Universität Bremen

AGRA

Fast and Accurate: Machine Learning Techniques for Performance Estimation of CNNs for GPGPUs

Christopher A. Metz[1], Mehran Goli[1], Rolf Drechsler[1,2]

[1]Institute of Computer Science, University of Bremen
[2]Cyber-Physical Systems, DFKI GmbH

Introduction

Machine Learning is now part of our daily life

It is used from embedded systems to supercomputers

Why Performance Estimation

Self-Driving Cars

Needs high performance

Fast executions

Needs powerful accelerator

Face recognition (e.g., Mobile Phone)

Needs low power

Do not need fast execution

Less powerful accelerator

Improve Design Process

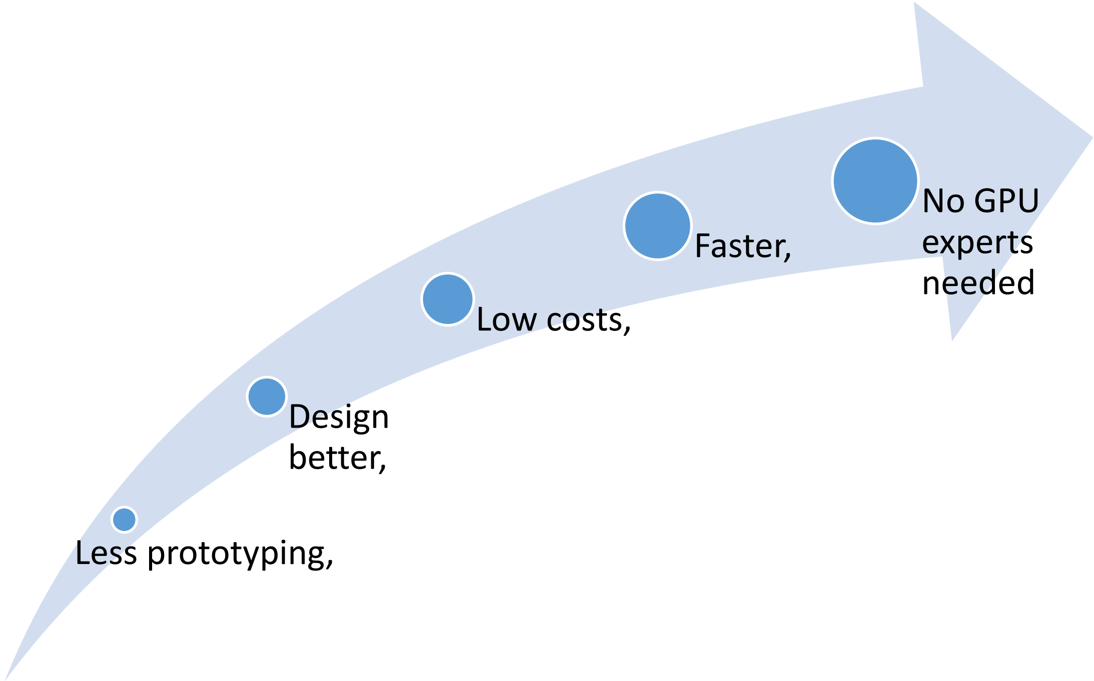
Classical Development

TIME INTENSIVE,

PRICY,

NEEDS GPU EXPERTS

Automatic Performance Estimation



State-of-the-Art Approaches

Need actual Device

- Use Performance Counter
- Performance Counter are not unified
- Profiling works on machine code

