

Transferable AutoML by Model Sharing over Grouped Dataset

Chao Xue¹, Junchi Yan², Rong Yan¹, Stephen M. Chu¹, Yonggang Hu³, Yonghua Lin¹

¹IBM Research – China, ³IBM Systems

²Department of CSE and MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University

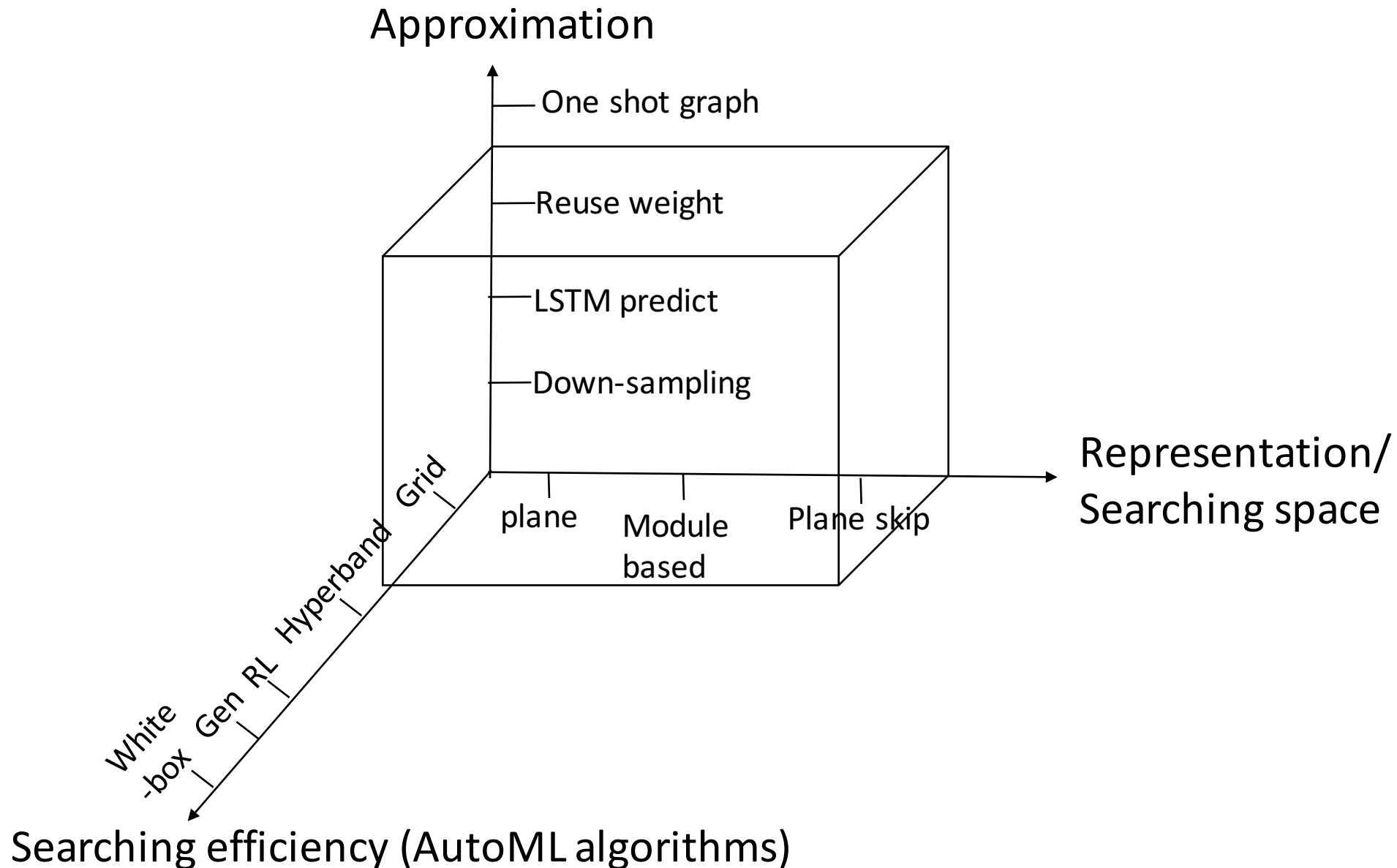
© IBM Corporation

IBM Research China



Outline

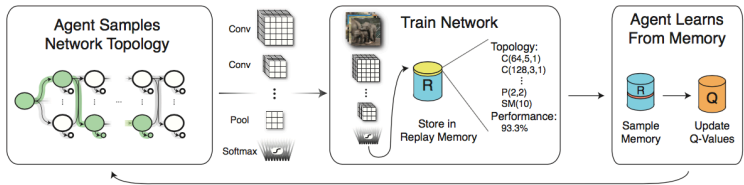
- Introduction of AutoML
- Transferable AutoML by Model Sharing over Grouped Dataset



