PSVL

Scaling up AI Model Deployments with the Limited and Imperfectly Labeled Data

April 13, 2021

Denis Gudovskiy
denis.gudovskiy@us.panasonic.com
Panasonic AI Lab, Mountain View

Panasonic AI

Panasonic Business Fields

Appliance Company


Air Conditioner


Hair Dryer


Television


Showcase

Life Solutions Company


Lighting Equipment


Wiring Equipment


Air Purifier


System Kitchen

Connected Solutions Company


Avionics Entertainment System


Factory Line Management System


Projector/Laptop/Tablet

Automotive Company


Cockpit System



Automotive Lithium-ion Battery

Industrial Solutions Company


Control Equipment

FA Devices

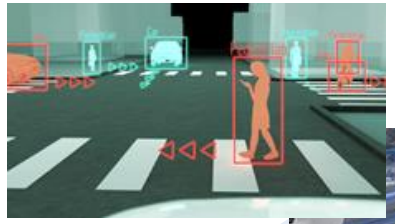

Electronic Devices


Multilayer Substrate Material


Dry Cell


Power Storage Module

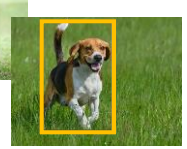
ADAS system for Car OEM
Autonomous Commuter




Security

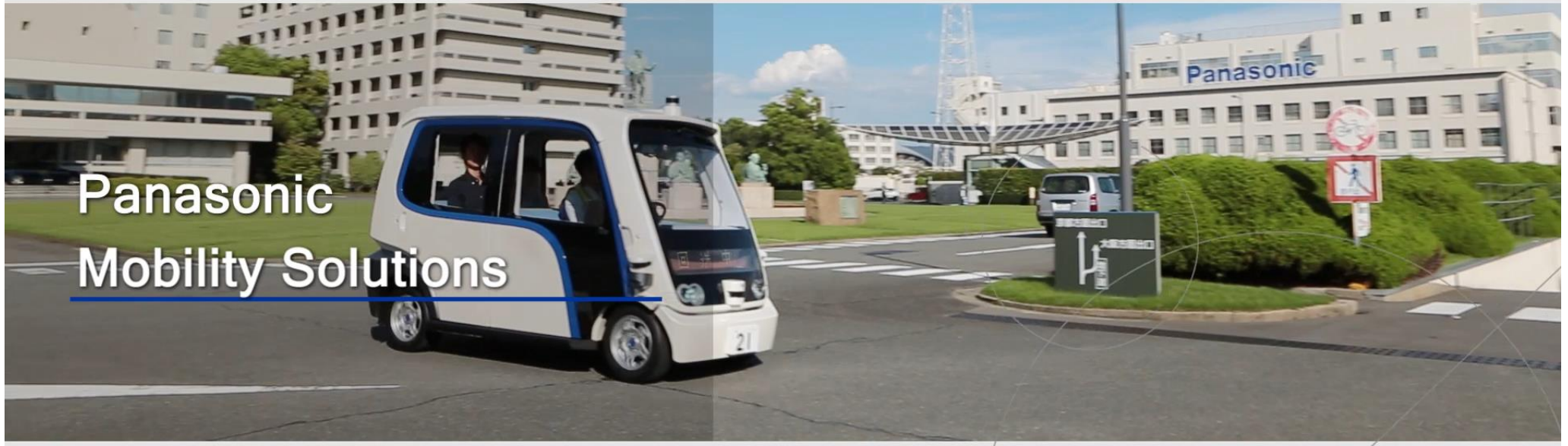


Imaging

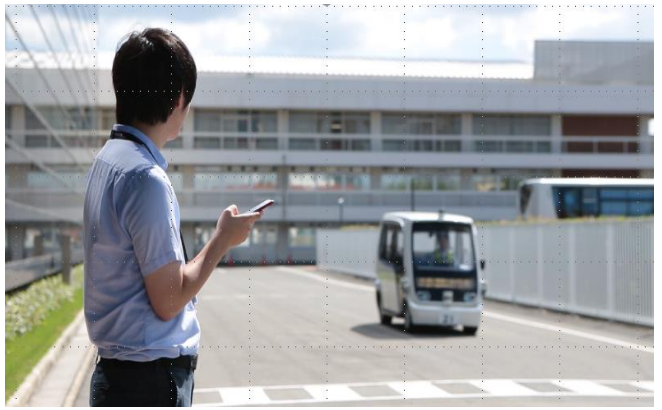

Air Conditioner
Cold Chain


Autonomous Ride Share Service



AI Development Process

Large scale multimodal dataset for living space



Real-time detection for self-driving mobility service on an embedded device

Data

Embedded AI

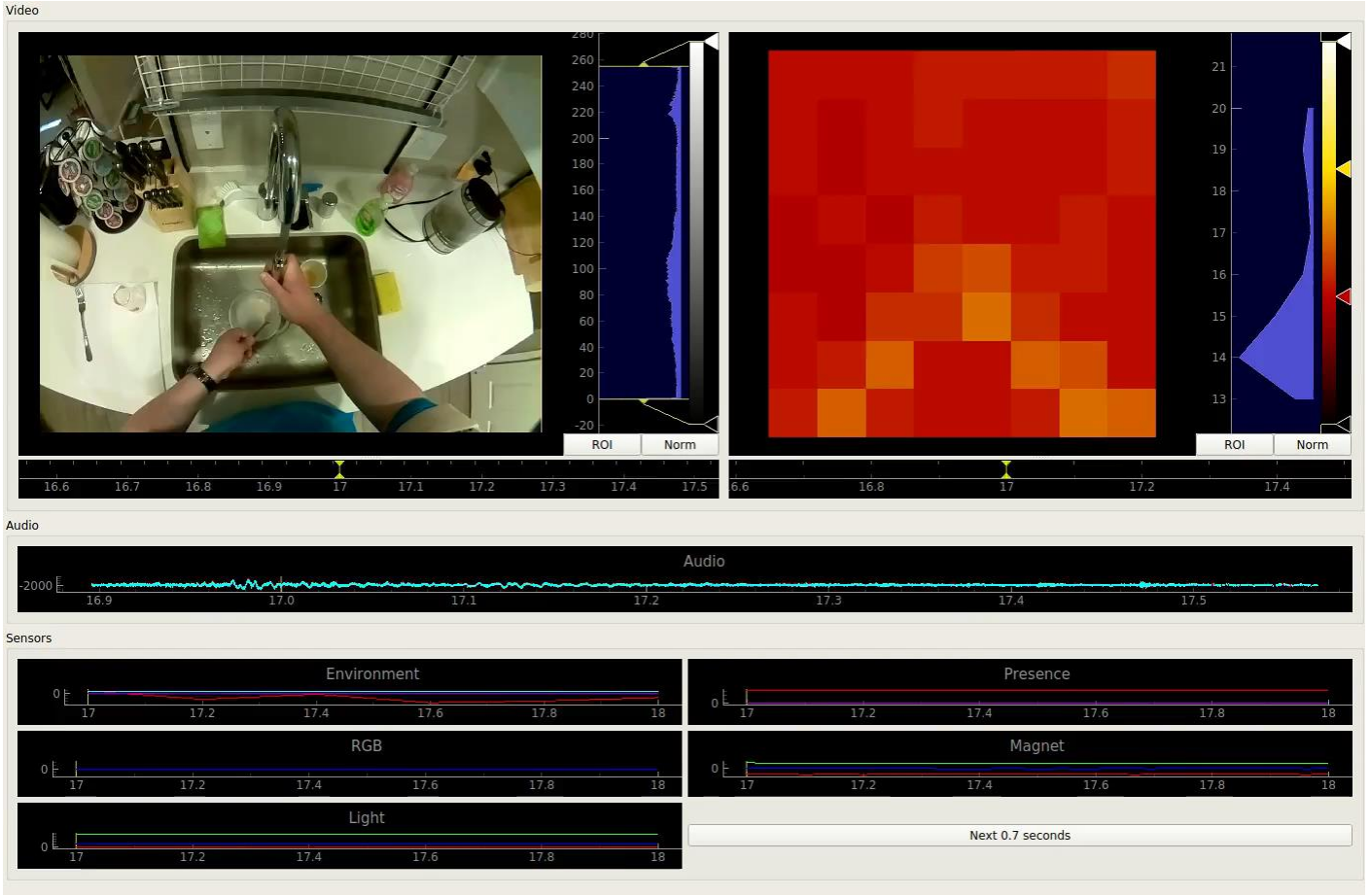
Planning → Data design/ Collection → PoC/ Modeling → System building/ Deployment → Quality Assurance → Operation

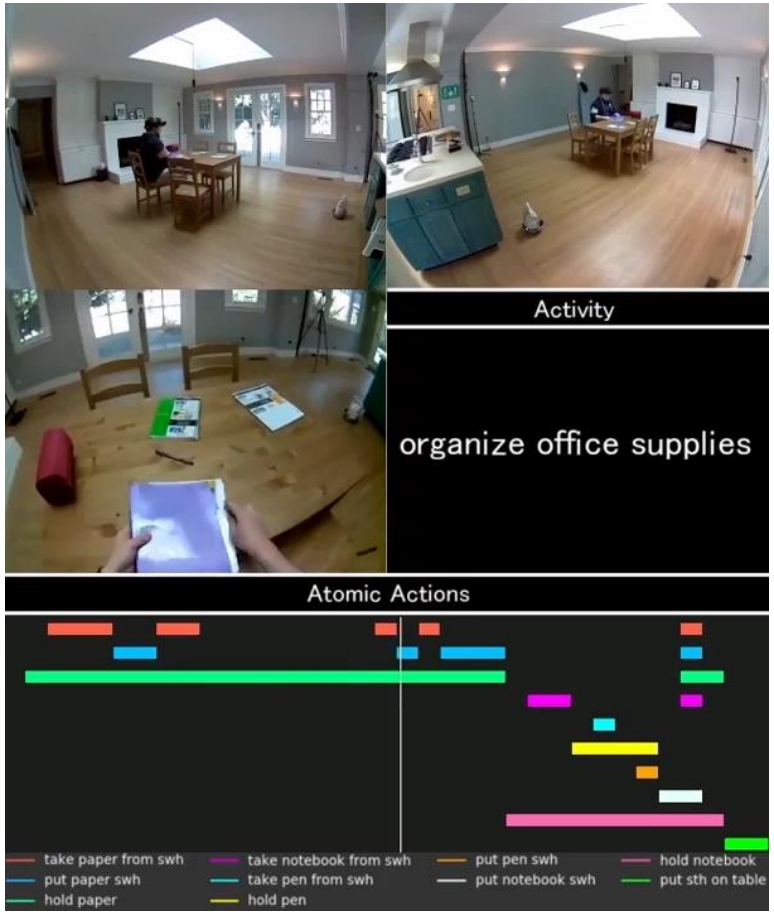
Home Action Genome Data Example

Home Action Genome (HOMAGE) [CVPR2021]:
A large-scale multi-view video database of indoor daily activities

