Transfer Learning for Performance Modeling of Deep Neural Network Systems



Md Shahriar Iqbal

UNIVERSITY OF SOUTH CAROLINA



Lars Kotthoff

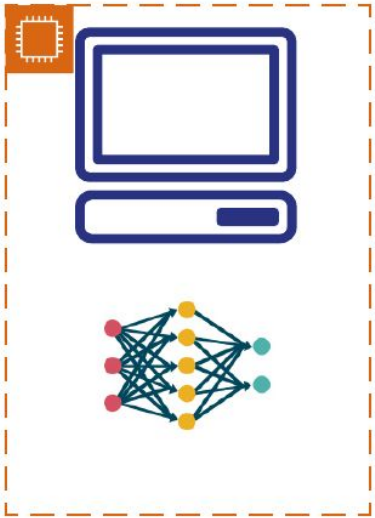
UNIVERSITY OF WYOMING



Pooyan Jamshidi

UNIVERSITY OF SOUTH CAROLINA

Performance can change across systems



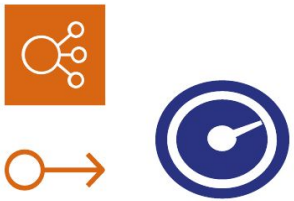
Environment 1.0
Dev/Staging



Push



Environment 2.0
Production

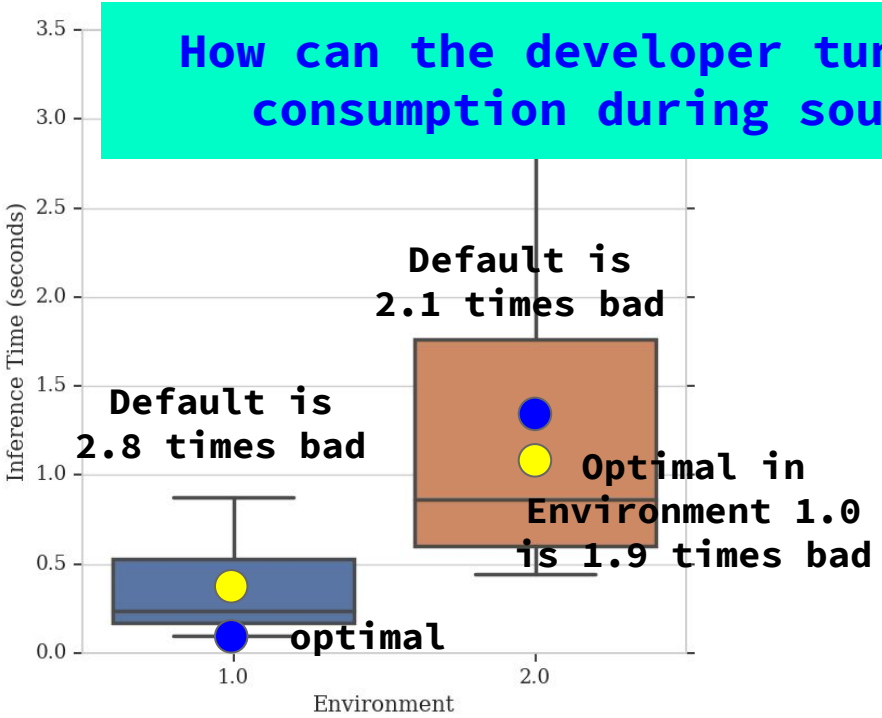


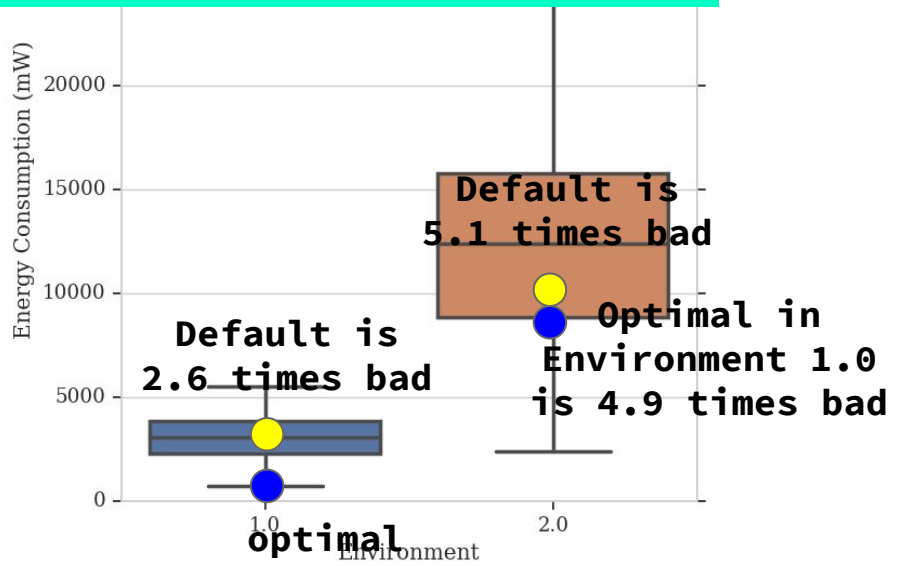
Performance
- Inference time
- Energy consumption

Can we expect similar performance in new environment?

Case study: Performance behavior changes across environments and so does optimal configuration