AutoML algorithms--from Random search, Bayesian Optimization to Evolution , Reinforcement Learning and Continuous space optimization

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}.$$

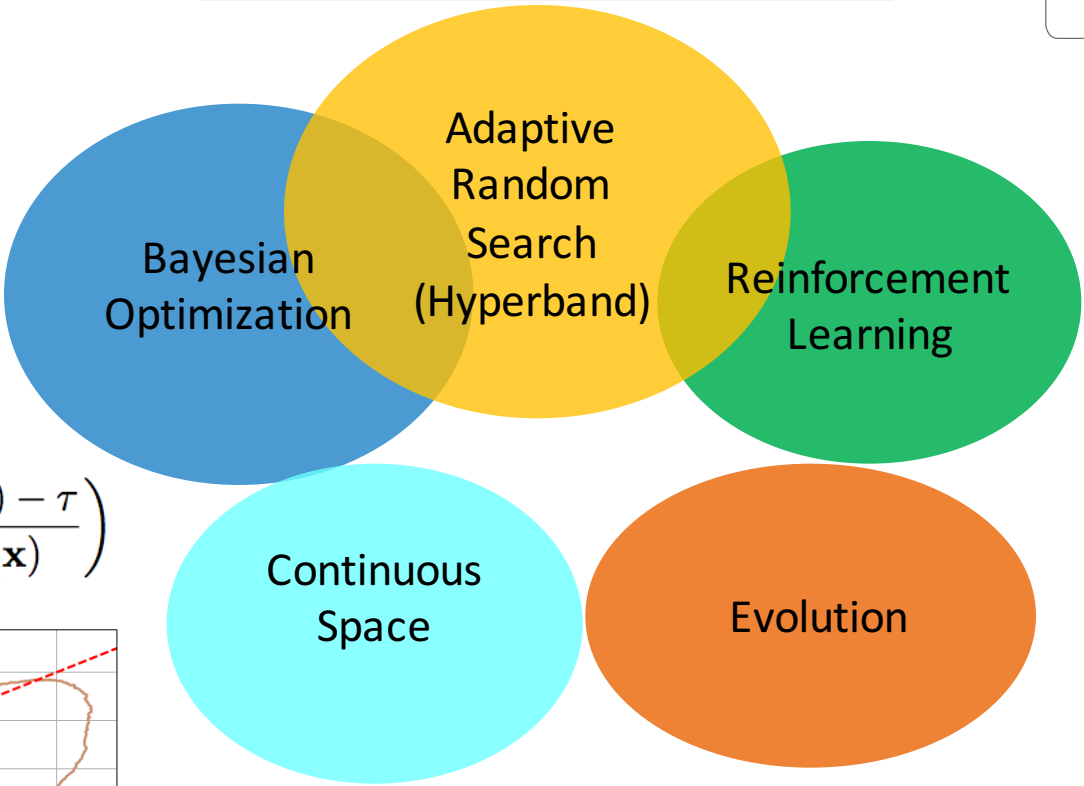
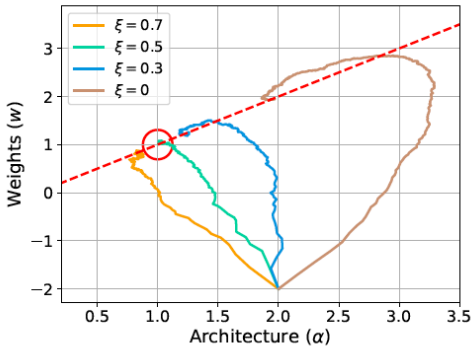
	$s = 4$		$s = 3$		$s = 2$		$s = 1$		$s = 0$	
i	n_i	r_i	n_i	r_i	n_i	r_i	n_i	r_i	n_i	r_i
0	81	1	27	3	9	9	6	27	5	81
1	27	3	9	9	3	27	2	81		
2	9	9	3	27	1	81				
3	3	27	1	81						
4	1	81								