Static-Code-Analysis

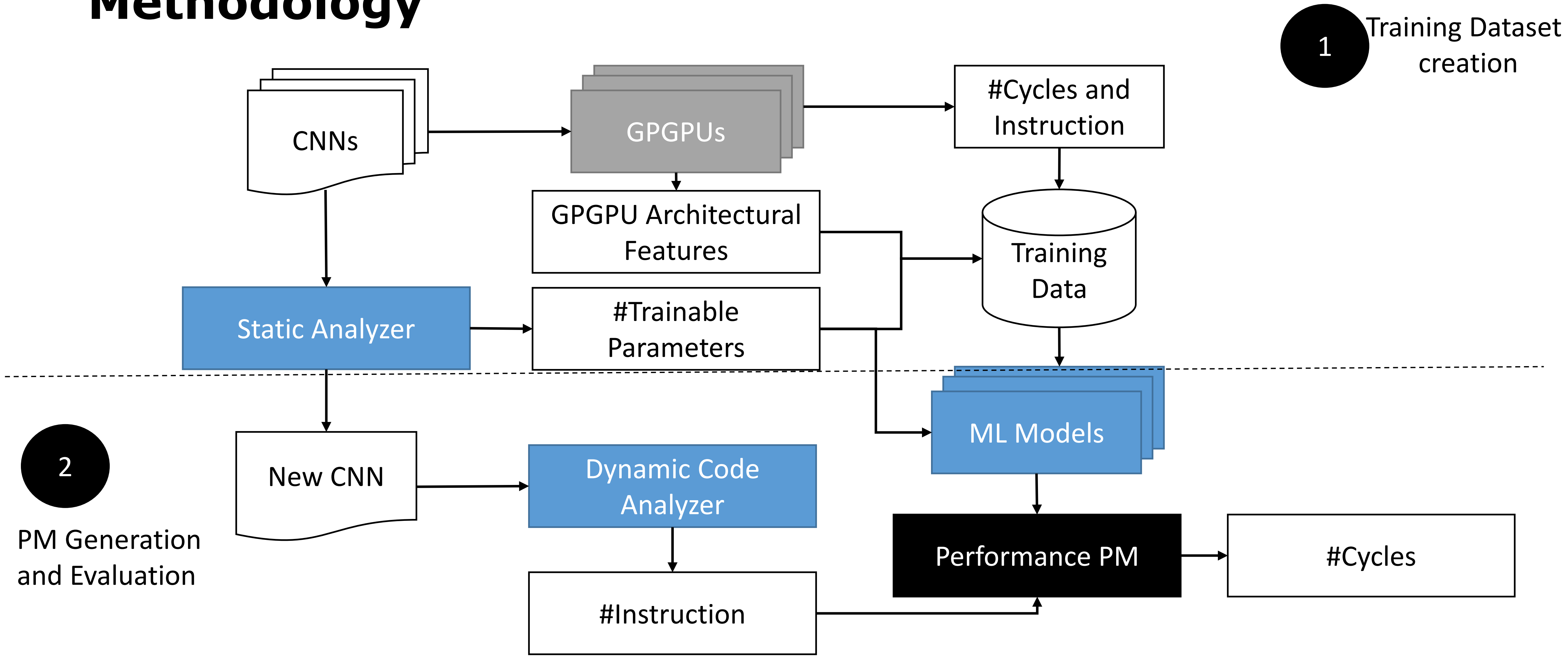
- Do not consider conditional jumps/branches
- Can over- or underestimate

Parallel Thread Execution

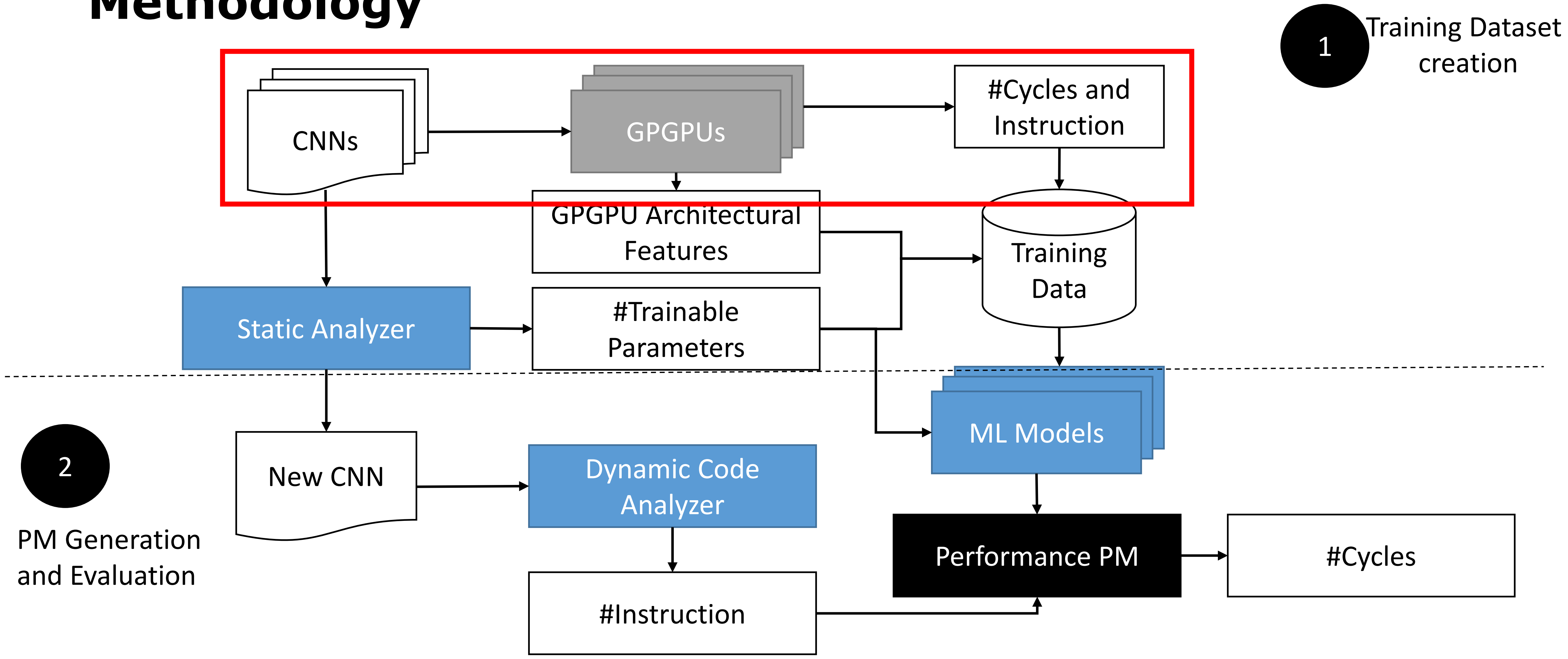
- CUDE → PTX
- Is an virtual ISA
- Portable between NVIDIA GPUs

```
// Generated by LLVM NVPTX Back-End
.version 6.0
.target sm_61
.address_size 64
...
.reqntid 256, 1, 1{
 .reg .pred %p<14>;
 ...
 mov.u32 %r13, %ctaid.x;
 mov.u32 %r14, %tid.x;
 shl.b32 %r15, %r13, 10;
 shl.b32 %r16, %r14, 2;
 or.b32 %r1, %r16, %r15;
 setp.lt.u32 %p1, %r1, 718296;
 @%p1 bra LBB0_2;
 bra.uni LBB0_1;
 LBB0_2:
 ld.param.u64 %rd10, [fusion_135_param_0];
 ...
 LBB0_1:
 ret;}
...
```