How can the developer tune parameters for energy consumption during source code development?





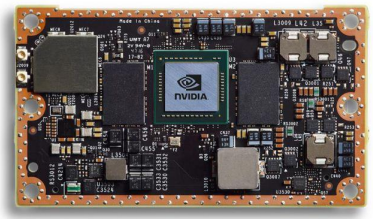
DNN system stack allows developers to tune parameters for energy consumption

Network Level

Model Compiler Level

Deployment Level

Hardware/OS Level



core status, core frequency,
gpu status, gpu frequency,
memory controller frequency etc

>100

2^{100} configurations possible

DNN Systems are highly configurable

Building configuration space of DNN systems

Hardware/OS Level

Core status Core status gpu frequency

$$C = O_1 \times O_2 \times O_3 \times O_4 \times \ldots \times O_n$$

Core frequency gpu status

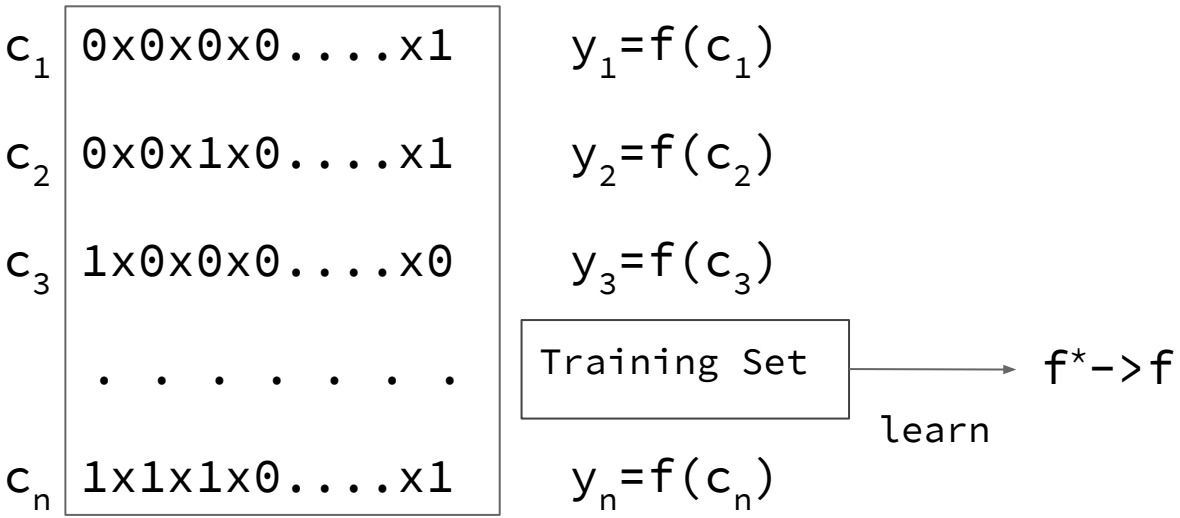
$$c_1 \in C \quad 0 \times 1 \times 204000 \times 1 \times \ldots \times 4996000$$

