ModelX: A Novel Transfer Learning Approach Across Heterogeneous Datasets

Arunavo Dey[1], Neil Anthony[3], Aakash Dhakal[1], Jayaram Thigarajan[2], Jae-Seung Yeom[2], Tapasya Patki[2], Tom Scogland[2], Tanzima Z. Islam[1]

[1]Texas State University
[2]Lawrence Livermore National Laboratory
[3]University of California Santa Barbara

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and was supported by the LLNL-LDRD Program under Project No. 24-SI-005 (LLNL-CFPRES-xxxxx).

The Evolution of Transfer Learning

1 — 1996

Foundational Work: Tom Mitchell's [1] "Machine Learning"

2 — Early 2000s

Rain et. al introduced sparse coding-based unsupervised pretraining for transfer learning. Ben-Davis et.al provided VC-theory based generalization bounds for domain adaptation — foundational for theoretical transfer learning.

3 — 2018-2020

NLP Breakthroughs: BERT [3], GPT [4], RoBERTa [5] and other transformer models revolutionize natural language processing, demonstrating immense transferability across diverse text tasks.

4 — 2023

Levin et. al [X] proposed transfer learning across tabular domains with feature mismatch leveraging transfer learning requiring a feature overlap between the domains.

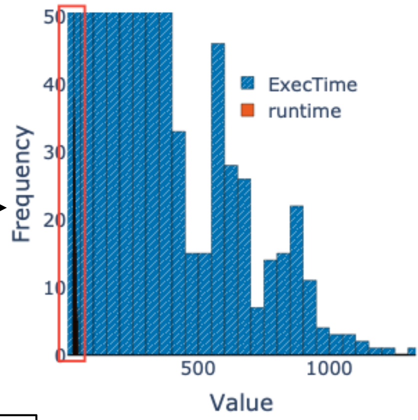
5 — 2025

Tabular Data: Shourav B. Rabbani's [X] "Transfer Learning of Tabular Data by Finetuning Large Language Models" pushes boundaries, extending transfer learning to complex tabular datasets.

6 — 2025

ModelX: Transfer learning across fully heterogeneous domains (This work)

Where Transfer Learning Excels and Falters



- Similar data distributions across datasets
- Need a lot of data for training supervised models
- Assumes overlap between feature names

Similar Domains

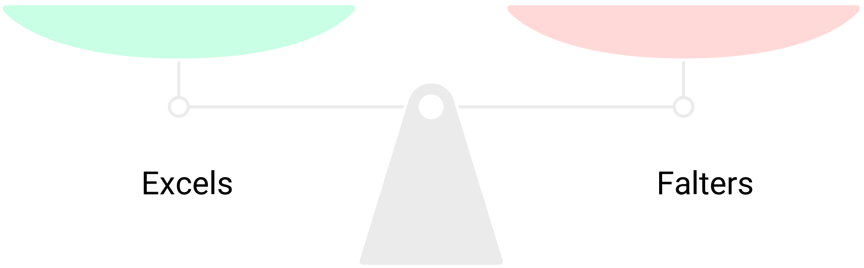
Abundant Source Data

Feature Overlap

Domain Divergence

Data Scarcity

Disjoint feature sets



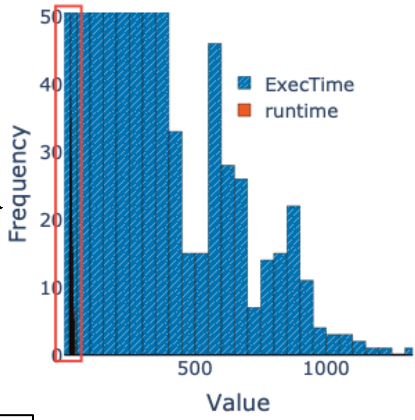
Excels

Falters

SOTA cannot do it!