https://homeactiongenome.org/

This is a collaboration work with Stanford Vision and Learning Lab





Home Action Genome Data Example

HOMAGE encompasses the indoor actions and multimodal data

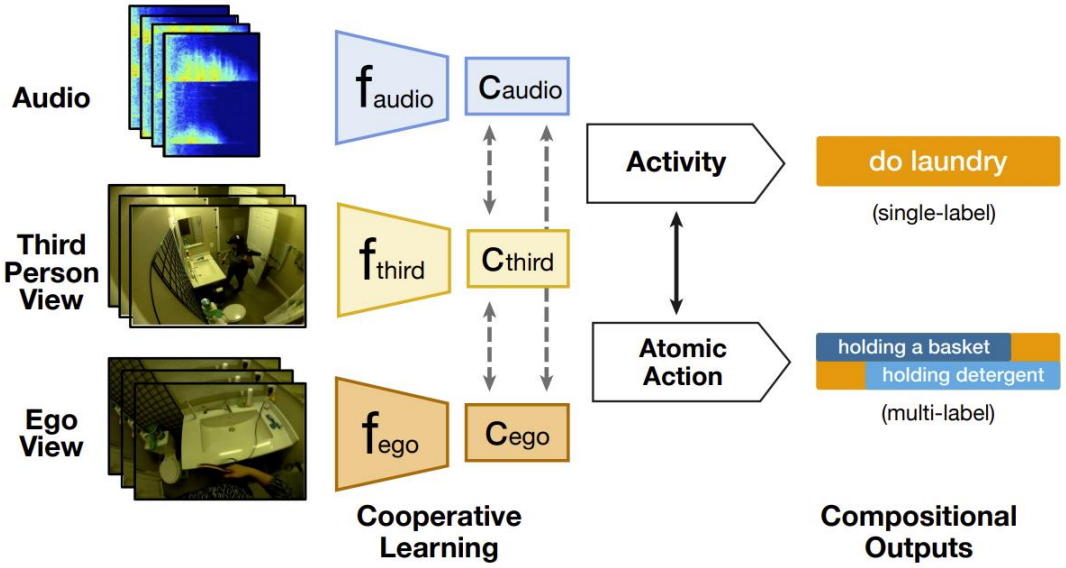
70 activity classes	Camera	Microphone	Thermal Sensor	RGB Light Sensor
453 atomic action classes				
Multi-View including egocentric view	Presence Sensor	9-axis IMU (accel, gyro, magnet)	Light Sensor	Environment Sensor

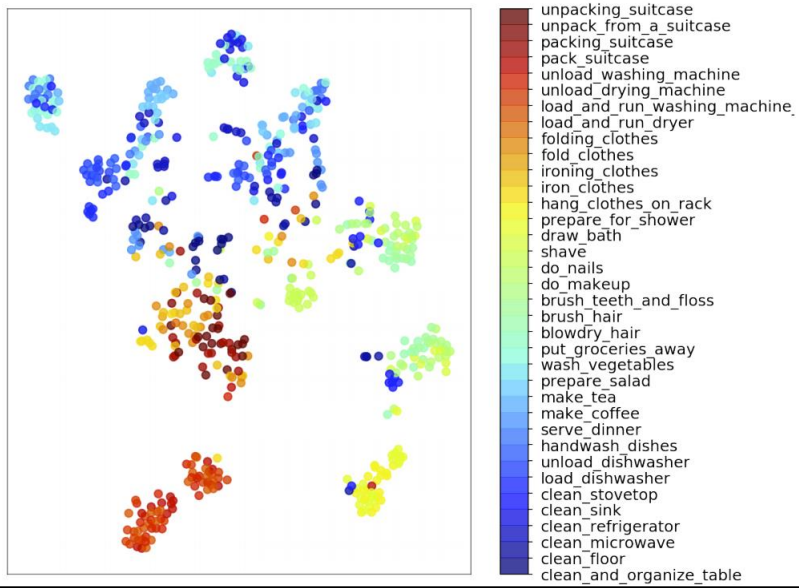
HOMAGE is a guest task of ActivityNet Challenge at CVPR2021 workshop

http://activity-net.org/challenges/2021/

HOMAGE workshop has been accepted to ICCV2021: http://campworkshop.org

Home Action Genome Data Example





Multimodal Cooperative Compositional Action Understanding (CCAU)

Method	Acc			mAP		
	Audio	Ego	3rd Person	Audio	Ego	3rd Person
Cooperative - Activity	28.3	31.1	17.0	-	-	-
Cooperative - Atomic Actions	-	-	-	5.9	18.5	9.5
Compositional	23.5	32.1	16.2	16.4	26.3	12.2
Cooperative Compositional	**29.3**	**34.9**	**19.2**	**21.7**	**29.3**	**13.8**

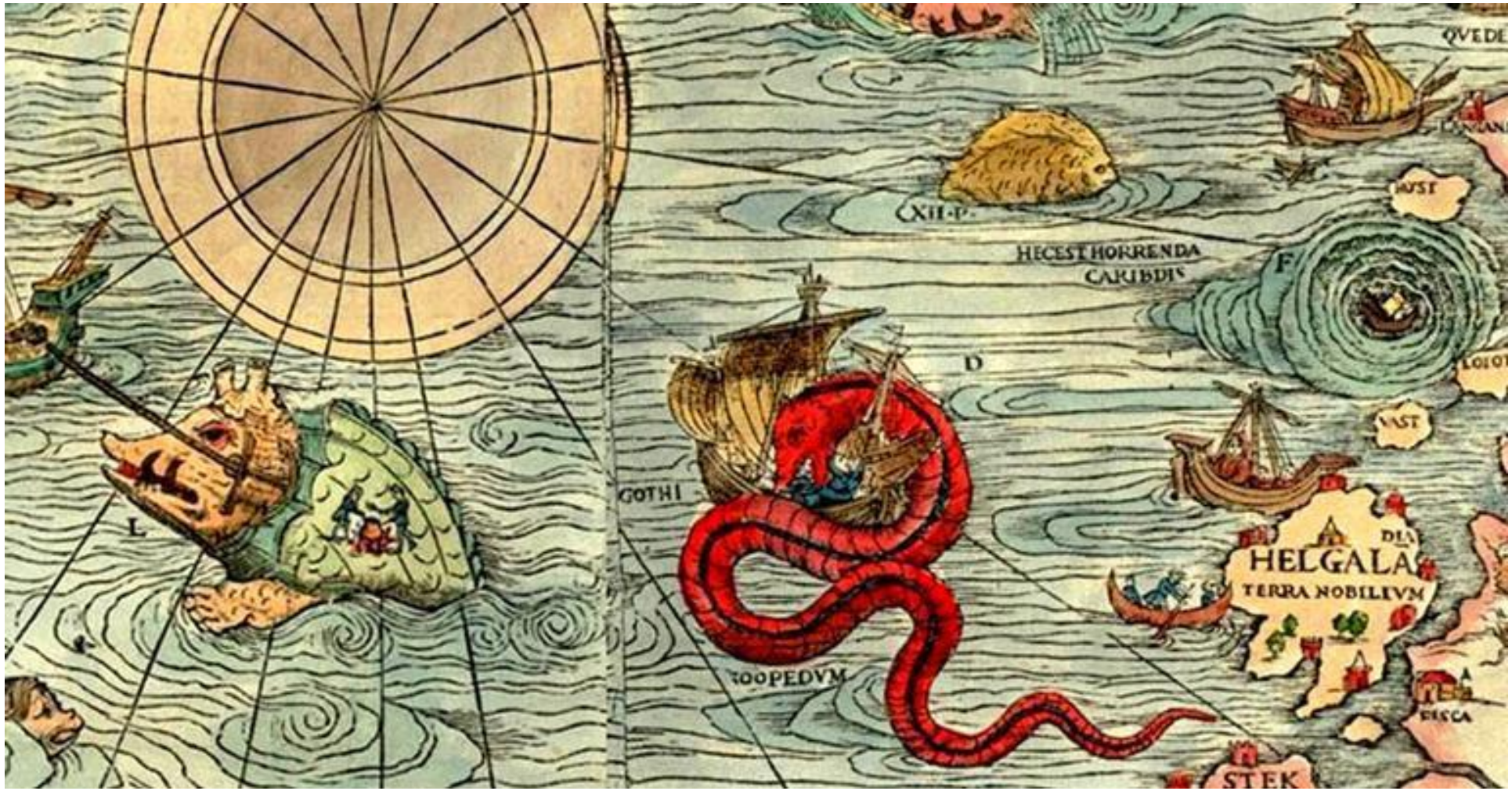
Few-shot learning

Method - Ego	Atomic Action - mAP			
	1 shot	5 shot	10 shot	20 shot
Single Modality	22.4	35.3	38.6	40.6
CCAU	**28.6**	**36.9**	**39.4**	**49.4**