Inference time $f_c(c_1) = 36.12ms$

Energy consumption $f_c(c_1) = 1456.98mW$

A typical approach to understand system behavior is sensitivity analysis

C $O_1xO_2xO_3xO_{4....}xO_{20}$

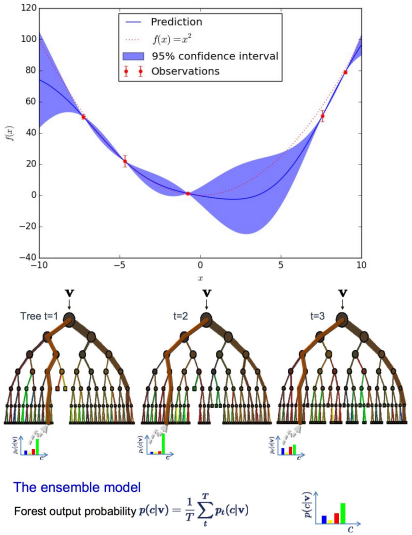


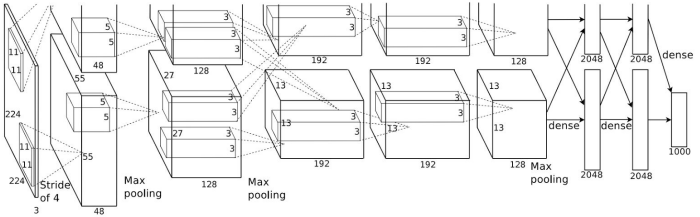
Performance models could be in any form of black box models

C $O_1xO_2xO_3xO_4....xO_{20}$

c_1	0x0x0x0....x1	$y_1=f(c_1)$
c_2	0x0x1x0....x1	$y_2=f(c_2)$
c_3	1x0x0x0....x0	$y_3=f(c_3)$
	. . . . .	
c_n	1x1x1x0....x1	$y_n=f(c_n)$

Training Set → $f^*->f$





Evaluating a performance model

C $O_1 x O_2 x O_3 x O_{4....} x O_{20}$

c_1	0x0x0x0....x1	$y_1=f(c_1)$
c_2	0x0x1x0....x1	$y_2=f(c_2)$
c_3	1x0x0x0....x0	$y_3=f(c_3)$
	. . . . .	Training Set
c_n	1x1x1x0....x1	$y_n=f(c_n)$

→ MAPE(f*,f)=$(|f*-f|/f)\times 100$

→ f*->f

A Performance Model contains useful information about option interactions

$f:C->R$

$$f(.)=1.2+30_1+50_2+70_7+80_1O_2-3.40_1O_7+50_3O_7+11.20_1O_2O_7$$

What are the roadblocks for performance modeling of DNN systems?

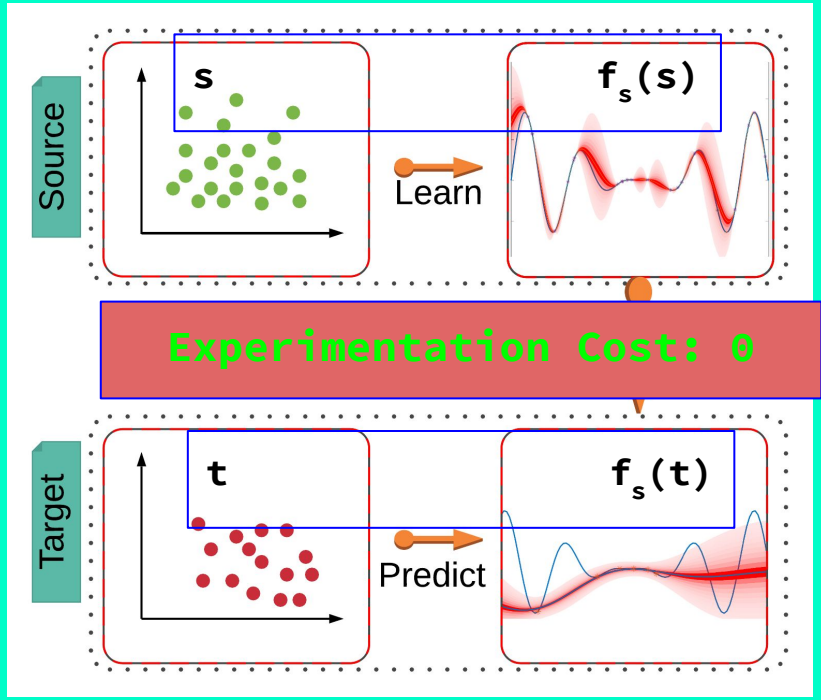
- Configuration space grows exponentially with addition of new options.
- Performance measurements are costly.