Where Transfer Learning Excels and Falters

- Similar data distributions across datasets — Similar Domains — Domain Divergence
- Need a lot of data for training supervised models — Abundant Source Data — Data Scarcity
- Assumes overlap between feature names — Feature Overlap — Disjoint feature sets



Data and structural heterogeneity could be caused by

CoMD on Intel Xeon E5-2695 v2

1	application	algorithm	bw_level	ipath_0	ipath_1	task_count	power_cap	runtime	thread_count
2	CoMD	pak	1	2	0	4096	64	54.4034	24
3	CoMD	pak	1	2	0	4096	80	46.5075	24
4	CoMD	pak	1	2	0	4096	115	42.4216	24
5	CoMD	pak	1	2	0	2048	64	41.7099	16

Kripke on Intel Xeon E5-2695 v2

	DRAMPowerPerNode	ProcessorPowerPerNode	Ranks	Nesting	Dset	Gset	OMP	PKG_LIMIT	AvgInst	AvgIpc	AvgArithFpu	AvgFreq	AvgTemp	ProcessorPower	DRAMPower	ExecTime
0	31.572256	86.904462	32	DGZ	8	1	4	50	6.829296e+10	3.7360	6.248262e+08	2.1110	34.2837	1390.4714	505.1561	7.6394
1	32.813519	95.389194	32	DGZ	8	1	4	55	6.088132e+10	4.2230	6.212548e+08	2.3428	33.1925	1526.2271	525.0163	6.0149
2	33.178587	103.230106	32	DGZ	8	1	4	60	5.925558e+10	4.5708	6.208049e+08	2.5125	35.6479	1652.4817	530.8574	5.4116
3	33.871169	113.629712	32	DGZ	8	1	4	65	5.633770e+10	4.8950	6.194195e+08	2.6936	35.8681	1818.0754	541.9387	4.8056

Agenda

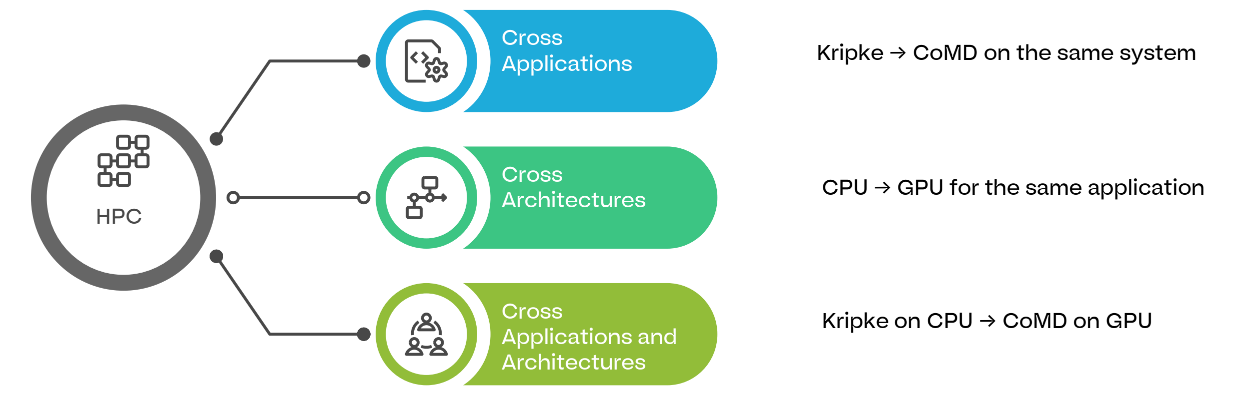
- Problem and motivation
- HPC scenarios & challenges
- Methodology
- Evaluations
- Application of *ModelX* for job scheduling
- Conclusions & Future work

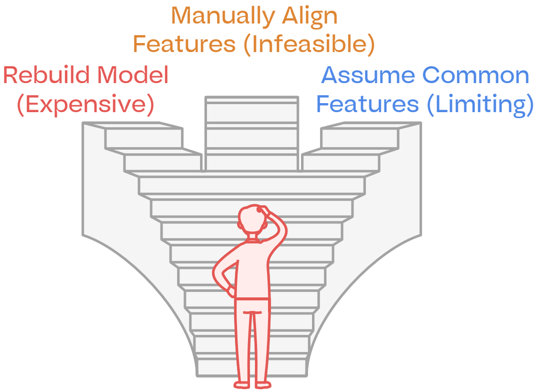
Transfer Learning Scenarios in HPC are Challenging Due to Heterogeneity

Challenges:

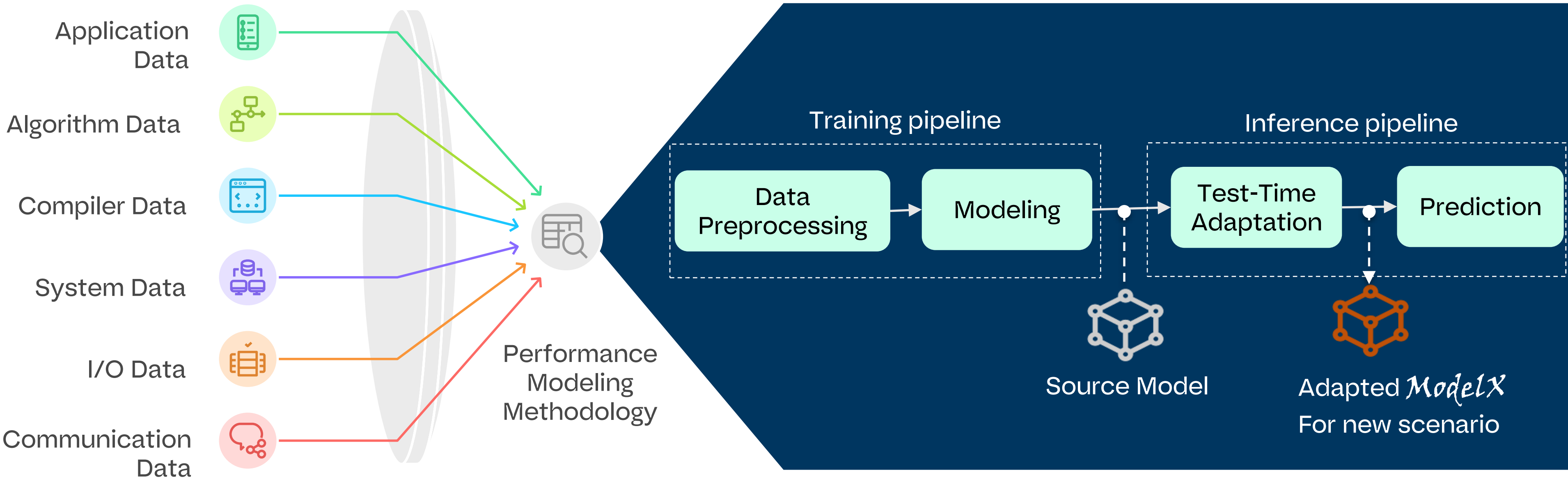
- Domain divergence: Data distribution shift across homogeneous datasets
- Data scarcity: Extensive data collection is expensive
- Feature heterogeneity (different names, counts, order of features)

How do *they* handle domain divergence, feature mismatch or disjoint feature sets today?





Performance Modeling Methodology using ML



- Assumption: All data sources during training are homogeneous.
- No assumption during inference time.

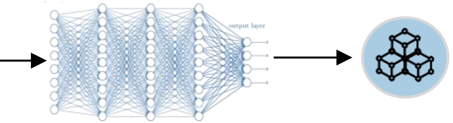
LLNL-CFPRES-xxxxx

Current Test-Time Adaptation Approaches

Zero-shot prediction

Train a source model and use it directly

X_{train}



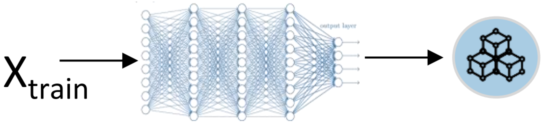
Current Test-Time Adaptation Approaches

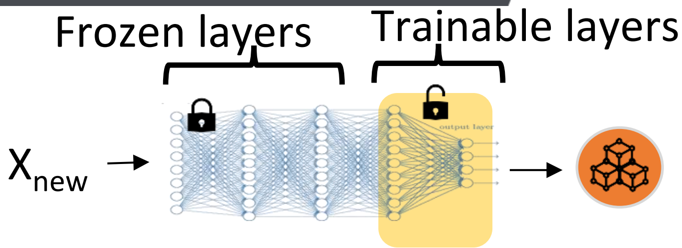
Zero-shot prediction

Train a source model and use it directly

Linear probing (few-shot adaptation)

Adapt a few layers of the source model using new samples





Current Test-Time Adaptation Approaches

Zero-shot prediction

Train a source model and use it directly

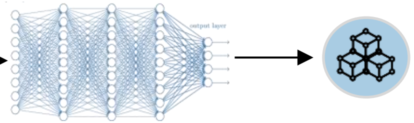
Linear probing (few-shot adaptation)

