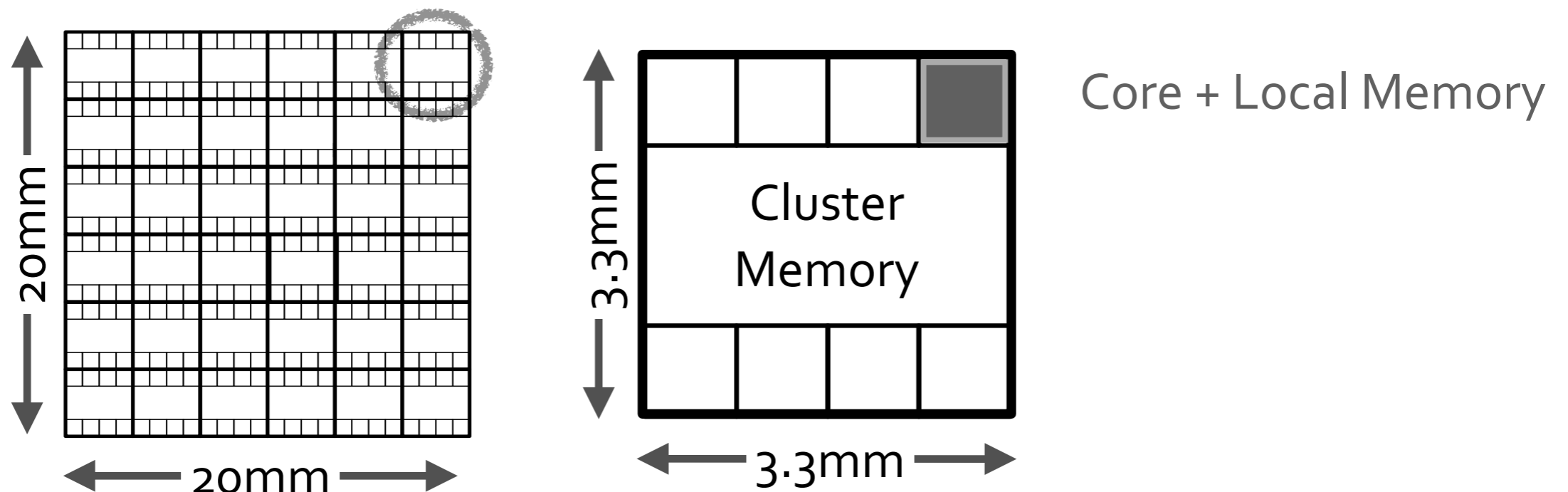
- Simulated 288 core chip at 11nm:
 - 36 clusters, 8 cores per cluster
 - Core: Single issue in-order
- VARIUS-NTV model to find the per-cluster min V_{dd} and max f



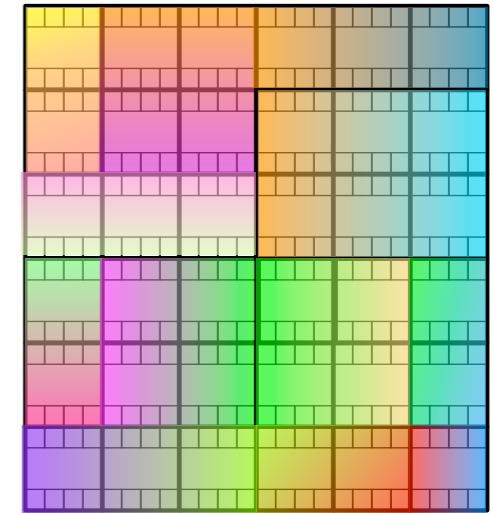
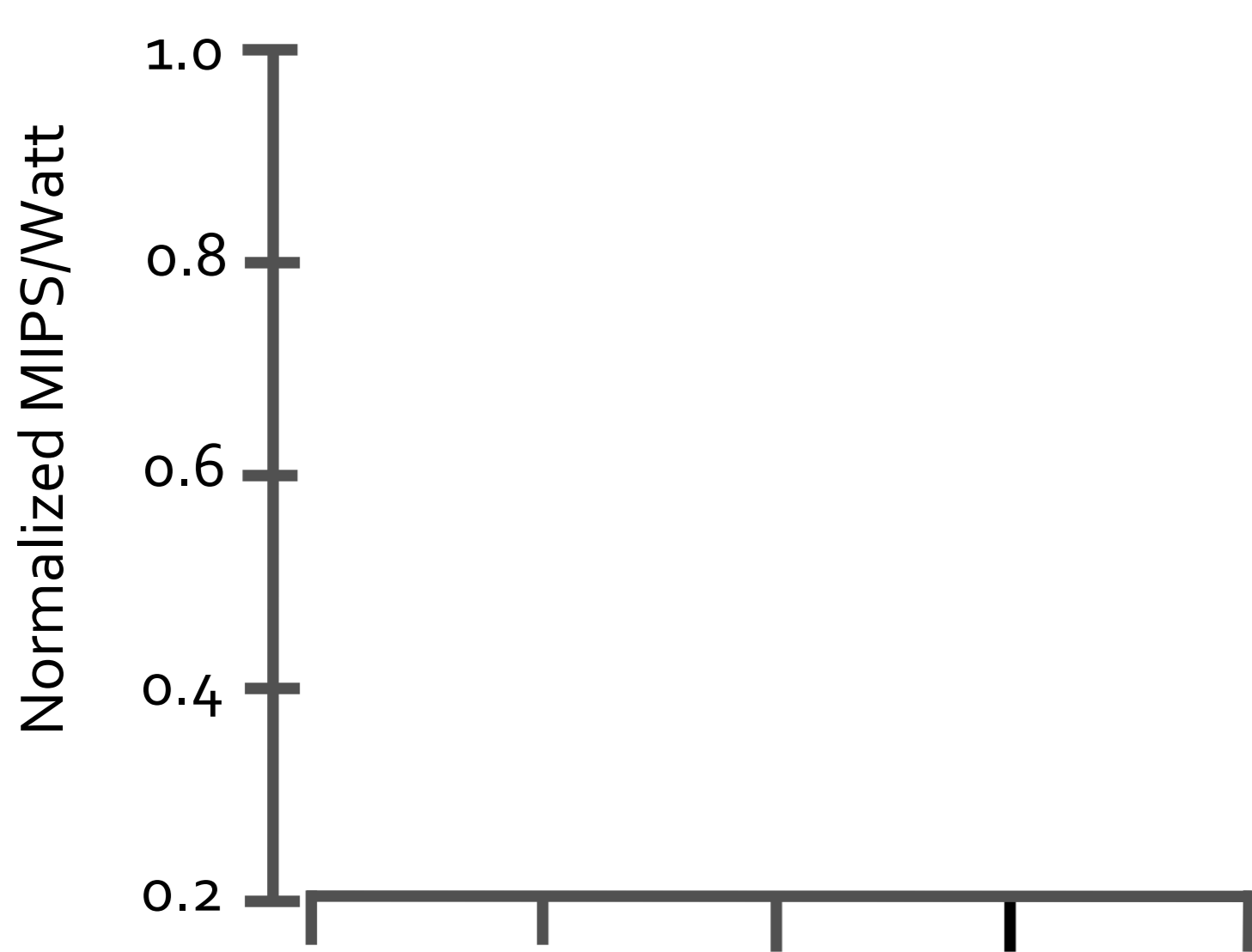
Core + Local Memory

