Drs.NAS: Ultra-Efficient Neural Architecture Search for Recommendation Systems

Ruixuan Wang [1,2], Xun Jiao [2]

[1] Arizona State University, [2] Villanova University

Outline

- Introduction
- Methodology
- Experimental Setup & Results
- Conclusion

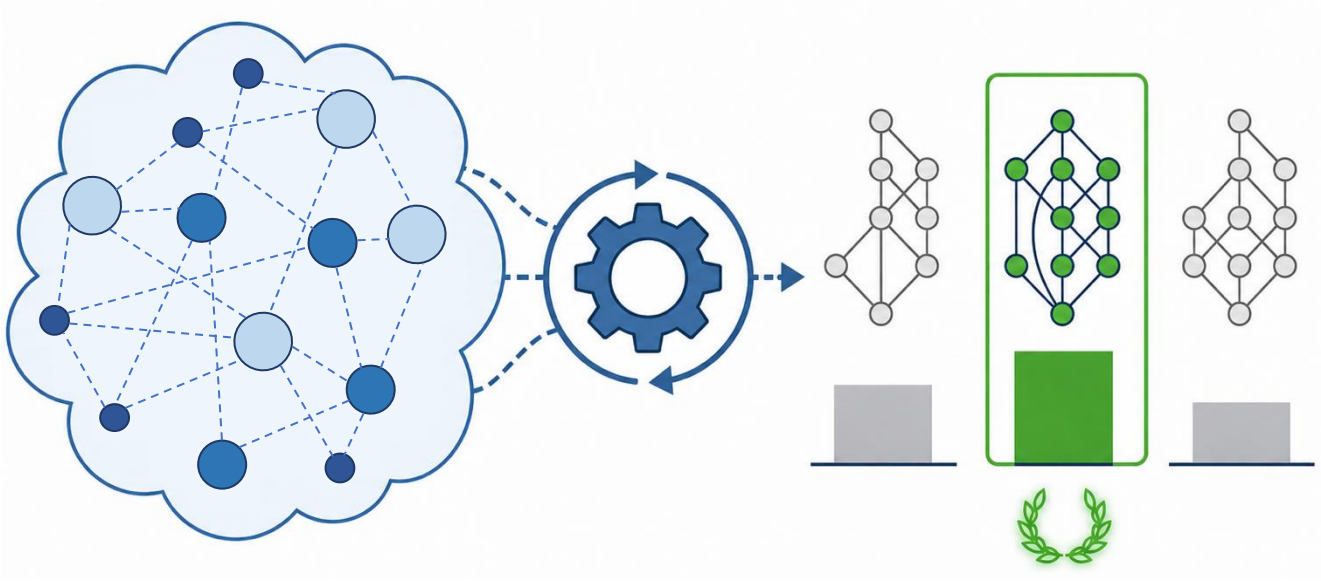
Deep Learning-based Recommendation Systems

- Deep learning–based recommendation systems (DRSs) are dominant workloads in hyperscale data centers.
- Designing efficient DRS architectures remains challenging because expert-driven exploration of large DRS design spaces is costly and unscalable.



Neural Architecture Search

Neural architecture search (NAS) automates the exploration of a pre-defined design space to identify the optimal model architecture for a given task, substantially reducing reliance on manual design effort.