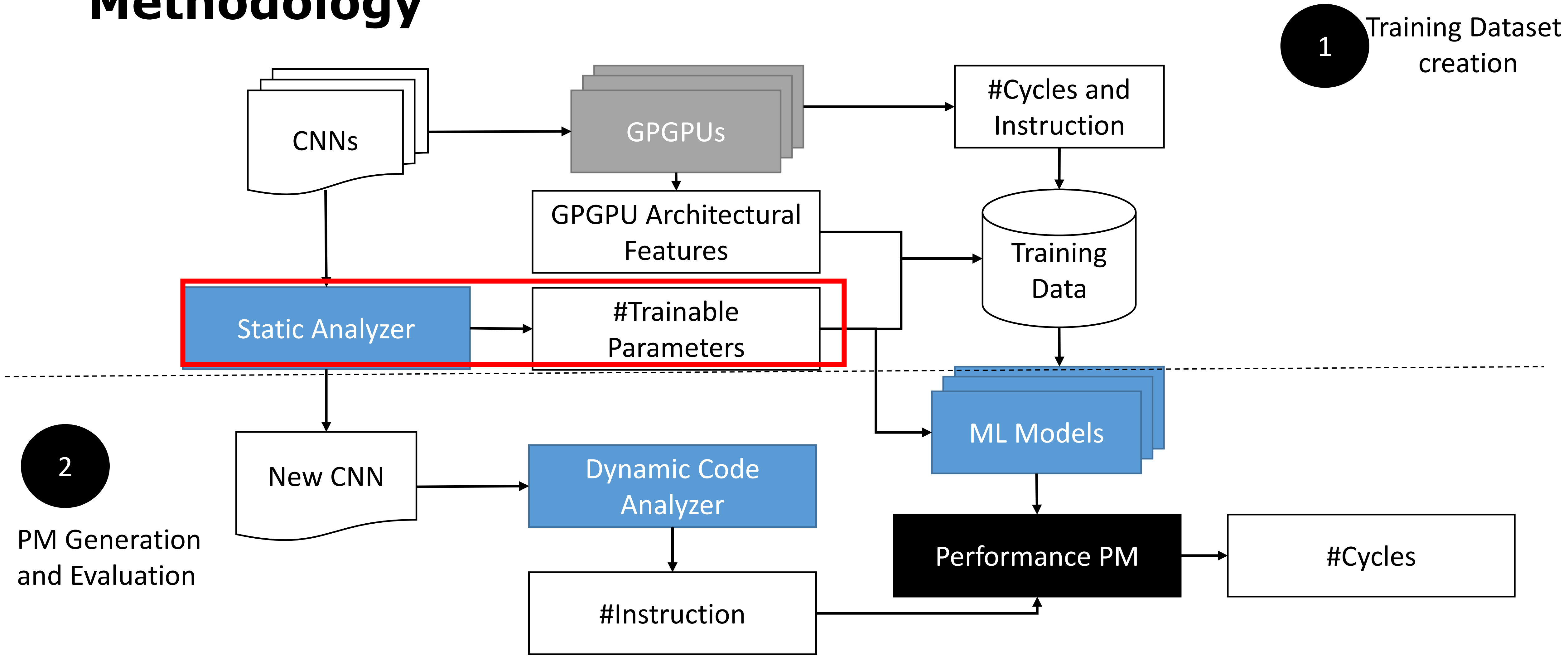
Methodology



Methodology

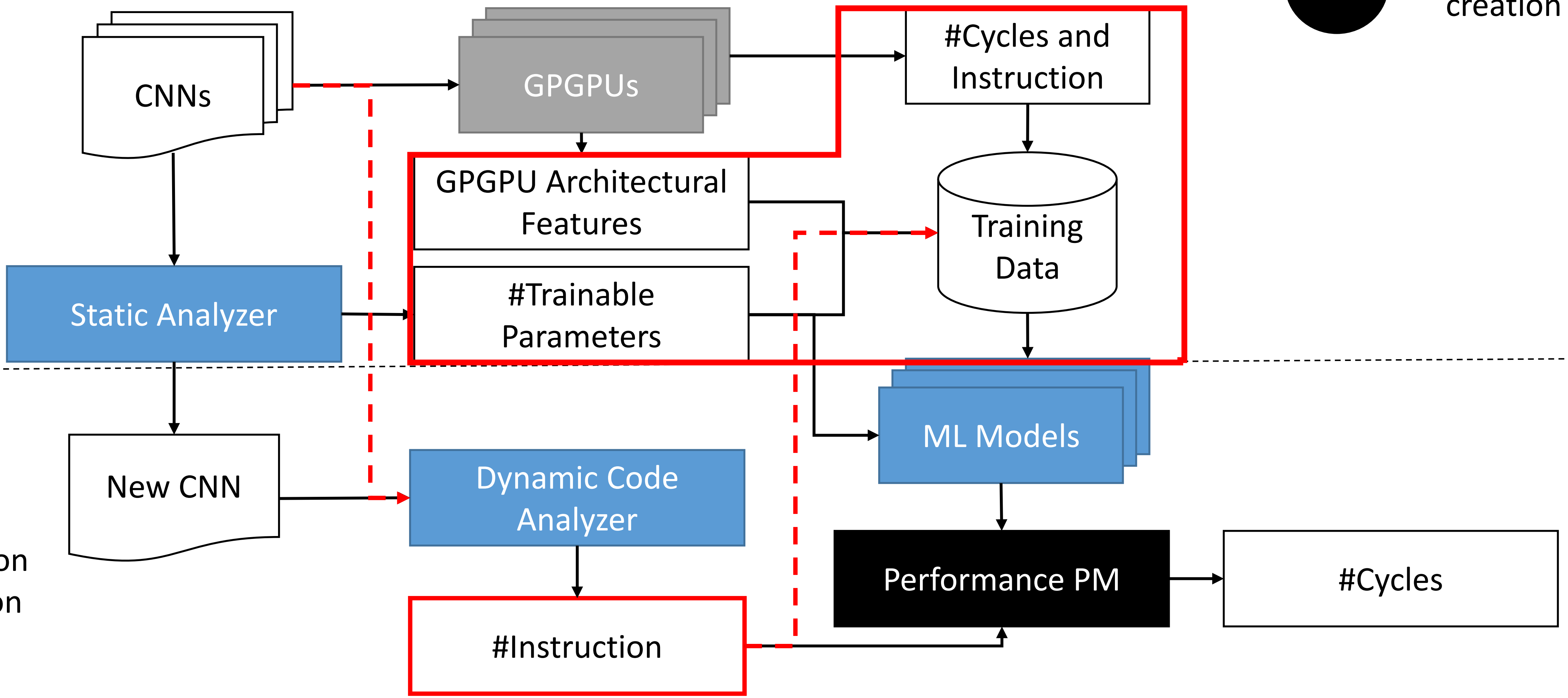


Methodology



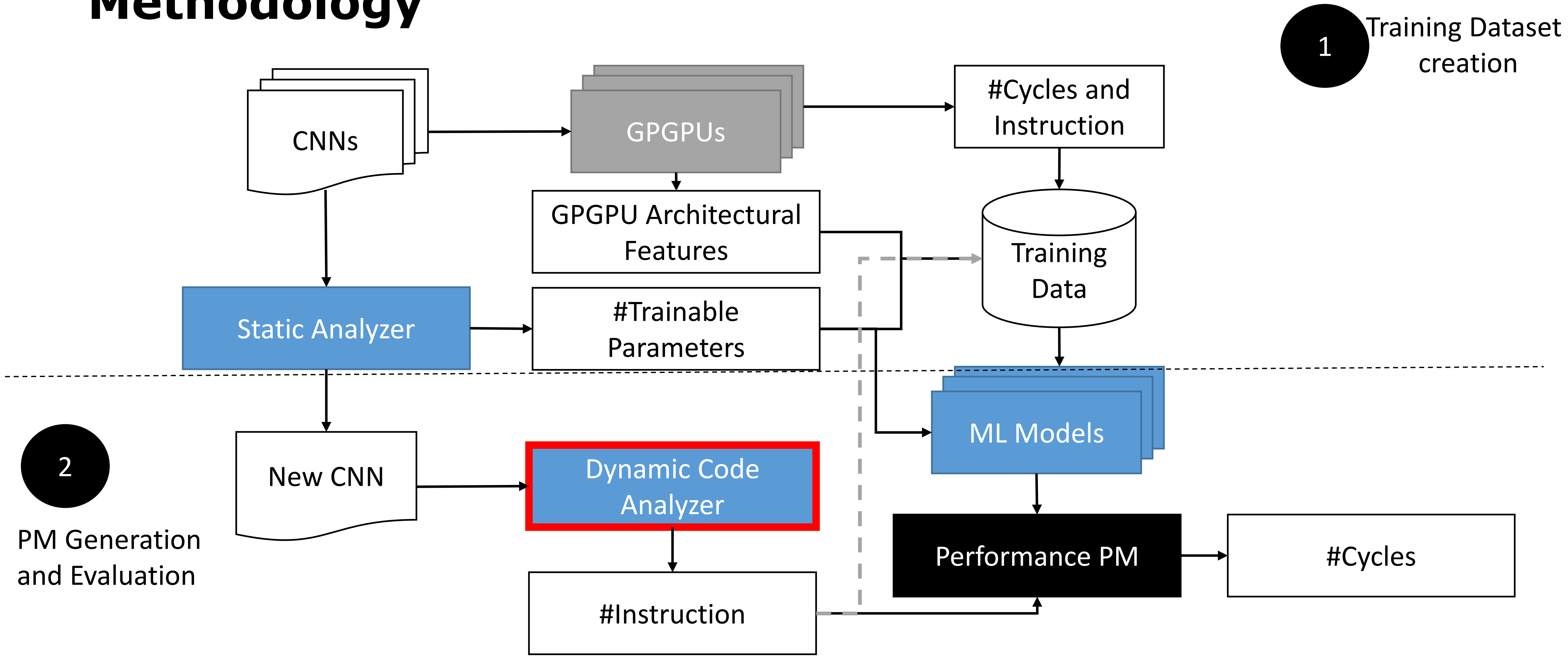
Methodology

1 Training Dataset creation

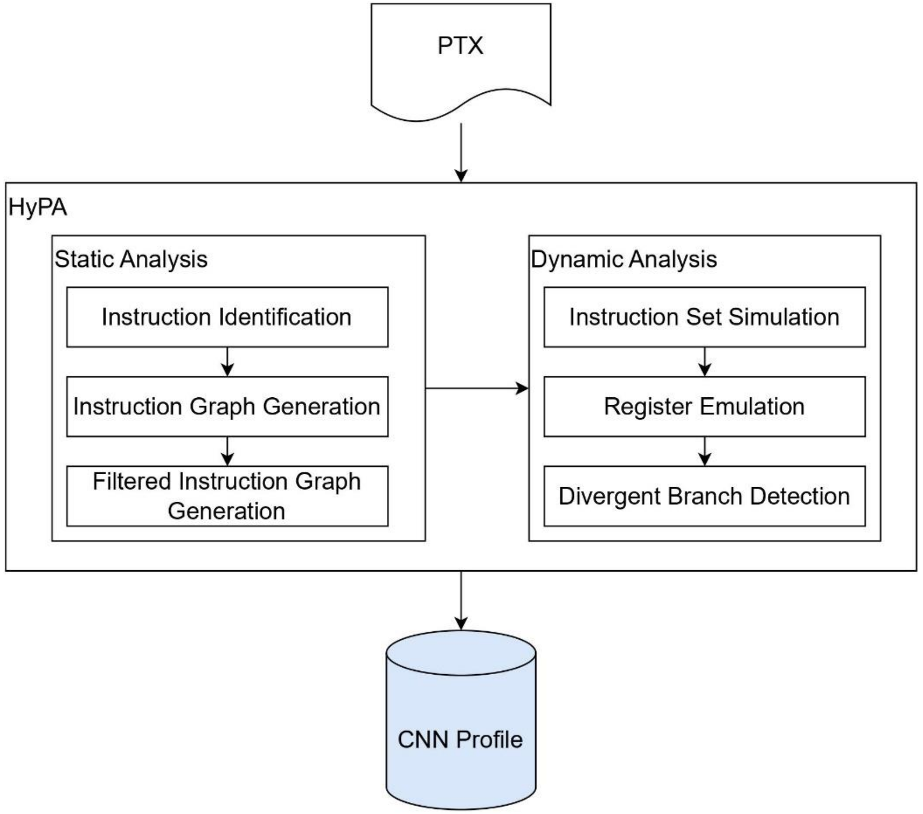


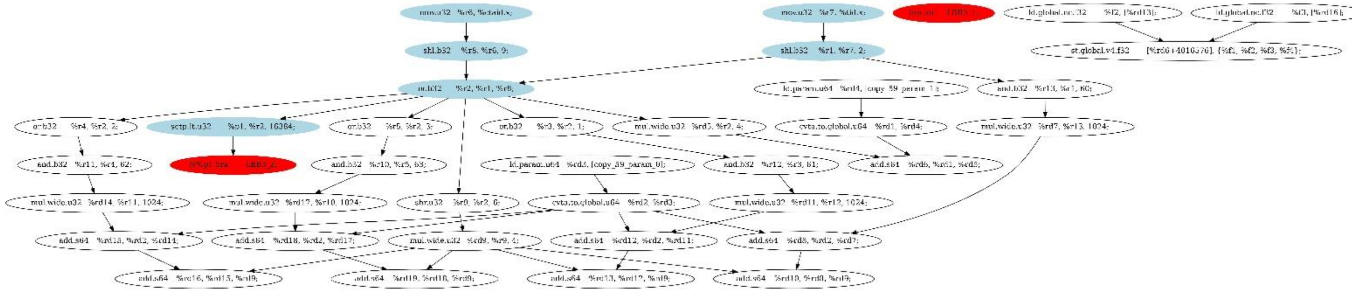
