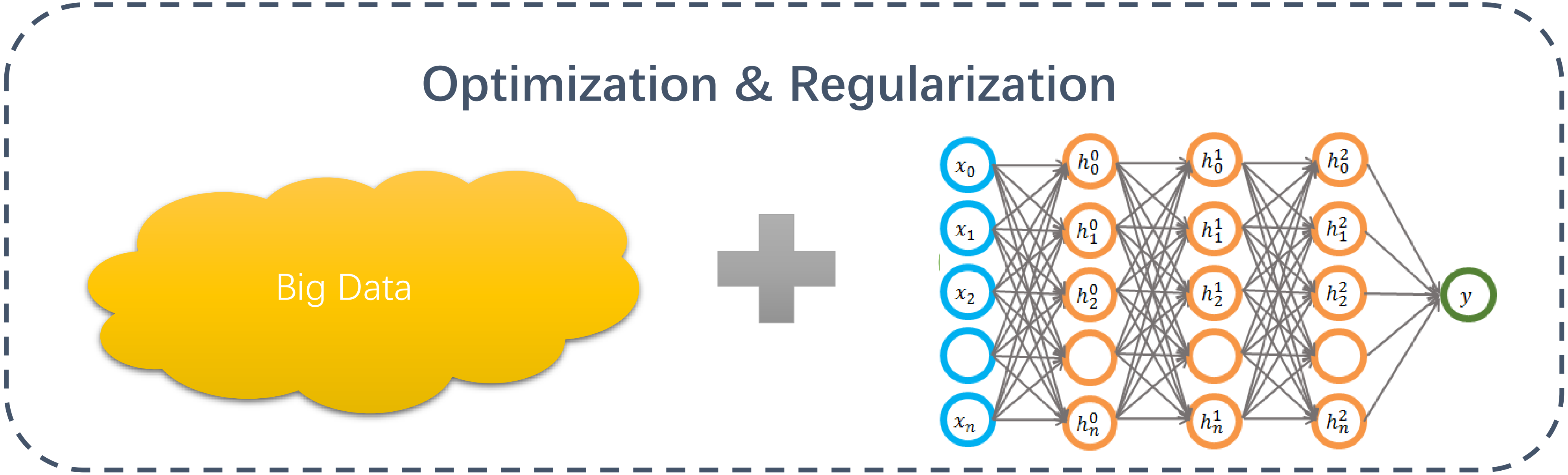
Optimization, Speed-up, and Out-of-distribution Prediction in Deep Learning

Wei Chen

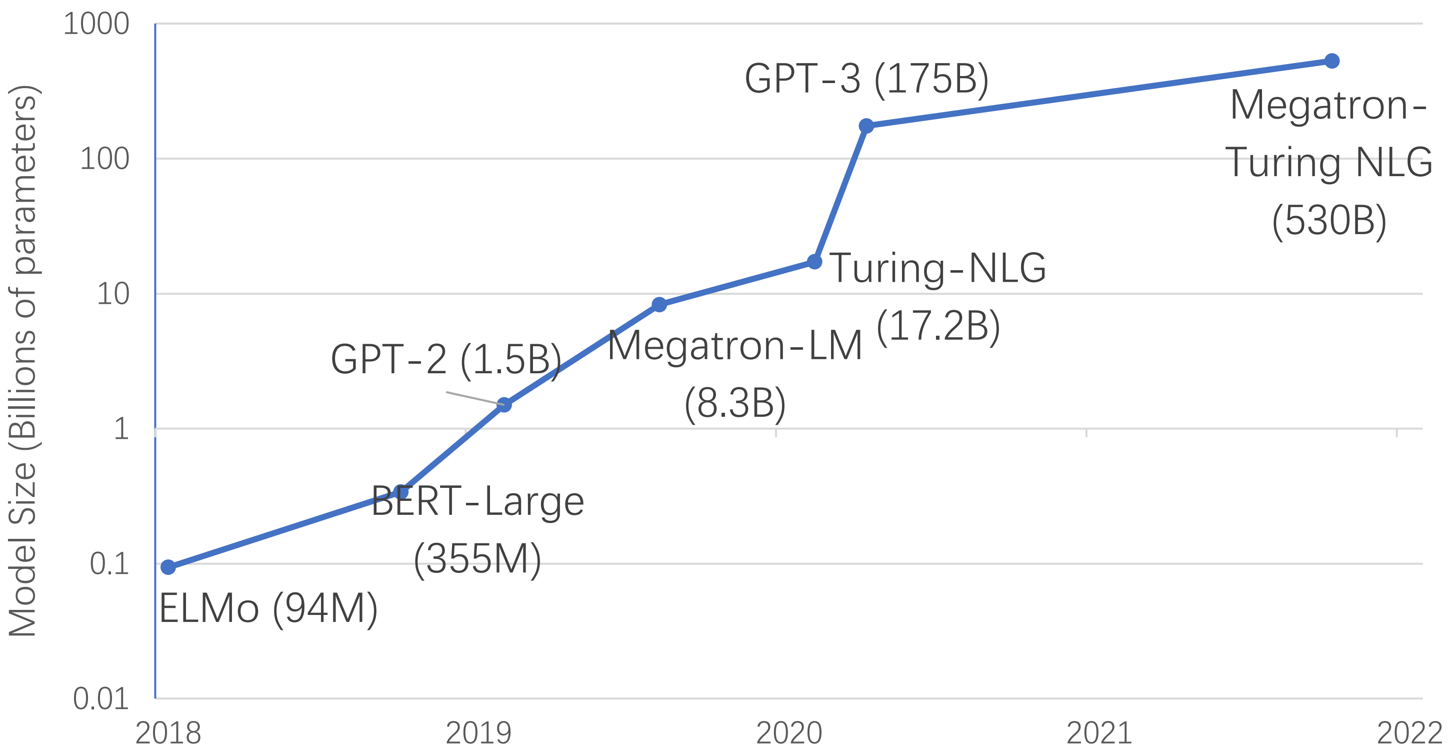
Institute of Computing Technology

Chinese Academy of Sciences

Deep Learning



Model is BIGGER

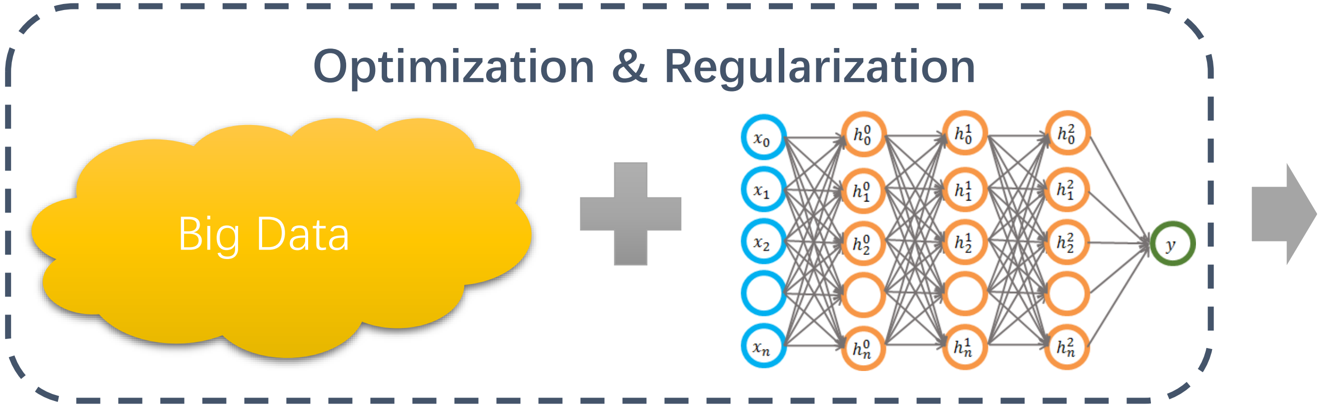


Machine Learning Taxonomy			
Deep Learning	Dynamics Learning	Reinforcement Learning	Data-driven Optimization

Tasks			
Prediction	Dynamical Evolution	Decision-making (control/search)	Optimization

Applications			
Perception (Vision, Speech, Language)	Science (Physics, Biology, Sustainability)	Robotics, Games, Logistics	Reasoning, Theorem proving, code generation

The Challenges in Machine Learning



The Learnt Model

Performance

Efficiency

Interpretability

Robustness

Fairness

Data Privacy

Convergence

Generalization

Theoretic ML

1. Optimization: G-SGD

Distributed ML

2. Synchronization: DC-ASGD

Trustworthy ML

3. OOD prediction: Causal Learning

Optimization: $\mathcal{G}$-SGD

Optimizing ReLU Neural Networks in its Positively Scale-Invariant Space (ICLR’19)

Joint work with Qi Meng, Shuxin Zheng, Huishuai Zhang, Qiwei Ye, Zhi-Ming Ma, Nenghai Yu, and Tie-Yan Liu.

Code

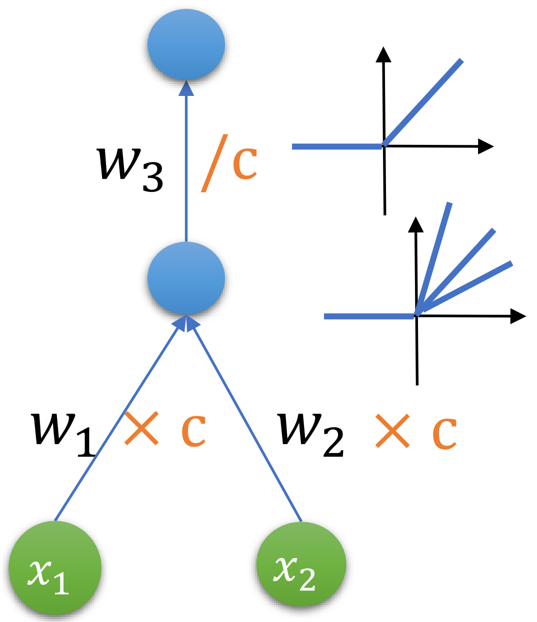
Weight Space is Actually Redundant⋯

- Positively scale-invariant (PSI) functions like ReLU, pReLU, max pooling and average pooling:

$$\sigma(c \cdot x) = c \cdot \sigma(x), \forall c > 0$$

- With PSI activation functions, neural networks with different weights may correspond to the same mathematical function, meaning that the weight space has redundancy.