Hic Sunt Dracones et AI - Here be Dragons and AI

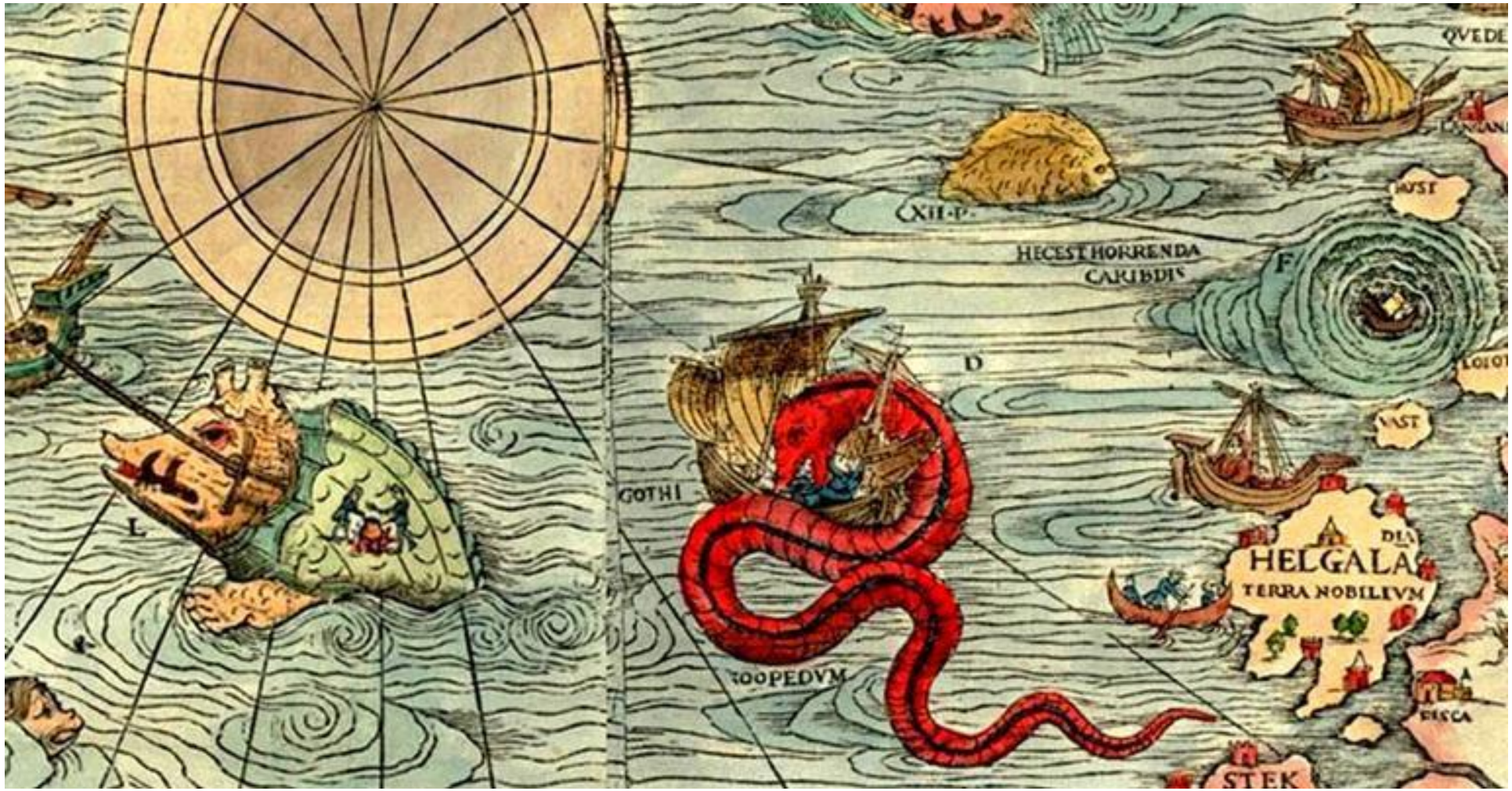
The explored world – train data

Scary dragons live in the test data



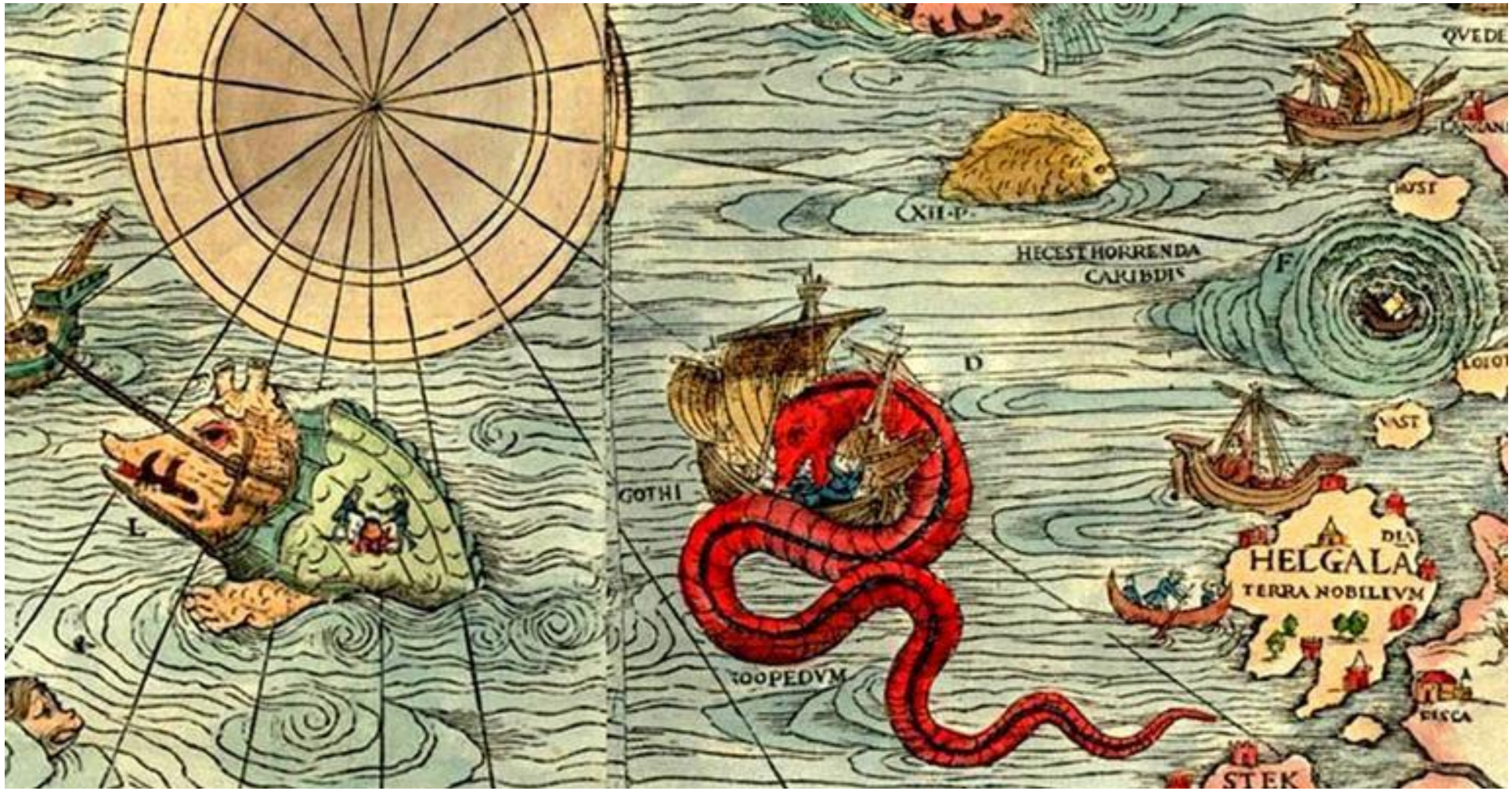
Hic Sunt Dracones et AI - Here be Dragons and AI

AI models minimize risk for the empirical train datasets…
and empirical datasets are always limited, biased and noisy…



Hic Sunt Dracones et AI - Here be Dragons and AI

…but it is our job to make them better and, hence, our models more robust.
Therefore, I encourage researchers to go beyond standard dataset setups!



Zoo of Relevant Methods and Industrial Applications

- Scaling AI applications to new domains is tough:
  - Labeling is noisy and expensive
  - Sometimes, labeling is not even possible
  - Models are not robust to unseen/rare test data
  - Safety-critical apps are the most vulnerable and leads to geo-fencing, remote operators etc.
- ML research community tries to catch up with:
  - Un/self/semi-supervised learning/pretraining
  - Active and zero/few-shot learning-assisted labeling
  - Augmentation methods optimize labeled datasets
  - Domain adaptation: train-time and test-time
  - Generative modeling: syn2real, image-to-image translation etc.
- However, such methods usually are limited to:
  - Simple tasks: classification vs. 2D/3D object detection/tracking/segmentation, scene parsing
  - Simple data domains: sparse unimodal vs. dense multimodal
  - Standard public datasets: balanced train/test vs. biased data with label noise

PSVL

CVPR SEATTLE WASHINGTON JUNE 16-18 2020

Deep Active Learning for Biased Datasets via Fisher Kernel Self-Supervision

Denis Gudovskiy[1] Alec Hodgkinson[1] Takuya Yamaguchi[2] Sotaro Tsukizawa[2]

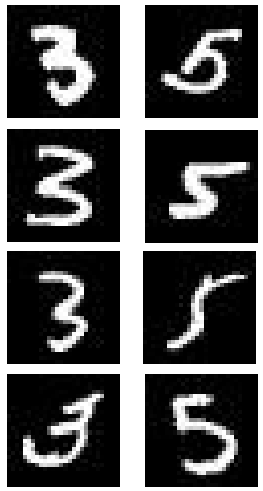
[1]Panasonic AI Lab, Mountain View, CA
[2]Panasonic AI Solutions Center, Osaka

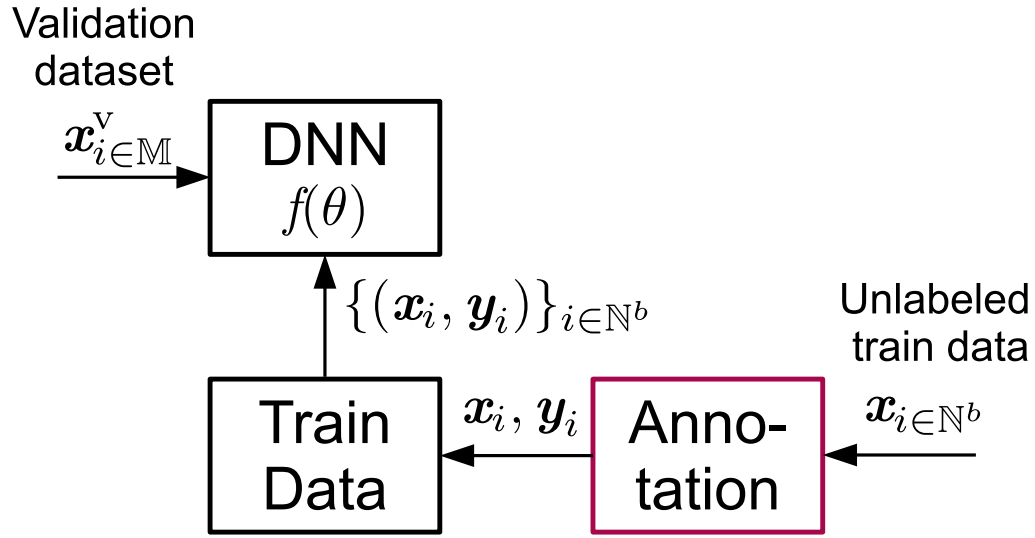
Code: github.com/gudovskiy/al-fk-self-supervision
Email: denis.gudovskiy@us.panasonic.com

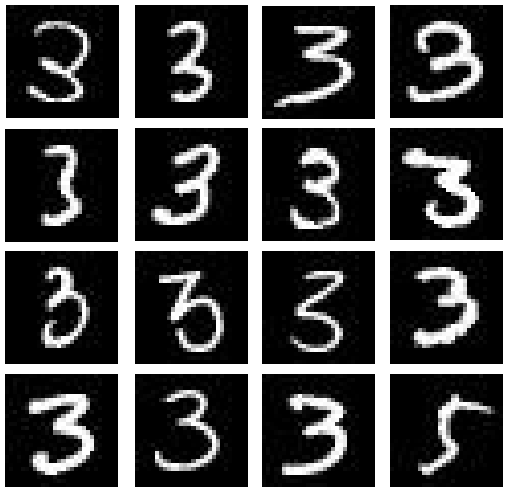
Panasonic AI

Motivation

- DNNs require large **annotated** train datasets
- Annotation process is **costly** and do not guarantee accuracy improvement