2 PM Generation and Evaluation

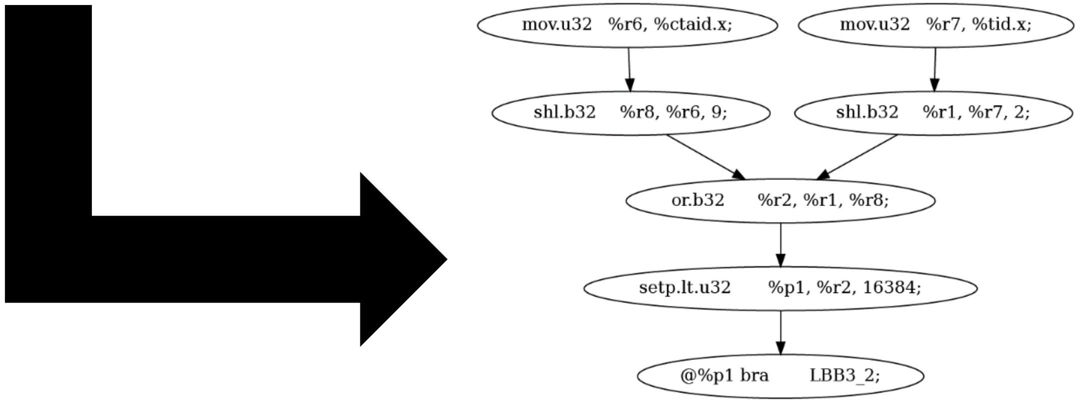
Methodology



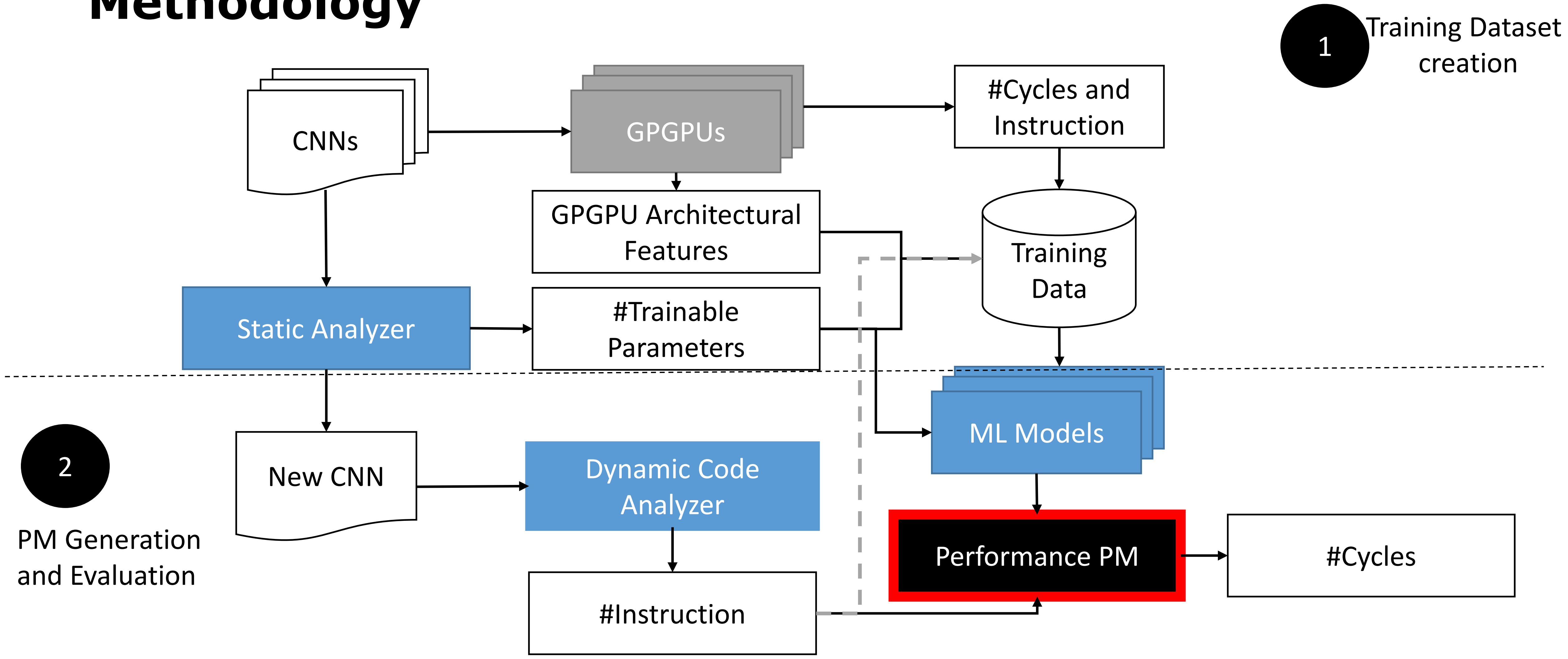
Dynamic Code Analysis





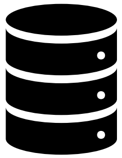


Methodology

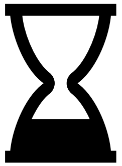


Predictive Model

- Five different Algorithm
  - Linear Regression