Evaluation Setup

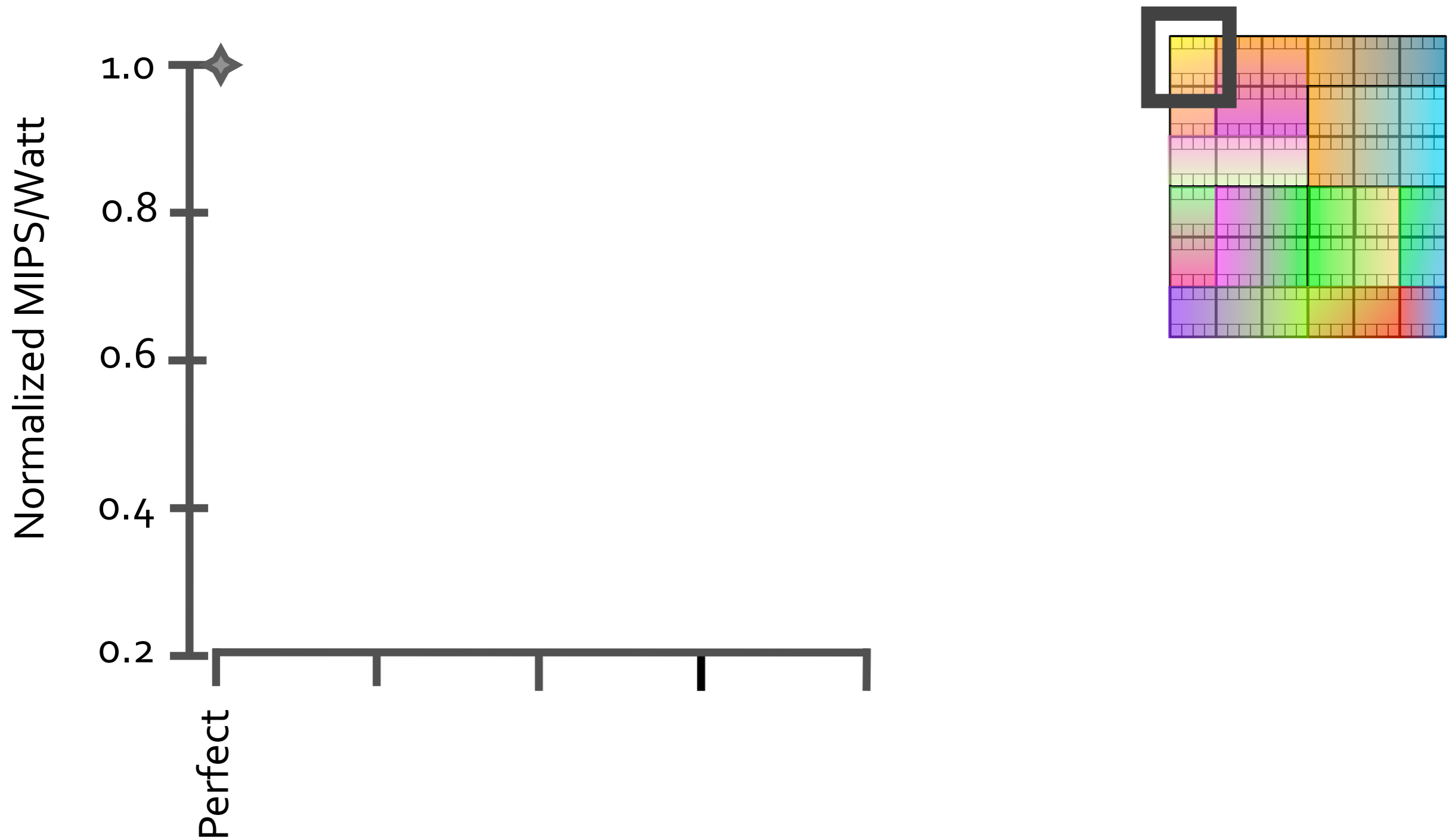
- Simulated 288 core chip at 11nm:
 - 36 clusters, 8 cores per cluster
 - Core: Single issue in-order
- VARIUS-NTV model to find the per-cluster min Vdd and max f
- Multi-programmed workload based on PARSEC



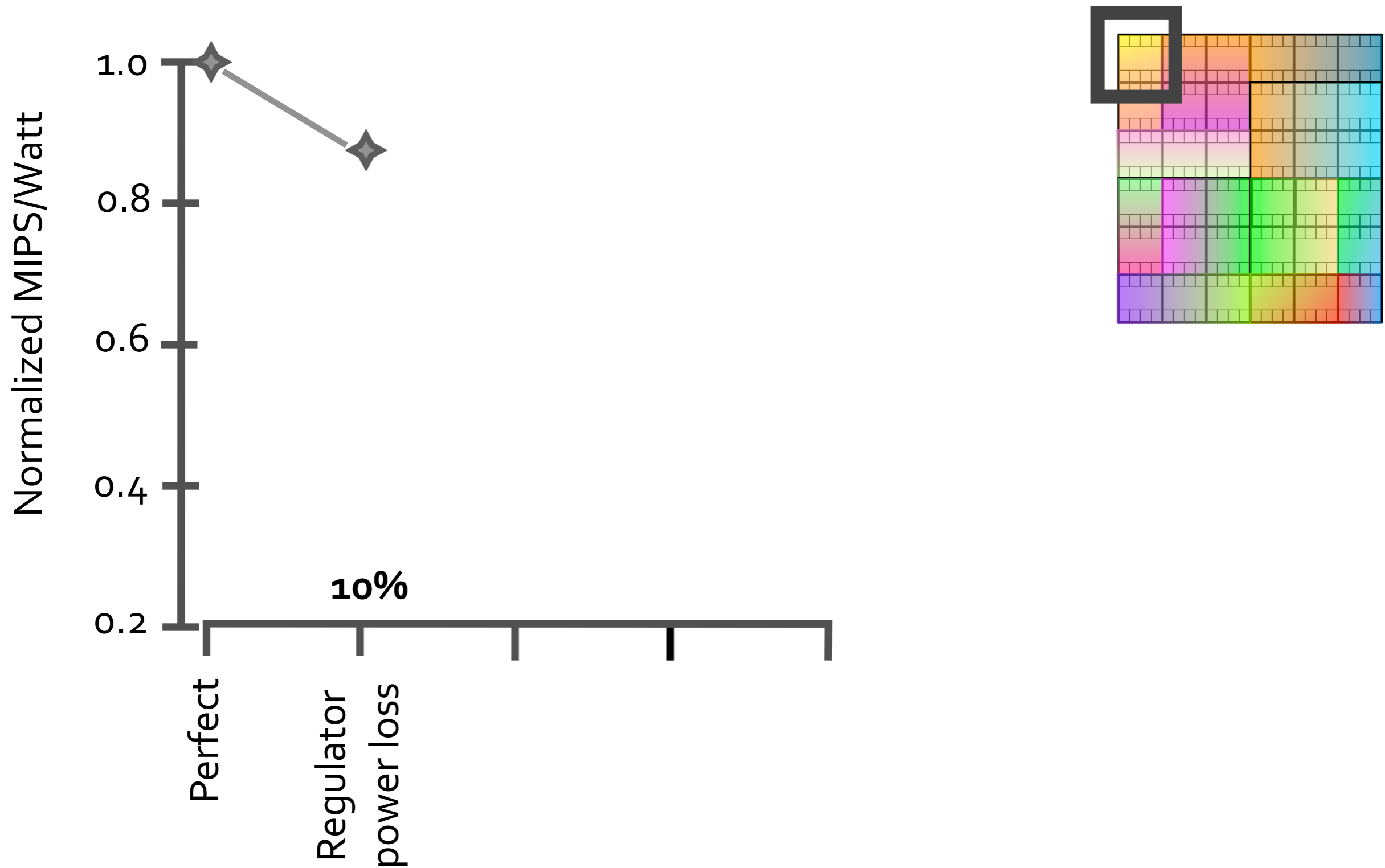
Effectiveness of Single Vdd Domain per Chip



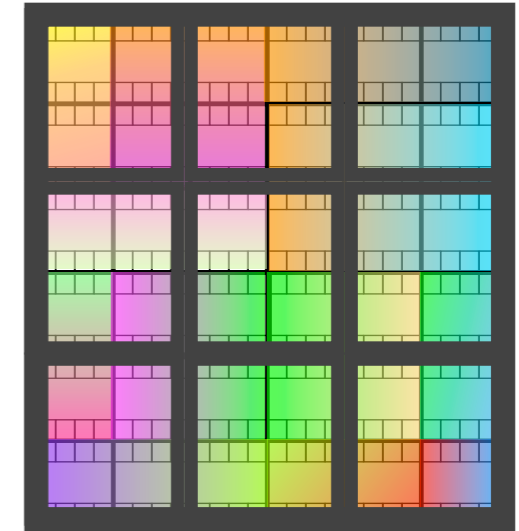
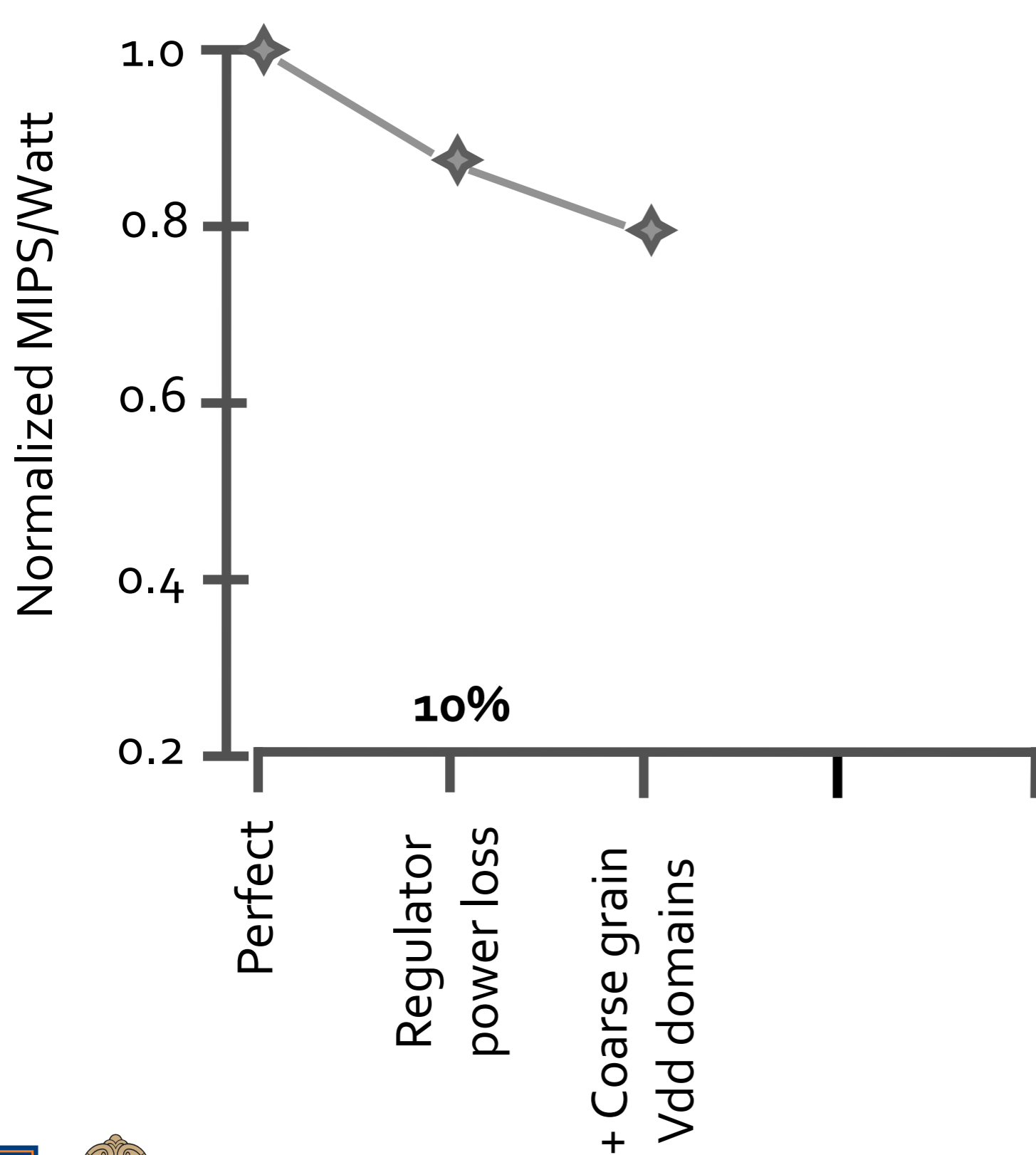
Effectiveness of Single Vdd Domain per Chip



