LitePred: Transferable and Scalable Latency Prediction for Hardware-Aware Neural Architecture Search

Chengquan Feng, Li Lyna Zhang, Yuanchi Liu, Jiahang Xu, Chengruidong Zhang, Zhiyuan Wang, Ting Cao, Mao Yang, Haisheng Tan

Inference latency of DNN is important

Face Recognition

Speech Recognition

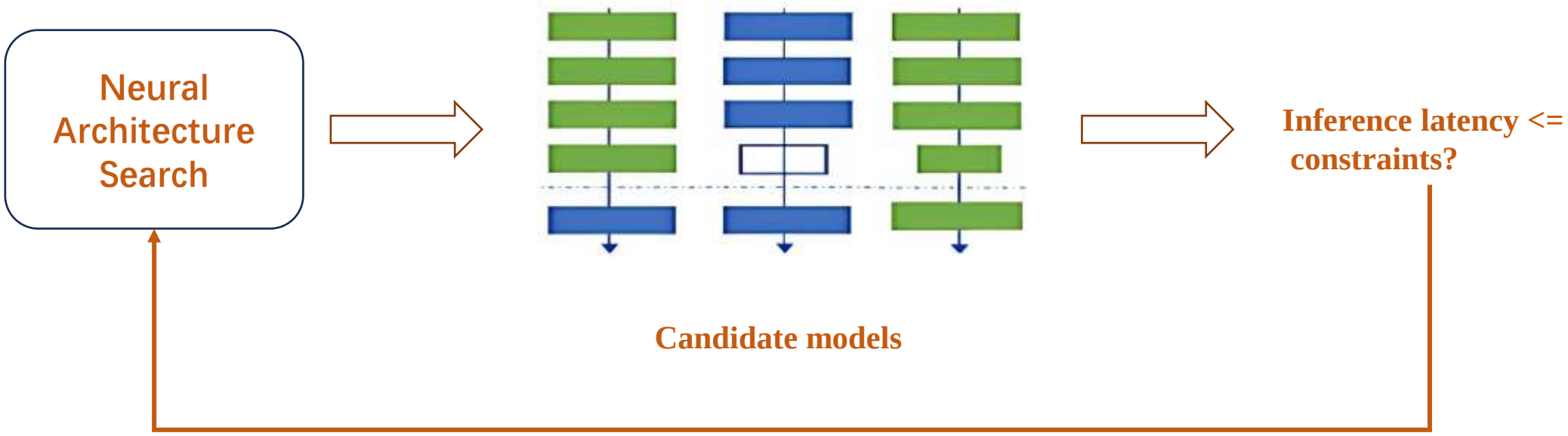
AR/VR

Edge device need low latency DNNs

Latency: the important model design metric

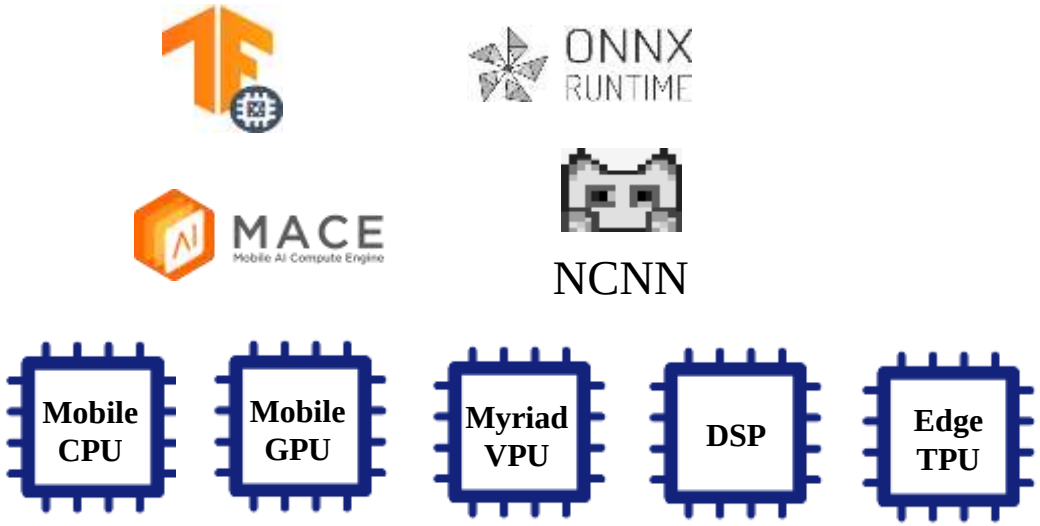
To get a model with high accuracy and low inference latency:

- Model design algorithms need consider the inference latency in the design process



Measurement latency is expensive and impractical

- Diverse devices and inference engines
  - More than 8318 smartphones
  - Various inference frameworks



- Large number of models in NAS space
  It will explore millions models in one search(eg,~0.3million in ProxylessNAS)

On-device measurement is expensive and impractical

Related Works

- FLOPs-based method

Not reflect the real inference latency

- Operator-level based method
  - Sum up all the operators'/ kernels'(fused operator) latencies

Need large cost to build predictors for new platform

- Model-level based method
  - Leverage GCN model to learn the graph optimization

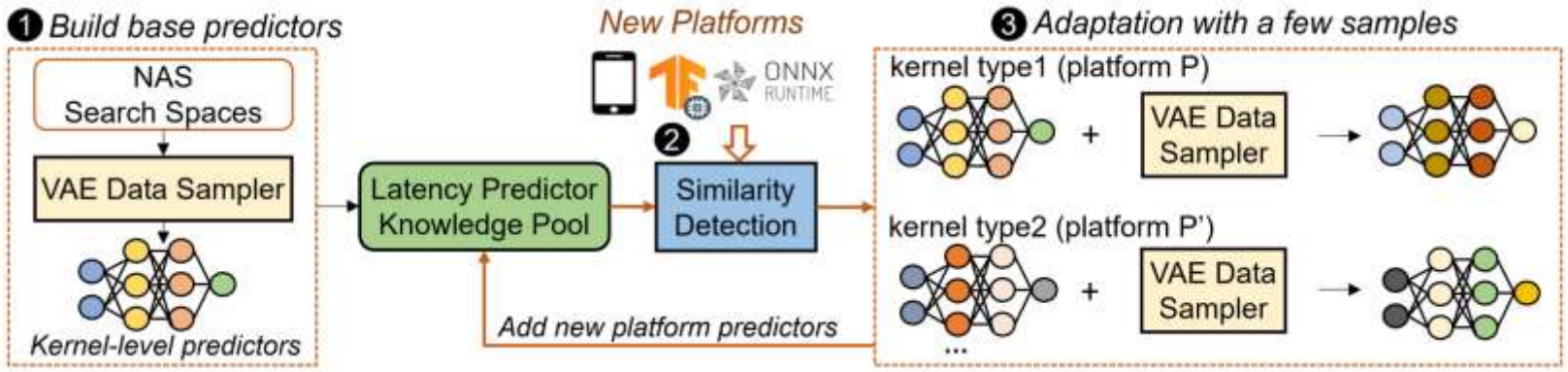
Hard to generalize to unseen devices and unseen models

Poor prediction accuracy on new platforms & Expensive rebuilding cost

System overview of LitePred

Key idea of LitePred： identify the most similar pre-existing latency predictors for each kernel on new platform, and then finetune them with just a few adaptation samples to achieve high prediction accuracy.

1. Build accurate base predictors from scratch as warmup ones
2. Perform similarity detection to identify most similar latency predictor for each kernel type.
3. Swiftly finetune with a few adaptation data