  - K-Nearest Neighbors
  - Random Forest Tree
  - Decision Tree
  - XG Boost



Small Dataset



Fast Execution

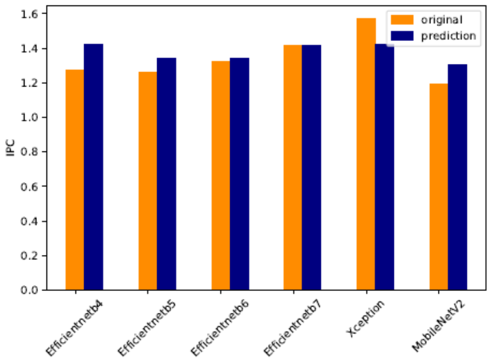
Results

How good are the predictive models?

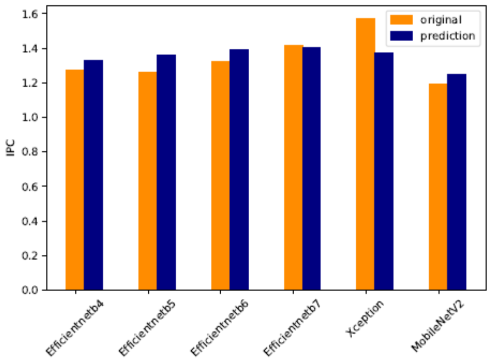
Comparing diffrent ML-Regression Models