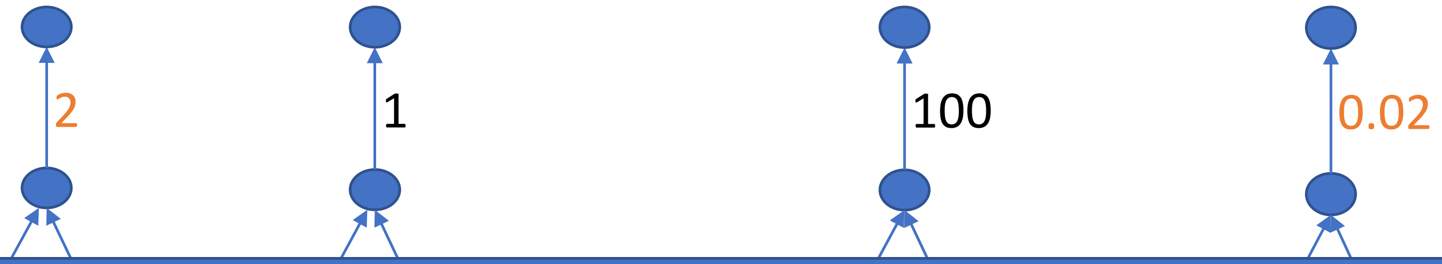
$$f_{w_1,w_2,w_3}(\cdot) = f_{cw_1,cw_2,\frac{w_3}{c}}(\cdot), \forall c > 0$$



$$\mathcal{G} = \{g_{C,O}(\cdot) \triangleq g_{c_1,O_1} \circ \cdots \circ g_{c_H,O_H}(w); C = (c_1, \dots, c_H) \in \mathbb{R}_+^H\}$$

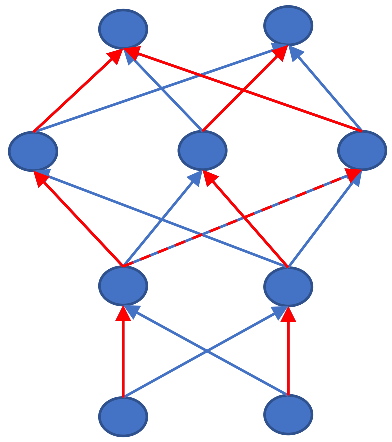
$\mathcal{G}$-invariant Networks vs. $\mathcal{G}$-variant Weights

- Neural networks with PSI activation functions are $\mathcal{G}$-invariant, however, the weights in such networks, as functions, are NOT $\mathcal{G}$-invariant.
  - Redundancy in weight space: gradients of equivalent networks could be different (Neyshabur, et al., 2015)
  - Problematic geometric measure in weight space (Dinh, Bengio, et al. ,2017)



Can we construct a **compact** space which is **sufficient** for representing **$\mathcal{G}$-invariant** neural networks?

$\mathcal{G}$-invariant Path Space



Path: $p = (i_0, \dots i_L)$

Value of path: the product of the weights over the path

$$v_w(p) = w_{i_0, i_1}, \dots, w_{i_{L-1}, i_L},$$

Activation status of path: only if all the nodes along the path are active, the path is active.

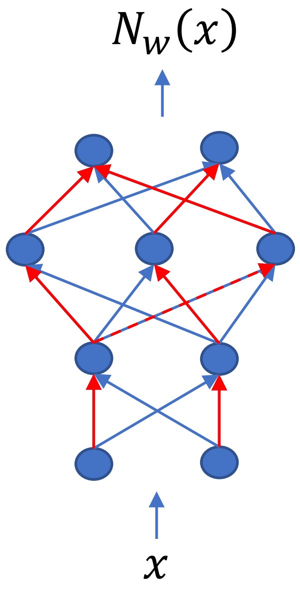
$$a_w(p, x) = \prod_{l=1}^{L} I[o_{i_l}(x; w) > 0]$$

Theorem 1: The values and activation status of paths are **$\mathcal{G}$-invariant,** i.e., for arbitrary path p, we have

$$v_w(p) = v_{g(w)}(p), \forall g \in \mathcal{G}$$
$$a_w(p, x) = a_{g(w)}(p, x), \forall g \in \mathcal{G}$$

Path Representation of Neural Networks

[Balduzzi, 2015]

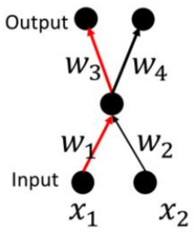


$$N_p^k(x) = \sum_{i_0=1}^{d} \sum_{p \in \mathcal{P}_{k,i_0}} \boxed{v_p(w)} \cdot \boxed{a_p(w,x)} \cdot \boxed{x_{i_0}}$$

Values of Paths — Activation Status of Paths — Input Feature

Objective: $\min_{w} L(w) \rightarrow \min_{v_p, a_p} L(v_p, a_p)$

Dimensionality of Path Space



$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

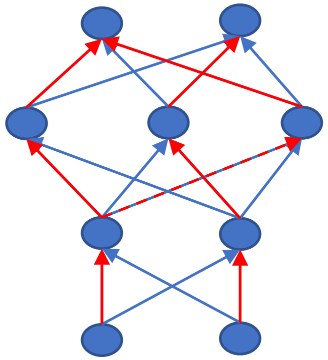
Theorem 2: Consider a ReLU neural network with m weights and structure matrix A, then,

$$Rank(A) = m - H,$$

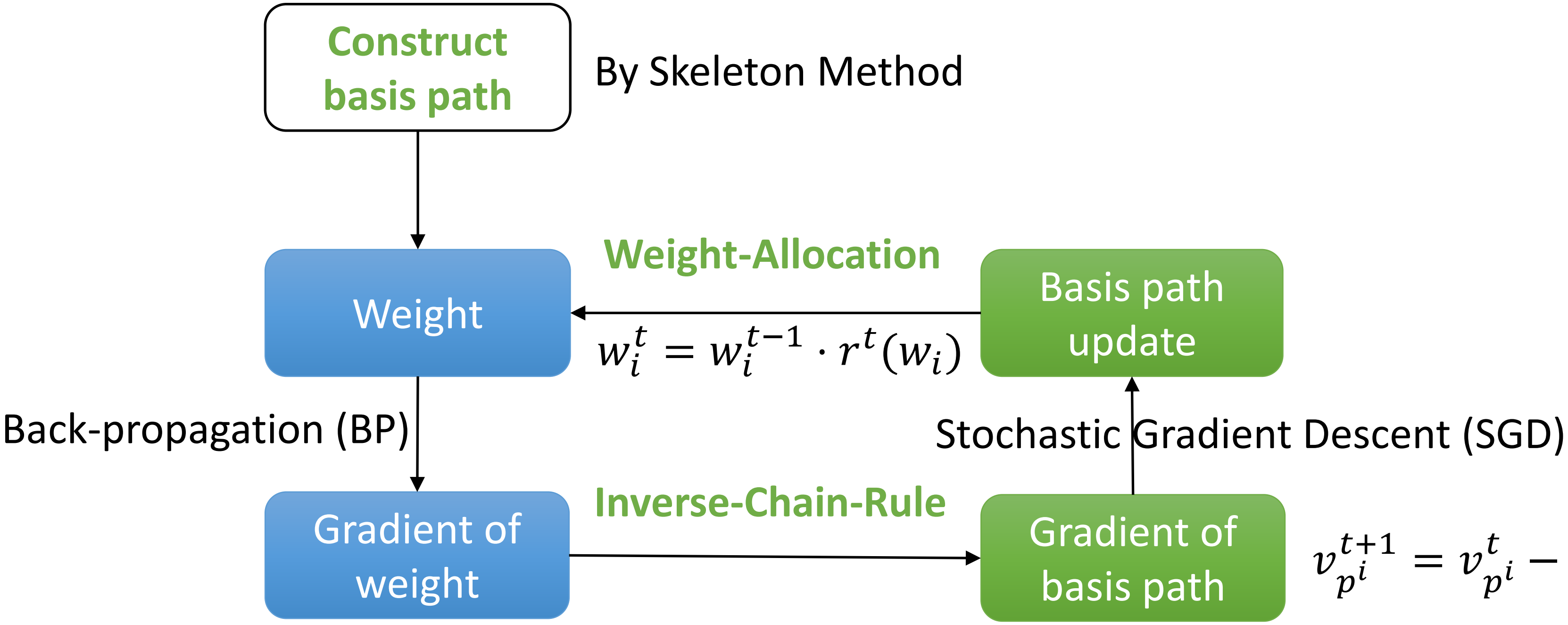
where H is the number of hidden nodes for fully-connected NN and the number of feature maps for CNN, respectively.

Definition 5: **(Basis Paths)** We define the basis paths of ReLU neural networks as the basis column vectors of the structure matrix.