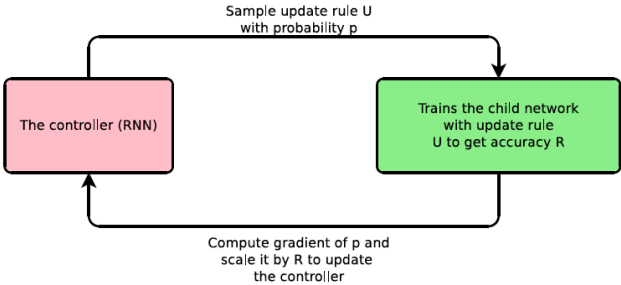
$$\begin{aligned} \mu_{a|b} &= \mu_a + \Sigma_{ab} \Sigma_{bb}^{-1} (\mathbf{x}_b - \mu_b) \\ \Sigma_{a|b} &= \Sigma_{aa} - \Sigma_{ab} \Sigma_{bb}^{-1} \Sigma_{ba}. \end{aligned}$$

$$\begin{aligned} \mu_n(\mathbf{x}) &= \mu_0(\mathbf{x}) + \mathbf{k}(\mathbf{x})^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} (\mathbf{y} - \mathbf{m}) \\ \sigma_n^2(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}(\mathbf{x}), \end{aligned}$$

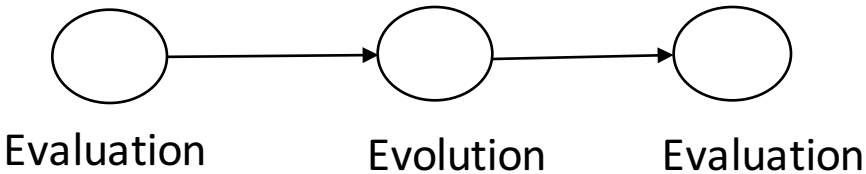
$$\alpha_{PI}(\mathbf{x}; \mathcal{D}_n) := \mathbb{P}[v > \tau] = \Phi \left(\frac{\mu_n(\mathbf{x}) - \tau}{\sigma_n(\mathbf{x})} \right)$$







Temporal Difference: Q-learning

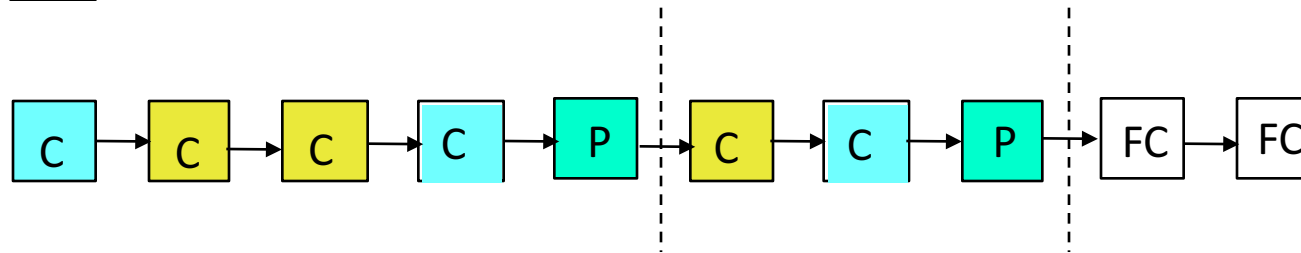


Policy Gradients



-  Conv, kernel size: 3, output channel 32
-  Dilated conv, kernel size: 5, output channel 64
-  Kernel size: 3, stride 3, max pooling
-  Output size: 512