Adapt a few layers of the source model using new samples

Fine tuning (few-shot adaptation)

Adapt all layers of the source model using new samples

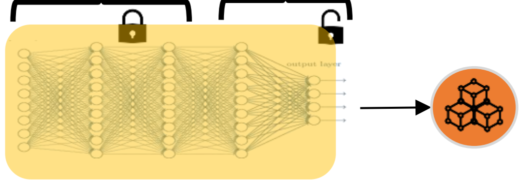
X_{train}



Frozen layers

Trainable layers

X_{new}



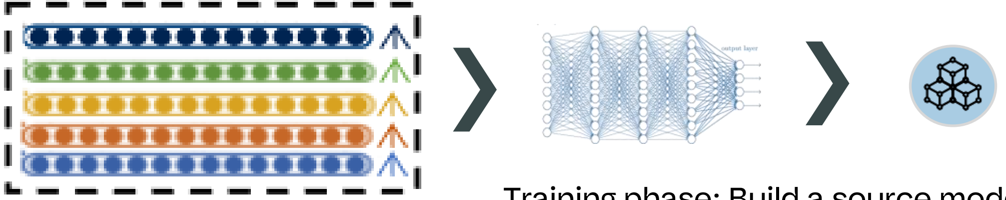
Proposed Solution: Bridging Heterogeneity During Inference Time

- Introducing a novel approach for robust performance prediction across diverse domains.

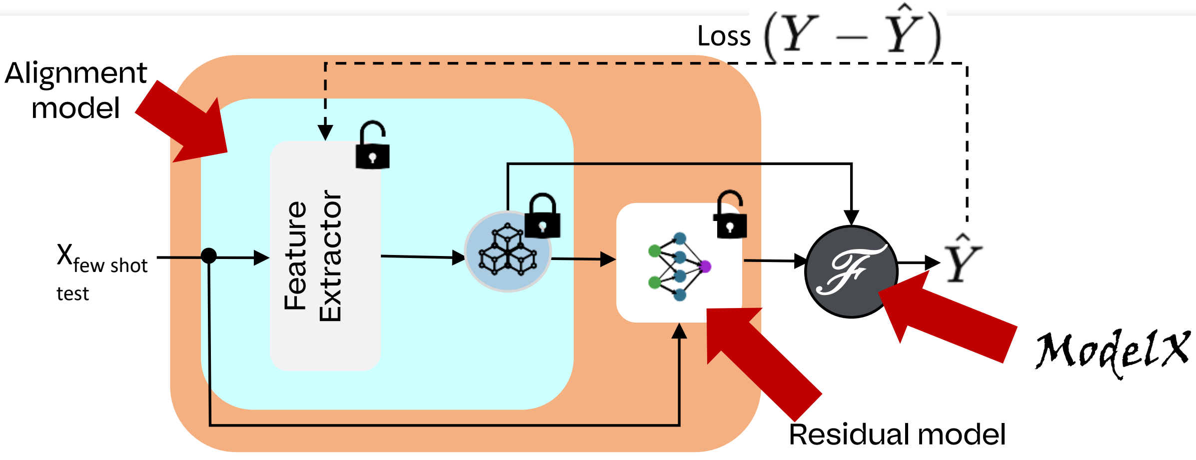
During training: Source model building using homogeneous datasets

During inference:

- Step 1: Train a feature extractor network (M_A) using new target few-shot samples using source model's prediction loss
- Step 2: Use just the new target few-shot samples to train an additional residual model M_R