  2^{20}=1048576 configurations takes nearly 2 years for measurement considering only 60s on average per configuration.

Transfer learning can help to reuse the cheap measurements from old environments to new environments to quickly learn system behavior using performance modeling.

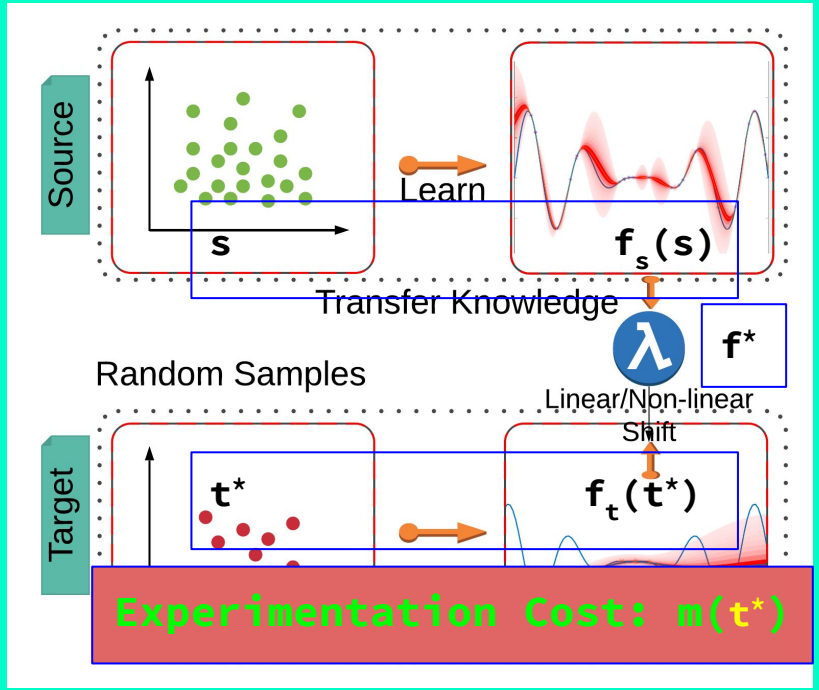
Direct Model Transfer Learning

Model learned in source environment is used directly in target environment



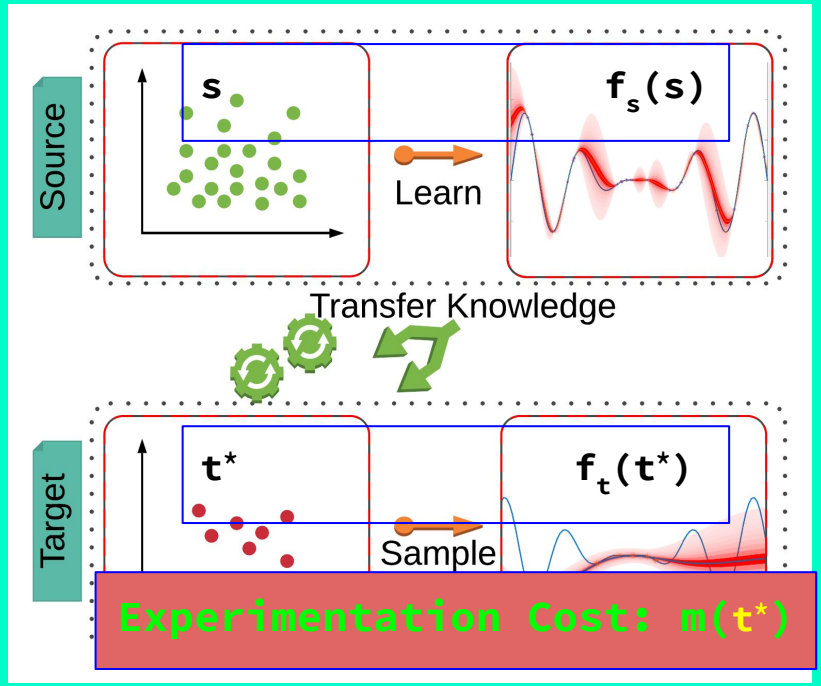
Linear/Non-Linear Model Shift Transfer Learning