NEUNETS: AN AUTOMATED SYNTHESIS ENGINE FOR NEURAL NETWORK DESIGN, Atin Sood, Benjamin Elder, Benjamin Herta and Chao Xue, etc., arXiv :1901.06261v1, 2019



Model representation:

Components: 2

FC numbers: 2

Conv stacks: [3, 2]

Conv type: [conv, dilated, dilated, conv, dilated, conv]

Conv kernel size: [3,5,5,3,5,3]





Conv output size: [32,64,64,32,64,32]

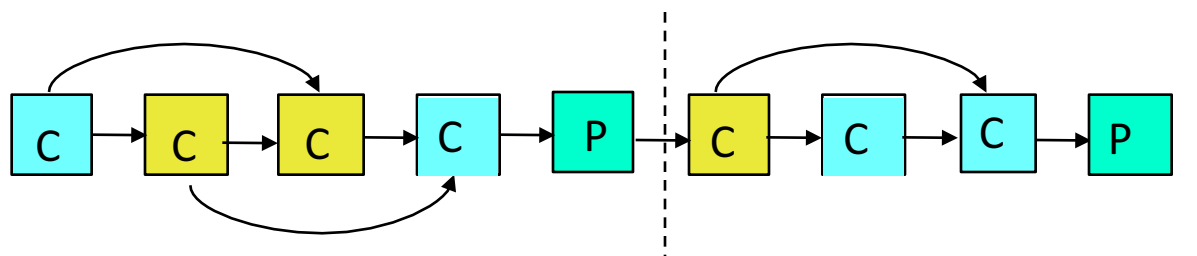
Pooling kernel size: [3,3]

Pooling stride size: [3,3]

Pooling type: [max, max]

FC size: [512, 512]

-  Conv, kernel size: 3, output channel 32
-  Dilated conv, kernel size: 5, output channel 64
-  Kernel size: 3, stride 3, max pooling
-  Output size: 512



NEUNETS: AN AUTOMATED SYNTHESIS ENGINE FOR NEURAL NETWORK DESIGN, Atin Sood, Benjamin Elder, Benjamin Herta and Chao Xue, etc., arXiv :1901.06261v1, 2019

Model representation:

Components: 2

FC numbers: 0

Conv stacks: [4, 3]

Skip pattern: [1-11-011, 1-11]

Conv type: [conv, dilated, dilated, conv, dilated, conv, conv]

Conv kernel size: [3,5,5,3,5,3,3]

Conv output size: [32,64,64,32,64,32,32]