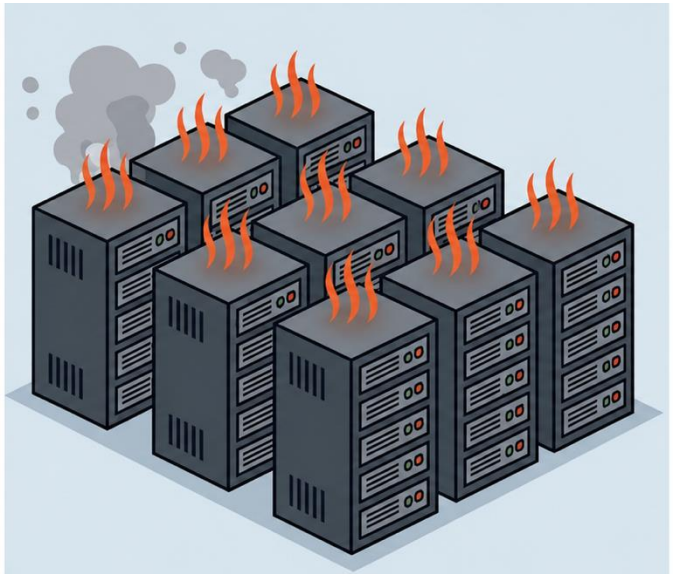
Design space

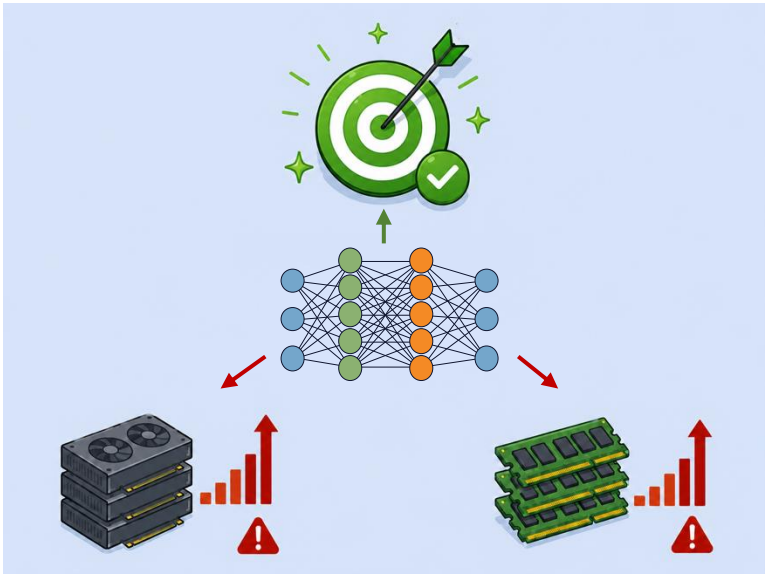
Search algorithm

Optimal architecture

Two Main Challenges of NAS for DRS

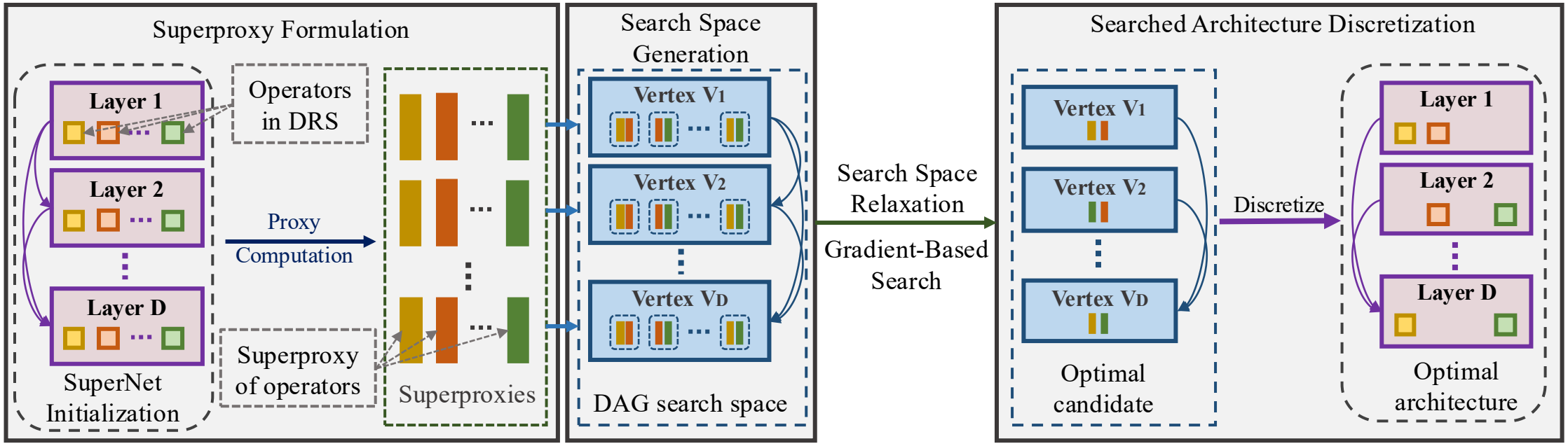
- **Prohibitive search costs**: Conventional NAS for DRS can require 5-18 hours of GPU computation.
- **Expensive deployment costs**: Accuracy-first search often produces computationally and memory-intensive architectures.





Our Solution: Drs.NAS

- We propose Drs.NAS, an ultra-efficient NAS framework for DRS architecture search.