Linear and non-linear model shift are learned using source model (all samples) and target model (random sub-samples).

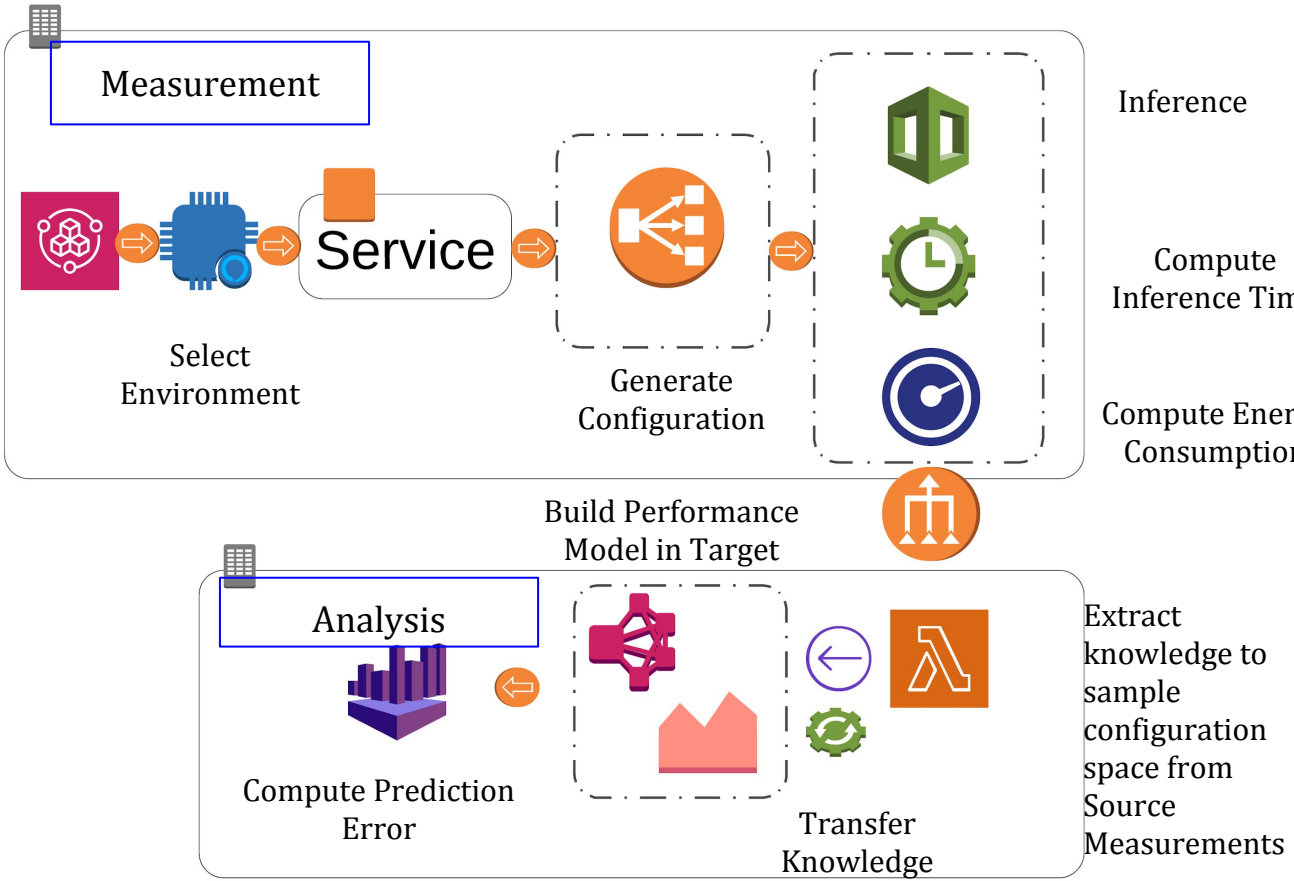


Guided Sampling Transfer Learning