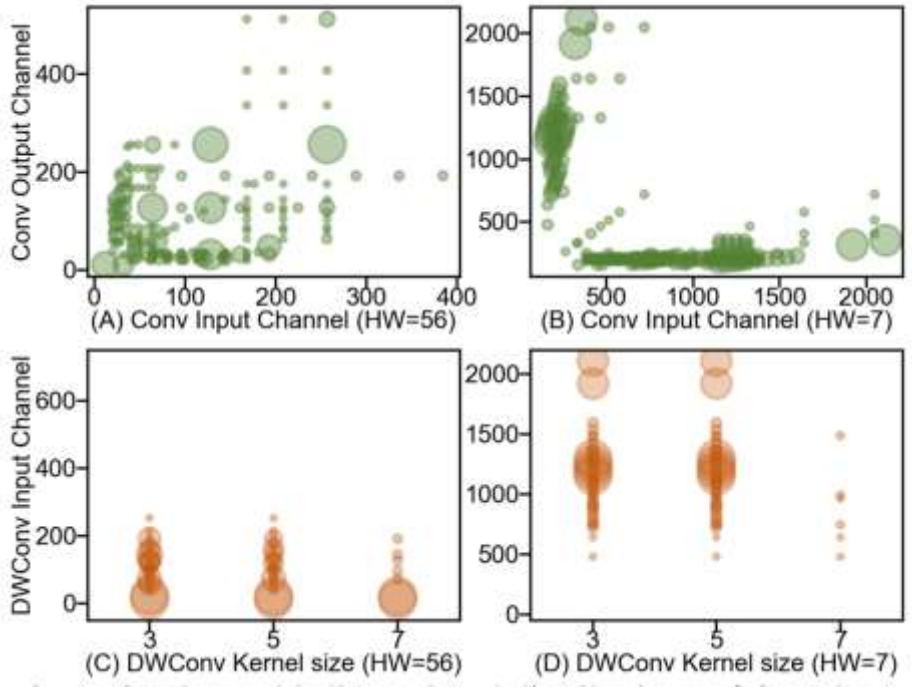


Predict arbitrary DNN models with Minimal adaptation cost

Challenges

- **Effective data collection**
  - Latency-dominant kernels exhibit multi-dimensional joint distribution.

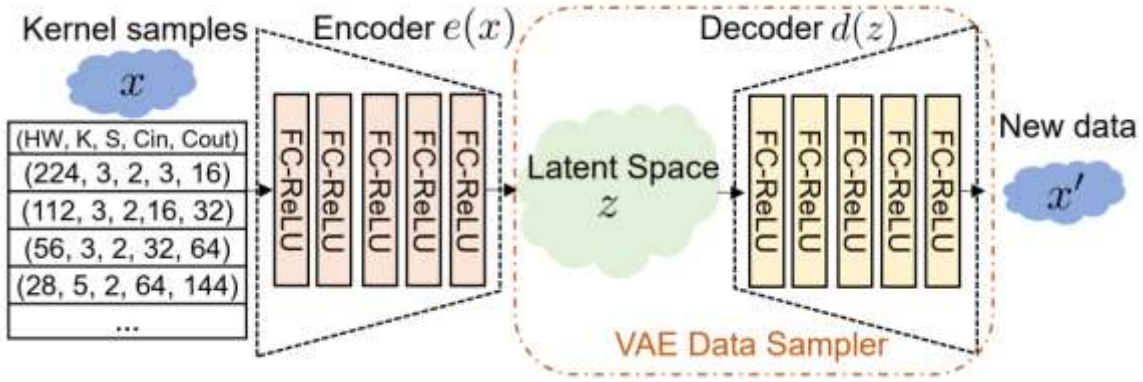
- **Similar predictor detection for each kernel**
  - Most edge platforms are black boxes



Build Accurate Base Latency Predictors

Key Tech #1 Efficient VAE data sampler

- **Collect data**
  - Latency-dominated configurations have low frequencies in NAS search spaces
  - Performing data normalization on latency-dominated kernels
- **Train model**
  - Mean Squared Error (MSE) reconstruction loss between x’ and x
  - Kullback-Leibler (KL) divergence loss between encoder output and distribution and standard normal distribution
- **Generate kernel configuration**
  - Sample N vectors from multivariate Gaussian distribution and pass them to decoder



Build Accurate Base Latency Predictors

Latency predictor model design

- A 16-layer Multilayer Perceptron (MLP) Network for every kernel
- Using configurations, FLOPs, and parameter size as prediction features
- Minimize the Mean Absolute Percentage Error (MAPE) loss

Platform	Method	Conv Acc.	DWConv Acc.
Xiaomi11 CPU	Adaptive data sampler	84.9%	52.8%
MindSpore	**VAE data sampler**	**91.4%**	**93.6%**
Xiaomi11 CPU	Adaptive data sampler	81.5%	95.5%
NCNN	**VAE data sampler**	**88.8%**	**98.3%**
Pixel5 GPU	Adaptive data sampler	61.6%	86.7%
TFLite 2.7	**VAE data sampler**	**76.7%**	**89.1%**
Pixel5 GPU	Adaptive data sampler	65.6%	79.6%
NCNN	**VAE data sampler**	**87.1%**	**81.7%**