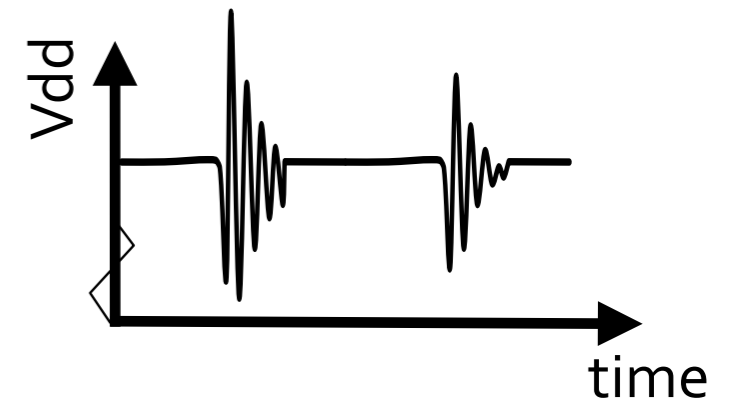
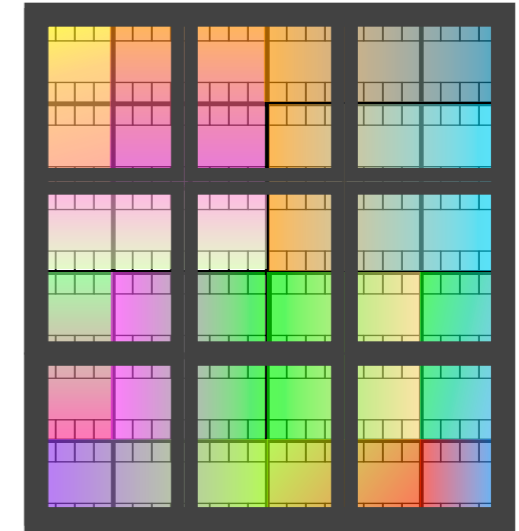
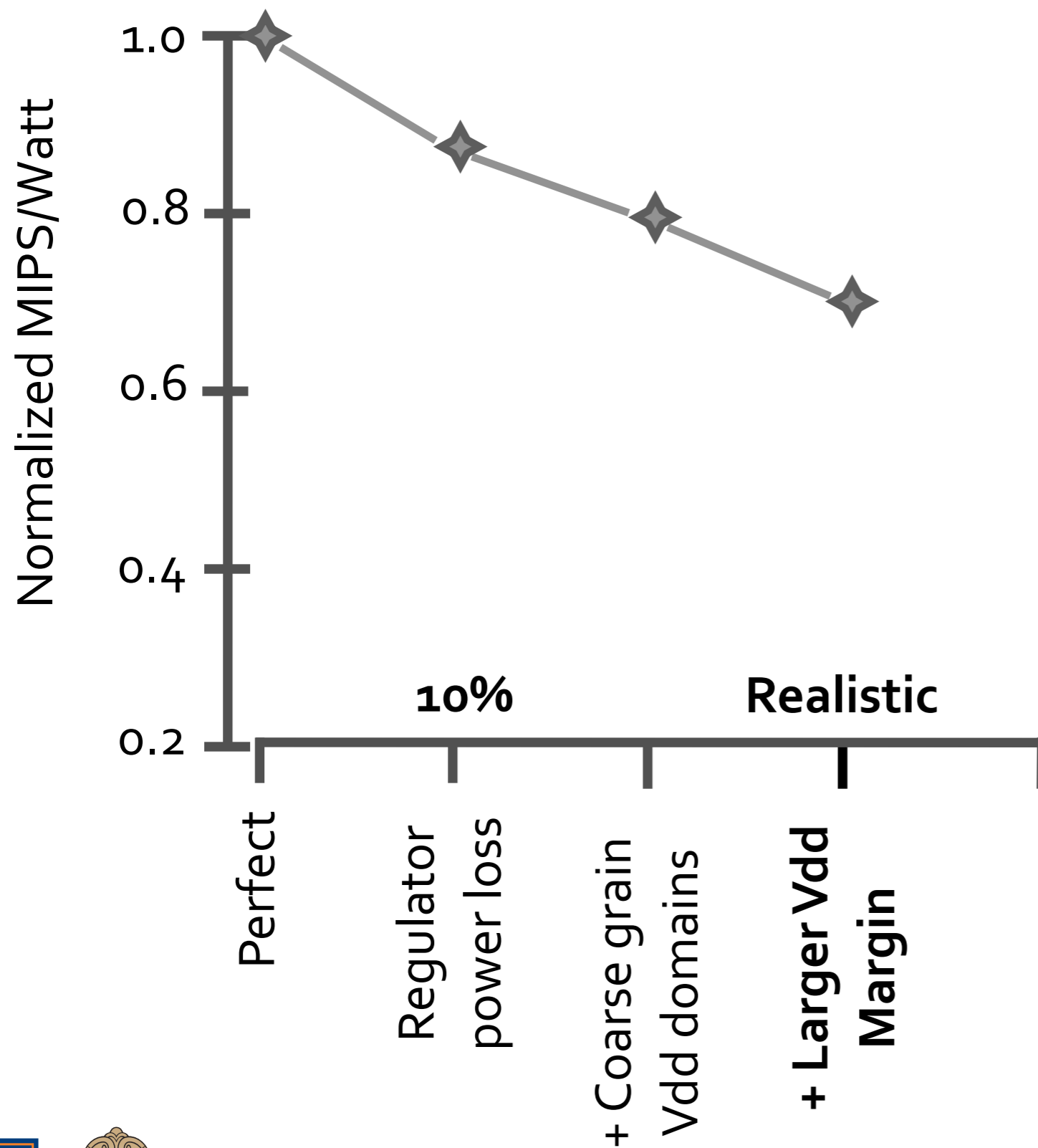
Effectiveness of Single Vdd Domain per Chip