$\mathcal{G}$-Space



$\mathcal{G}$-SGD: the SGD in PSI Space



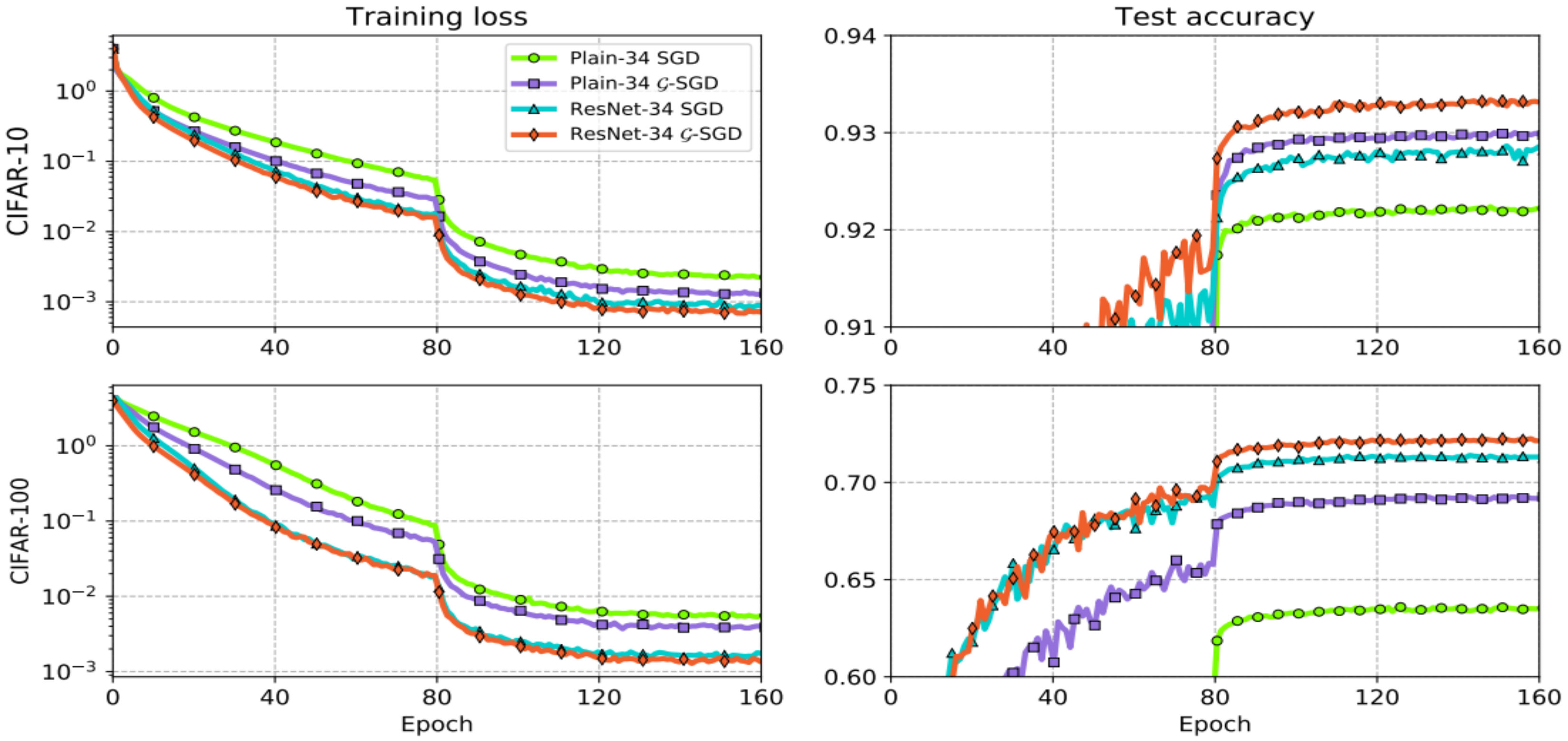
$$v_{p^i}^{t+1} = v_{p^i}^{t} - \eta_t \frac{\partial l(v;x,y)}{\partial v_{p^i}}\Big|_{v=v^t}$$

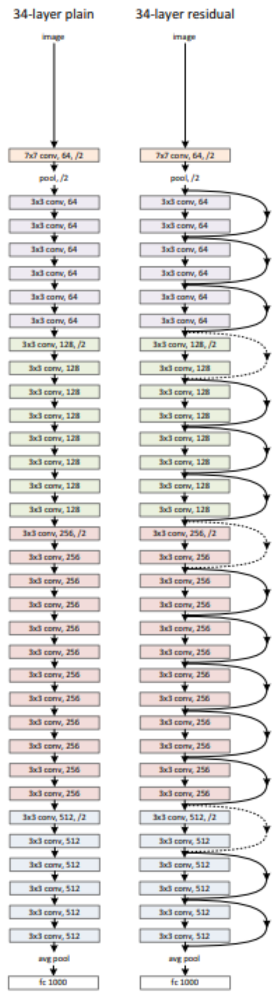
$$\left(v_w(p^1), \cdots, v_w\left(p^{m-H}\right)\right) = (w_1, \cdots, w_m) \odot \tilde{A}, \text{ where} \tilde{A} = \begin{bmatrix} p_1^1 \cdots\ p_1^{m-H} \\ p_2^1 \cdots\ p_2^{m-H} \\ \cdots\cdots \\ p_m^1 \cdots\ p_m^{m-H} \end{bmatrix}$$

Experimental Results

		C10	C100
Plain-34	SGD	7.76 ($\pm$0.17)	36.41($\pm$0.54)
	$\mathcal{G}$-SGD	**7.00** ($\pm$0.10)	**30.74** ($\pm$0.29)
ResNet-34	SGD	7.13 ($\pm$0.22)	28.60($\pm$0.26)
	$\mathcal{G}$-SGD	**6.66** ($\pm$0.13)	**27.74** ($\pm$0.06)

Dataset: CIFAR10
Model: (1) deep convolutional net (plain34) ;(2) Deep residual net (resnet34)
Learning rate: 1.0





Discussion

$\mathcal{G}$-Space

Algorithm

- **Structures**: ResNet (√), DenseNet, RNN, LSTM, transformer (√)
- **Optimizers**: G-SGD (√), G-MomSGD, G-Adam, G-RMSProp
- **Optimization tricks**: BN(√), LN, initialization, weight decay
- **Activations**: ReLU (√), sigmoid, tanh

Theory

- **Generalization:** Basis Path Norm (√), flatness (√)
- Loss Surface & Convergence Rate (√)
- Initialization
- Relation with manifold and quotient space

Application

- Large scale image classification
- NMT (√)
- Reinforcement learning

References:

- Qi Meng et al, G-SGD: Optimizing ReLU Neural Networks in its Positively Scale-Invariant Space, ICLR'19
- Shuxin Zheng et al, Capacity Control of ReLU Neural Networks by Basis-path Norm, AAAI'19.
- Xufang Luo et al, Path-BN: Towards Effective Batch Normalization in the Path Space for ReLU Networks. UAI'21
- Yue Wang et al, The Scale-Invariant Space for Attention Layer in Neural Network. Neurocomputing 392 (2020): 1-10.
- Yue Wang et al, Positively Scale-Invariant Space for Recurrent Neural Networks with ReLU Activations. Preprint
- Juanping Zhu et al, Interpreting the Basis Path Set in Neural Networks. Journal of Systems Science and Complexity 2021, Pages 1-13

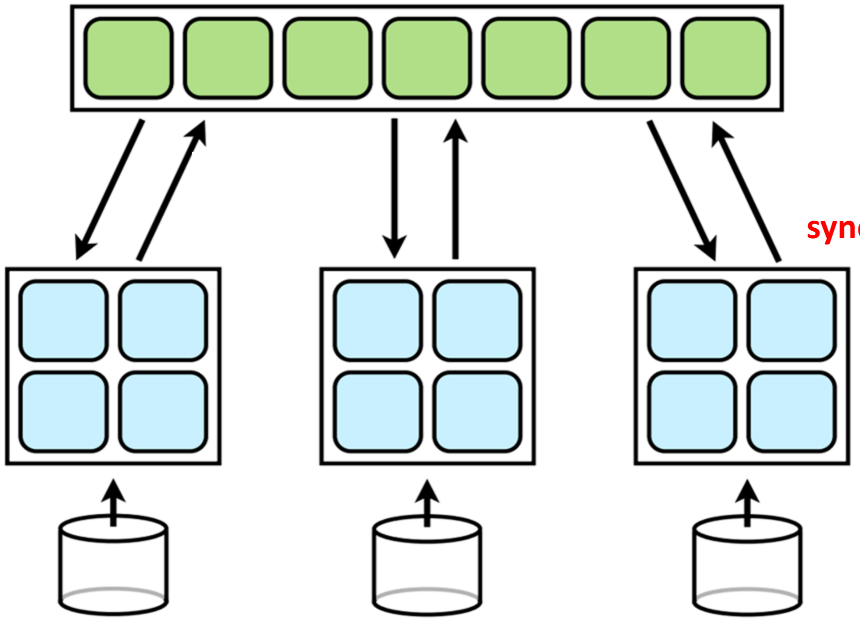
Speed-up: DC-ASGD

Asynchronous Stochastic Gradient Descent with Delay Compensation (ICML'17)

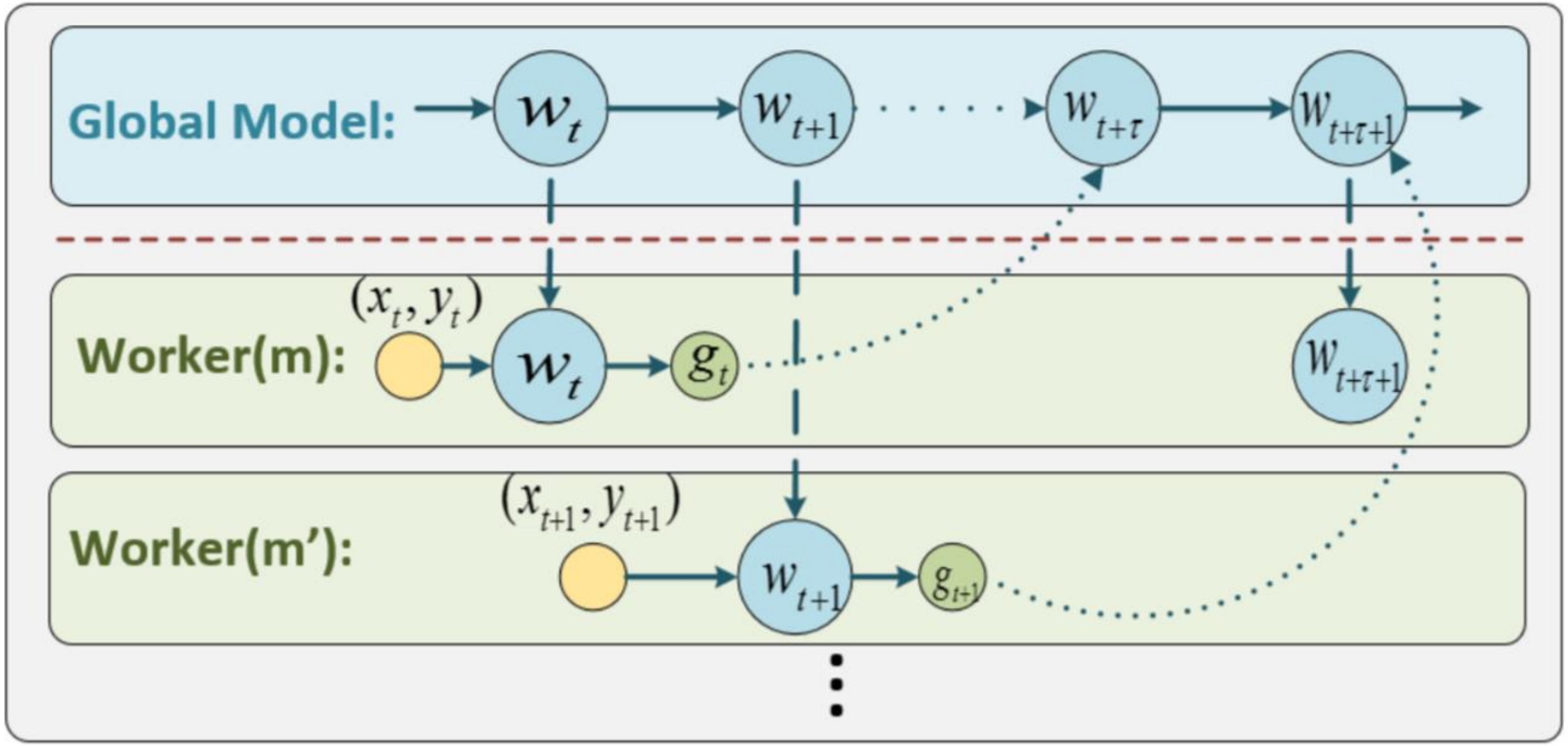
Joint work with Shuxin Zheng, Qi Meng, Taifeng Wang, Zhi-Ming Ma, and Tie-Yan Liu.