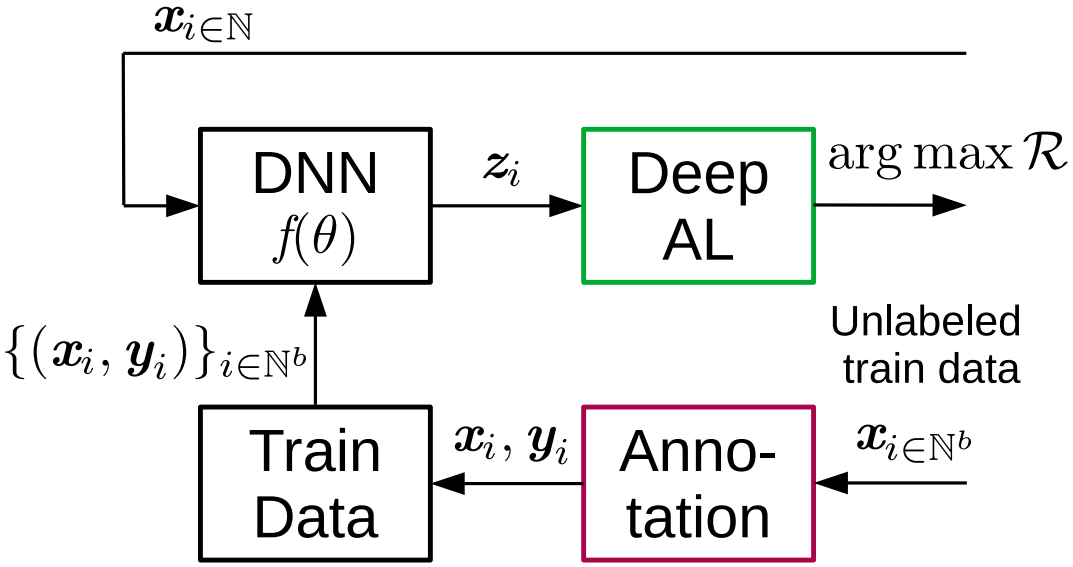


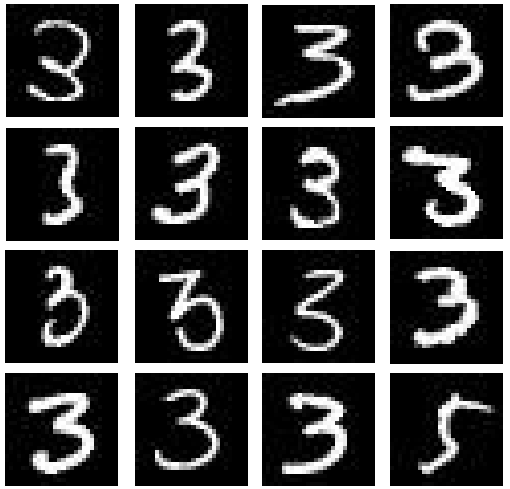


- Challenges:
  - ❖ How to **select train examples** for annotation?
  - ❖ How to **increase accuracy** while **minimizing annotation** costs?
  - ❖ How to utilize all **unlabeled data**?

Active Learning

- Active learning (AL) aims to select only **relevant train examples** for annotation

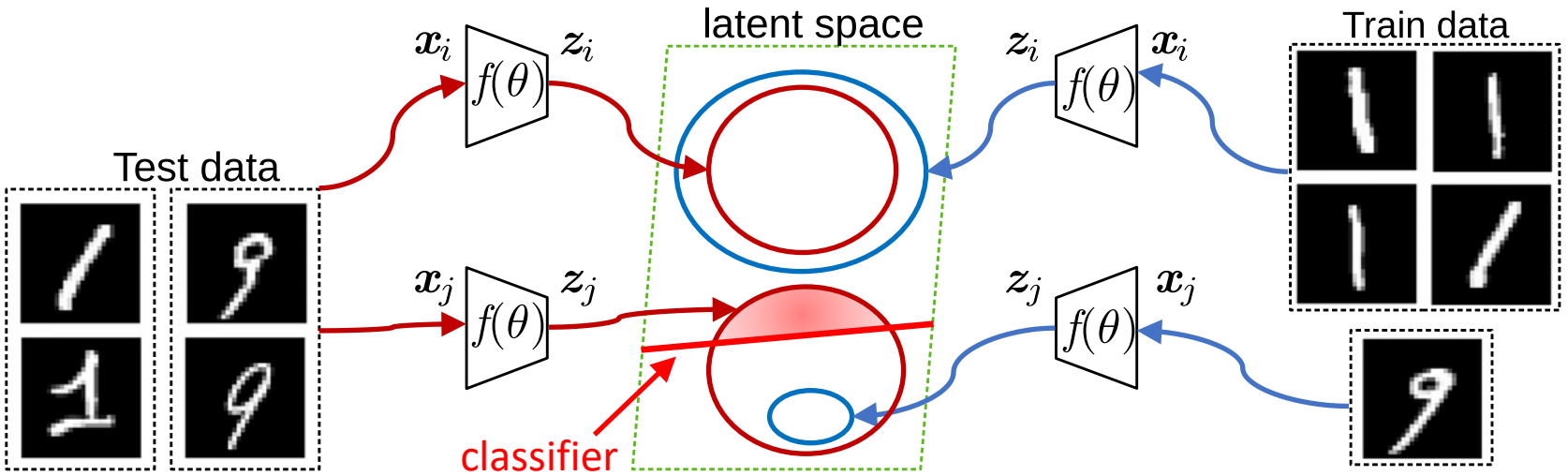




- How active learning works:
  - ✓ A **representation** $\boldsymbol{z}$ is pooled from the task model
  - ✓ AL **acquisition function** $\mathcal{R}(\boldsymbol{z})$ is calculated
  - ✓ A b^{th} **batch** $\mathbb{N}^b$ of examples **is selected** for annotation using $\mathcal{R}(\boldsymbol{z})$
  - ✓ Steps are repeated upon reaching **target accuracy**

Problem Statement for Biased Datasets

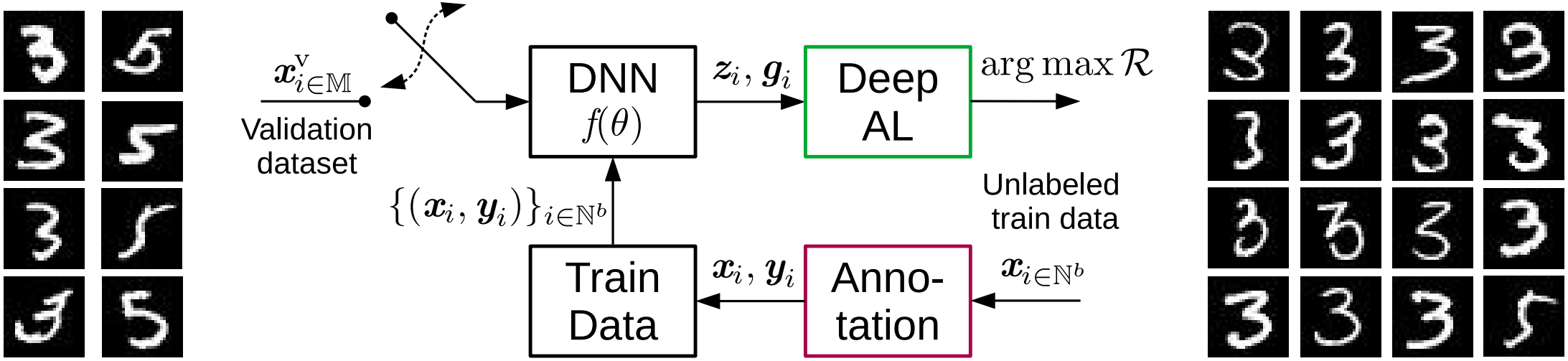
- **Prior methods** assume: $Q_x^{\text{test}} = Q_x$ and only train data is accessed
- Then, a trained classifier **misses on underrepresented** test instances



- Case #1: autonomous vehicle in a rare traffic situation
  - ❖Vehicle may have an accident with **unfamiliar objects**
- Case #2: face recognition with gender and race biases
  - ❖Photo cannot be recognized for very **rare types** of faces
- **Our idea**: explicitly minimize val/train distribution shift $\mathcal{R}_{opt} = \arg\min D_{KL}\left(\hat{Q}_x^{\text{v}} \parallel \hat{Q}_x\right)$

Our AL Method via Self-Supervised Fisher Kernel

- Main approach: **density matching in feature space** using Fisher kernel

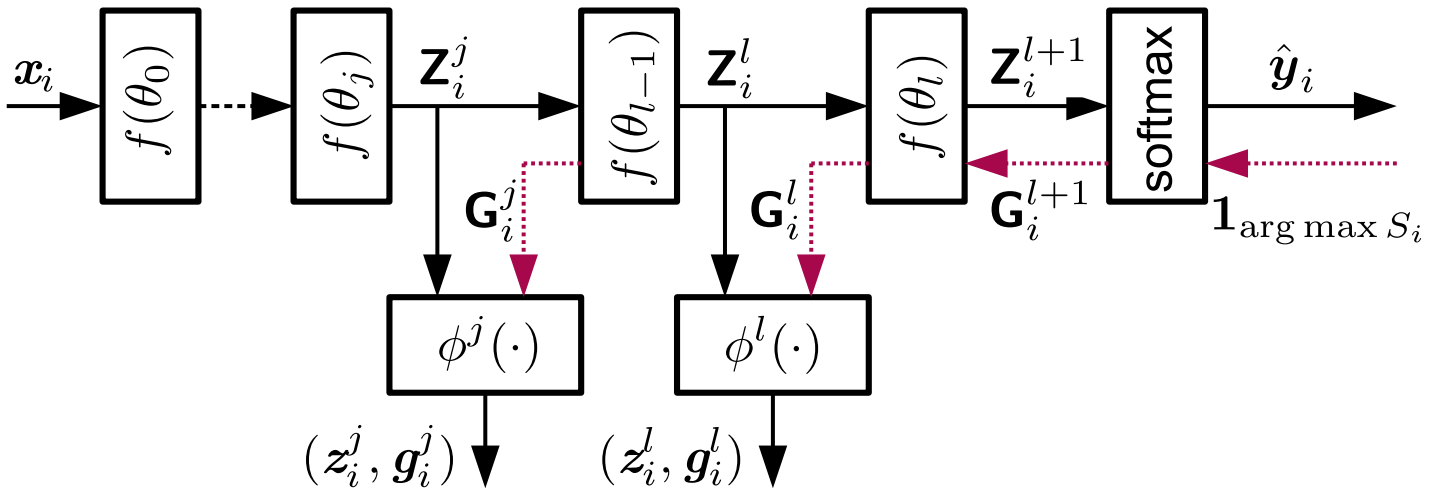


- Initially, we pretrain classifier using unsupervised learning [1]
- Next, our algorithm repeats the following steps:
  1) Pool compact features $\boldsymbol{z}$ and gradients $\boldsymbol{g}$ for val/unlabeled data
  2) Estimate practical Fisher kernel (PFK) between validation and unlabeled data
  3) Annotate and add selected examples to train data that maximize PFK