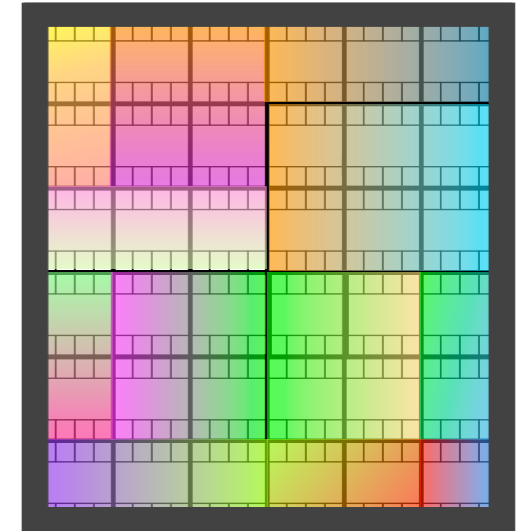
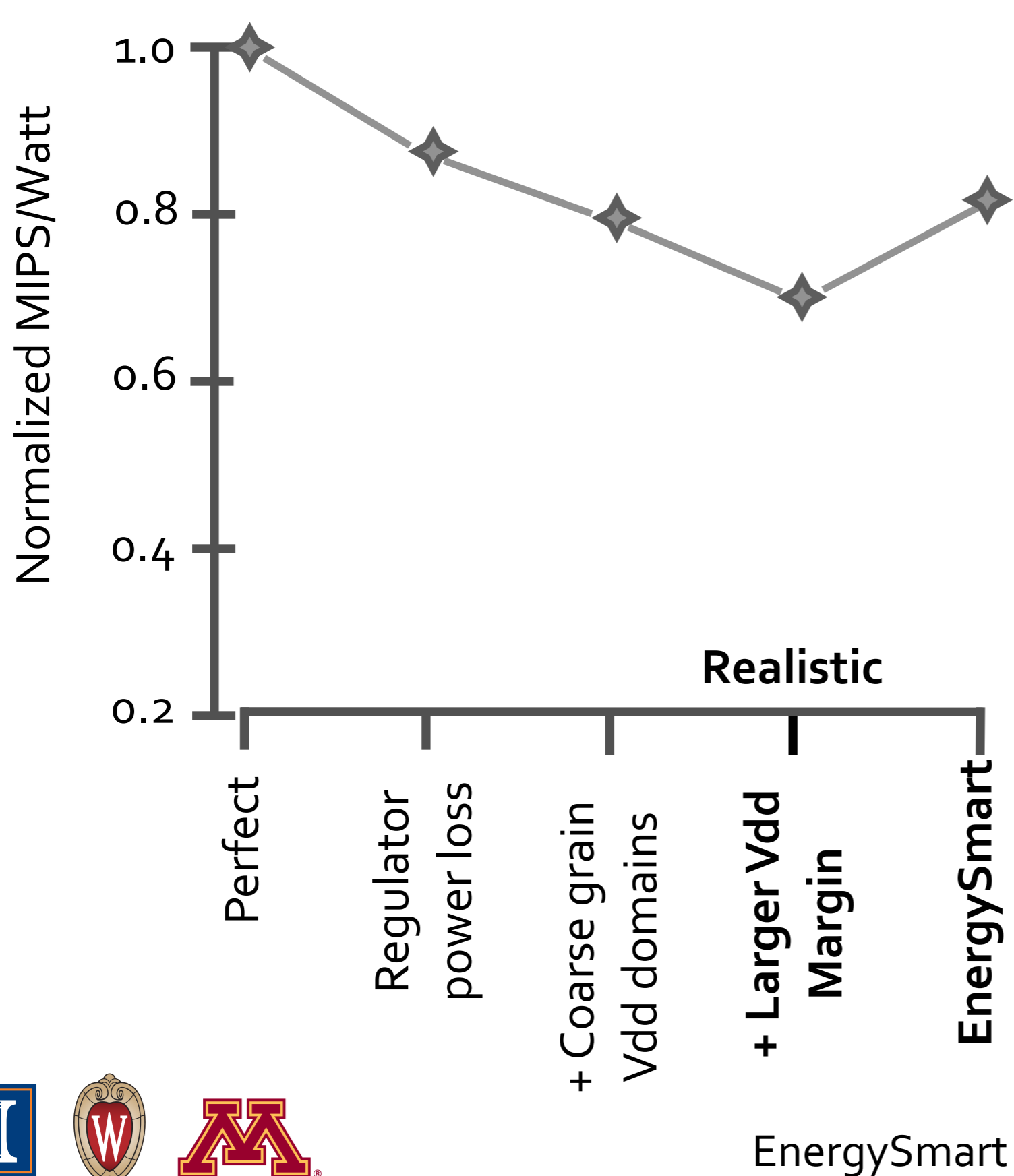
Effectiveness of Single Vdd Domain per Chip



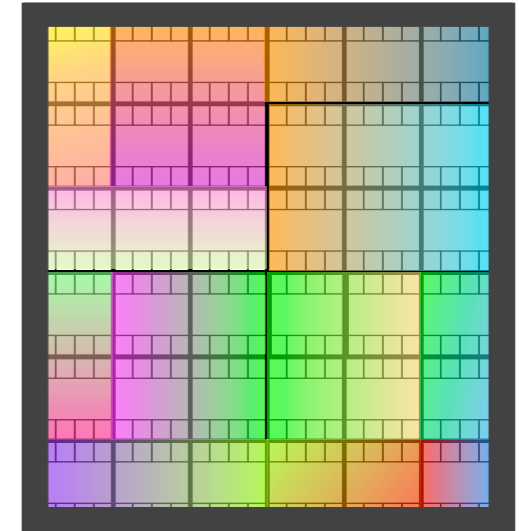
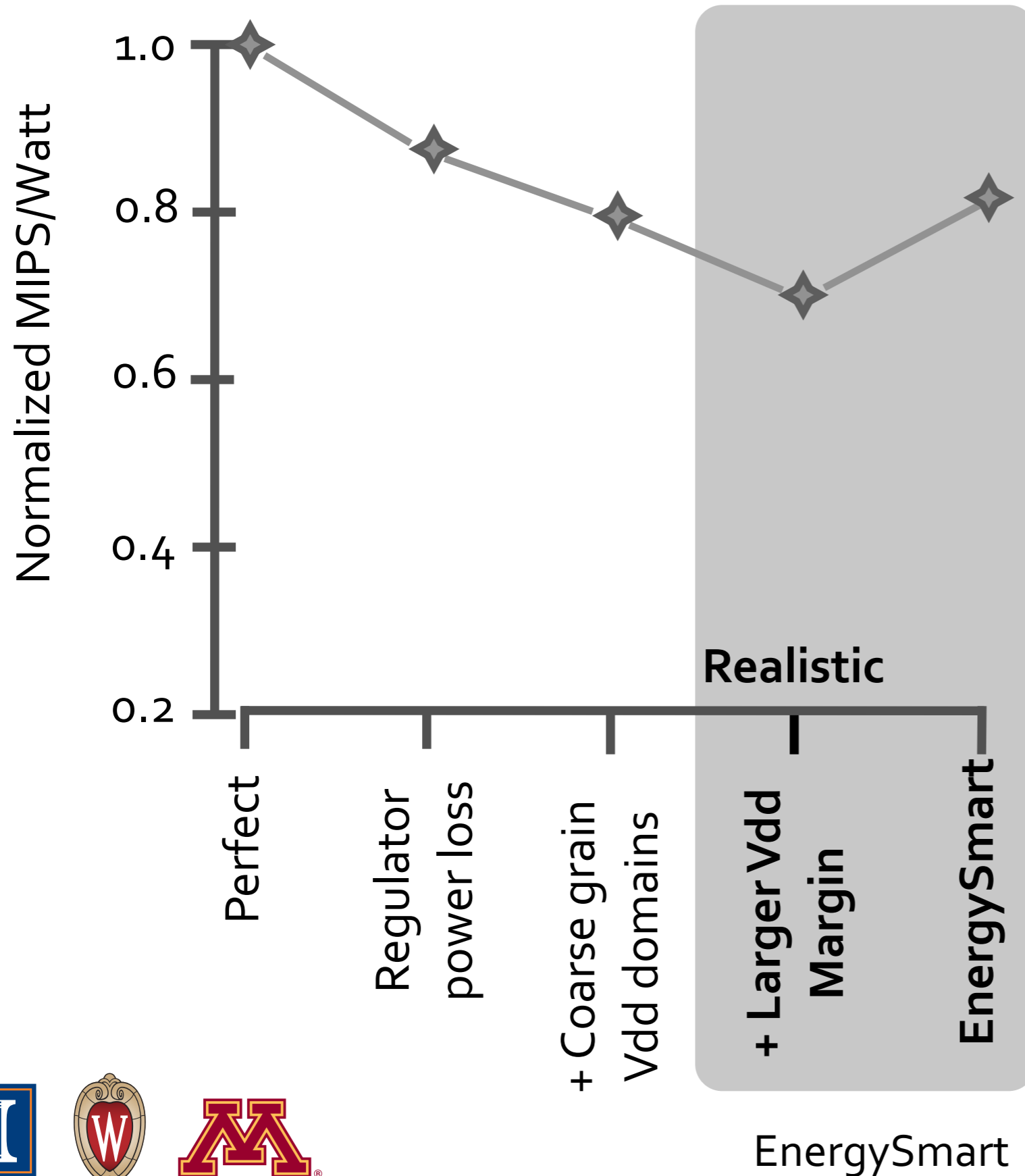
Effectiveness of Single Vdd Domain per Chip