Superproxy

- We construct a seven-dimensional superproxy for each operator by aggregating zero-cost proxies.

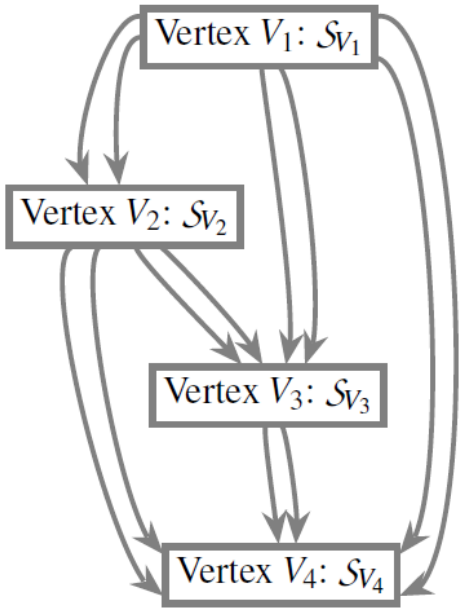
Proxy set	Proxies	Description
Overhead-relevant	#Params, #FLOPs, W_{norm}, G_{norm}	Reflect capacity/ resource-related tendencies
Overhead-irrelevant	SynFlow, ZiCo, MeCo	Estimate trainability/ predictive potential

- The superproxy serves as a performance/efficiency indicator, enabling architecture search without iterative model training and candidate architecture validation.

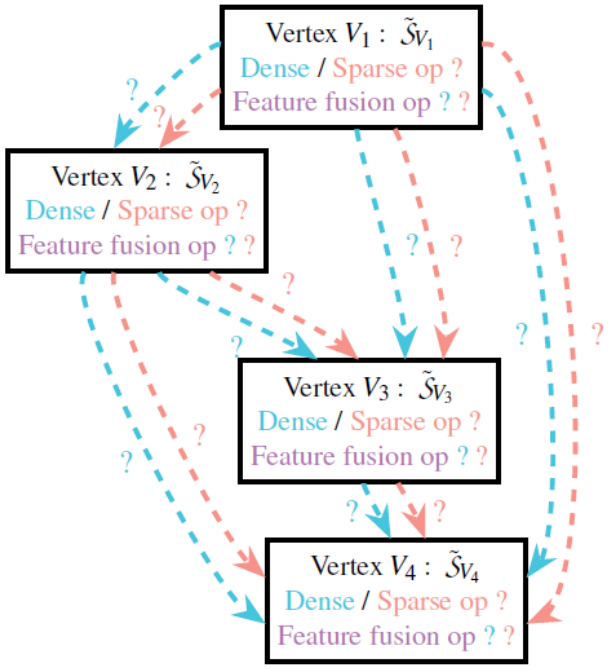
$$S_o = [S_{param}(o), S_{flops}(o), \dots, S_{zico}(o), S_{meco}(o)]$$

Drs.NAS Search Space

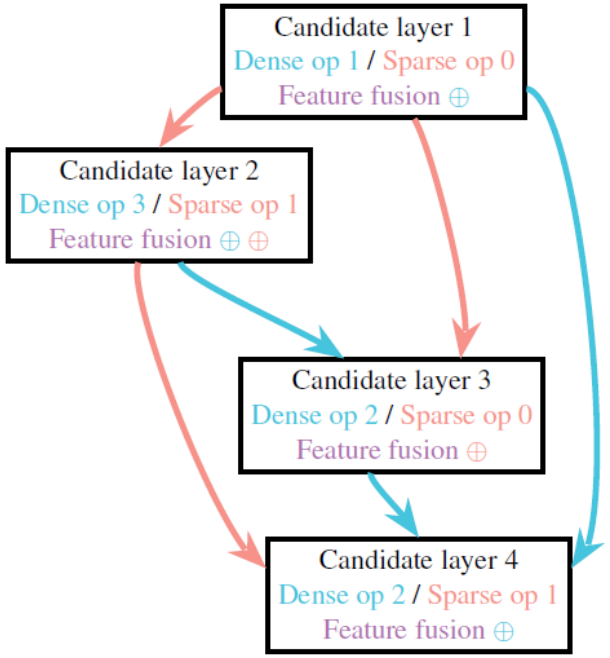
- Drs.NAS reformulates architecture search as an optimal subgraph sampling problem over a directed acyclic graph (DAG) search space.