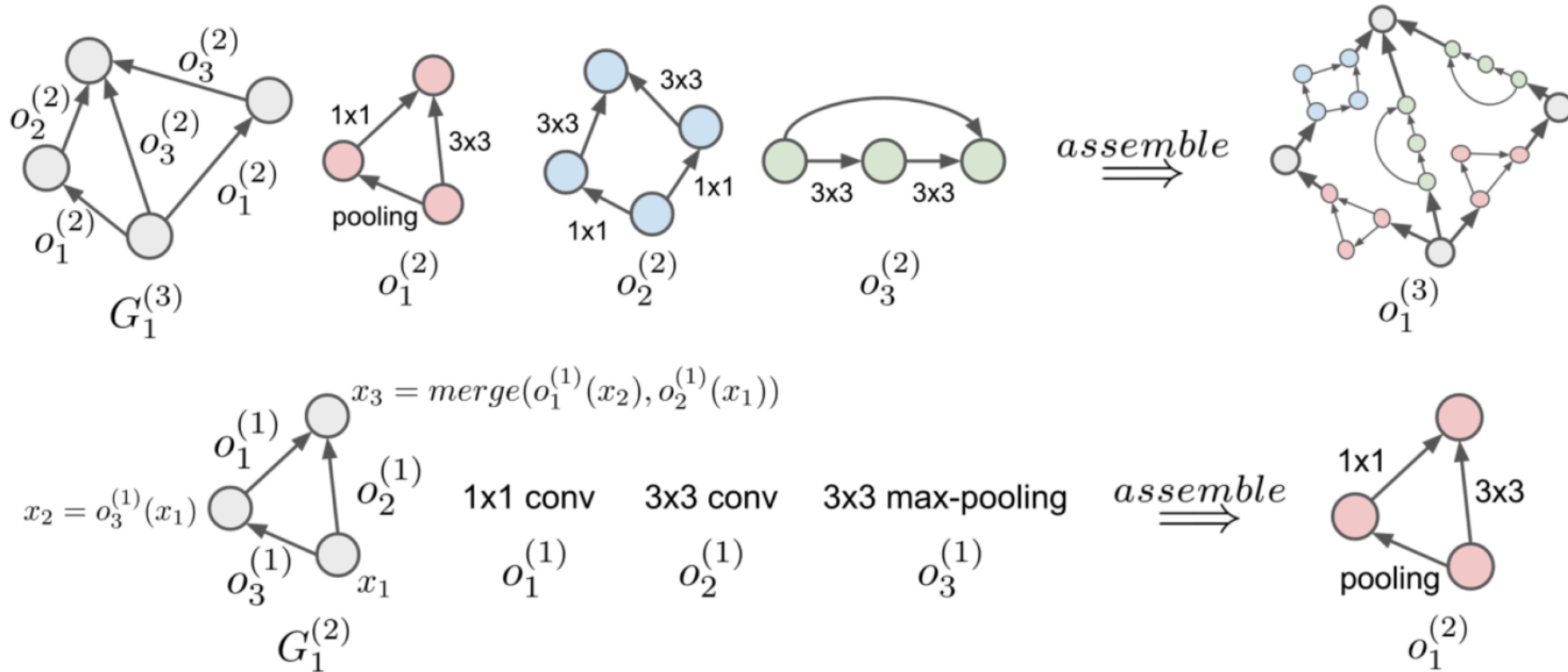
Pooling kernel size: [3,3,3]