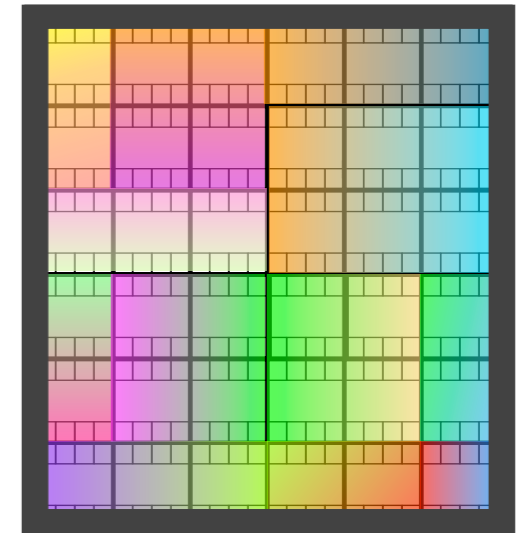
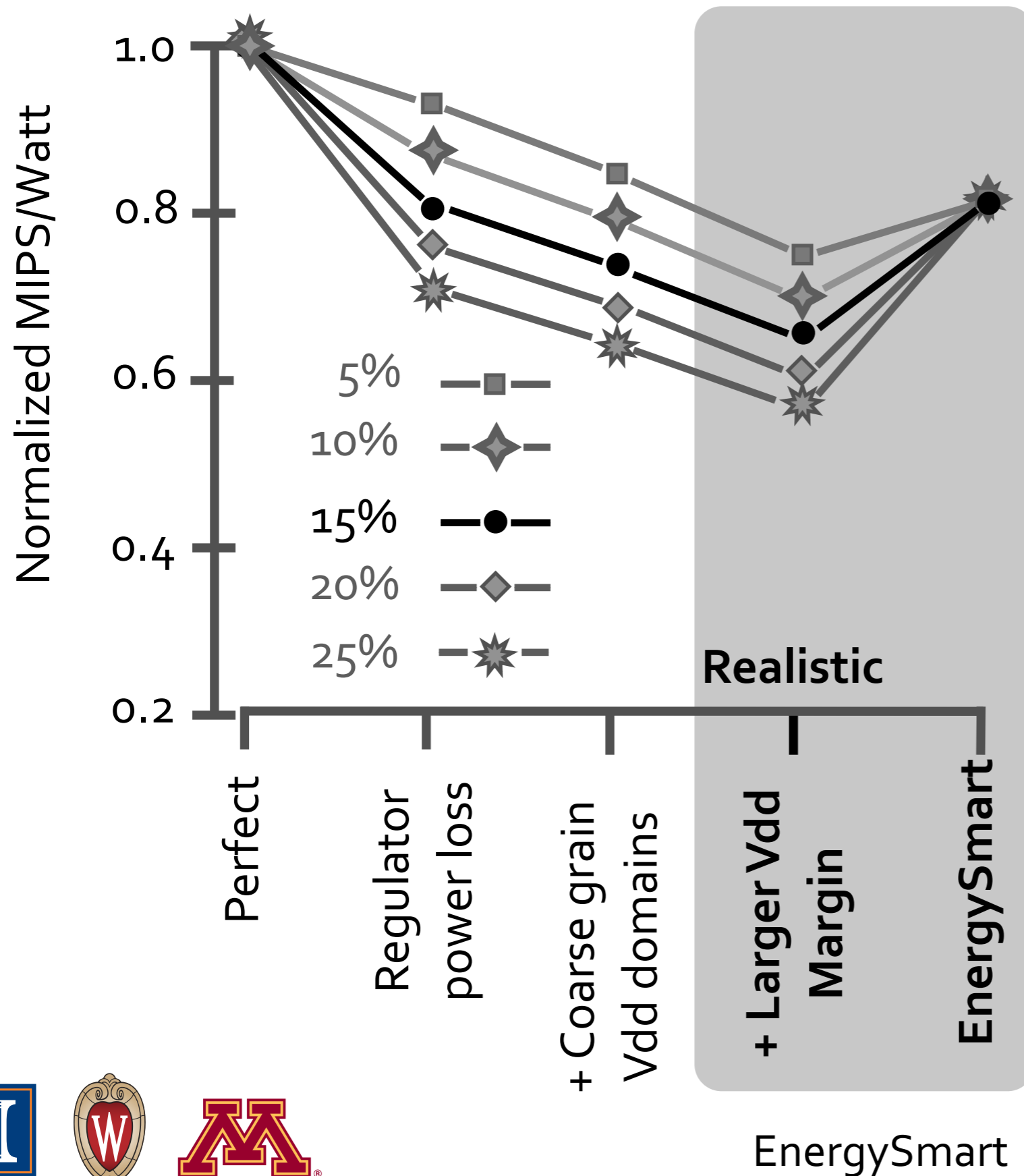
Effectiveness of Single Vdd Domain per Chip



Effectiveness of Single Vdd Domain per Chip



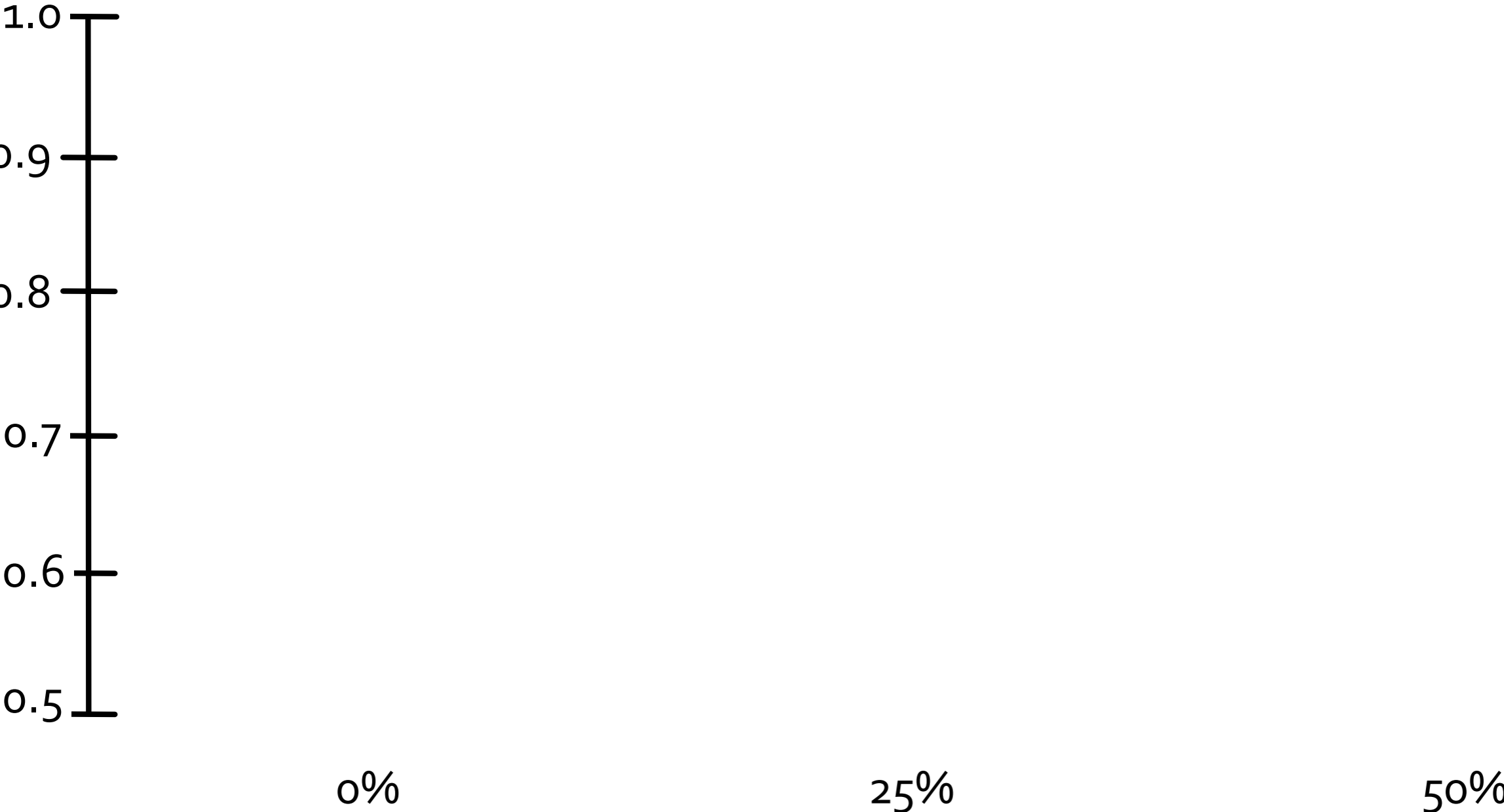
Effectiveness of Single Vdd Domain per Chip