Distributed Deep Learning

- Big data + Big model ≫ Capacity of a single machine



Asynchronous SGD

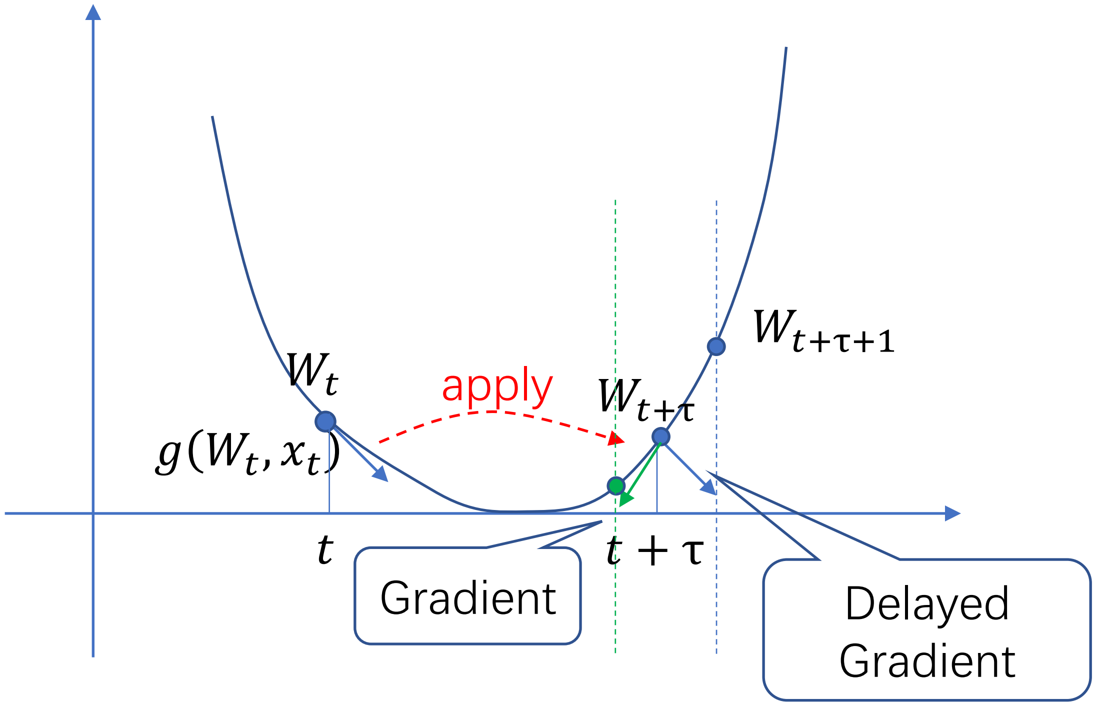


ASGD Training Process

Delayed Gradient

- SGD
  - $W_{t+\tau+1} = W_{t+\tau} - \eta * g(W_{t+\tau}, x_t)$
- Async SGD
  - $W_{t+\tau+1} = W_{t+\tau} - \eta * g(W_t, x_t)$

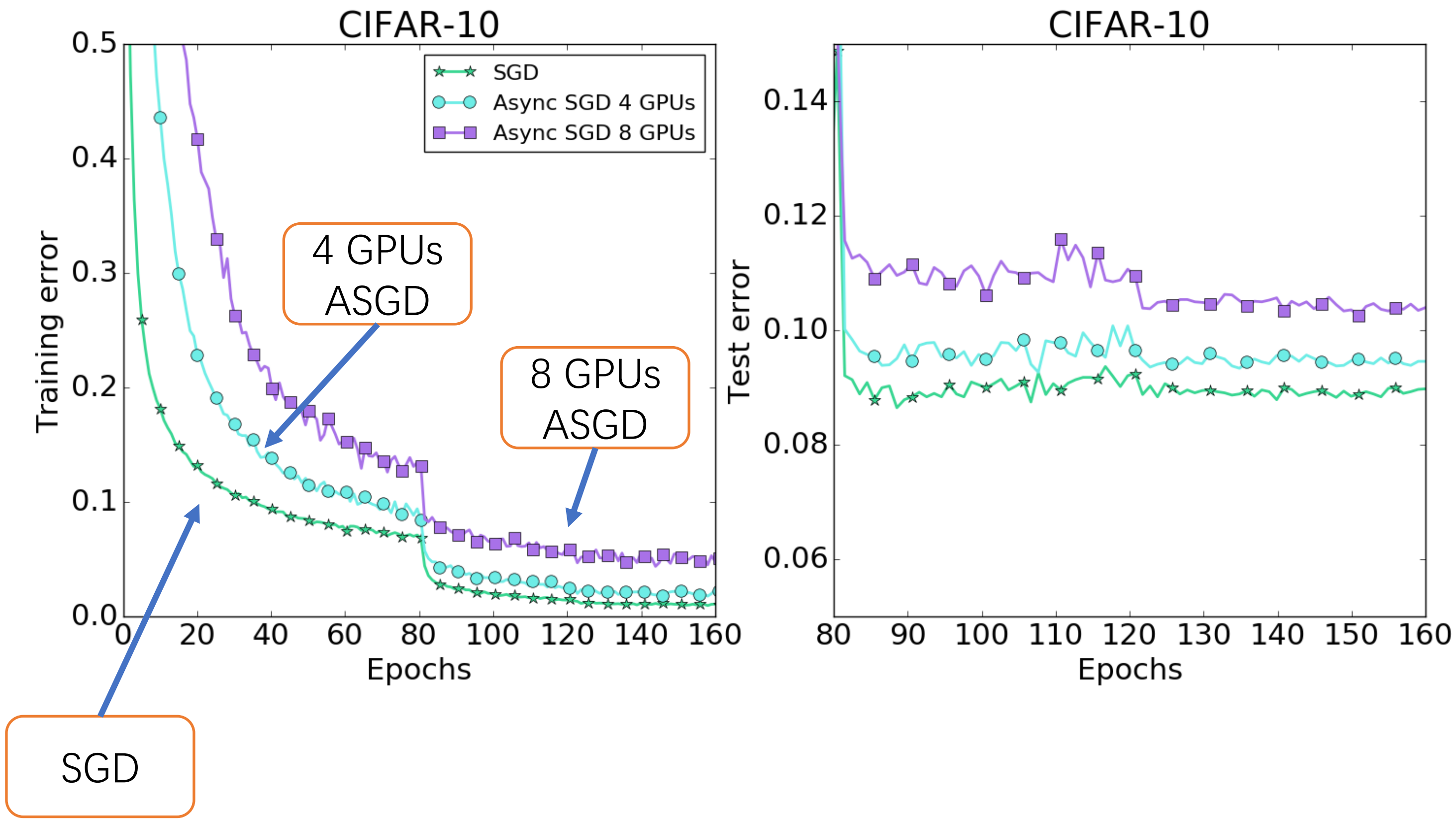
$g(W_{t+\tau}, x_t) \neq g(W_t, x_t)$



Delayed Gradient

- ResNet 20
- CIFAR-10
- 1/4/8 GPUs

# workers	algorithm	error(%)
1	SGD	8.65†
4	ASGD	9.27
8	ASGD	10.26



Delay Compensation in ASGD

$$g(W_{t+\tau}) \neq g(W_t)$$

- Taylor Expansion at w_t

$$g(W_{t+\tau}) = g(W_t) + \nabla g(W_t) \cdot (W_{t+\tau} - W_t) + \frac{1}{2}(W_{t+\tau} - W_t)^T \boldsymbol{H} g(W_t) \cdot (W_{t+\tau} - W_t) + O(||W_{t+\tau} - W_t||^3)$$

Async SGD : 0^{th} Order Approximation

$$\boldsymbol{g(W_{t+\tau}) = g(W_t)}$$

Delay

Delay Compensation in ASGD

$$g(W_{t+\tau}) \neq g(W_t)$$

- Taylor Expansion at w_t

$$g(W_{t+\tau}) = g(W_t) + \nabla g(W_t) \cdot (W_{t+\tau} - W_t) + \frac{1}{2}(W_{t+\tau} - W_t)^T \boldsymbol{H} g(W_t) \cdot (W_{t+\tau} - W_t) + O(||W_{t+\tau} - W_t||^3)$$

Delay Compensated Gradient: 1st Order Approximation

$$g(W_{t+\tau}) = g(W_t) + \nabla g(W_t) \cdot (W_{t+\tau} - W_t)$$

where $g(W_t) = \nabla f(W_t)$ and $\nabla g(W_t) = \boldsymbol{H} f(W_t)$,

$\boldsymbol{H}$ is Hessian Matrix.