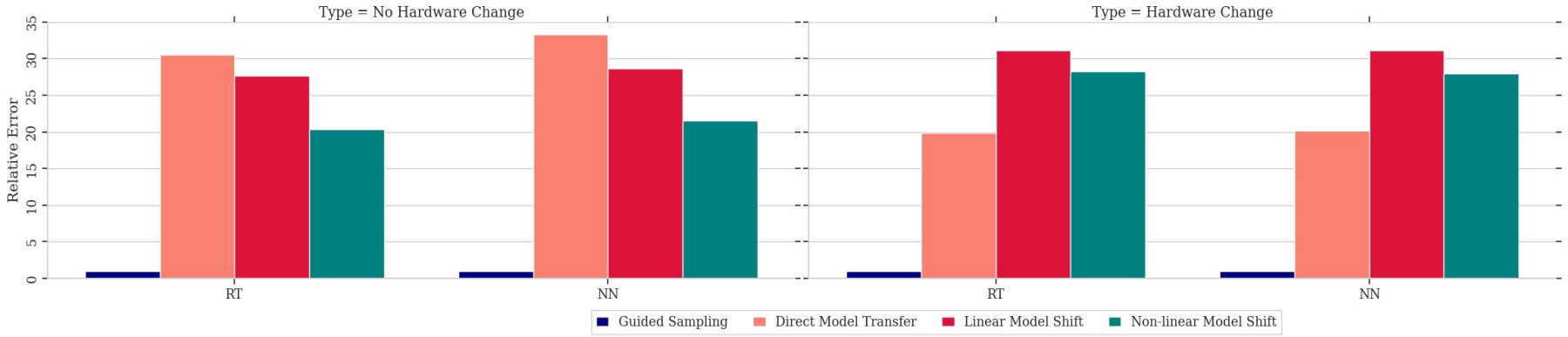
Influential configurations are sampled in target environment by studying options interactions from source environment to build performance model.