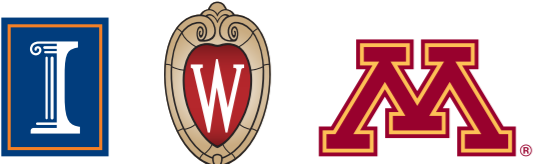
EnergySmart is more efficient

Cluster Assignment

Normalized MIPS/Watt



% Unavailable Clusters

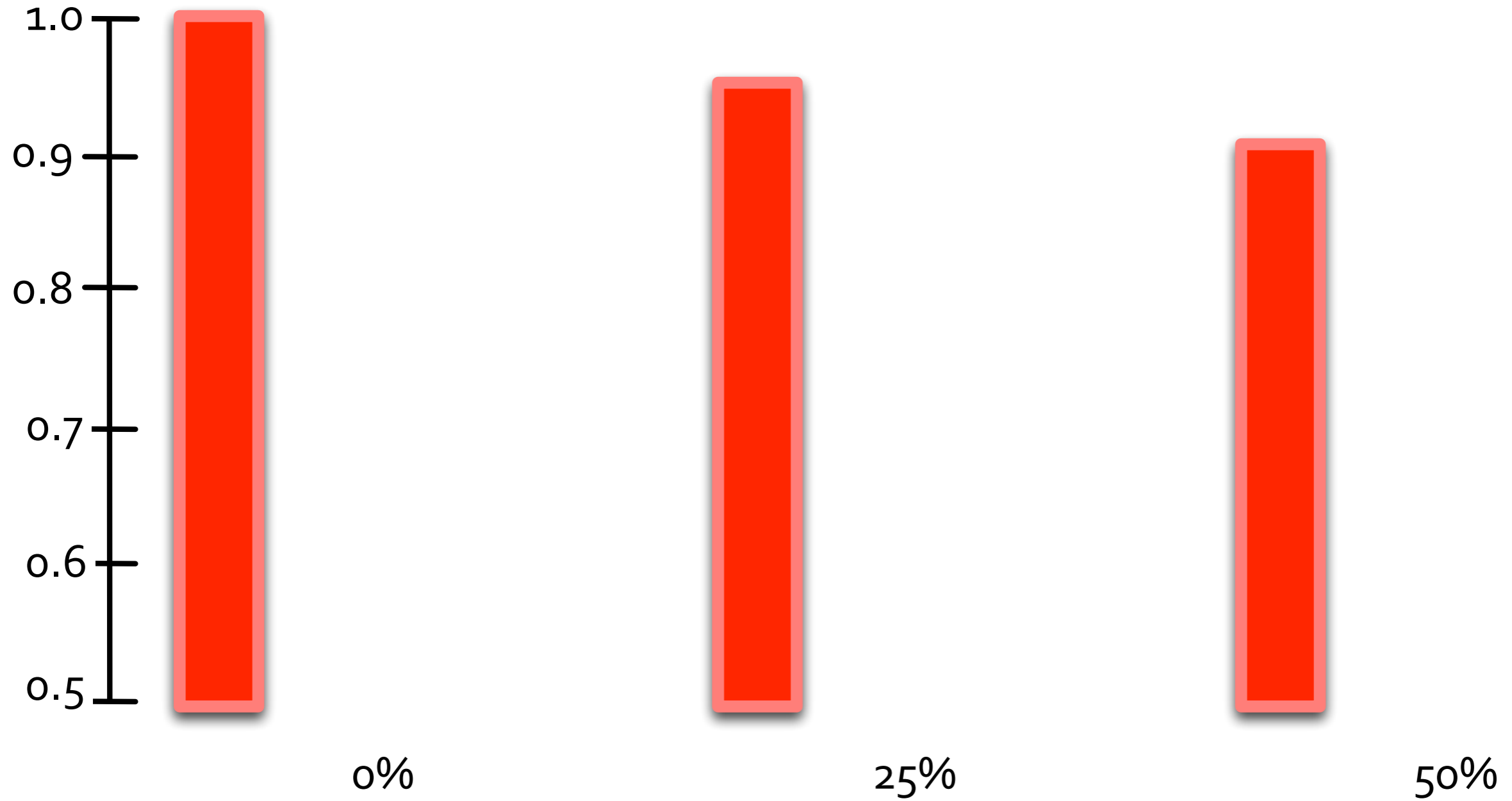


EnergySmart

Cluster Assignment

 EnergySmart

Normalized MIPS/Watt



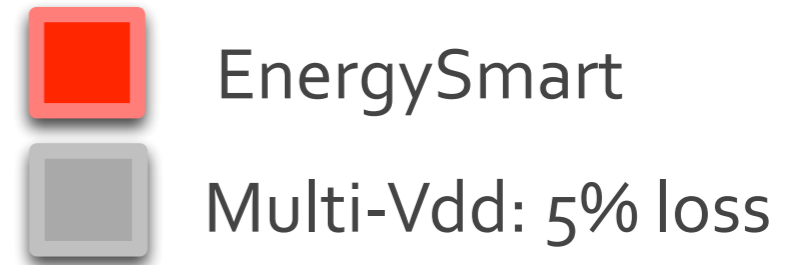
% Unavailable Clusters

EnergySmart

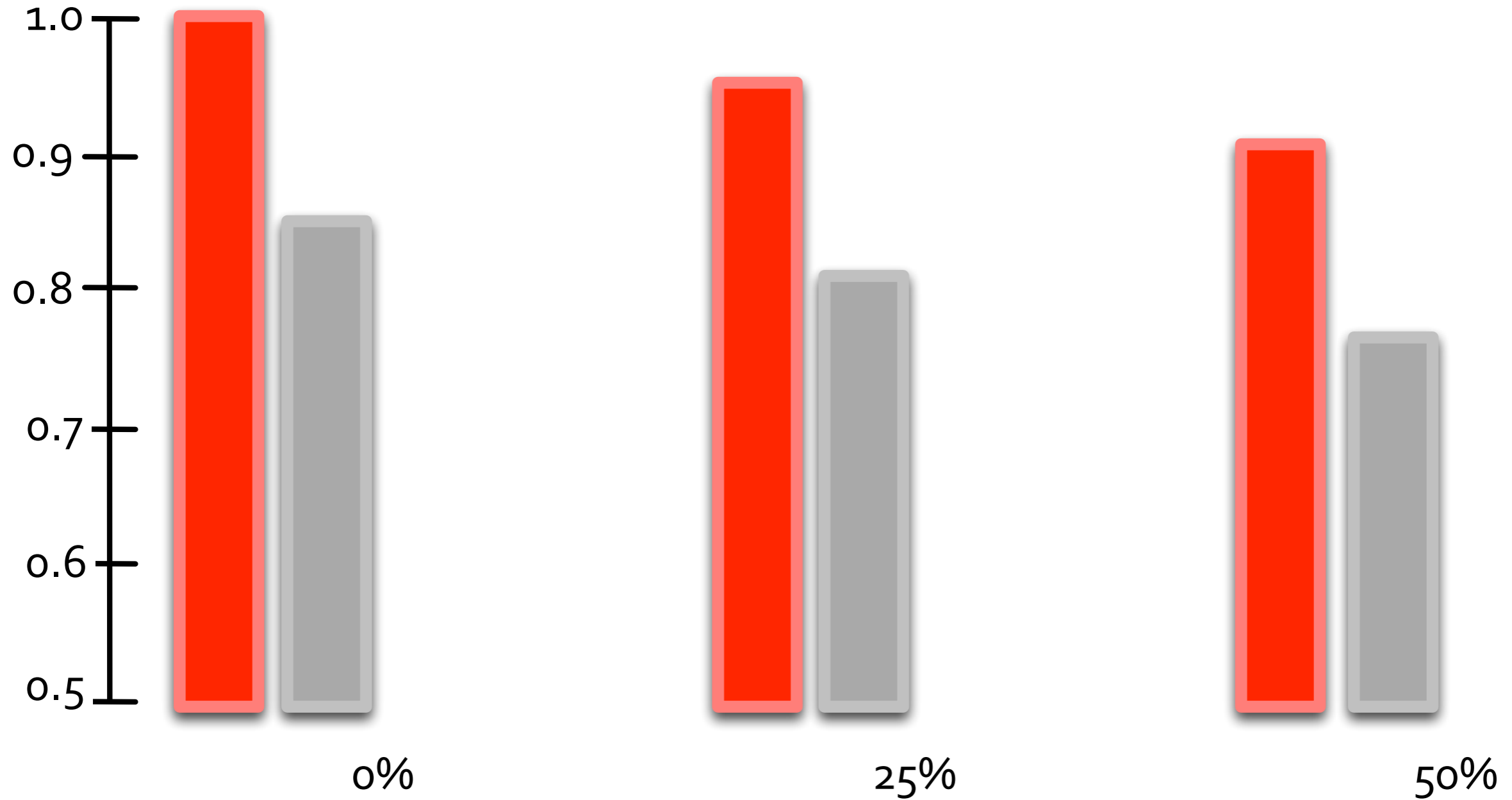
16



Cluster Assignment



Normalized MIPS/Watt



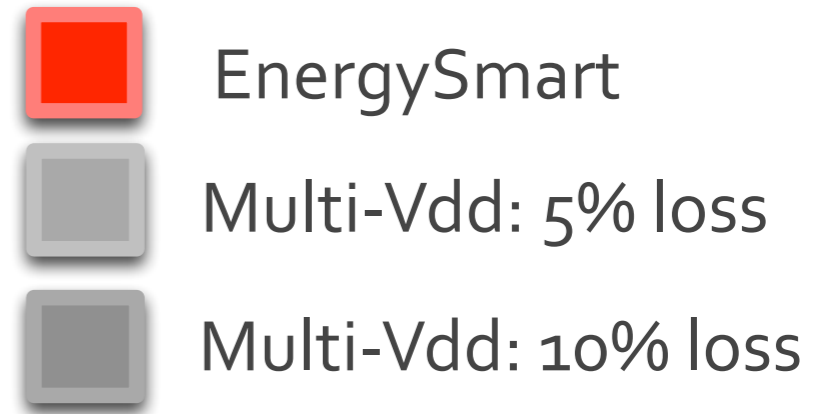
% Unavailable Clusters

EnergySmart

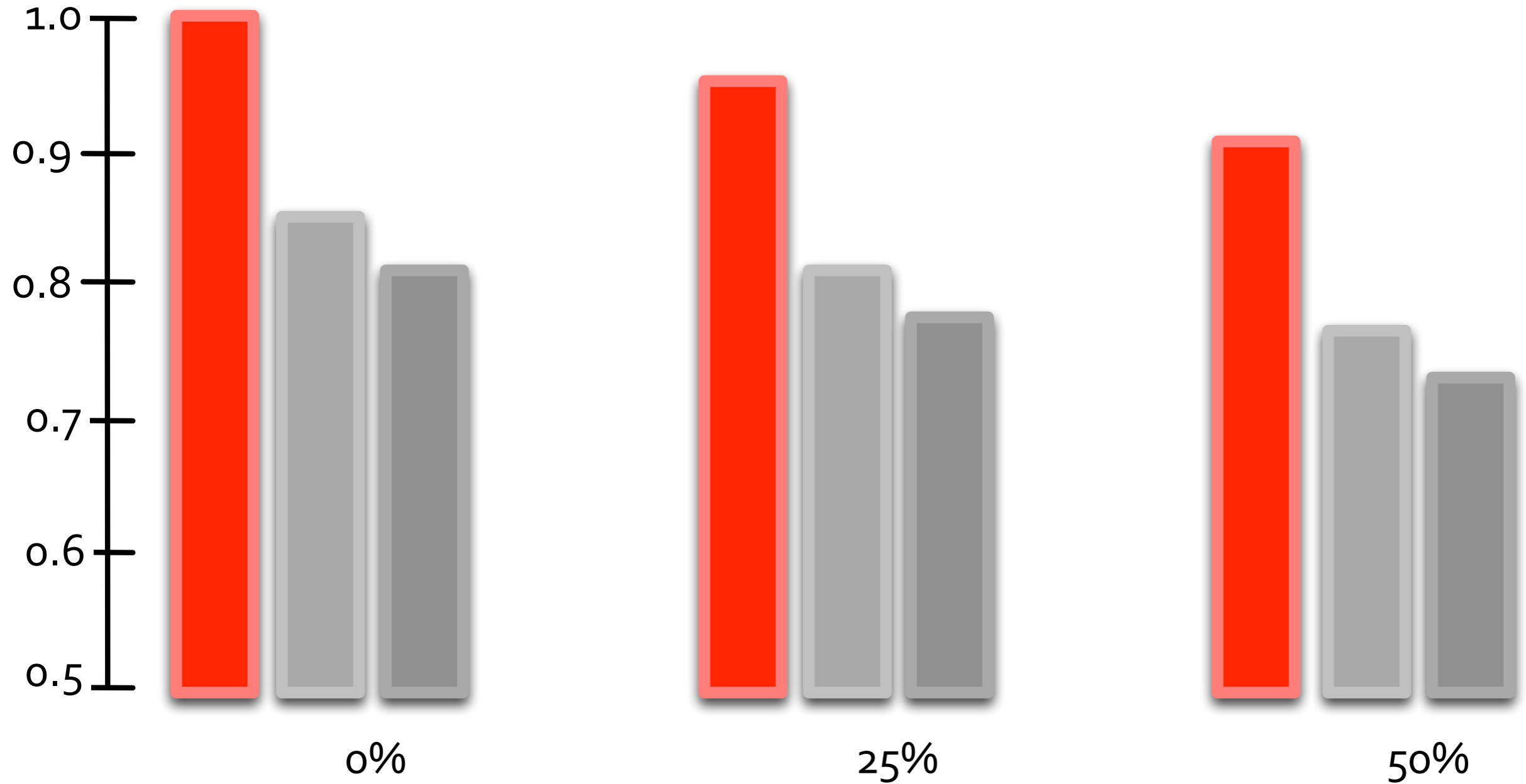
16



Cluster Assignment



Normalized MIPS/Watt



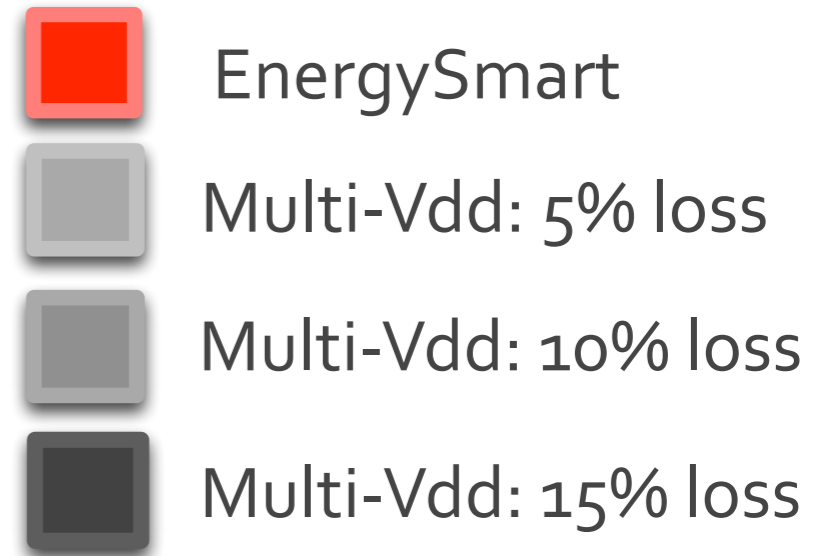
% Unavailable Clusters

EnergySmart

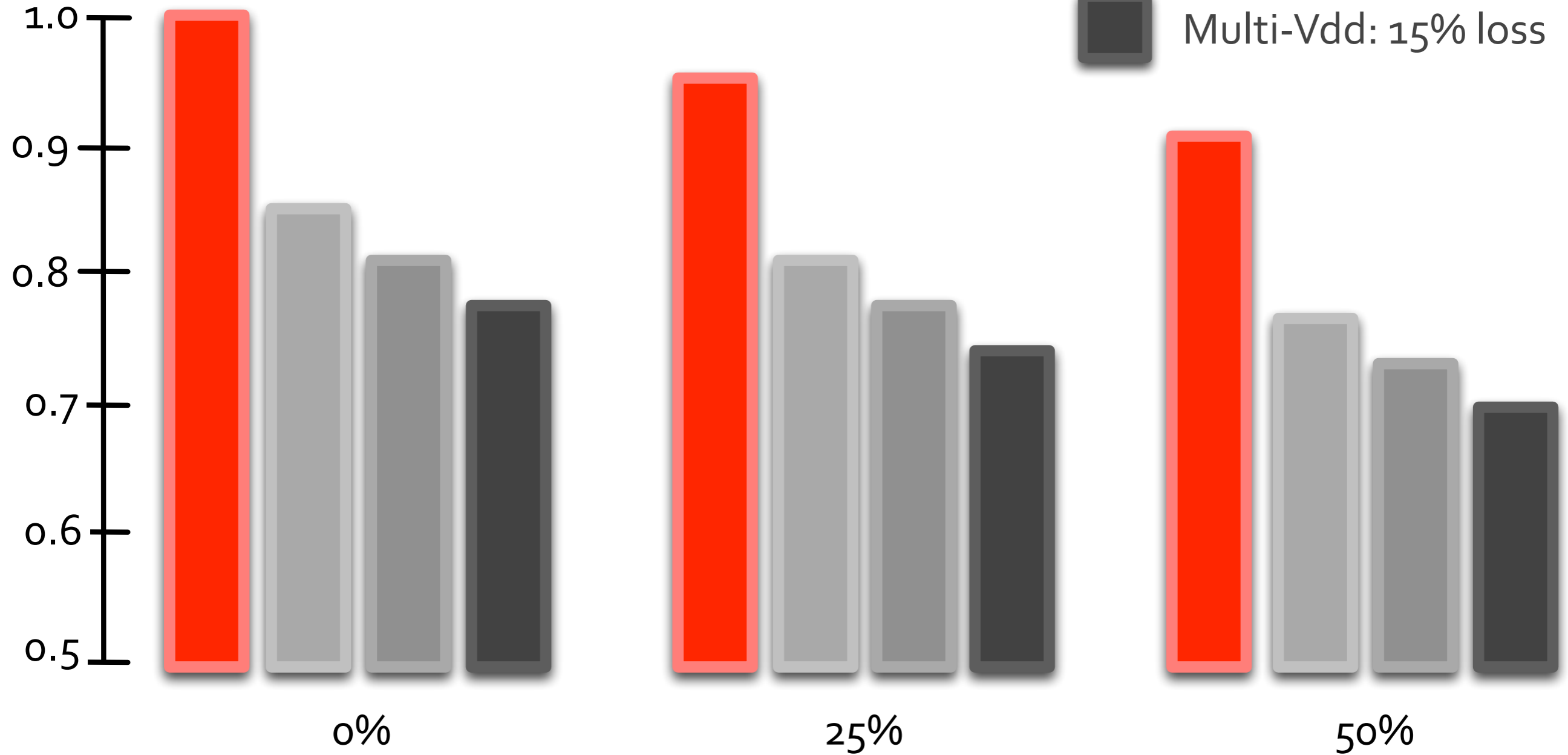
16



Cluster Assignment



Normalized MIPS/Watt



% Unavailable Clusters

EnergySmart

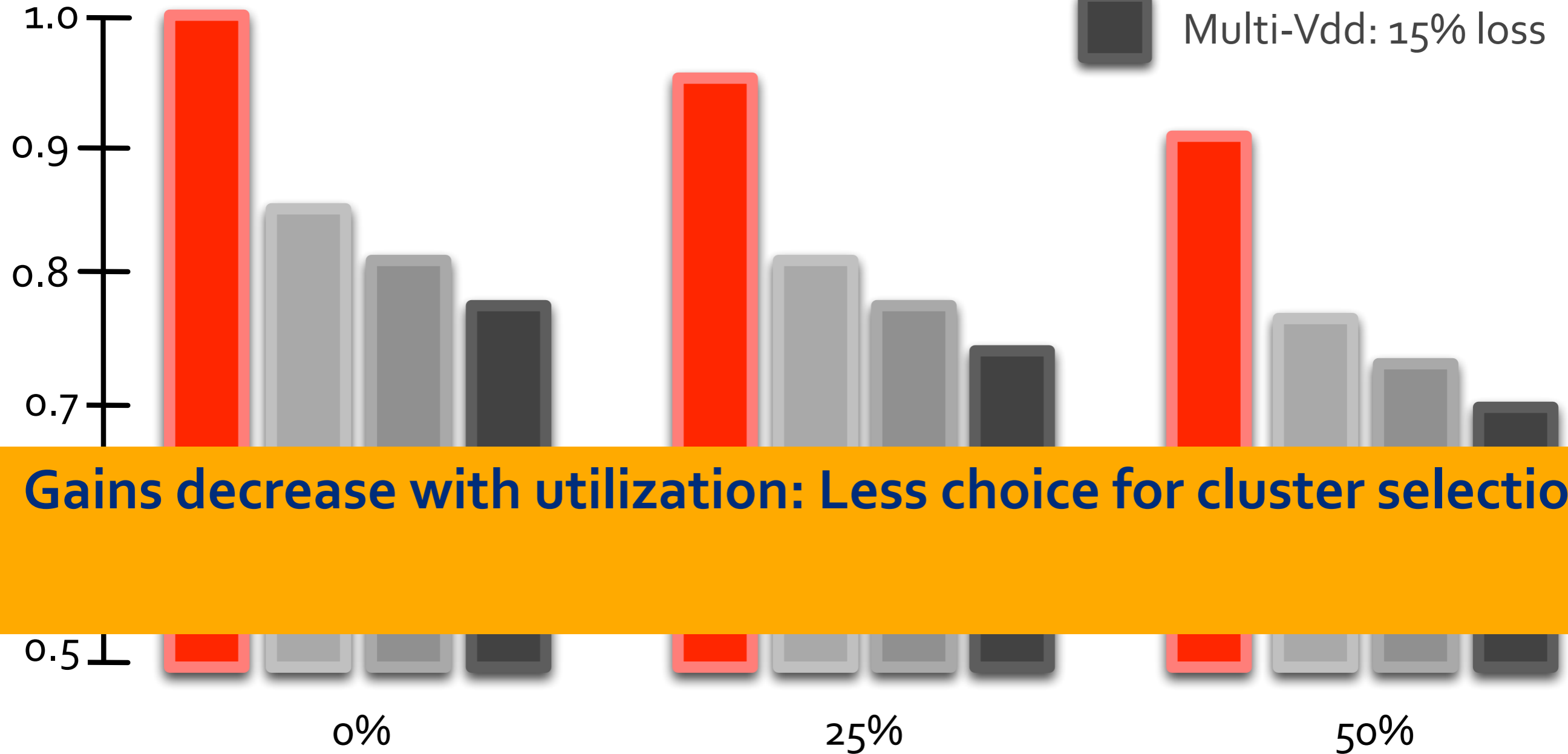
16



Cluster Assignment

- EnergySmart
- Multi-Vdd: 5% loss
- Multi-Vdd: 10% loss
- Multi-Vdd: 15% loss

Normalized MIPS/Watt



Gains decrease with utilization: Less choice for cluster selection

% Unavailable Clusters

EnergySmart

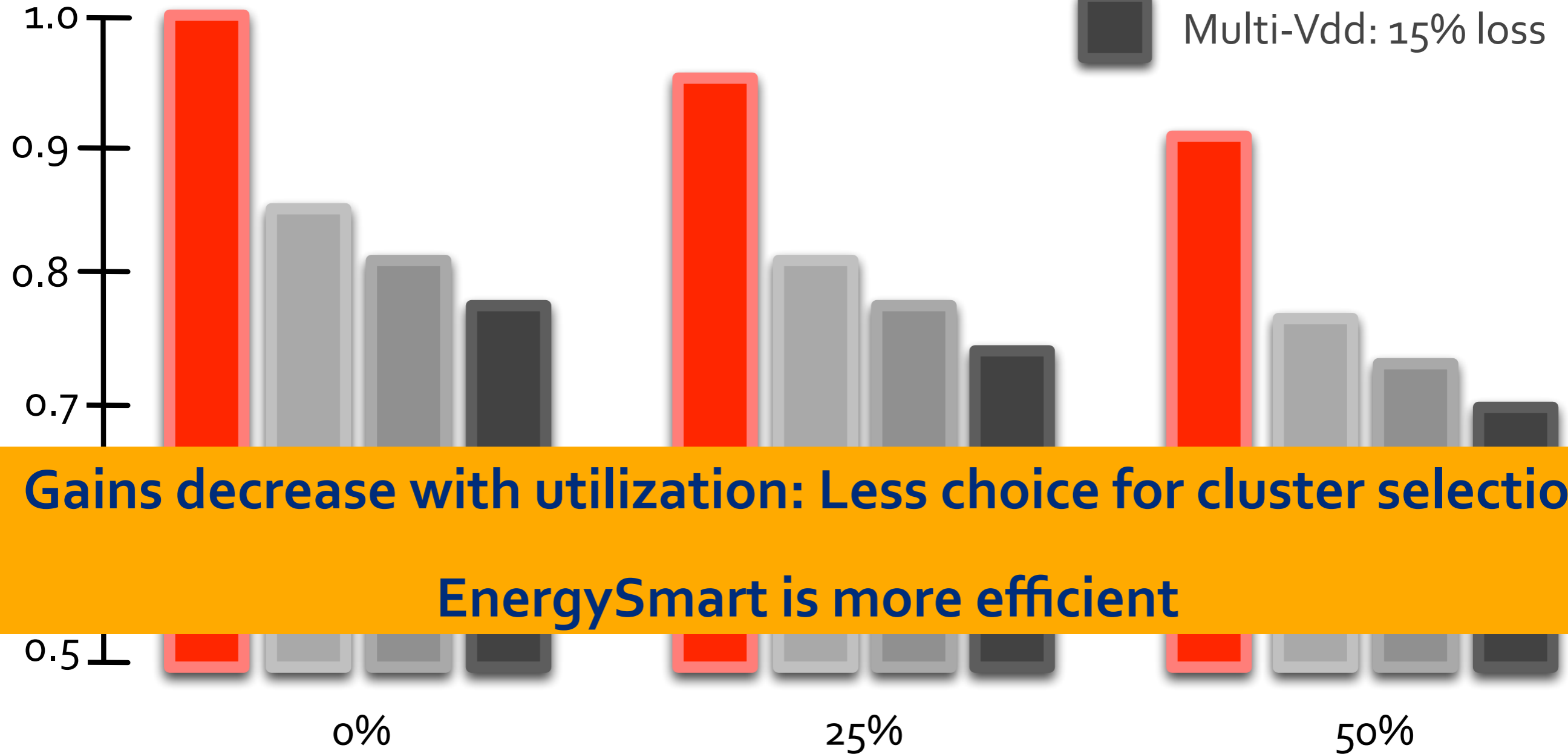
16



Cluster Assignment

- EnergySmart