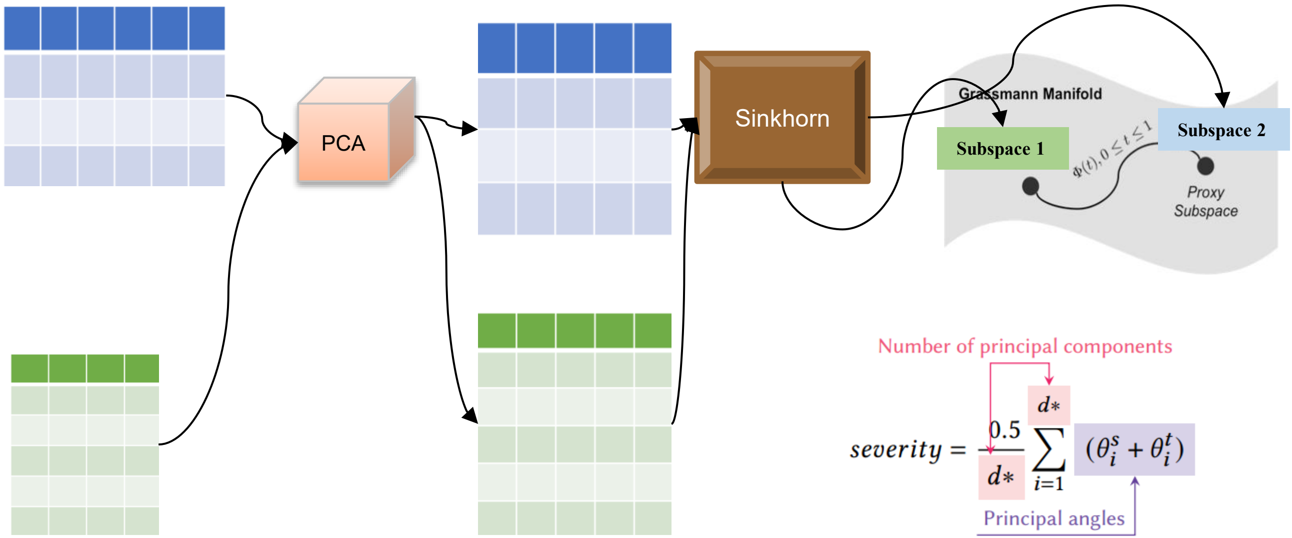
Training phase: Build a source model from one or more homogeneous source datasets



Inference phase: A few-shot test-time adaptation method that can learn from new samples that may have disjoint features, data divergence

Proposed Explainability Measure for Quantifying the Divergence between Datasets

- Design a distance measure to quantify the “difficulty” of transferring knowledge between two datasets
- This measure can explain why SOTA does not work, and when different components of our solution is necessary or sufficient



Number of principal components

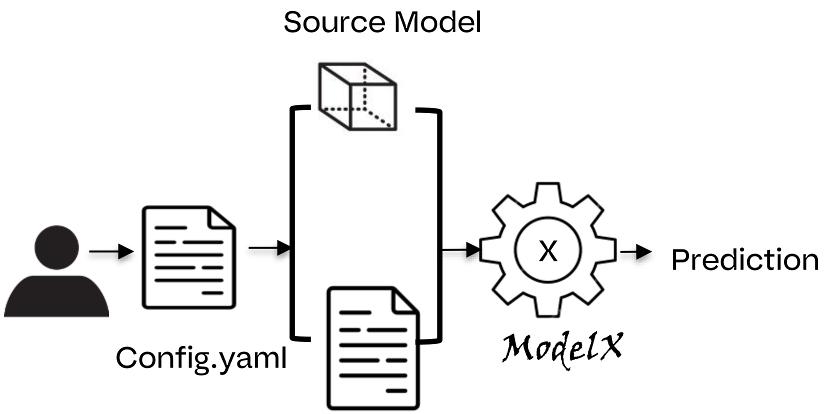
$$severity = \frac{0.5}{d*} \sum_{i=1}^{d*} (\theta_i^s + \theta_i^t)$$