Hessian Matrix

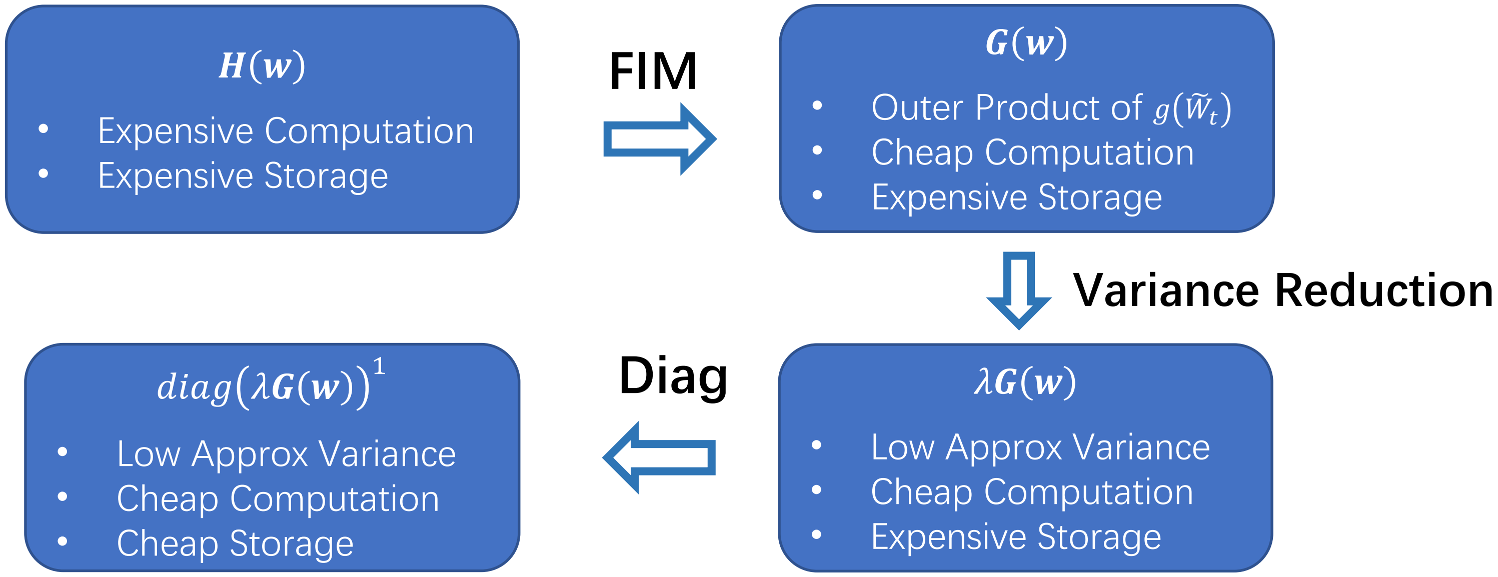
$$\mathbf{H}f(\mathbf{w}) = \begin{pmatrix} \frac{\partial^2 f}{\partial w_1 \partial w_1}(\mathbf{w}) & \frac{\partial^2 f}{\partial w_1 \partial w_2}(\mathbf{w}) & \cdots & \frac{\partial^2 f}{\partial w_1 \partial w_n}(\mathbf{w}) \\ \frac{\partial^2 f}{\partial w_2 \partial w_1}(\mathbf{w}) & \frac{\partial^2 f}{\partial w_2 \partial w_2}(\mathbf{w}) & \cdots & \frac{\partial^2 f}{\partial w_2 \partial w_n}(\mathbf{w}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial w_n \partial w_1}(\mathbf{w}) & \frac{\partial^2 f}{\partial w_n \partial w_2}(\mathbf{w}) & \cdots & \frac{\partial^2 f}{\partial w_n \partial w_n}(\mathbf{w}) \end{pmatrix}$$

$$G(\mathbf{w}_t) = \left(\frac{\partial}{\partial \mathbf{w}} f(x, y, \mathbf{w}_t)\right) \left(\frac{\partial}{\partial \mathbf{w}} f(x, y, \mathbf{w}_t)\right)^T$$

Fisher Information Matrix

$$\epsilon_t \triangleq \mathbb{E}_{(y|x,\mathbf{w}^*)} ||G(\mathbf{w}_t) - H(\mathbf{w}_t)|| \to 0, t \to \infty$$

Hessian Matrix



1. Becker S, Le Cun Y. Improving the convergence of back-propagation learning with second order methods[C]//Proceedings of the 1988 connectionist models summer school. 1988: 29-37.

DC-ASGD

ASGD:

$$W_{t+\tau+1} = W_{t+\tau} - \eta\, g(W_t)$$

Delay Compensated ASGD:

$$W_{t+\tau+1} = W_{t+\tau} - \eta\underbrace{\left(g(W_t) + \underbrace{\lambda_t g(W_t) \odot g(W_t)}_{diag(\lambda G)} \odot (W_{t+\tau} - W_t)\right)}_{\text{DC Gradient}}$$

Algorithm

Algorithm 1 DC-ASGD: worker m

repeat
  Pull $\mathbf{w}_t$ from the parameter server.
  Compute gradient $g_m = \nabla f_m(\mathbf{w}_t)$.
  Push g_m to the parameter server.
until $forever$

Algorithm 2 DC-ASGD: parameter server

Input: learning rate η, variance control parameter λ_t.
Initialize: $t = 0$, $\mathbf{w}_0$ is initialized randomly, $\mathbf{w}_{bak}(m) = \mathbf{w}_0$, $m \in \{1, 2, \cdots, M\}$
repeat
  if receive "g_m" **then**
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta \cdot \left(g_m + \lambda_t g_m \odot g_m \odot (\mathbf{w}_t - \mathbf{w}_{bak}(m))\right)$
    $t \leftarrow t + 1$
  else if receive "pull request" **then**
    $\mathbf{w}_{bak}(m) \leftarrow \mathbf{w}_t$
    Send $\mathbf{w}_t$ back to worker m.
  end if
until $forever$

Convergence Rate

- DC-ASGD and ASGD will converge at the same rate.

$$O\left(\frac{V}{\sqrt{Tb}}\right)$$

*More details in paper

- DC-ASGD has larger tolerant for delay τ.

$$\tau \leq \min\left\{\frac{L_2\gamma}{C_\lambda}, \frac{\gamma}{C_\lambda}, \frac{\sqrt{T}\gamma}{\tilde{C}}, \frac{L_2 T\gamma}{4\tilde{C}}\right\}$$

  - where $\gamma = \sqrt{\frac{L_2 T V^2}{2D_0 b}}$ is the upper-bound of ASGD τ.

Experiments

- ResNet 20
- CIFAR-10
- 1/4/8 GPUs

# workers	algorithm	error(%)
1	SGD	8.65†
4	ASGD	9.27
	SSGD	9.17
	DC-ASGD-c	8.67
	DC-ASGD-a	**8.19**
8	ASGD	10.26
	SSGD	10.10
	DC-ASGD-c	9.27
	DC-ASGD-a	**8.57**

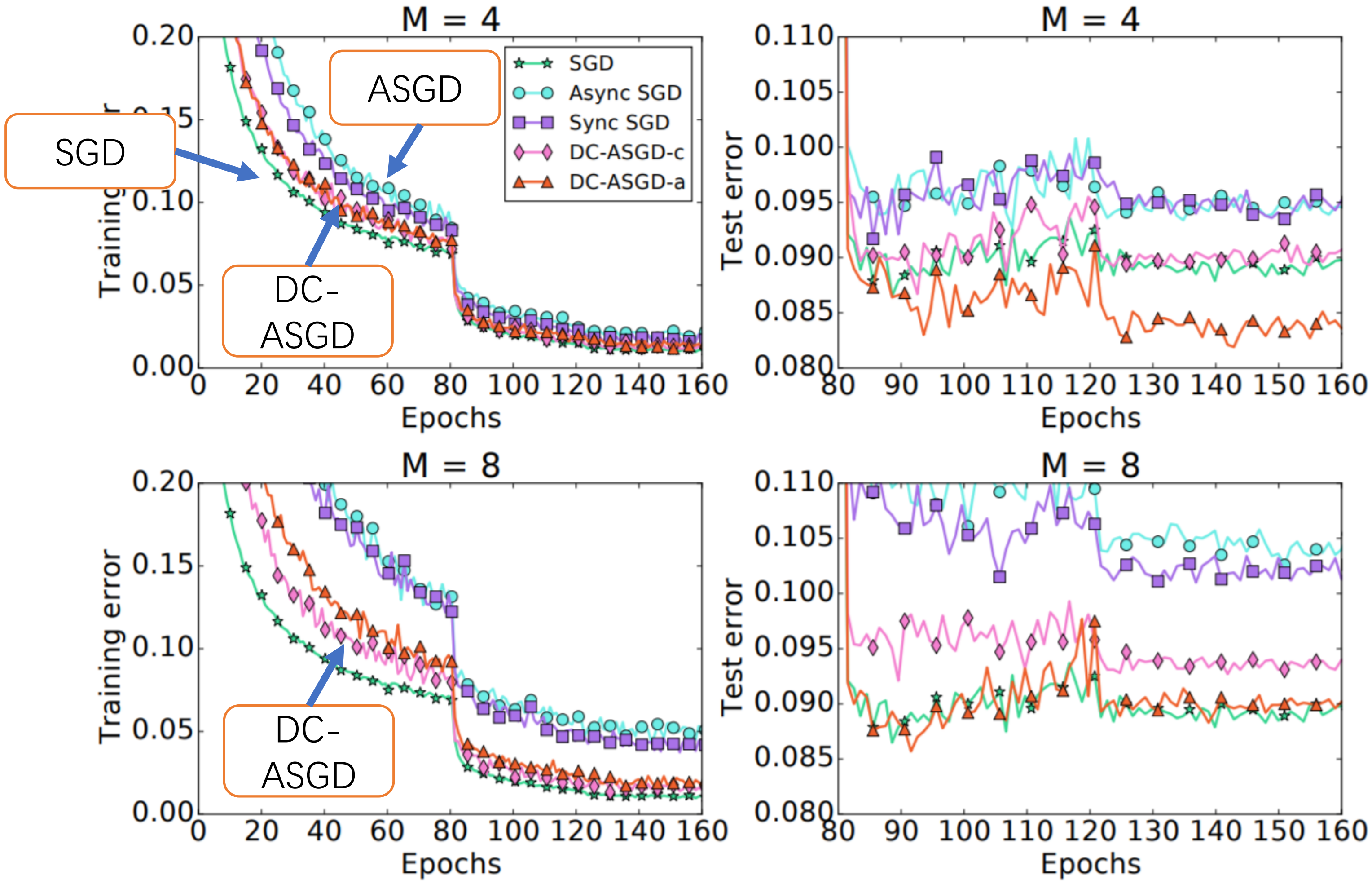


Figure 2. Error rates of the global model w.r.t. number of effective passes of data on CIFAR-10