- Multi-Vdd: 5% loss
- Multi-Vdd: 10% loss
- Multi-Vdd: 15% loss

Normalized MIPS/Watt



Gains decrease with utilization: Less choice for cluster selection

EnergySmart is more efficient

% Unavailable Clusters

EnergySmart

16



Conclusion



Conclusion

Energy Smart:

Eschewing multi-Vdd domains to tolerate variation at NTV



Conclusion

Energy Smart:

Eschewing multi-Vdd domains to tolerate variation at NTV



Conclusion

Energy Smart:

Eschewing multi-Vdd domains to tolerate variation at NTV

- Multiple f domains rather than Vdd domains



Conclusion

Energy Smart:

Eschewing multi-Vdd domains to tolerate variation at NTV

- Multiple f domains rather than Vdd domains
 - Cluster selection replaces multi-Vdd adaptation



Conclusion

Energy Smart:

Eschewing multi-Vdd domains to tolerate variation at NTV

- Multiple f domains rather than Vdd domains
 - Cluster selection replaces multi-Vdd adaptation
- Simple clustered hardware architecture



Conclusion

Energy Smart:

Eschewing multi-Vdd domains to tolerate variation at NTV

- Multiple f domains rather than Vdd domains
 - Cluster selection replaces multi-Vdd adaptation
- Simple clustered hardware architecture
- Simple, variation-aware cluster assignment



Conclusion

Energy Smart:

Eschewing multi-Vdd domains to tolerate variation at NTV

- Multiple f domains rather than Vdd domains
 - Cluster selection replaces multi-Vdd adaptation
- Simple clustered hardware architecture
- Simple, variation-aware cluster assignment
- Energy-efficiency within 81% of perfect organization



Conclusion

Energy Smart:

Eschewing multi-Vdd domains to tolerate variation at NTV

- Multiple f domains rather than Vdd domains
 - Cluster selection replaces multi-Vdd adaptation
- Simple clustered hardware architecture
- Simple, variation-aware cluster assignment
- Energy-efficiency within 81% of perfect organization
 - Realistic multi-Vdd organization achieves only 69% (at 90% regulator efficiency)



EnergySmart: Toward Energy-Efficient Many-Cores for Near-Threshold Voltage Computing

Ulya R. Karpuzcu*, Abhishek Sinkar*, Nam Sung Kim*, Josep Torrellas❖
*University of Minnesota ❖University of Illinois *University of Wisconsin



UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN



WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON



UNIVERSITY OF MINNESOTA