DAG
search space



Relaxed
search space



Discretized
candidate architecture

Gradient-based Architecture Search

- We formulate a composite loss function and jointly optimize the predictive performance and efficiency of candidate architectures based on gradient descent.

$$\mathcal{L}_A = \lambda_1 * \mathcal{L}_{vertex} + \lambda_2 * \mathcal{L}_{edge} + \lambda_3 * \mathcal{L}_{cost}$$

Loss	Description
$\mathcal{L}_{vertex}$	**Vertex selection**: Encourages effective operators in each layer to improve predictive performance.
$\mathcal{L}_{edge}$	**Edge selection:** Regularizes computational flow to improve data and gradient propagation.
$\mathcal{L}_{cost}$	**Architecture selection:** Balances model capacity against computational cost.

Experimental Setup

- We evaluate three widely adopted click-through-rate (CTR) prediction benchmarks.

Benchmark	Dense feature	Sparse feature	Samples
Criteo	13	26	45,840,617
Avazu	0	23	40,428,967
KDD	3	10	149,639,104

- We evaluate **Log Loss** and **AUC score** for predictive performance, and **#Params** and **#FLOPs** for memory and computational efficiency.

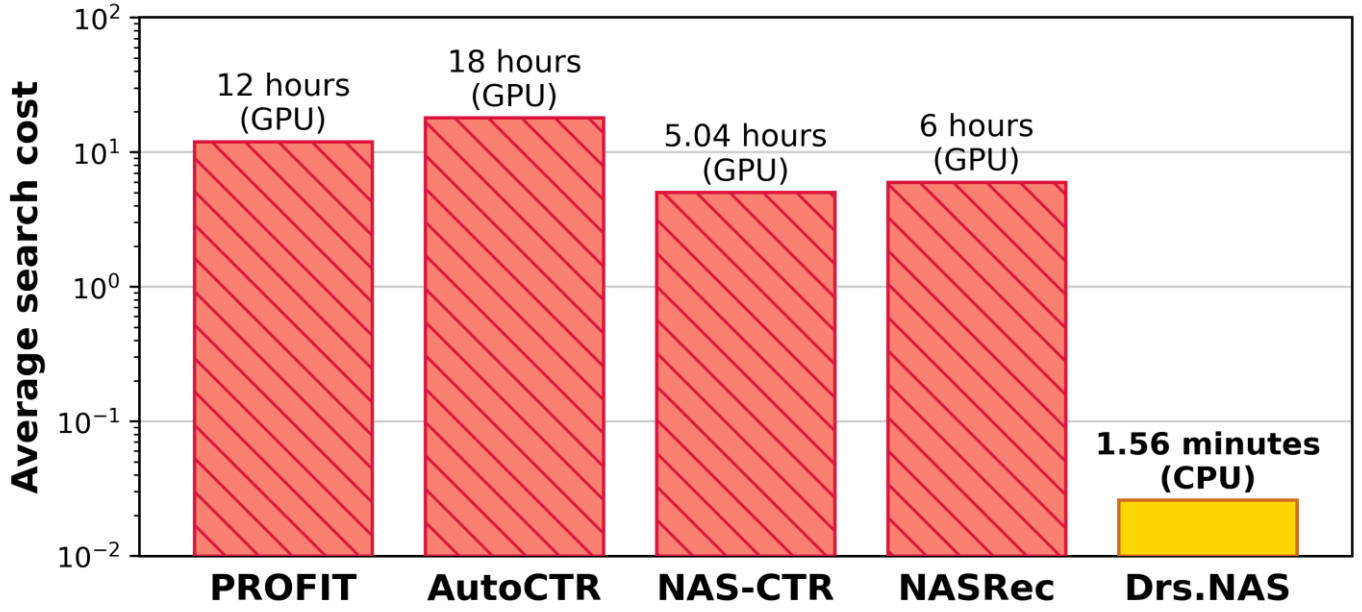
Drs.NAS Search Space Configurations

- **Full search space**:
  - Dense operators: Fully connected (FC), sigmoid gating, summation, and dot-product
  - Sparse operators: Embedded FC and transformer
  - Feature fusion

- **Small search space**:
  - Dense operators: FC and dot-product
  - Sparse operators: Embedded FC
  - Feature fusion