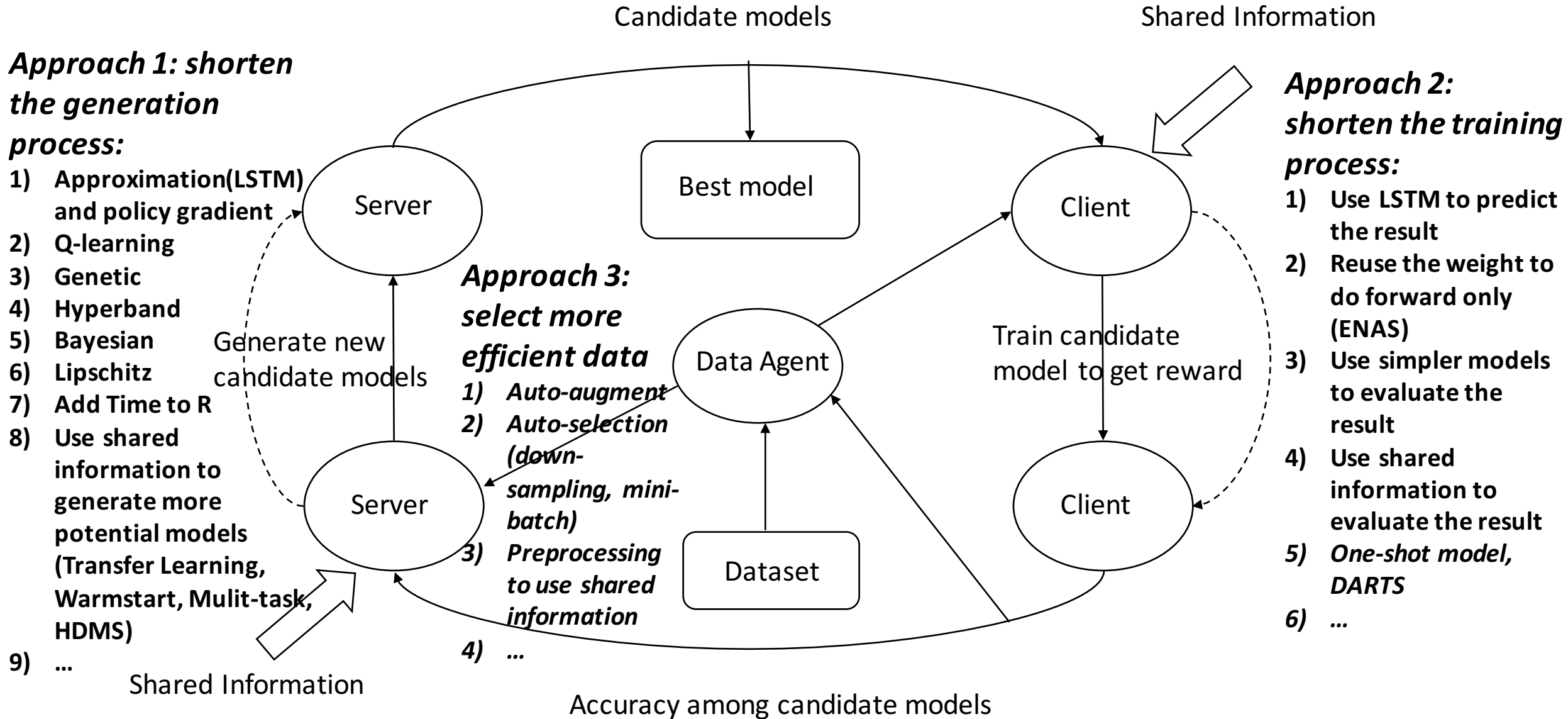
Pooling stride size: [3,3,3]