References:

[1] Gidaris et al. Unsupervised representation learning by predicting image rotations. In ICLR18

Details of Self-Supervised Fisher Kernel



- We pool **multi-scale** features $\boldsymbol{z}_i \in \mathbb{R}^L$ and gradients $\boldsymbol{g}_i \in \mathbb{R}^L$ from the task DNN
- Pseudo-labels for gradients are **estimated** using $S = \hat{p}(\boldsymbol{y}|\boldsymbol{z})$ metric
- Practical Fisher kernel is a **tractable** similarity matrix for DNNs:

$$\boldsymbol{R}_{\boldsymbol{z},\boldsymbol{g}} = \boldsymbol{R}_{\boldsymbol{z}} \circ \boldsymbol{R}_{\boldsymbol{g}} = \sum_j \left(\left(\boldsymbol{Z}_{\mathrm{v}}^j \right)^T \boldsymbol{Z}^j \circ \left(\boldsymbol{G}_{\mathrm{v}}^j \right)^T \boldsymbol{G}^j \right)$$

- Examples similar to clustered misclassified validation data are added to train dataset

Complexity Estimates

- Comparison in terms of forward and backward DNN passes
  - M – size of validation dataset
  - N – size of all unlabeled train data
  - N^b – current labeled train data size
  - $\acute{N}^b$ – current unlabeled data size
  - K – number of stochastic samples
  - I – number of train epochs
  - E – number of ensembles

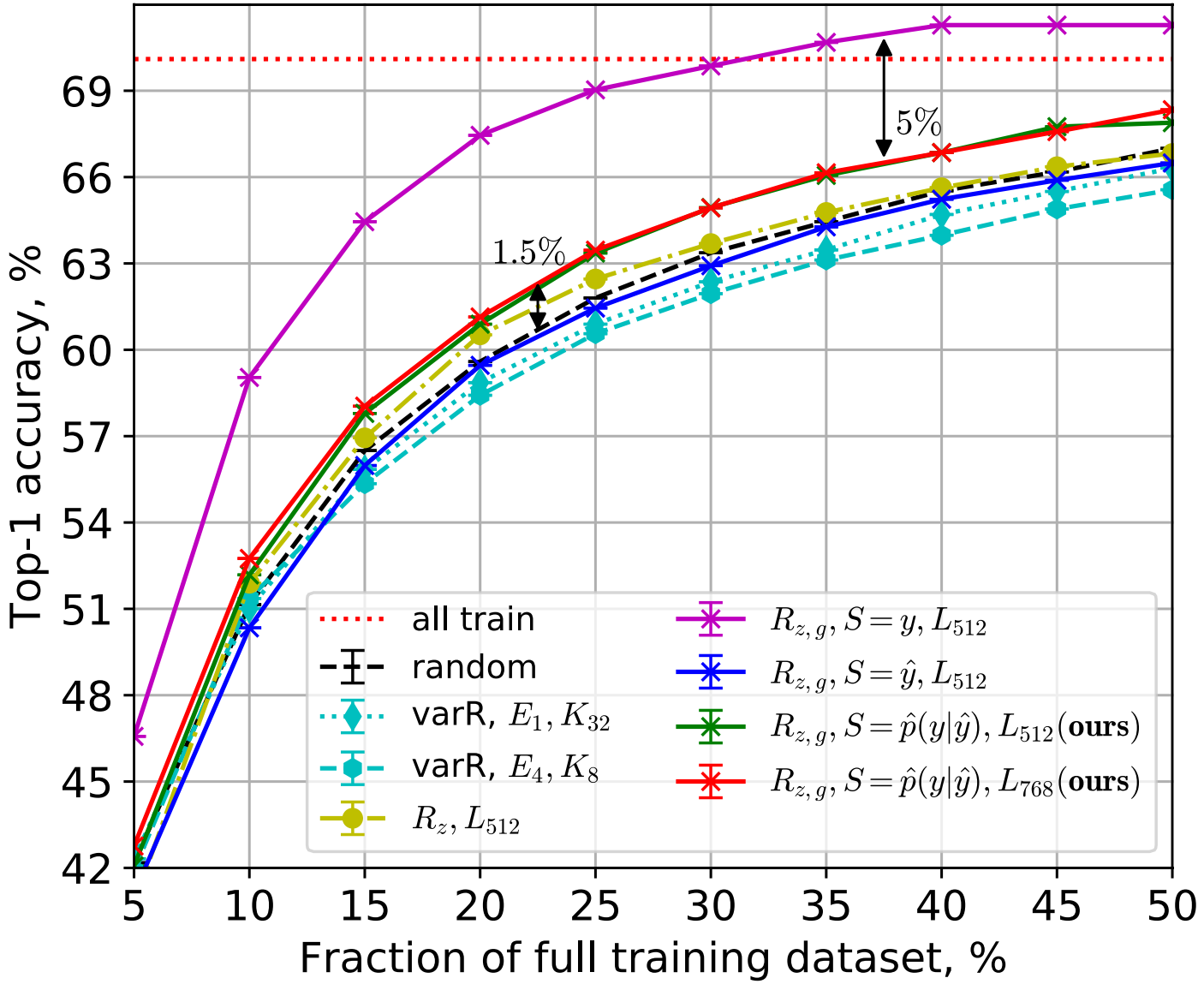
Method	AL	Train
Uncert. (varR) [2]	$K\acute{N}^b$	$2IN^b$
Ensembl. uncert. [3]	$EK\acute{N}^b$	$2EIN^b$
VAAL [4]	$\acute{N}^b + 2NI_{\text{VAE,D}}$	$2IN^b$
PCC: $\boldsymbol{R_z}$	$M + \acute{N}^b$	$2IN^b$
PFK: $\boldsymbol{R_{z,g}}$ **(ours)**	$2(M + \acute{N}^b)$	$2IN^b$

- **Our AL speedup** compared to [2,3] $\frac{EK\acute{N}^b}{2(M+\acute{N}^b)} \approx \frac{EK}{2}$, since $\acute{N}^b \gg M$
- Typically, complexity of our method is at least **10× lower**

References:
[2] Gal et al. Deep Bayesian active learning with image data. In ICML17
[3] Beluch et al. The power of ensembles for active learning in image classification. In CVPR18
[4] Sinha et al. Variational adversarial active learning. In ICCV19

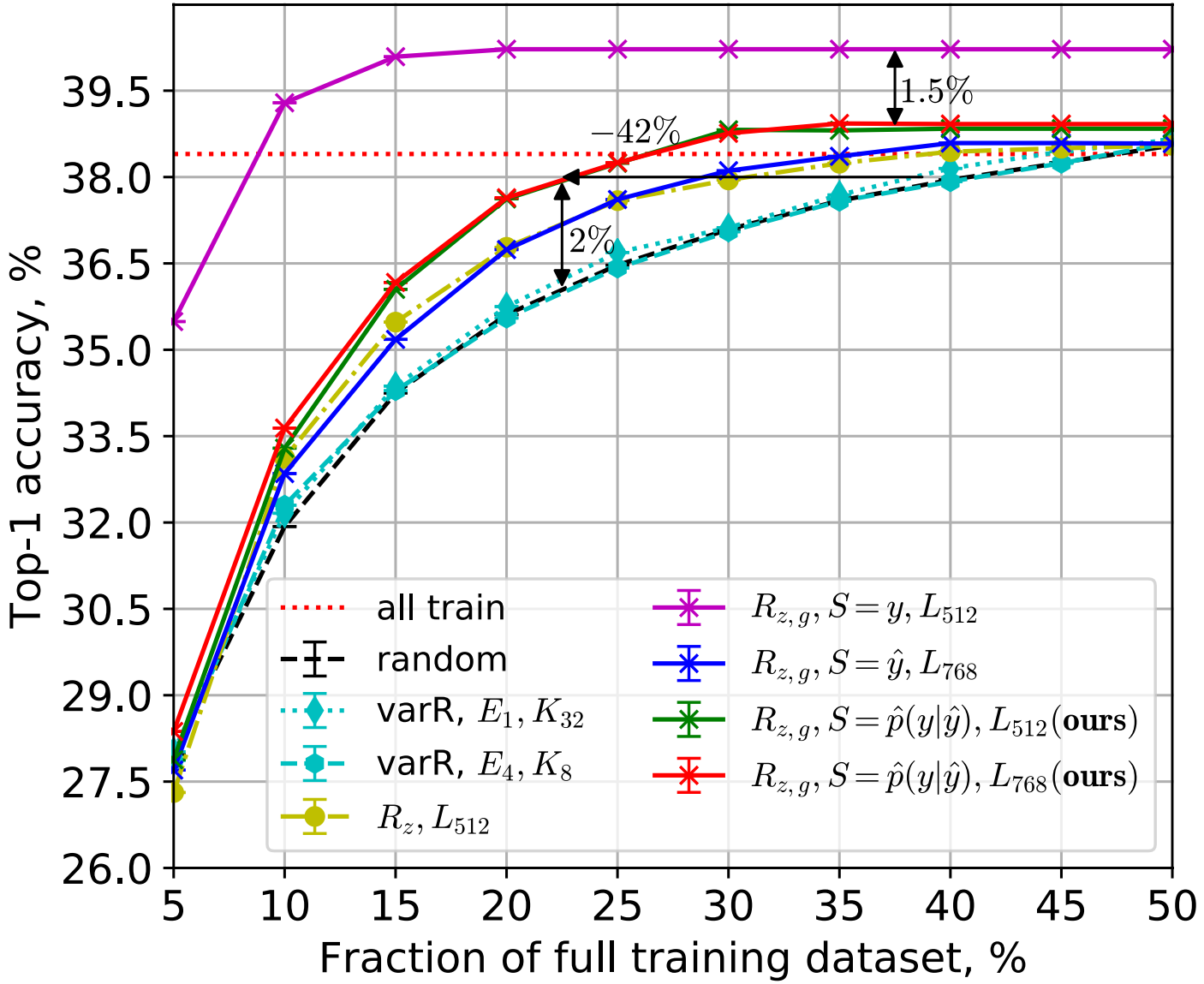
ImageNet w/o Class Imbalance

- ResNet-18 model
- Practically unrealistic data setup
- Our AL method with pseudo-labels:
  - 1.5% accuracy ↑
  - 20% labels ↓
  - 16× speedup ↑
- Theoretical limit with true labels:
  - 7% accuracy ↑
  - 70% labels ↓