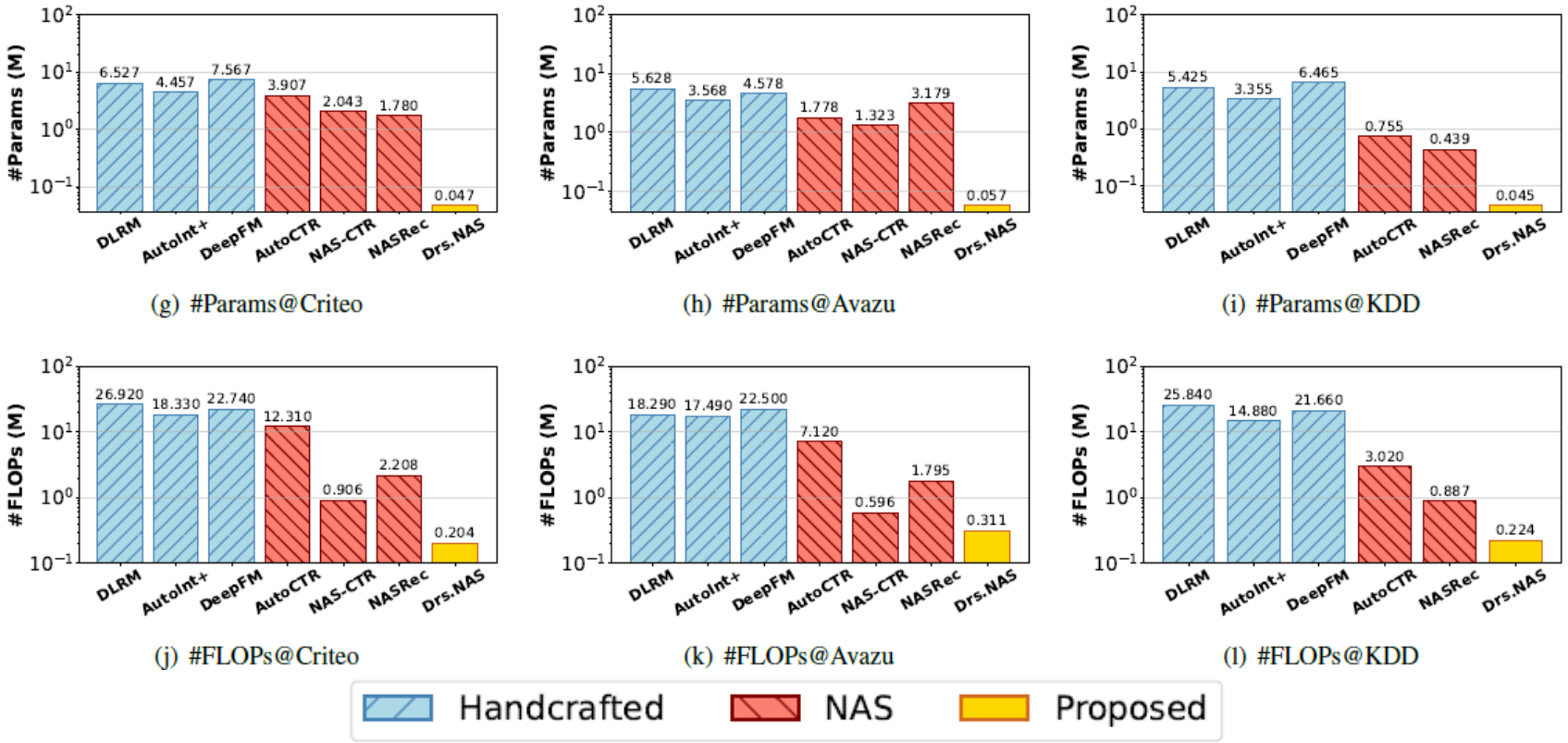
Search Cost Reduction

- By eliminating the requirement for iterative model training and candidate architecture validation, Drs.NAS substantially reduces search time and cost, providing up to 692× faster search than NAS baselines.



Efficiency Evaluation

- Drs.NAS searched architectures are both computationally and memory efficient, reducing model size and FLOPs by 34.9× and 14.7×, on average, relative to NAS baselines.