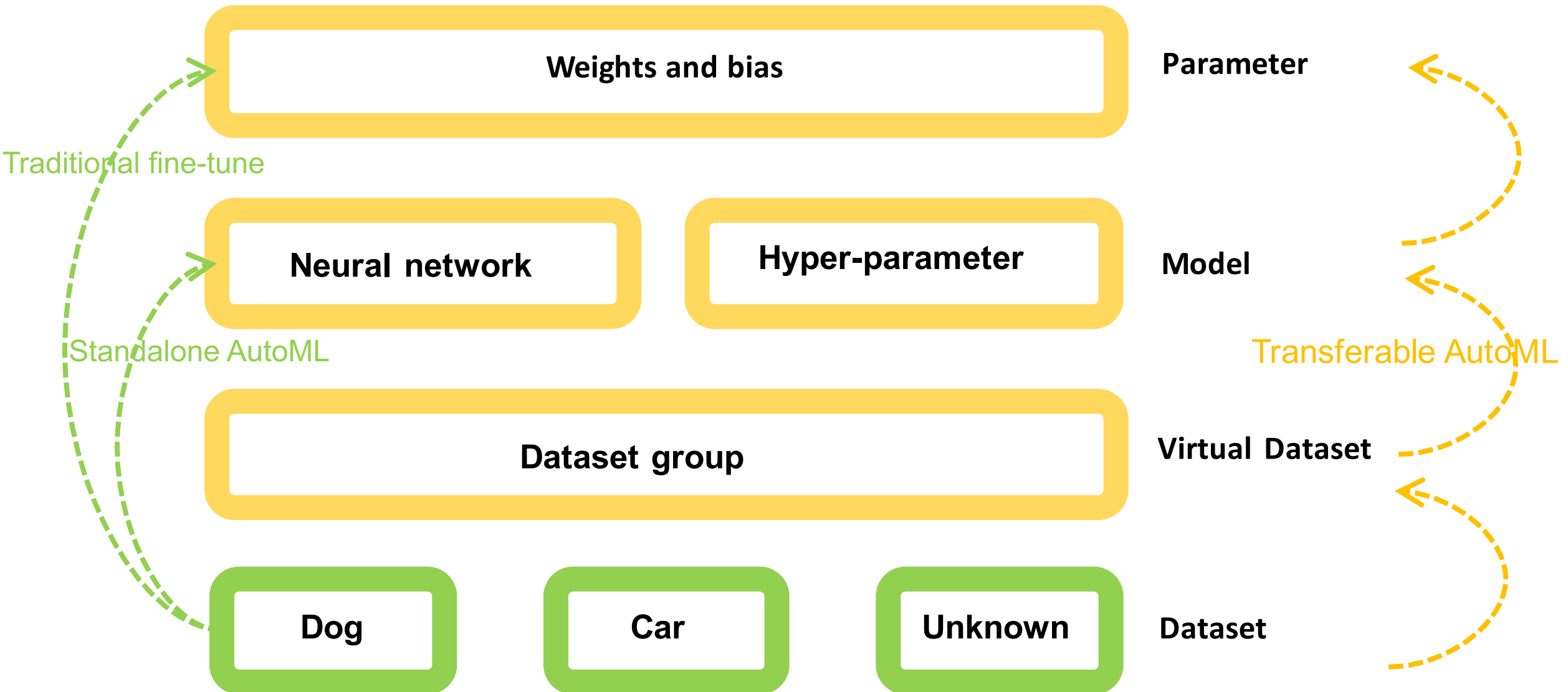
Pooling type: [max, max]

Neural Network Encoding

HIERARCHICAL REPRESENTATIONS FOR EFFICIENT ARCHITECTURE SEARCH, Hanxiao Liu, ICLR 2018







Meta-learning:

We introduce a meta learning method to express a dataset d in a dataset feature space Ω_d . To prove this representation can work well for AutoML, we consider the AutoML problem first. The basic idea for AutoML is to identify the model m from a given dataset d :

$$m^* = \arg \max_m p(m|d) \quad (1)$$

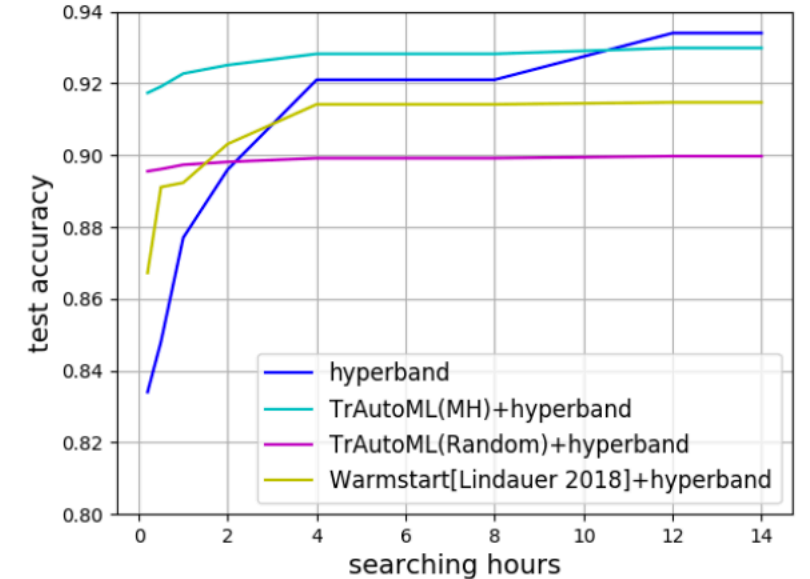
It is reasonable to share m^* among the datasets that above posterior distributions conditioned on are closely approximated. To compare the posterior distributions over models between two different datasets, $p(m|d_1)$ and $p(m|d_2)$, we use Kullback-Leibler (KL) divergence:

$$\begin{aligned} & KL(p(m|d_1) || p(m|d_2)) \\ &= \int p(m|d_1) \ln \left\{ \frac{p(m|d_1)}{p(m|d_2)} \right\} dm \\ &\approx \sum_{b_i} \frac{p(b_i)p(d_1|b_i)}{\sum_{b_j} p(b_j)p(d_1|b_j)} \ln \left(\frac{p(d_1|b_i)}{p(d_2|b_i)} \frac{\sum_{b_j} p(b_j)p(d_2|b_j)}{\sum_{b_j} p(b_j)p(d_1|b_j)} \right) \end{aligned} \quad (2)$$

Transfer Architecture:

Markov process and hypothesis test mechanism for dataset clustering.

These two components can handle Type II error and Type I error for dataset grouping (incorrectly accepting grouping and incorrectly rejecting grouping), respectively.



(d) FASHION-MNIST

Table 2: Total search time (in days, including the overhead of running benchmark models), total classification relative errors (TRE) and benchmark overhead on 7 dataset: in model search from scratch setting.

Techniques combination	Total search time	TRE	Overhead
Hyperband [22]	10.40	0	0
Warmstart [24] + Hyperband	6.23	0.412	0
Meta-learning [26] + Hyperband	3.85	0.118	0
Tr-AutoML(Random) + Hyperband	2.96	1.653	0
Tr-AutoML(Kmeans) + Hyperband	5.44	0.059	0.2
Tr-AutoML(Seq. Kmeans [13]) + Hyperband	4.48	0.061	0.2
Tr-AutoML(MH) + Hyperband	3.17	0.062	0.2
MetaQNN [3]	16.29	0	0
Warmstart [24] + MetaQNN	7.19	0.276	0
Meta-learning [26] + MetaQNN	5.46	0.075	0
Tr-AutoML(Random) + MetaQNN	4.68	1.149	0
Tr-AutoML(Kmeans) + MetaQNN	7.87	0.036	0.2
Tr-AutoML(Seq. Kmeans [13]) + MetaQNN	6.32	0.041	0.2
Tr-AutoML(MH) + MetaQNN	4.85	0.039	0.2
ENAS [31]	12.22	0	0
Warmstart [24] + ENAS	5.10	0.132	0
Meta-learning [26] + ENAS	4.82	0.044	0
Tr-AutoML(Random) + ENAS	4.02	0.471	0
Tr-AutoML(Kmeans) + ENAS	6.17	0.017	0.2
Tr-AutoML(Seq. Kmeans [13]) + ENAS	5.04	0.019	0.2
Tr-AutoML(MH) + ENAS	4.22	0.019	0.2

Table 4: Searching time (in days) and total classification relative errors on four datasets: in model search from pre-defined model setting.

Algorithm	Time	TRE
MetaQNN	8	0
Tr-AutoML(Random) + MetaQNN	2.8	1.78
Tr-AutoML(Kmeans) + MetaQNN	4.5	0.069
Tr-AutoML(MH) + MetaQNN	3.0	0.074

Table 5: Search time (in days, including overhead). Test accuracy: in transfer from basic cells setting.

Target dataset	Techniques	Search time	Accuracy
FASHION-MNIST	Hyperband [22]	1.67	0.942
	Meta-learning [26]	0.001	0.939
	Tr-AutoML(Random)	0	0.936
	Tr-AutoML(MH)	0.013	0.939
STL10	ENAS [31]	1.08	0.734
	Meta-learning [26]	0.001	0.680
	Tr-AutoML(Random)	0	0.692
	Tr-AutoML(MH)	0.017	0.725
ImageNet	NASNet-A [43]	1800	0.740
	Meta-learning [26]	2.56	0.717
	Tr-AutoML(Random)	2.54	0.712
	Tr-AutoML(MH)	2.81	0.734

Thank you for listening