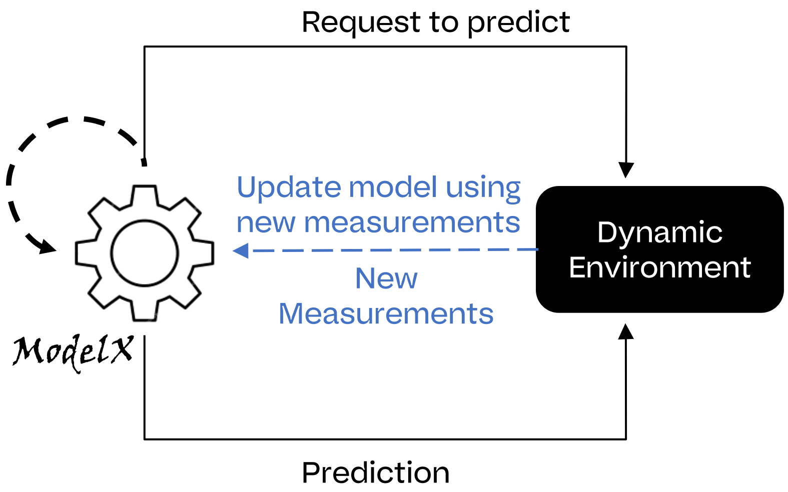
Principal angles

Distance between subspaces using Grassmanian manifold

UNFINISHED

ModelX Can be Use in both Online and Offline Scenarios





Scenario 1: ModelX can be used during offline scenarios

Scenario 2: ModelX can be used during online scenarios

Experimental Setup

Metrics	Mean Squared Error
Datasets	11 HPC and 4 machine learning datasets
Scenarios	Cross applications Cross architectures Online job scheduling

App.	Example Features
HPC Datasets	
CG, LU, FT, Kripke1, CoMD, N=3180, p=7	Power Cap, Task Count, Core Count, Placement, and Bandwidth, **runtime**
Kripke2, N=17386, p=23	DRAMPowerPerNode, ProcessorPowerPerNode, Ranks, App. specific input parameters, OMP, PKG_LIMIT, DRAM_LIMIT, AvgInst, AvgIpc, AvgArithFpu, AvgFreq, AvgTemp, ProcessorPower, DRAMPower, Nesting Order, **ExecTime**
Hypre, N=50396, p=21	DRAMPowerPerNode, ProcessorPowerPerNode, Ranks, OMP, PMX, NS, MU, AvgIPC, Smoother, AvgTSC, AvgTemp, ProcessorPower, DRAMPower, Solver-related parameters, **ExecTime**
XSBench and OpenMC on SB, N=200, p=121	NumThread, InputSize, **EfficiencyLoss**, perf::[MEM|DTLB|LLC]_[MISS|STALL], perf::[L_1|L_2|L_3|]_[MISS|STALL]
XSBench and OpenMC on BGQ, N=200, p=145	NumThread, InputSize, **Efficiency-Loss**, PEVT_[XU|AXU|L1P|STL]_[MISS|], PAPI_[BR|STL|SYC])_[STALL|CYC|MISS]
ML Datasets	
Airfoil, N=1503, p=5	Frequency, Angle of attack, Chord length, Free-stream velocity, Suction side, **Scaled sound pressure level**
NO2, N=500, p=8	**NO2**, Cars per hour, temperature, wind speed, temperature difference, wind direction, hour of day, day number
Crime, N=1994, p=127	population, householdsize, PctEmploy, PctIlleg, medIncome, perCapInc, PctPopUnderPov, **ViolentCrimesPerPop**
SkillCraft, N=3395, p=16	GameID, APM, SelectByHotkeys, AssignToHotkeys, MinimapRightClicks, NumberOfPACs, **ActionLatency**

ModelX Improves Prediction Accuracy Across Applications 93.5%

- Domain Divergence and Disjoint Features
- Overhead: Average test time adaptation 45.83s, average inference time 0.78s per query
- Number of features between 7 and 21
- For heterogeneous cases, ModelX has been compared against a supervised model with 100x data

Source	Target	Severity	Improvement	Winner	Best of Others
Airfoil	Airfoil	0	-82%	Source	Source
CoMD, CG,FT	Kripke[1]	0.5	56%	ModelX (Input Alignment)	Linear Probing
CoMD, FT. Kripke[1]	CG	0.65	69%	ModelX (Input Alignment)	Linear Probing
ComD, CG, Kripke[1]	FT	0.66	60%	ModelX (Input Alignment)	Fine Tuning
ComD, CG, LU, FT	Kripke[2]	0.60	95%	ModelX (Residual augmentation)	X
Kripke[2]	Hypre	0.70	99%	ModelX (Residual augmentation)	X
Hypre	Kripke[2]	0.70	99%	ModelX (Residual augmentation)	X

Homogeneous datasets

Heterogeneous

ModelX Improves Prediction Accuracy by 77% Across Architectures Compared to the Oracle using only 1-5% of the data