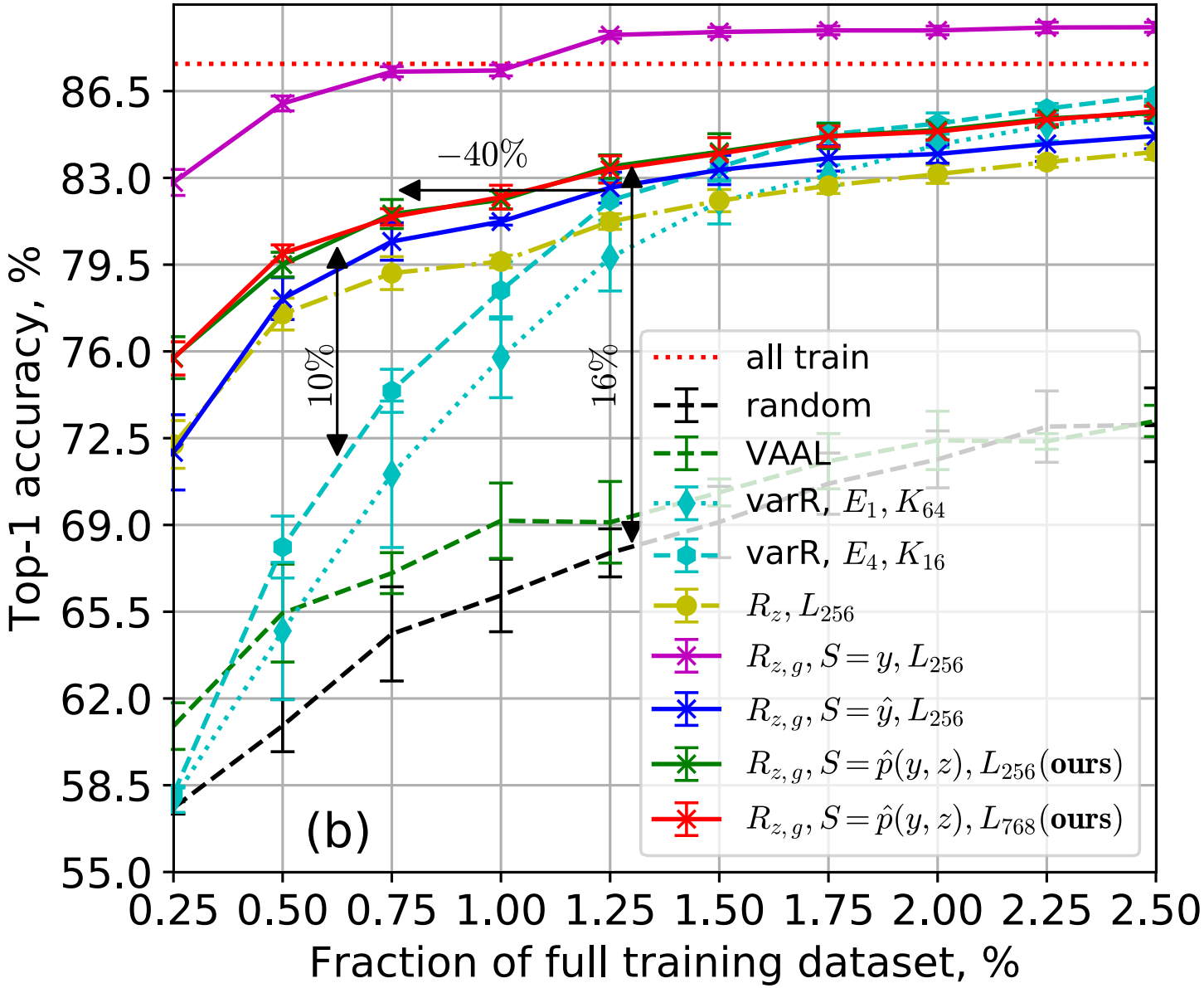
ImageNet with 100× Class Imbalance

- ResNet-18 model
- Practical data setup case
- Imbalance = {500 random classes}/{500 other classes} images
- Our AL method with pseudo-labels:
  - 2% accuracy ↑
  - 42% labels ↓
  - 16× speedup ↑
- Theoretical limit with true labels:
  - 6% accuracy ↑
  - 90% labels ↓



SVHN with 100× Imbalance

- ResNet-10 DNN model
- Imbalance = {0…4}/{5…9} images
- Our AL method with pseudo-labels:
  - 10% accuracy ↑
  - 40% labels ↓
  - 32× speedup ↑
- Theoretical limit with true labels:
  - 17% accuracy ↑
  - 80% labels ↓



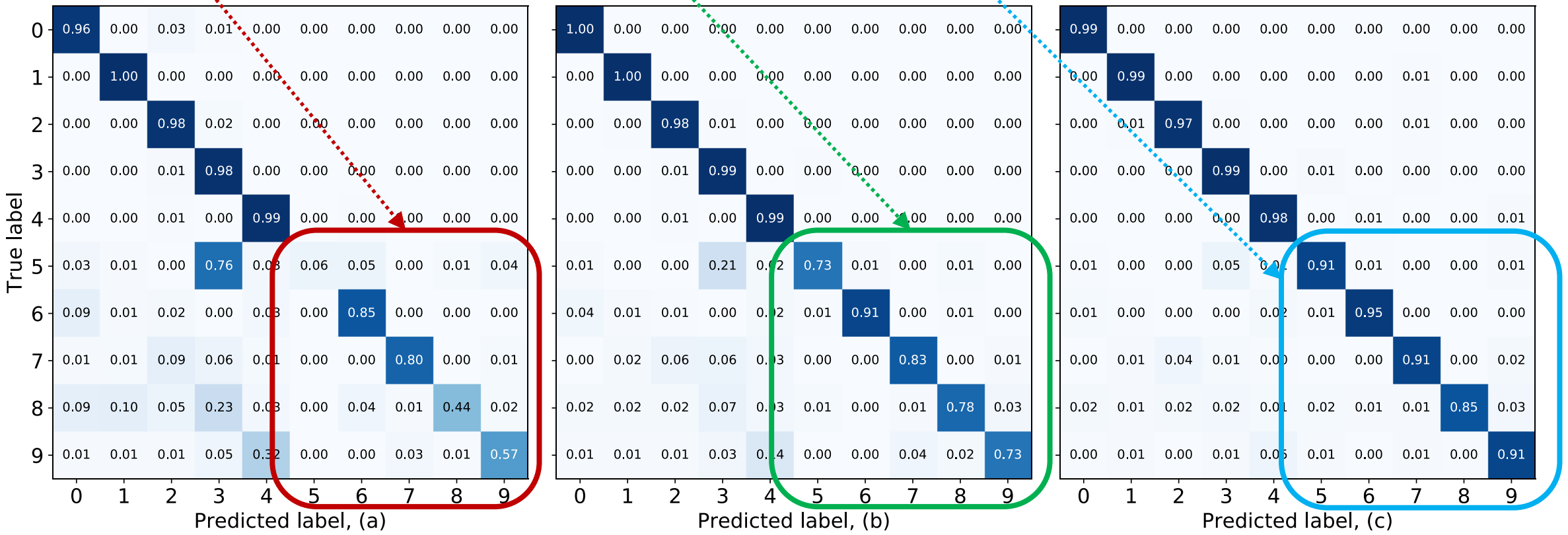
Confusion Matrices for 100× Imbalanced MNIST

Imbalance = {0…4}/{5…9} images = 100× at AL iteration $b = 3$

a) Prior varR (uncertainty-based) method: **36%** accuracy for digits {5,8,9}

b) Our method with estimated pseudolabels: **75%** accuracy for digits {5,8,9}

c) Our method with all true labels: **89%** theoretical limit for PFK



T-SNE Clustering: MNIST 100× Imbalance

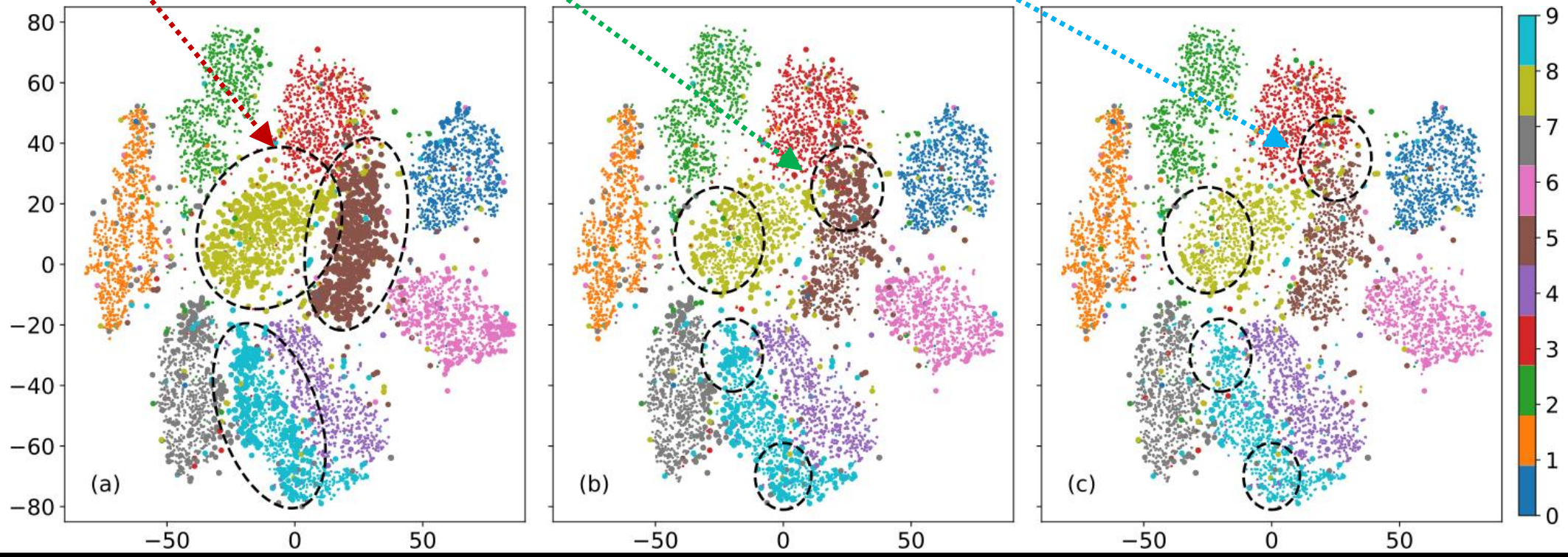
Imbalance = {0…4}/{5…9} images = 100× at AL iteration $b = 3$