Regression Model	MAPE	R2	Adj. R2
Linear Regression	8.07%	-0.0034	-0.4439
K-Nearest Neighbors	5.94%	0.34	0.08
Random Forest Tree	7.12%	0.22	-0.12
Decision Tree	5.73%	0.45	0.19
XG Boost	7.59%	0.14	-0.24

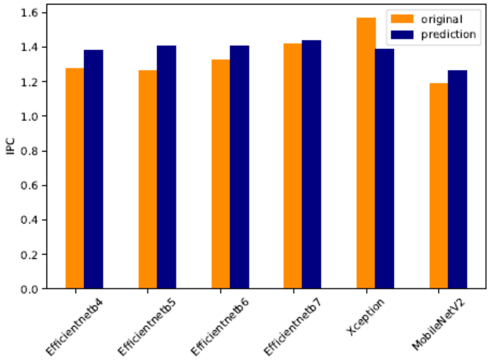
Results



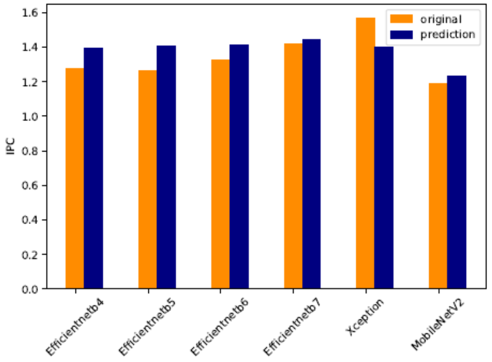
(a) Decision Tree predicted



(b) KNN predicted



(c) Gradient Boosting predicted



(d) Random Forest Tree predicted

Regression Model	MAPE	R2	Adj. R2
Linear Regression	8.07%	-0.0034	-0.4439
K-Nearest Neighbors	5.94%	0.34	0.08
Random Forest Tree	7.12%	0.22	-0.12
Decision Tree	5.73%	0.45	0.19
XG Boost	7.59%	0.14	-0.24