Experimental Setup

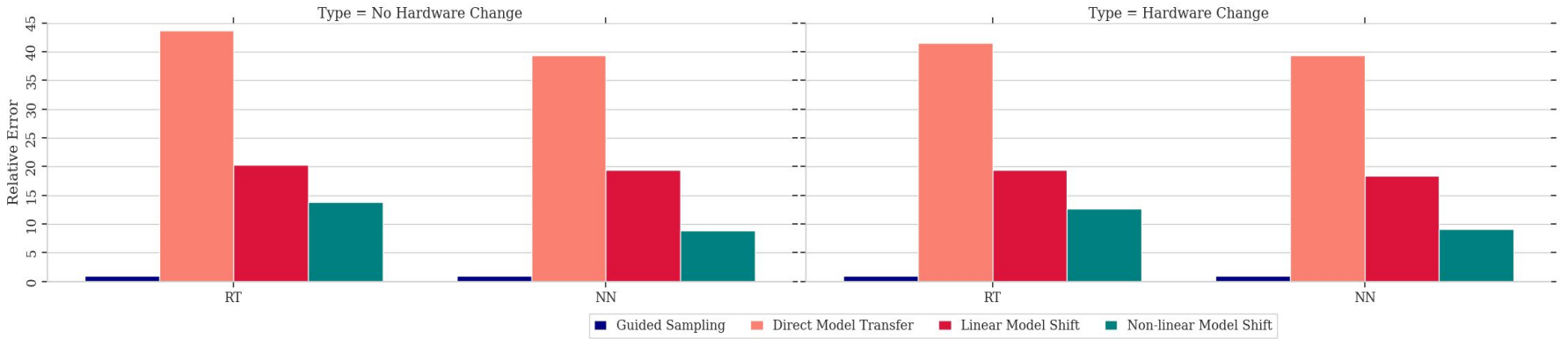


Results: Inference Time



	Cost	Reduction of Error
Direct Model Transfer	0	1
Linear Model Shift	2.48 hours (0.15%)	3.48
Non-Linear Model Shift	2.48 hours (0.15%)	5.76
Guided Sampling	2.48 hours (0.15%)	22.93

Results: Energy Consumption



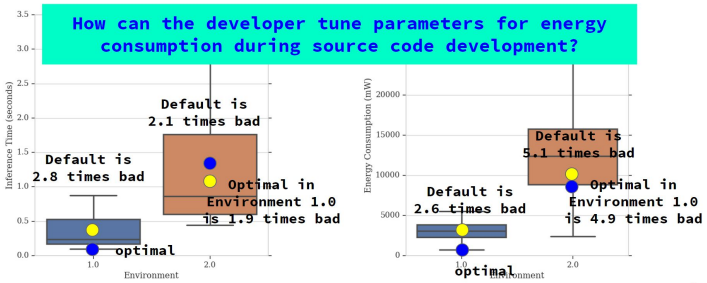
	Cost	Reduction of Error
Direct Model Transfer	0	1
Linear Model Shift	2.48 hours (0.15%)	7.19
Non-Linear Model Shift	2.48 hours (0.15%)	16.31
Guided sampling	2.48 hours (0.15%)	42.85

Useful tool for practitioners

- To build performance models in new environments using only 2.44% of the configuration space.
- To avoid invalid configurations to quickly find optimal configurations.
- To learn the performance landscape of a system for performance debugging e.g., performance regression etc.

Summary

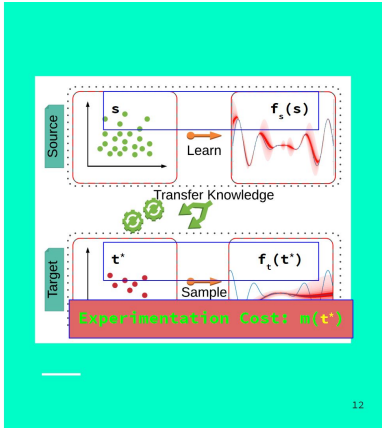
Case study: Performance behavior changes across environments and so does optimal configuration



2

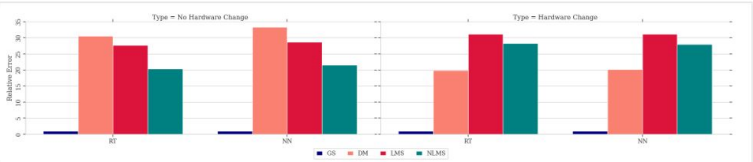
Guided Sampling Transfer Learning

Influential configurations are sampled in target environment by studying options interactions from source environment to build performance model.



12

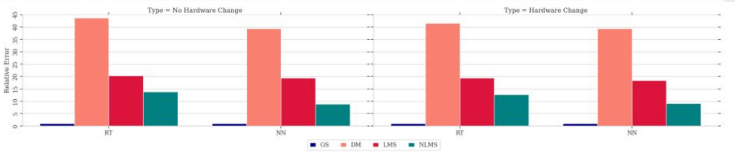
Results: Inference Time



	Cost	Performance boost
Direct Model Transfer	0	1
Linear Model Shift	2.48 hours (0.15%)	3.48
Non-Linear Model Shift	2.48 hours (0.15%)	5.76
Guided Sampling	2.48 hours (0.15%)	22.93

12

Results: Energy Consumption



	Cost	Performance Boost
Direct Model Transfer	0	1
Linear Model Shift	2.48 hours (0.15%)	7.19
Non-Linear Model Shift	2.48 hours (0.15%)	16.31
Guided sampling	2.48 hours (0.15%)	42.85

13