Experiments

- ResNet 20
- CIFAR-10
- 1/4/8 GPUs

# workers	algorithm	error(%)
1	SGD	8.65†
4	ASGD	9.27
	SSGD	9.17
	DC-ASGD-c	8.67
	DC-ASGD-a	**8.19**
8	ASGD	10.26
	SSGD	10.10
	DC-ASGD-c	9.27
	DC-ASGD-a	**8.57**

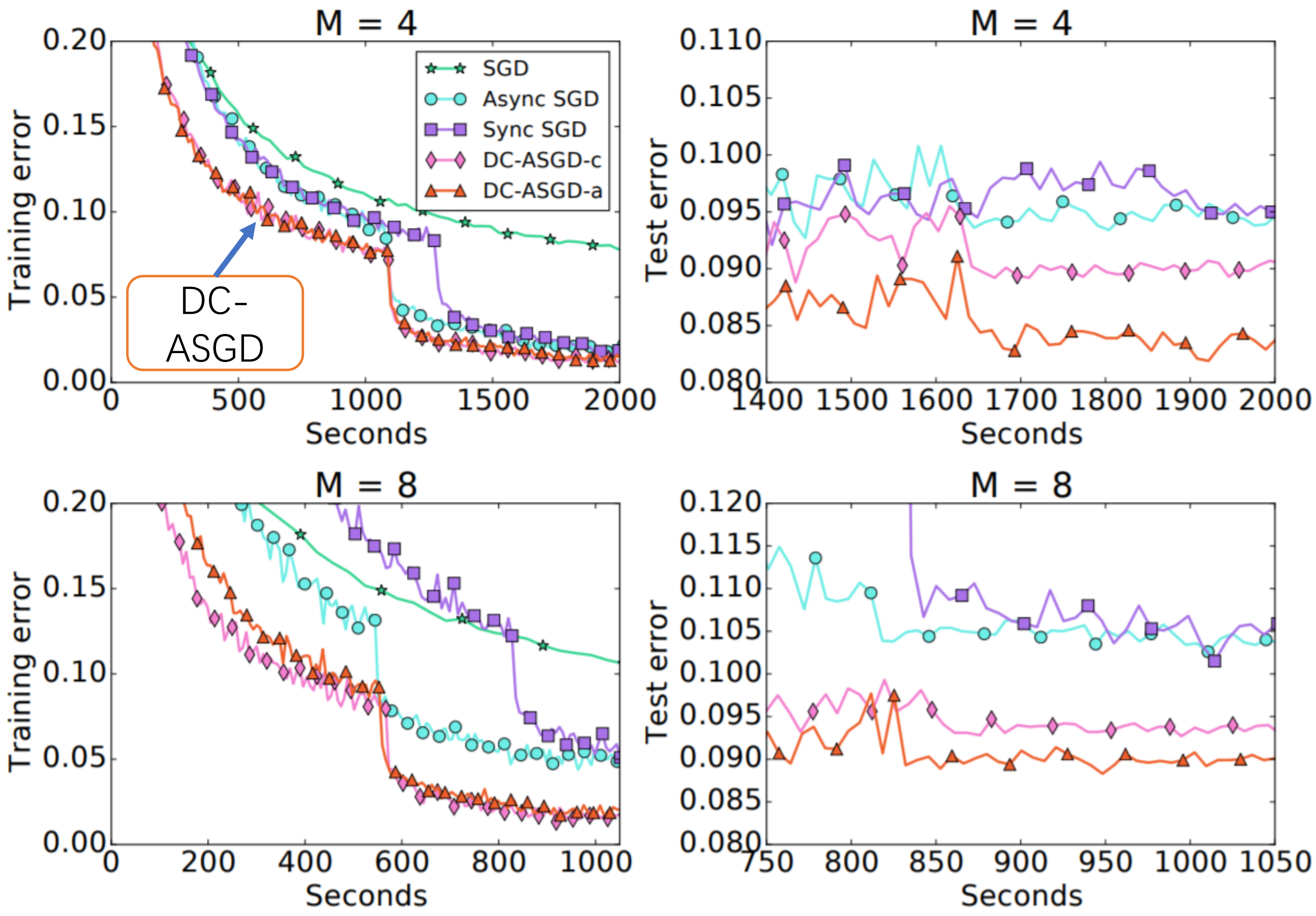


Figure 3. Error rates of the global model w.r.t. wallclock time on CIFAR-10

Experiments

- ResNet 50
- ImageNet1K
- 16 GPUs

# workers	algorithm	error(%)
16	ASGD	25.64
	SSGD	25.30
	DC-ASGD-a	**25.18**

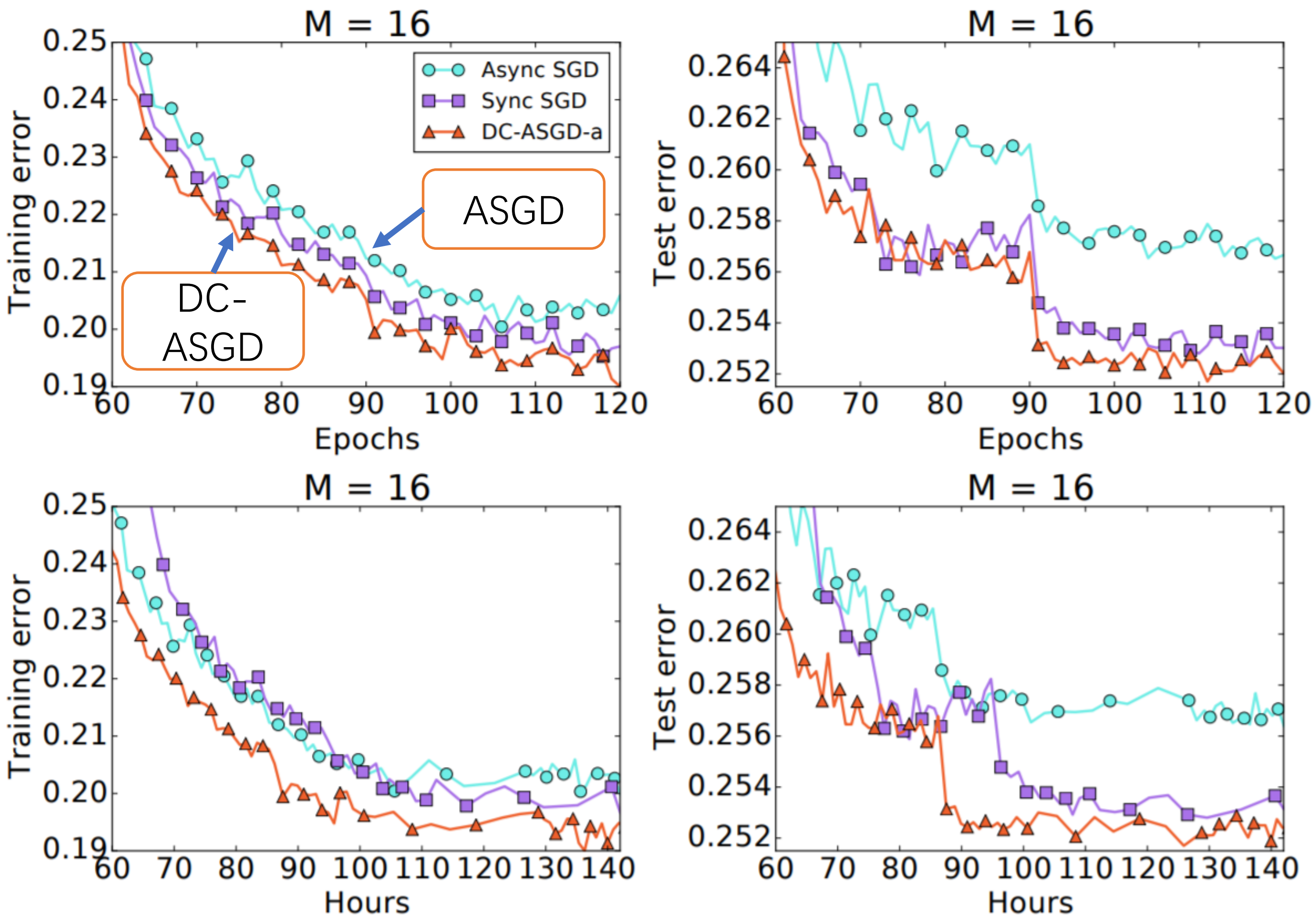


Figure 4. Error rates of the global model w.r.t. both number of effective passes and wallclock time on ImageNet