VAE data sampler VS state-of-the-art methods

VAE data sampler VS state-of-the-art methods

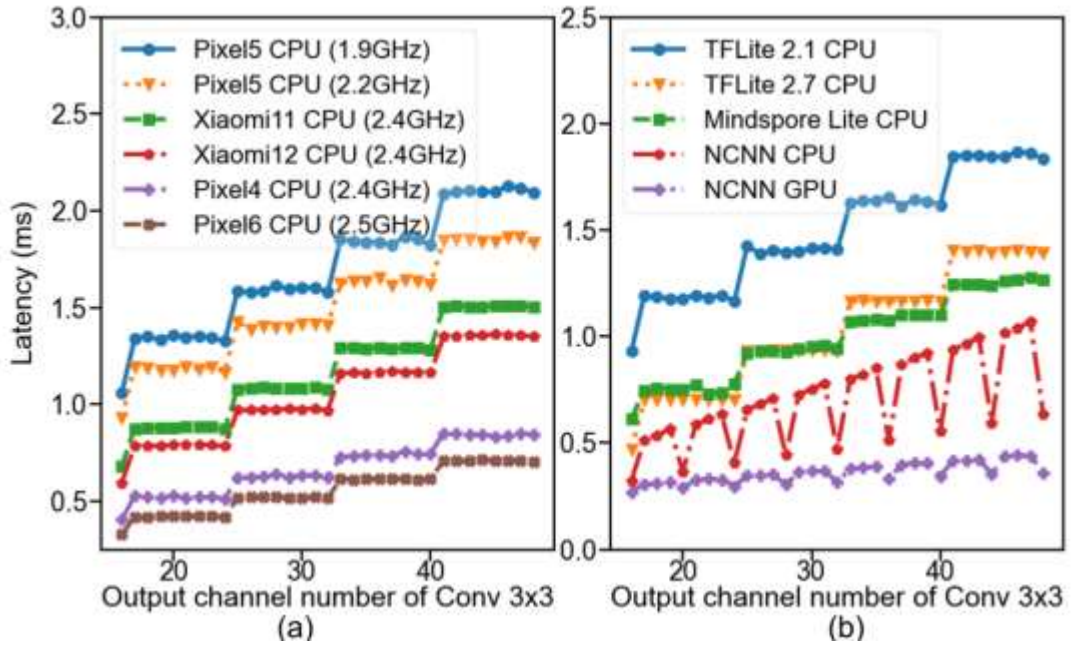
Transfer Predictors to New Platforms

Core principle of litepred :

Knowledge from a pre-existing latency predictor for one platform can be transferred to new platforms that share similarities.

(i) Mobile hardware of the same type exhibit similar latency patterns.

(ii) Despite varying optimizations and implementations, there exist inference frameworks display similar latency patterns.



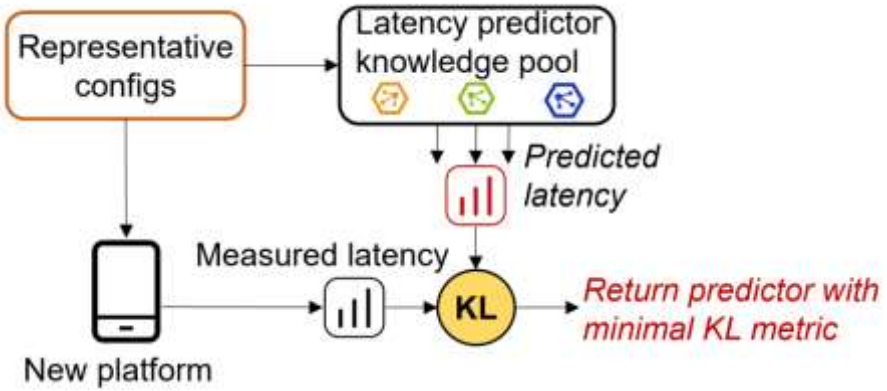
Mobiles CPUS under the same TFLite 2.1

Xiaomi 11 under various frameworks

Transfer Predictors to New Platforms

Key Tech #2 Similar platform detection

- **latency-distribution based similarity detection**
  - A pre-existing kernel latency predictor is considered similar for the target platform if its predicted latency displays a similar distribution to the real latency
- **Representative configs**
  - Configurations in the search spaces that reflect the underlying distribution
  - Designed configurations that capture the latency patterns
- **Similar score**
  - Kullback-Leibler (KL) divergence



$$D_{KL}(P||Q) = \sum_{y_r \in Y_r, y_p \in Y_p} P(y_p) \log \frac{P(y_p)}{Q(y_r)}$$