Inference Time Comparison

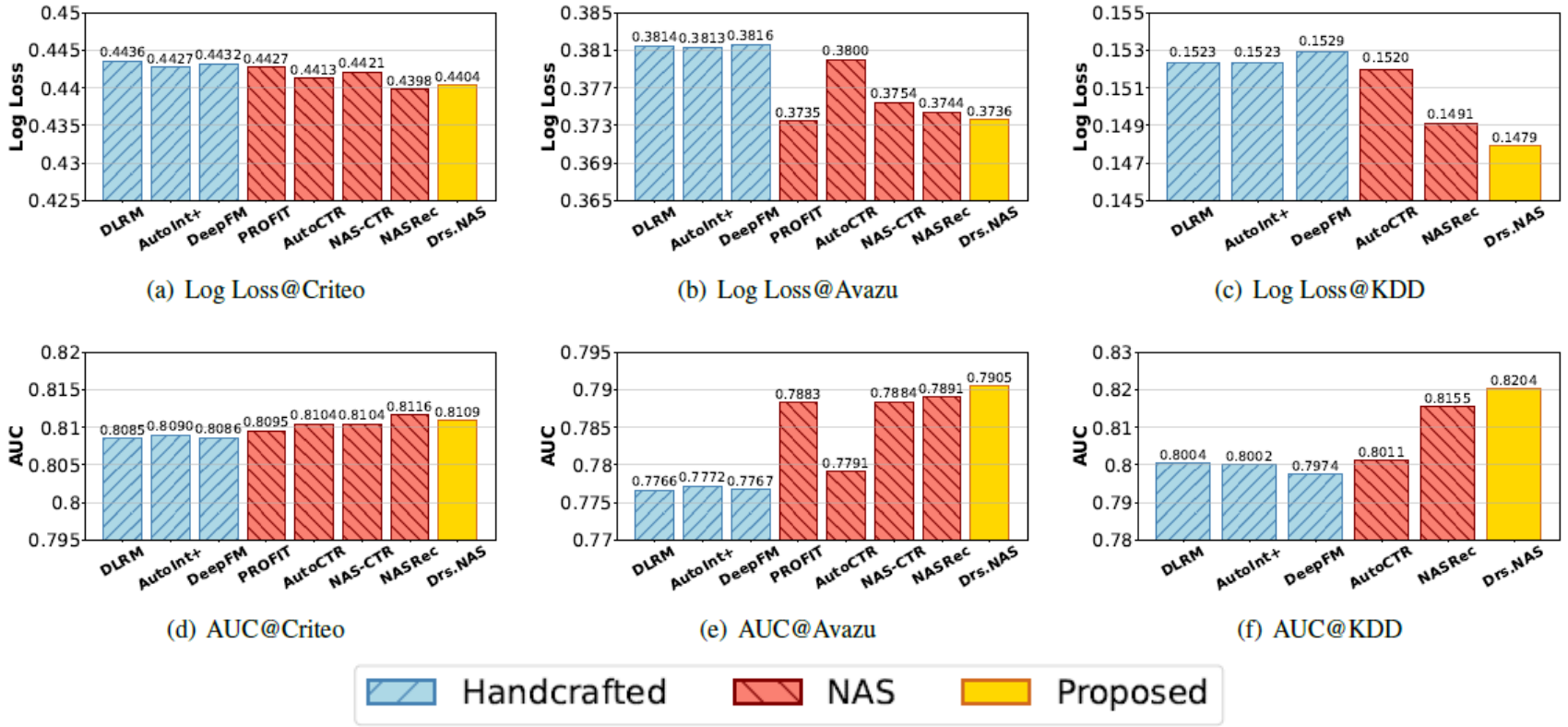
- Drs.NAS searched architectures achieve 60.8% and 24.0% lower per-sample inference time on CPU and GPU, respectively, than the state-of-the-art NAS baseline.

Method - Search space	Criteo (GPU)	Criteo (CPU)	Avazu (GPU)	Avazu (CPU)	KDD (GPU)	KDD (CPU)
NASRec - Small	4.96	43.26	3.78	44.15	2.47	53.94
NASRec - Full	8.18	86.36	5.99	69.36	6.19	76.42
Drs.NAS - Small	**4.69**	**19.72**	**3.44**	**20.33**	**1.84**	**18.56**
Drs.NAS - Full	5.21	25.92	3.98	24.91	4.06	33.25

Per-sample inference time (μs) comparison across different benchmarks and search space configurations.