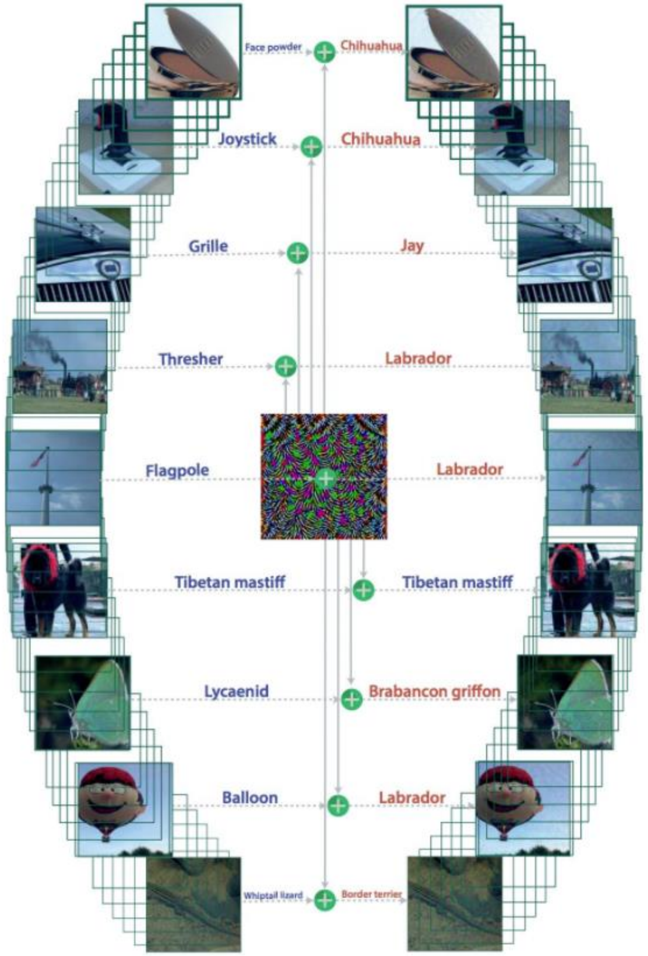
OOD Prediction: Causal Learning

Recovering Latent Causal Factor for Generalization to Distributional Shifts (NeurIPS’ 2021)

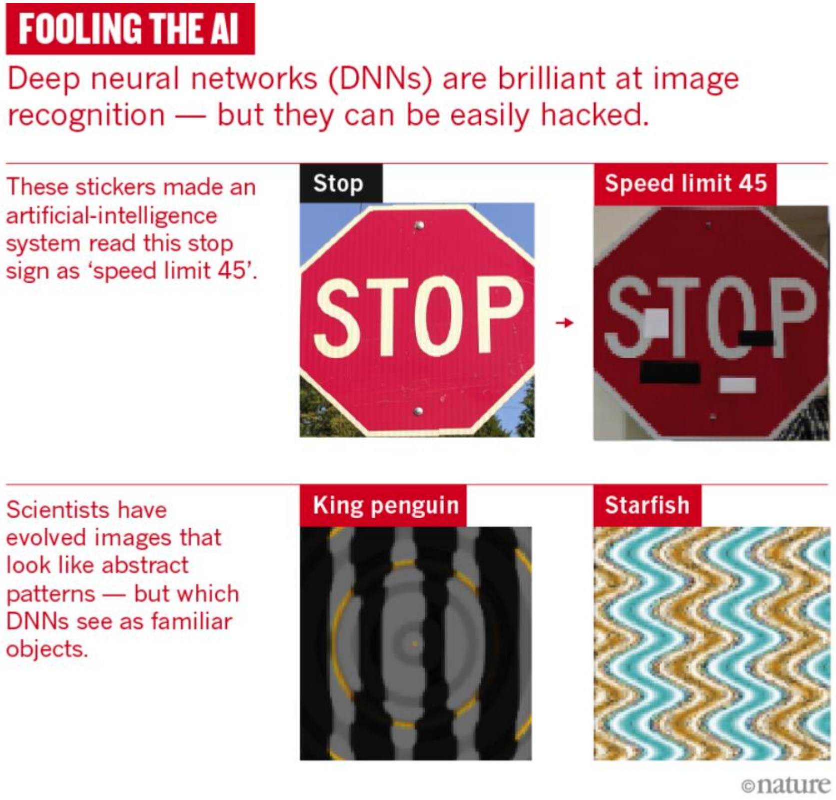
Joint work with Xinwei Sun, Chang Liu, Botong Wu, Xiangyu Zheng, Tao Qin, and Tie-Yan Liu

Code

Interpretability, Robustness, and Reliability



"[Universal adversarial perturbations]", by Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard.

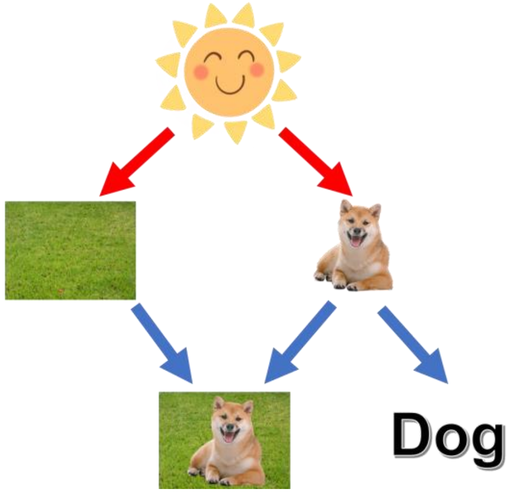


Adversarial Perturbations
=> "Reliability"

Out-of-distribution Instances
=> "Robustness"

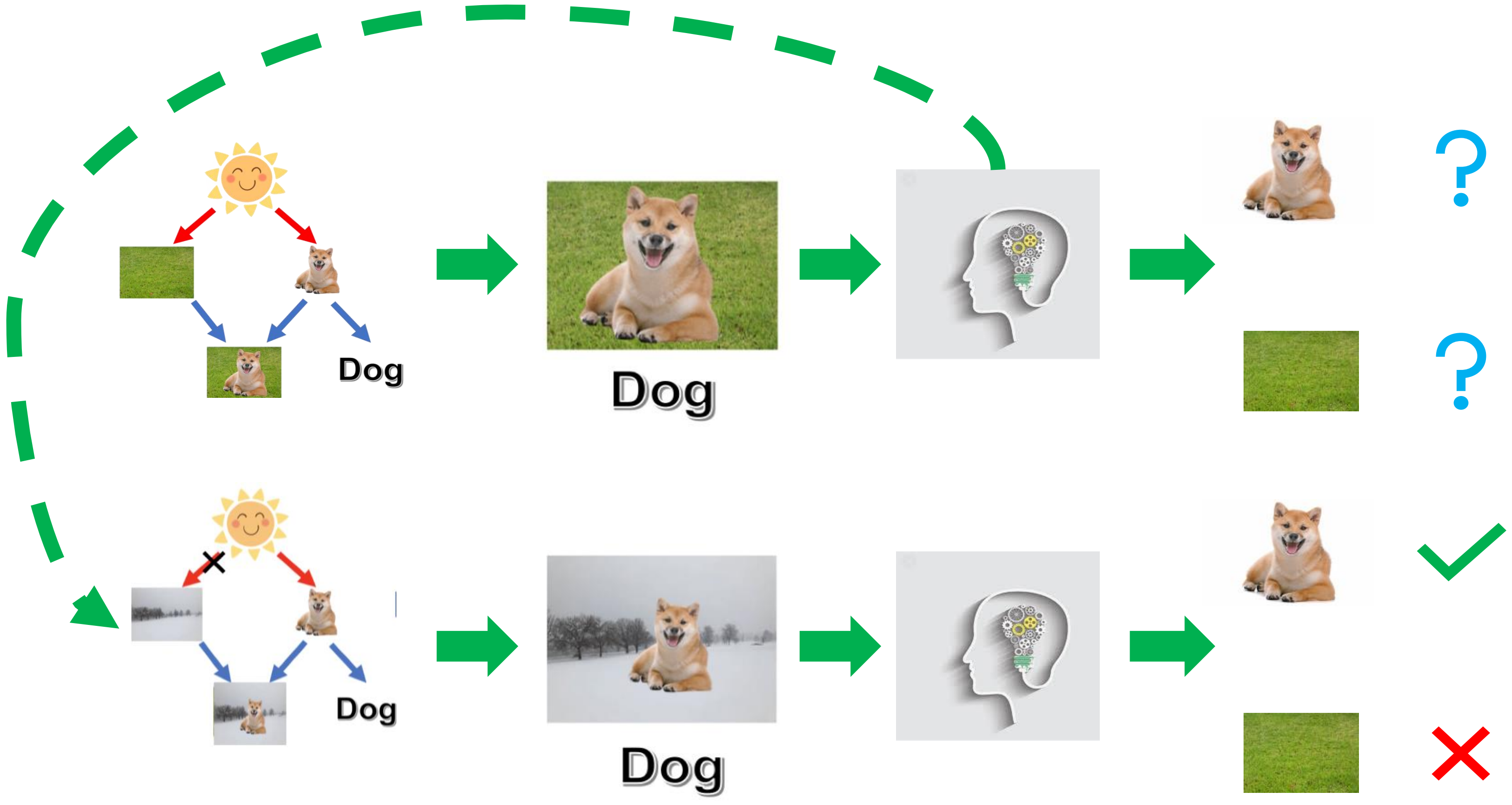
Why “interpretability” matters?

- To avoid DNN models learn the spurious correlations



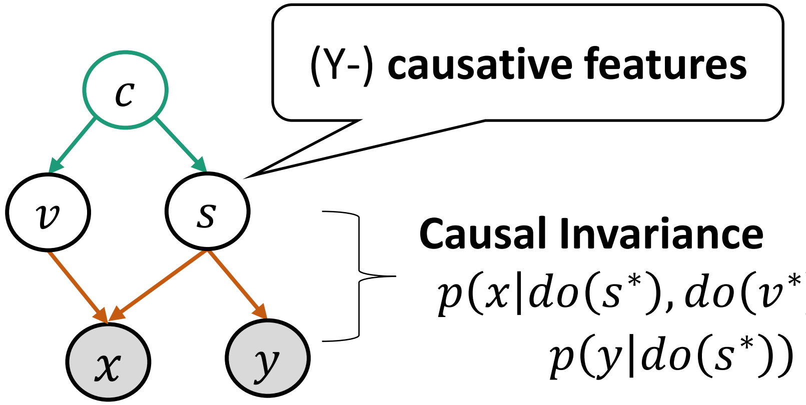
An example for sampling bias inherited from data.