Y_r : real kernel latency

Y_p : predicted latencies

Evaluation

Experiment Setup

- **Platforms**
  - 10 different hardware and CPU frequencies, 5 popular inference frameworks on edge and 2 data precision, totally 85 platforms
- **Evaluation datasets**
  - 5 widely-used CNN and 1 vision transformer NAS search space
  - 1.86 million model and latency pairs (4k models from each search space on 85 platforms)

Device	CPU	GPU	CPU Frequency
Pixel 4	Qualcomm Snapdragon 855	Adreno 640	2.4GHz, 2.1GHz
Pixel 5	Qualcomm Snapdragon 765G	Adreno 620	2.2GHz, 1.9GHz
Pixel 6	Google Tensor SoC	Mali-G78	2.5GHz, 2.2GHz
Xiaomi 11	Qualcomm Snapdragon 888	Adreno 660	2.4GHz, 2.1GHz
Xiaomi 12	Qualcomm Snapdragon 8 Gen 1	Adreno 730	2.4GHz, 2.1GHz
Inference engines	TFLite 2.1, TFLite 2.7, NCNN, Mindspore Lite, Onnxruntime		
Precision	FP32, INT8		

Evaluation

Transferable latency prediction of LitePred on diverse new platforms

- We achieve an average of **99.3%** transfer accuracy, with **87.0%** of models having prediction errors within a negligible **5%** margin on CNN models, with minimal adaptation cost (**0.05 to 1.73 hours**) on new platforms
- We achieve high transferable prediction accuracy on transformers

(a) Selecting most similar kernel predictors from the whole knowledge pool						
Platform	**Similar Platforms**		**#Adaptation Data**	**Time Cost**	**Prediction Accuracy ±5%Acc**	**±10%Acc**
	Conv kernel	DWConv kernel				
Xiaomi11CPU, ORT	Xiaomi12 CPU, ORT	Xiaomi12 CPU, ORT	1400	0.48h	90.5%	98.9%
Pixel5 GPU, NCNN	Xiaomi11 GPU, NCNN	Xiaomi11 GPU, NCNN	17400	0.96h	84.3%	99.1%
Xiaomi11 CPU, MindSpore	Pixel5 CPU, MindSpore	Xiaomi12 CPU, MindSpore	4800	0.35h	90.4%	99.9%
Xiaomi11 GPU, TFLite 2.7	Xiaomi12 GPU, TFLite 2.7	Xiaomi12 GPU, TFLite 2.7	11000	0.17h	83.7%	98.6%
Xiaomi11 CPU, NCNN	Xiaomi11 CPU, MindSpore	Pixel5 CPU, NCNN	11400	0.88h	80.3%	98.9%
Pixel6 CPU, TFLite 2.1	Xiaomi12 CPU, TFLite 2.1	Xiaomi12 CPU, TFLite 2.1	3500	0.16h	79.4%	100%
Pixel5 CPU, TFLite 2.7	Xiaomi11 CPU, TFLite 2.7	Xiaomi11 CPU, TFLite 2.7	3400	0.13h	79.6%	99.2%
Xiaomi12 CPU, TFLite 2.7, INT8	Xiaomi11 CPU, ORT	Pixel5 GPU, TFLite 2.7	3100	0.05h	95.7%	100%
(b) Similarity detection of kernel predictors *Excluding* same inference frameworks						
Xiaomi11 CPU, ORT	Pixel5 CPU, MindSpore	Pixel5 GPU, NCNN	2400	0.72h	84.2%	99.2%
Xiaomi12 GPU, TFLite 2.7	Pixel5 GPU, NCNN	Xiaomi12 CPU, MindSpore	16100	0.22h	79.4%	98.7%
Xiaomi11 CPU, Mindspore	Pixel5CPU, TFLite 2.7	Pixel5GPU, TFLite 2.7	9700	0.80h	98.1%	99.2%
Pixel5 GPU, NCNN	Xiaomi12 CPU, TFLite 2.1	Xiaomi11 CPU, ORT	18500	1.73h	86.5%	99.3%
Xiaomi12 CPU, TFLite 2.1, low Freq	Xiaomi11 GPU, NCNN	Pixel5 CPU, MindSpore	1800	0.18h	94.7%	100%
Xiaomi12 CPU, TFLite 2.1	Pixel4 CPU, TFLite 2.7	Pixel5 CPU, MindSpore	1800	0.10h	97.6%	99.9%