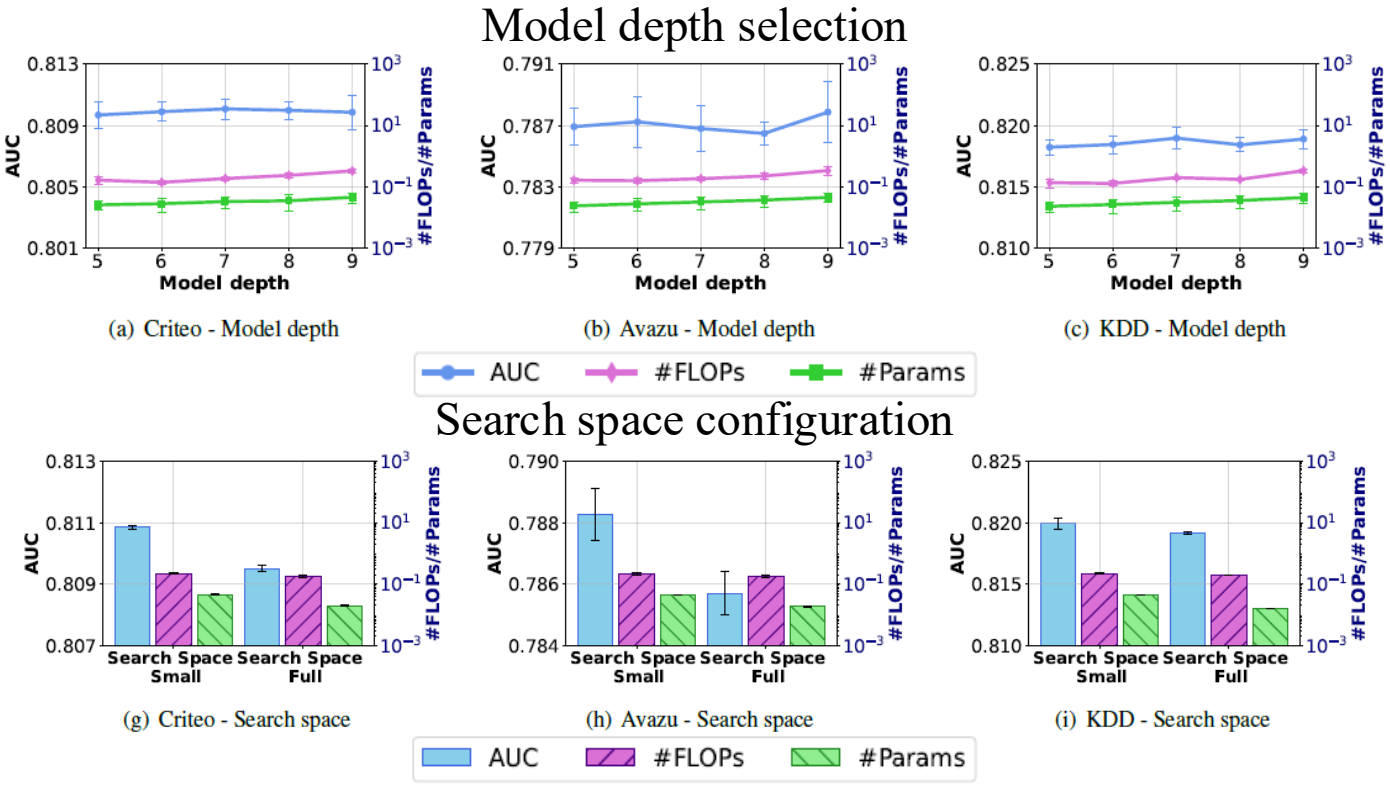
Predictive Performance Evaluation

- While significantly reducing search cost and improving architectural efficiency, Drs.NAS delivers on par or better predictive performance than NAS baselines.



Design Space Exploration

- Drs.NAS discovers competitive architectures across different model depths and search spaces, enabling systematic exploration of diverse performance–efficiency trade-offs.



Conclusion

- We propose Drs.NAS, an ultra-efficient NAS framework for DRS architecture search.
- Unlike prior NAS approaches that rely on costly training and validation, Drs.NAS uses superproxy-guided approach to reduce architecture search from GPU-hours to CPU-minutes.
- Drs.NAS searched architectures are 34.9× smaller model sizes and 14.7× fewer FLOPs, on average, while delivering on par or superior predictive performance than NAS baselines.

Thank you