Balls are **misclassified** and dots are **correctly** classified

a) Prior varR [2] method: **36%** accuracy for digits {5,8,9}

b) Our method with estimated pseudolabels: **75%** accuracy for digits {5,8,9}

c) Our method with all true labels: **89%** theoretical limit for PFK



PSVL

To appear in CVPR2021

AutoDO: Robust AutoAugment for Biased Data with Label Noise via Scalable Probabilistic Implicit Differentiation

Denis Gudovskiy[1] Luca Rigazio[1] Shun Ishizaka[2] Kazuki Kozuka[2] Sotaro Tsukizawa[2]

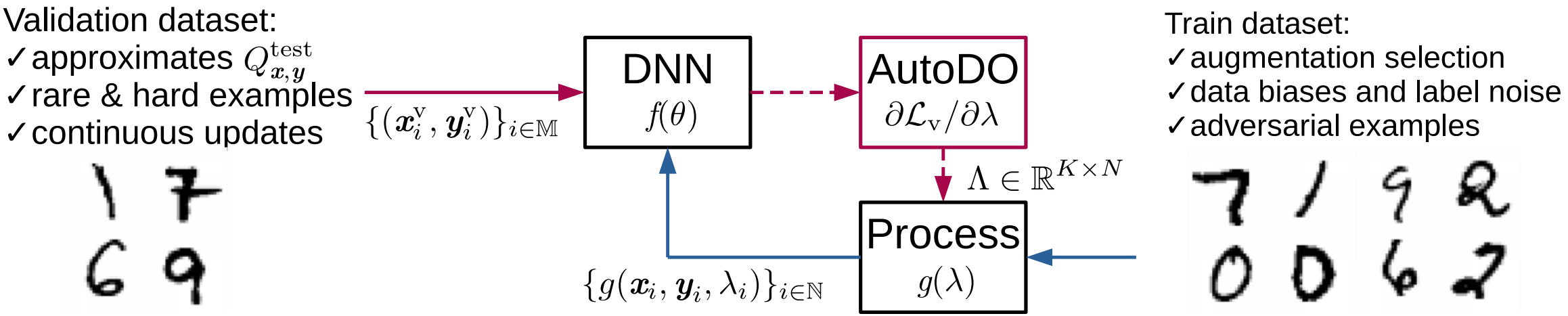
[1]Panasonic AI Lab, Mountain View, CA
[2]Panasonic Technology Division, Osaka

Code: github.com/gudovskiy/autodo
Email: denis.gudovskiy@us.panasonic.com

Panasonic AI

Labeled Data Optimization

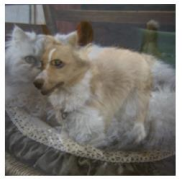
- After AL-assisted labeling, **efficiency of labeled data** can be improved
- Data pre-processing can **virtually increase train dataset** and improve generalization

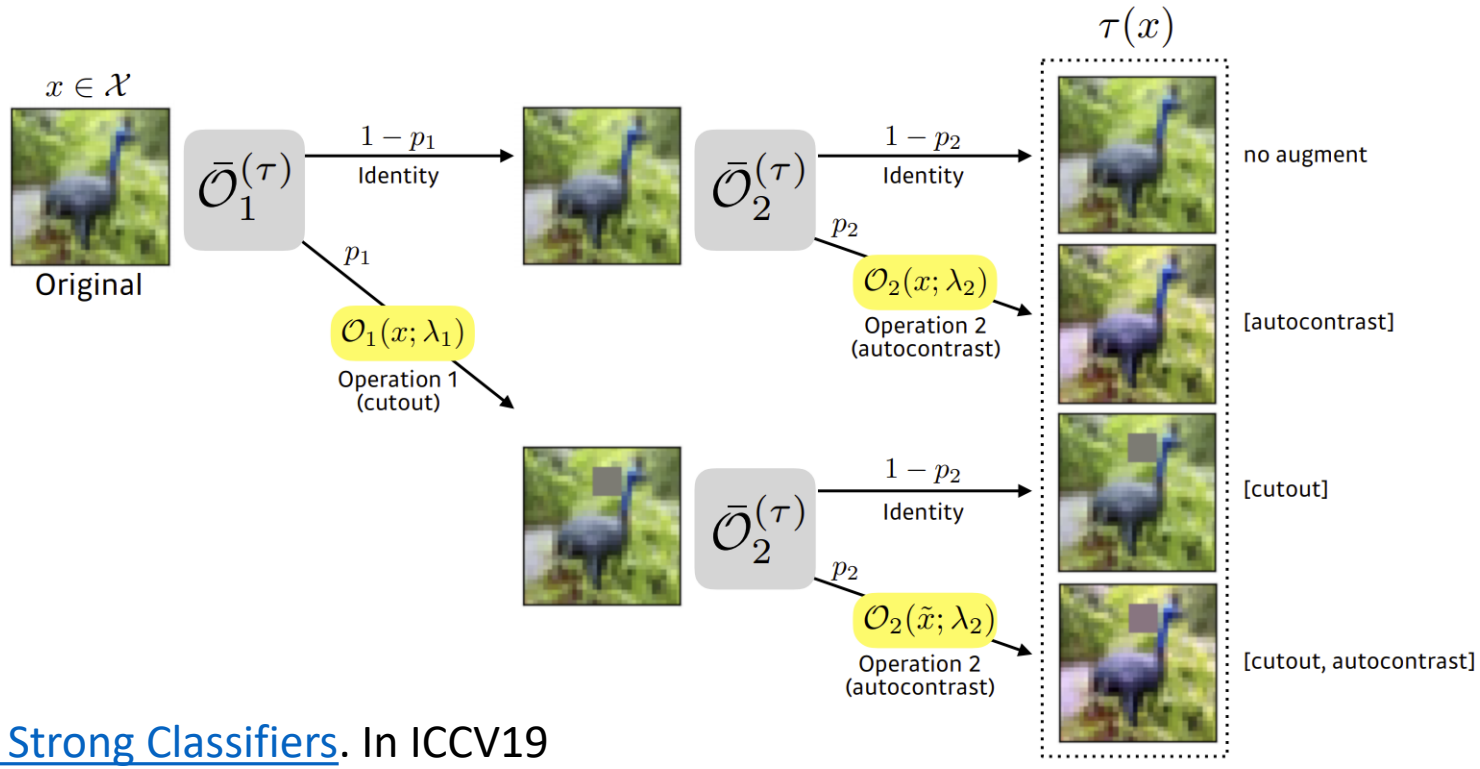


- Labeled data optimization steps:
  - ✓ Select **pre-processing functions** (*e.g.* augmentations) to increase data variability
  - ✓ Set processing **hyperparameters** or learn them automatically (AutoAugment/AutoDO)
  - ✓ Train AI model with the virtually **expanded train dataset** to improve generalization
  - ✓ Continue optimizing data processing functions and their hyperparameters

Labeled Data Optimization

- Typical data pre-processing functions: augmentations, noise, mixup, CutMix [1]
- Manual selection is time-consuming and requires domain knowledge
- Recent methods aim to **search or learn pre-processing** automatically: AutoAugment [2]
- Learnable pre-processing can be achieved using automatic **hyperparameter optimization**

	ResNet-50	Mixup [48]	Cutout [3]	CutMix
Image				
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0	Dog 0.6 Cat 0.4
ImageNet Cls (%)	76.3 (+0.0)	77.4 (+1.1)	77.1 (+0.8)	**78.6** **(+2.3)**
ImageNet Loc (%)	46.3 (+0.0)	45.8 (-0.5)	46.7 (+0.4)	**47.3** **(+1.0)**
Pascal VOC Det (mAP)	75.6 (+0.0)	73.9 (-1.7)	75.1 (-0.5)	**76.7** **(+1.1)**

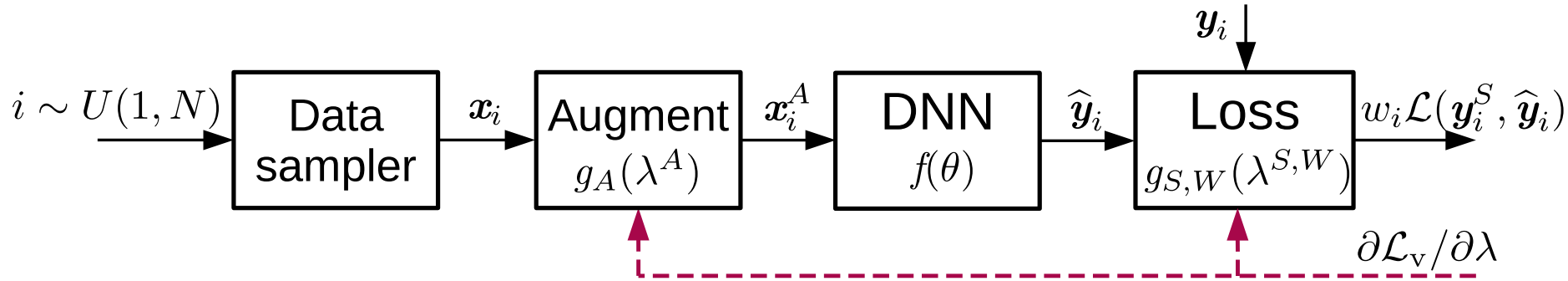


References:

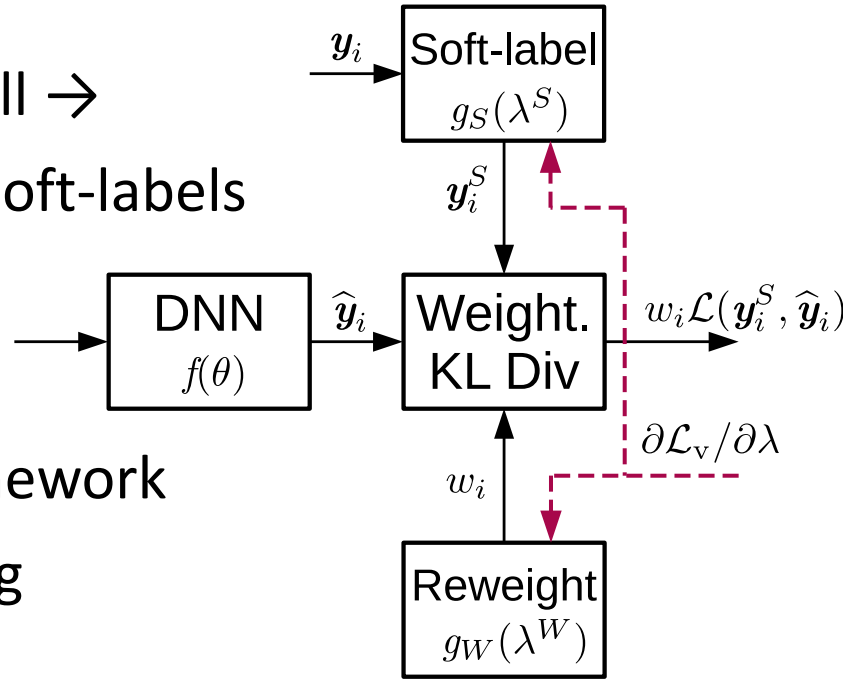
[1] Yun et al. CutMix: Regularization Strategy to Train Strong Classifiers. In ICCV19