Intervention brings Causation



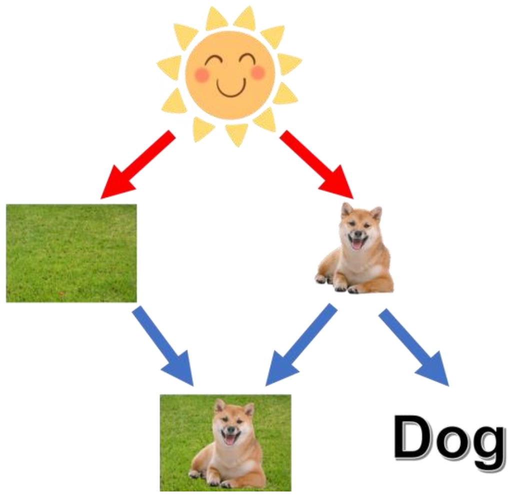
Ingredients for identifying causation:
(soft)-intervention or diverse extent of correlation.

The Causal Model



Causal Invariance

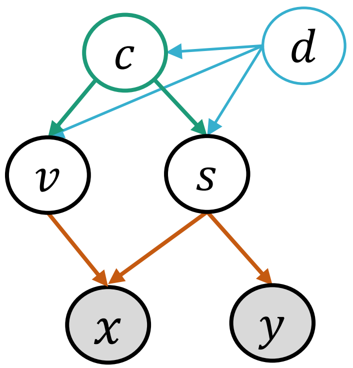
$$p(x|do(s^*), do(v^*)) = p_{f_x}(x|s^*, v^*)$$
$$p(y|do(s^*)) = p_{f_y}(y|s^*)$$



The structural causal model $M \coloneqq \big(G, \mathcal{F}, P(\epsilon)\big)$,

- G is the causal graph
- $\mathcal{F} := (f_x, f_y, f_s, f_v, f_c)$ is the data generating function (e.g., $f_s(c, \epsilon_s)$ denotes the generation of S)
- ϵ denotes the values of all the unobservable variables

Out-of-Distribution Prediction



Given $\{\mathcal{D}^e := \{x_i^e, y_i^e\}_{i\in[n_e]}\}_{e\in\mathcal{E}_{train}}$.

Goal: Learn f that generalizes well to $\mathcal{E} \sqsupset \mathcal{E}_{train}$

Latent **C**ausal **I**nvariant **M**odel (LaCIM)

Causal Invariance

$p_{f_x}(x|s^*, v^*) \to x$

$p_{f_y}(y|s^*) \to y$

Given x, inference s^* from $p_{f_x}(x|s,v)$ and $p_{f_y}(y|s^*)$ for Prediction

A set of structural causal models $M^e := (G, \boldsymbol{\mathcal{F}^e}, P(\epsilon))$ augmented with the **domain variable** $\boldsymbol{D}$

- G is the causal graph
- For each $e \in \mathcal{E}$, the $\boldsymbol{\mathcal{F}^e} := \{f_x, f_y, f_s^e, f_v^e, f_c^e\}$ denotes the generating mechanism of X, Y, S, V, C.
- ϵ denotes the values of all the unobservable variables,

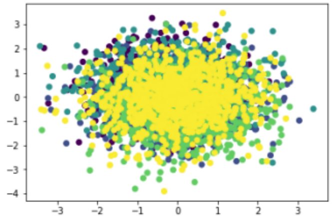
Identifiability of LaCIM

Definition: (Pearl, 2000) A quantity $Q(M)$ is identifiable, given a set of (causal) assumptions A, if two models M_1 and M_2 that satisfy A, we have

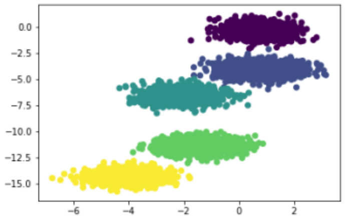
$$P(M_1) = P(M_2) \Rightarrow Q(M_1) = Q(M_2)$$

Theorem: Suppose the S- V correlation in multiple datasets $\mathcal{D}^e := \{x_i^e, y_i^e\}_{e \in \mathcal{E}}$ are diverse enough and the noise in additive, then for any $x \leftarrow f_x(s^\star, v^\star), y \leftarrow f_y(v^\star)$, such that:

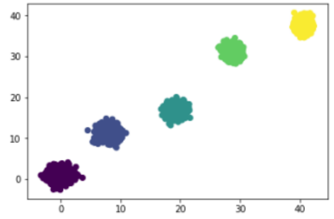
- Identifiability of Causal Factor: there exists an invertible function $h, \text{s.t.}, \tilde{s} = h(s^\star)$.
- Identifiability of Invariant Predictor: $\tilde{p}(y|\tilde{s}) = p^\star(y|s^\star)$



Pool-LaCIM

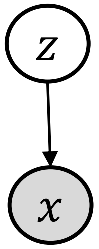


LaCIM



$p_{\theta^*}(s|D)$

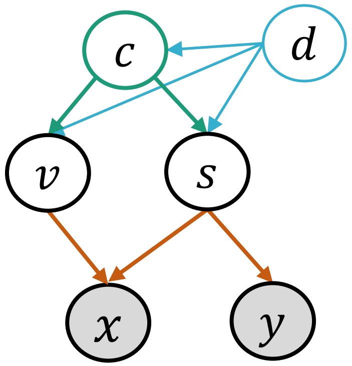
Learning Method



Prior $p_\theta(z)$
Posterior $p_\theta(z|x)$
Variational distribution $q_\psi(z|x)$
Likelihood $p_\theta(x|z)$

Variational AutoEncoder (VAE)
Max ELBO:

$$-\mathcal{L}_{\theta,\psi} = E_{p(x)}\left[E_{q_\psi(z|x)} \log \frac{p_\theta(x,z)}{q_\psi(z|x)}\right]$$



LaCIM

Variational distribution
$q_\psi^e(s,v|x,y)$
$= q_\psi(s,v|x,y,d)$

Train

$$\mathcal{L}^e(\theta,\psi) = \mathbb{E}_{p^e(x,y)}\left(\mathbb{E}_{q_\psi^e(s,v|x,y)} \log \frac{p_\theta^e(x,y,s,v)}{q_\psi^e(s,v|x,y)}\right)$$

where $p_\theta^e(x,y,s,v) = p_\theta(x|s,v)p_\theta(y|s)p^e(s,v)$.

By our causal model, we re-parameterize q_ψ^e as below,

$$q_\psi^e(s,v|x,y) = \frac{q_\psi^e(s,v|x)q_\psi^e(y|s)}{q_\psi^e(y|x)}$$

Inference

1. Infer s^* from $\max_{s,v} \log p_\theta(x|s,v) + \lambda \mathrm{J}(s,z)$
2. Predict via $\mathrm{argmax}_y \mathrm{p}_\theta(y|s^*)$

Experimental Results

Table 1: Accuracy (%) on test domain. Average over 10 runs.

Dataset / Method	NICO				CMNIST		ADNI ($m=2$)		
	$m=8$		$m=14$		$m=2$		D: Age	D: TAU	# Params
	ACC	# Params	ACC	# Params	ACC	# Params	ACC	ACC	
ERM	60.3 ± 2.8	18.08M	59.3 ± 2.1	18.08M	91.9 ± 0.9	1.12M	62.1 ± 3.2	64.3 ± 1.0	28.27M
DANN	58.9 ± 1.7	19.13M	60.1 ± 2.6	26.49M	84.8 ± 0.7	1.1M	61.0 ± 1.5	65.2 ± 1.1	30.21M
MMD-AAE	60.8 ± 3.4	19.70M	64.8 ± 7.7	19.70M	92.5 ± 0.8	1.23M	60.3 ± 2.2	65.2 ± 1.5	36.68M
DIVA	58.8 ± 3.4	14.86M	58.1 ± 1.4	14.87M	86.1 ± 1.0	1.69M	61.8 ± 1.8	64.8 ± 0.8	33.22M
IRM	61.4 ± 3.8	18.08M	62.8 ± 4.6	18.08M	92.9 ± 1.2	1.12M	62.2 ± 2.6	65.2 ± 1.1	28.27M
sVAE	60.4 ± 2.1	18.25M	64.3 ± 1.2	19.70M	93.6 ± 0.9	0.92M	62.7 ± 2.5	66.6 ± 0.8	37.78M
LaCIM (**Ours**)	$\mathbf{63.2 \pm 1.7}$	18.25M	$\mathbf{66.4 \pm 2.2}$	19.70M	$\mathbf{96.6 \pm 0.3}$	0.92M	$\mathbf{63.8 \pm 1.1}$	$\mathbf{67.3 \pm 0.9}$	37.78M

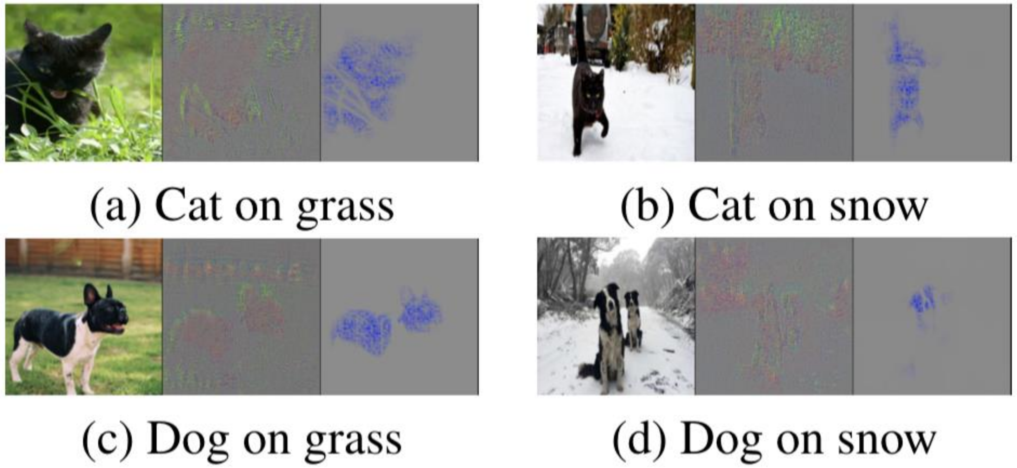
Visualization

LaCIM

S:

V:

ERM LaCIM ERM LaCIM



Discussions

- All models are wrong, but some are useful. The causal graph should rely on the belief of generating process and the definition of Y.
- Duality b/w causal discovery (“link the nodes”) and causal representation learning (“fill in the blanks”).

Thanks!

chenwei2022@ict.ac.cn
https://weichen-cas.github.io/