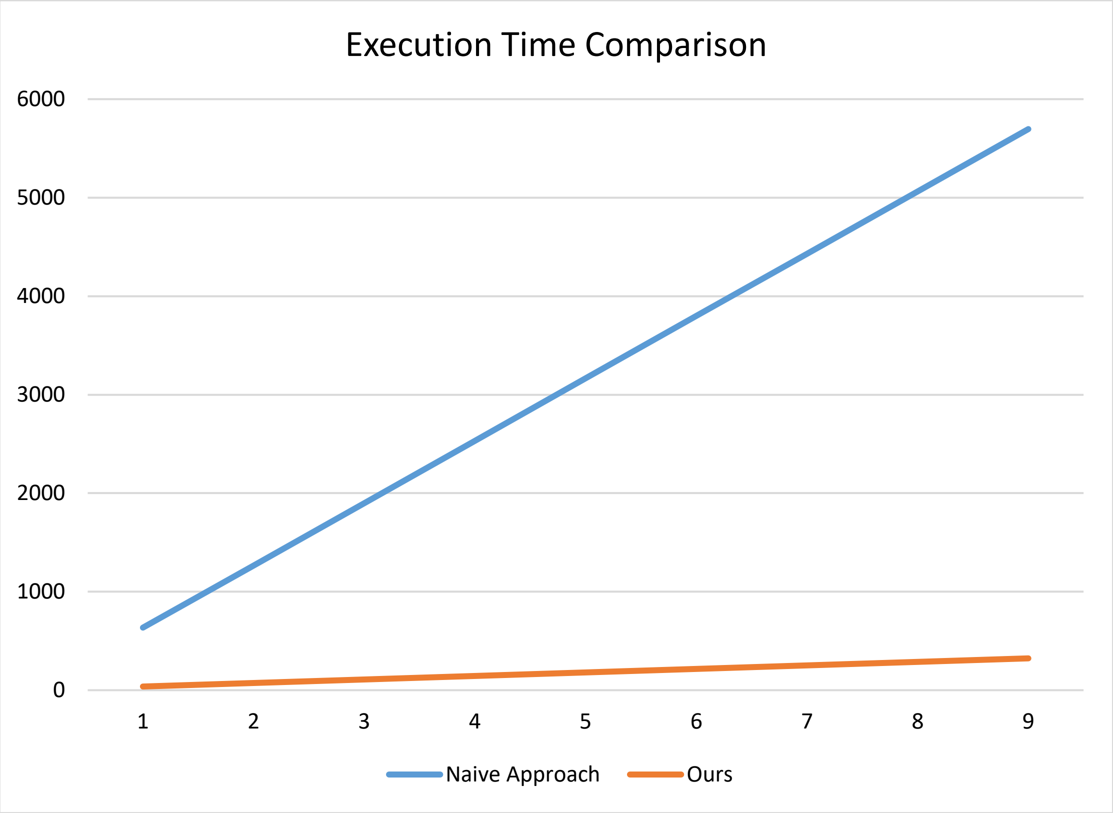
Most Influencing Predictors

Executed Instructions

Trainable Parameters

Memory Bandwidth

Execution Time



- Time of Profiling (nvprof):
  - Min 314s
  - Max 1037s
- Time Dynamic Code Analysis
  - Min 6.8s
  - Max 60.2s
- Time Predictive Model
  - Min 1s
  - Max 11s

CNN	Naive Approach (s)								Ours (s)								
	t_p	n=1	n=2	n=3	n=4	n=5	n=6	n=7	t_{pm}	t_{dca}	n=1	n=2	n=3	n=4	n=5	n=6	n=7
efficientnet b3	663	663	1,326	1,989	2,652	3,315	3,978	4,641	11	24.8	35.8	46.8	57.0	68.8	79.8	90.8	101.8
efficientnet b4	778	778	1,556	2,334	3,112	3,890	4,668	5,446	9	24.0	33.0	42.0	51.0	60.0	69.0	78.0	87.0
efficientnet b5	950	950	1,900	2,850	3,800	4,750	5,700	6,610	8	40.3	48.3	56.3	64.3	72.3	80.3	88.3	96.3
efficientnet b6	936	936	1,872	2,808	3,768	4,680	5,616	6,552	8	60.2	68.2	76.2	84.2	92.2	100.2	108.2	116.2
efficientnet b7	1,037	1,037	2,074	3,111	4,148	5,185	6,222	7,259	1	6.8	7.8	8.8	9.8	10.8	11.8	12.8	13.8
Xception	314	314	628	942	1,256	1,570	1,884	2,198	8	23.6	31.6	39.6	47.6	55.6	63.6	71.6	79.6
MobileNet V2	343	343	686	1,029	1,372	1,715	2,058	2,401	8	12.2	20.2	28.2	36.2	44.2	52.2	60.2	68.2

Conclusion

Future Work

- Combining with power estimation
- Multi-Objective Optimization
- Improve dynamic code analyzer

Thank you for your attention

Christopher Metz
cmetz@uni-bremen.de