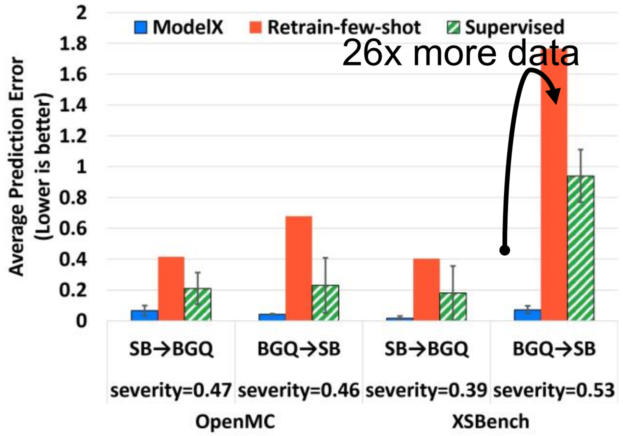
UNFINISHED

IBM BGQ – 143 features

B	C	E	F	G	H	J
PEVT_XU_BR_MISPRED_C	PEVT_LSU_ST_	PEVT_IU_AXU	PAPI_INT_INS	PEVT_LSU_LD_LA	PEVT_INST_QFP	PEVT_INST_XU
363275749.5	0	1.6723E+11	12205190201	11572	2449262430	2.5
181640446.8	6.625	8.3415E+10	6102393911	35687.5	1224932300	0.75
121043941.1	0	5.5493E+10	4065870710	18142.91667	816190208.8	0.166666667
90747775.13	488.9375	4.1512E+10	3048010431	3095.5625	611864963.6	0.125
73443709.1	542.3	3.3557E+10	2438308036	4484274.3	489484843.8	0.1
61696261.88	860.5416667	2.8135E+10	2032515138	6241053.542	408034358.4	0.083333333
53169009.32	1114.214286	2.4245E+10	1741726499	6890697.179	349656812.4	0.071428571

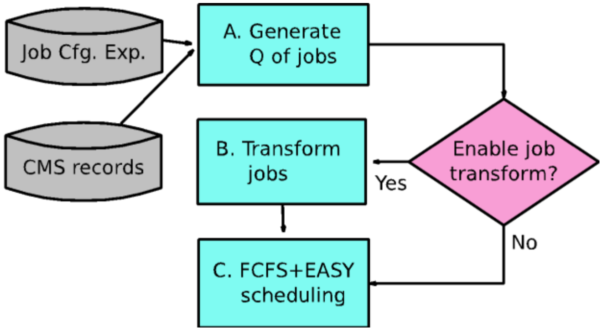
Intel Sandy Bridge – 121 features

B	C	D	E	F	G
MEM_LOAD_RETI	PAPI_L2_TCA	perf::INSTRUCTIONS	perf::NODE-STORI	perf::DTLB-STORE-M	INSTS_WRITTEN
2576406999	1577467923	24327658760	2	38137	103802174
1276757083	801386841	13069615784	5246383	10682.5	61142597
852924026.3	535450194.3	8700637454	2671912.333	4611.666667	56508432.67
635040848.8	401738887.3	6521345416	2300212	2898	40303665.5

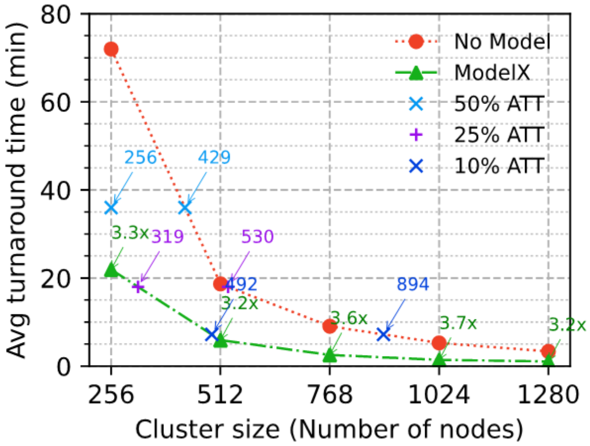


ModelX Reduces Average Turnaround Time by 71%

- Based on real-world job logs and performance measurement data from 6 HPC proxy applications
- Assumption: Jobs can run with a modified number of nodes than requested
- Scheduler asks *ModelX* to predict the execution time of a job using lesser number of nodes
- 3.4x time shorter turnaround time
- The state-of-the-practice scheduling method can perform as good by using up to 55% more nodes per job



- The Lassen job logs* collected over 2.5 year
- Extracted 70K jobs → 1-week's worth job
- Use the statistics of that week's job to create a stream of jobs



Add lassen's paper

Summary