LitePred prediction on 5 CNN spaces

Platform	Similar Platform	Time Cost	Prediction Acc ±5%	±10%
Xiaomi11 CPU TFLite 2.7	Xiaomi11 CPU TFLite 2.1	0.05h	100%	100%
Xiaomi12 CPU TFLite 2.1	Pixel5 CPU TFLite 2.7, LowFreq	0.08h	83.9%	99.9%
Xiaomi12 CPU TFLite 2.7, INT8	Pixel5 CPU TFLite 2.7, LowFreq	0.02h	41.4%	99.9%

LitePred prediction on vision transformer space

Evaluation

Comparison with baseline methods

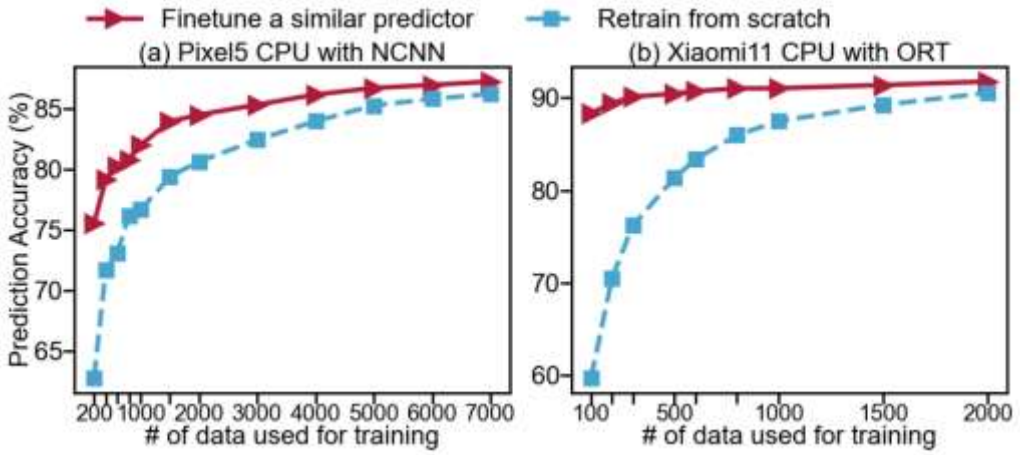
- LitePred outperforms both state-of-the-art platform-specific and platform-agnostic baselines by achieving up to **5.3%** higher prediction accuracy with a significant **50.6×** profiling cost reduction