[2] Lim et al. Fast AutoAugment In NeurIPS19

Automatic Dataset Optimization (AutoDO)



- Other hyperparameters can be automatically optimized as well →
- AutoDO model [1] includes augmentations, loss weights and soft-labels
- **Main idea**: learn hyperparameters for each train data point
- Hence, AutoDO model contains millions of hyperparameters!
- Implicit differentiation [2] provides scalable optimization framework
- Model and dataset optimization are alternating during training



References:

[1] Gudovskiy et al. AutoDO: robust AutoAugment for biased data with label noise. To appear in CVPR21

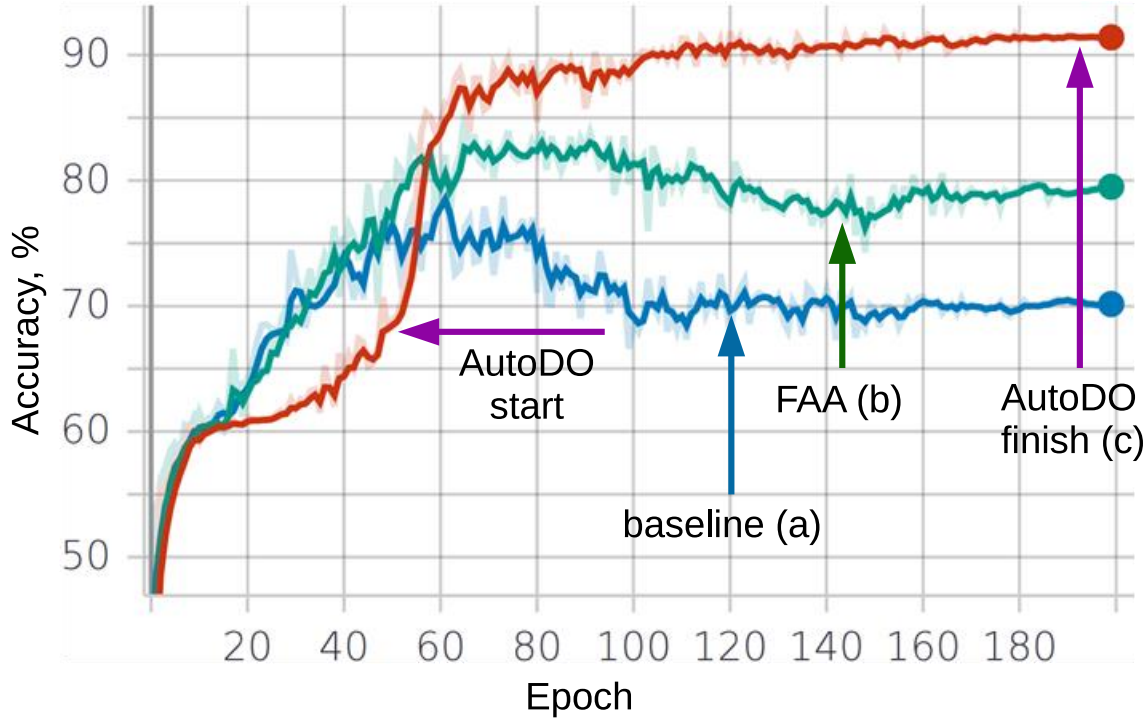
[2] Lorraine et al. Optimizing millions of hyperparameters by implicit differentiation. In AISTATS20

Automatic Dataset Optimization (AutoDO)

- **Main focus**: optimization of realistic train datasets with data biases and noisy labels
- Train data distribution should be adjusted to test data in actual systems
- AutoDO can flexibly change train distribution due to per-point model
- This prevents overfitting to train data and increases generalization!

Table 1. WRNet28-10 SVHN top-1 test error rate, $\mu_{\pm\sigma}$ %.

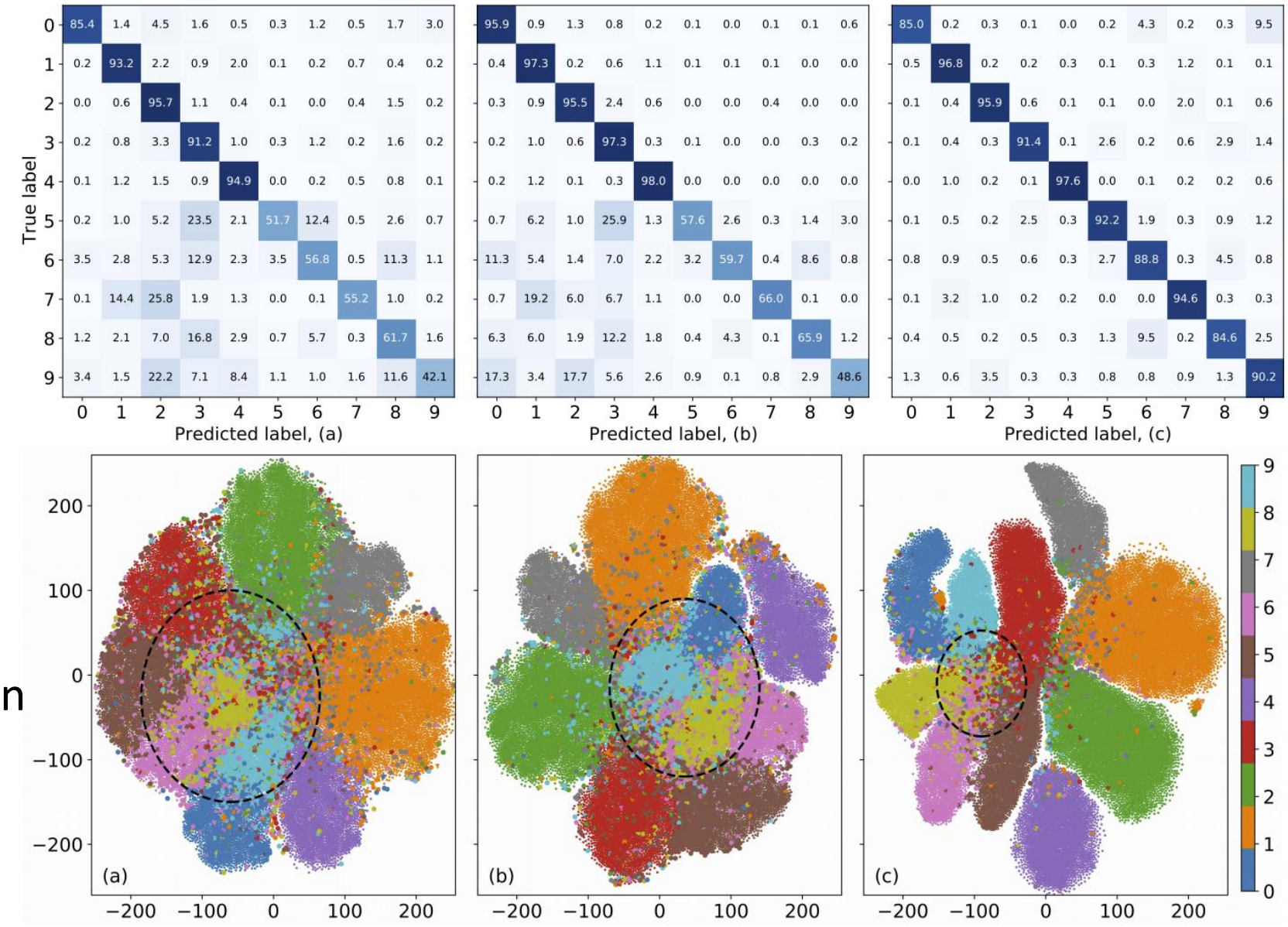
Alg./IR-NR	1-0.0	100-0.0	1-0.1	100-0.1
Baseline	$3.6_{\pm 0.10}$	$13.6_{\pm 0.69}$	$5.3_{\pm 0.27}$	$20.0_{\pm 1.92}$
RAA [6]	$2.7_{\pm 0.04}$	$10.9_{\pm 0.66}$	$3.4_{\pm 0.11}$	$13.6_{\pm 0.96}$
FAA [17]	$2.8_{\pm 0.02}$	$11.5_{\pm 0.32}$	$3.7_{\pm 0.08}$	$15.3_{\pm 1.07}$
DADA [16]	$2.9_{\pm 0.03}$	$12.2_{\pm 0.54}$	$4.1_{\pm 0.13}$	$16.5_{\pm 1.51}$
$\lambda^{A_{\text{SHA}}}$ (ours)	$2.8_{\pm 0.10}$	$12.6_{\pm 1.53}$	$3.0_{\pm 0.17}$	$13.7_{\pm 0.77}$
λ^{A} (ours)	$2.7_{\pm 0.09}$	$10.2_{\pm 0.50}$	$3.0_{\pm 0.07}$	$12.3_{\pm 0.80}$
$\lambda^{A,W}$ (ours)	$2.8_{\pm 0.04}$	$6.1_{\pm 0.22}$	$2.8_{\pm 0.07}$	$8.1_{\pm 0.14}$
$\lambda^{A,W,S}$ (ours)	$\mathbf{2.5}_{\pm 0.04}$	$\mathbf{5.3}_{\pm 0.21}$	$\mathbf{2.6}_{\pm 0.05}$	$\mathbf{6.3}_{\pm 0.57}$



References:
[6] Cubuk et al. RandAugment. In CVPRW19
[17] Lim et al. Fast AutoAugment. In NeurIPS19
[16] Li et al. DADA. In ECCV20

Automatic Dataset Optimization (AutoDO)

- WideResNet28-10 model
- SVHN dataset
  - 100× class imbalance
  - 10% noisy labels
- Confusion matrices (top)
- T-SNE of embeddings (bottom)
  a) Standard augmentations
  b) Fast AutoAugment
  c) AutoDO
- AutoDO improves generalization when train dataset is biased, and labels are noisy



Conclusions

- I presented recent Panasonic AI-related projects:
  - ✓ Autonomous ride share service (big thanks to BDD)
  - ✓ Our new challenging multimodal Home Action Genome Dataset
  - ✓ Recent research about robust active learning and AutoAgument
- I encourage research community to:
  - ✓ Consider a realistic imperfect data collection and labeling process
  - ✓ Introduce some data bias when develop your models
  - ✓ Research new methods to overcome limitations of empirical datasets
- Main takeaways:
  - ✓ Methods to deal with the imperfect data not only save corp money…
  - ✓ They are good for publications[1] and discovering all unknown yet dragons!

[1]CVPR2021: L2ID and TCV workshops. ICLR2021: S2D workshop

Questions?

Panasonic AI