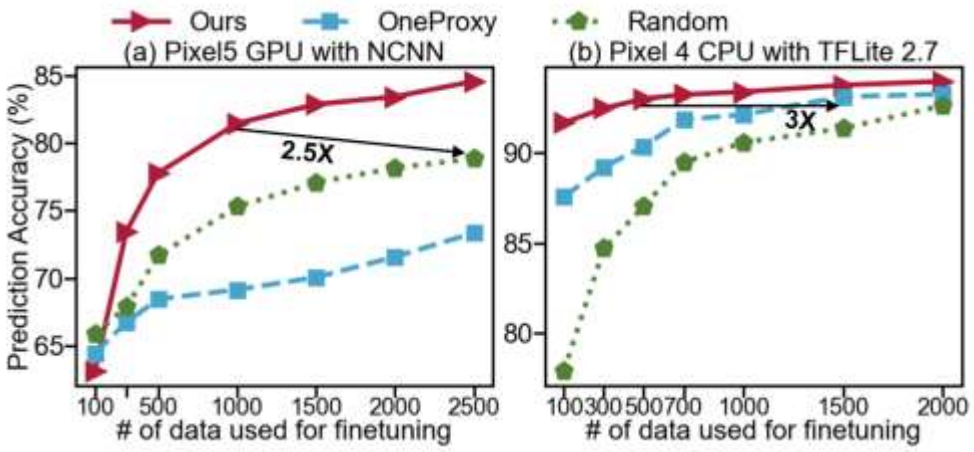
Platform	Method	Train Data	Cost	RMSE	Prediction Acc ±5%	Prediction Acc ±10%
Xiaomi11 CPU Mindspore	HELP	10 models	12.44s	6.6 ms	11.5%	22.5%
	HELP*	1030 models	0.35h	4.1 ms	39.3%	48.7%
	nn-Meter	234997 kernels	16.23h	0.8 ms	78.0%	98.9%
	Ours	4800 kernels	**0.35h**	**0.4 ms**	**95.4%**	**100%**
Xiaomi11 CPU NCNN	HELP	10 models	10.87s	9.5 ms	15.4%	23.0%
	HELP*	3000 models	0.88h	6.7 ms	37.1%	49.1%
	nn-Meter	169305 kernels	20.17h	0.4 ms	96.4%	100%
	Ours	11400 kernels	**0.88h**	**0.3 ms**	**99.5%**	**100%**
Pixel 5 GPU TFLite 2.7	HELP	10 models	2.66s	1.2 ms	13.9%	28.0%
	HELP*	2500 models	0.62h	0.8 ms	51.6%	61.1 %
	nn-Meter	104996 kernels	7.94h	0.8 ms	37.7%	95.8%
	Ours	11900 kernels	**0.62h**	**0.3 ms**	**95.9%**	**99.9%**
Pixel 5 GPU NCNN	HELP	10 models	16.41s	12 ms	7.9%	16.8%
	HELP*	2100 models	0.96h	7.5 ms	33.5%	50.8 %
	nn-Meter	397384 kernels	48.60h	1.6 ms	52.2%	94.7%
	Ours	17400 kernels	**0.96h**	**0.9 ms**	**92.6%**	**100%**

Evaluation

The effectiveness of our similarity detection technique

- Finetuning a pre-trained predictor from similar platform yields higher accuracy than training a new predictor from scratch on the same platform
- By our similarity detection, we achieve higher accuracy with **2.5×** less adaptation data than baseline method





Evaluation

Hardware-aware NAS with LitePred

- By integrating LitePred into OFA, we achieve up to **4.4%** accuracy compared to MobileNets across various edge platforms

Method	Pixel6 CPU, TFLite 2.1		Xiaomi11 CPU, Mindspore		Pixel5 GPU, NCNN		Xiaomi11 GPU, TFLite 2.7	
	Top1 Acc. ↑	Latency ↓	Top1 Acc. ↑	Latency ↓	Top1 Acc. ↑	Latency ↓	Top1 Acc. ↑	Latency ↓
MobileNetV2	72.0	45.9ms	72.0	22.7ms	72.0	31.3ms	72.0	5.0 ms
OFA [8] + LitePred	**76.4**	**44.4 ms**	**73.4**	**21.8 ms**	**75.3**	**31.1 ms**	**75.8**	**5.0 ms**
MobileNetV3× 0.75	73.3	29.7ms	73.3	24.4ms	73.3	34.4ms	73.3	4.0 ms
OFA [8] + LitePred	**74.8**	**29.6 ms**	**74.5**	**23.9 ms**	**76.0**	**34.3 ms**	**74.4**	**3.9 ms**
MobileNetV3	75.2	37.2ms	75.2	33.4ms	75.2	30.3ms	75.2	4.7ms
OFA [8] + LitePred	**75.5**	**36.8 ms**	**75.6**	**33.2 ms**	**75.6**	**29.8 ms**	**75.5**	**4.6 ms**

Summary

- **LitePred: a lightweight transferrable approach for accurately predicting DNN inference latency**
  - Principle: knowledge from a pre-existing latency predictor for one platform can be transferred to new platforms that share similarities
  - Key Tech #1: Efficient VAE data sampler
  - Key Tech #2: Similar platform detection
- **Extensive experiments on 85 edge platforms and 6 NAS search spaces**
- **Impressive results**
  - LitePred achieves an average latency prediction accuracy of **99.3%** with less than an hour of adaptation cost
  - LitePred achieves up to **5.3%** higher accuracy with a significant **50.6×** reduction in profiling cost
  - By integrating LitePred with NAS, achieving an impressive up to **4.4%** higher accuracy on ImageNet.