Self-supervised Label Augmentation via Input Transformations

Hankook Lee, Sung Ju Hwang, Jinwoo Shin

Korea Advanced Institute of Science and Technology (KAIST)

International Conference on Machine Learning (ICML 2020)

2020. 06. 15.

Outline

Self-supervised Learning

- What is self-supervised learning?
- Applications of self-supervision
- *Motivation:* How effectively utilize self-supervision in fully-supervised settings?

Self-supervised Label Augmentation (SLA)

- *Observation:* Learning invariance to transformations
- *Main idea:* Eliminating invariance via joint-label classifier
- ***Aggregation*** across all transformations & **Self-distillation** from aggregation

Experiments

- Standard fully-supervised / few-shot / imbalance settings

Outline

Self-supervised Learning

- What is self-supervised learning?
- Applications of self-supervision
- *Motivation:* How effectively utilize self-supervision in fully-supervised settings?

Self-supervised Label Augmentation (SLA)

- *Observation:* Learning invariance to transformations
- *Main idea:* Eliminating invariance via joint-label classifier
- ***Aggregation*** across all transformations & **Self-distillation** from aggregation

Experiments

- Standard fully-supervised / few-shot / imbalance settings

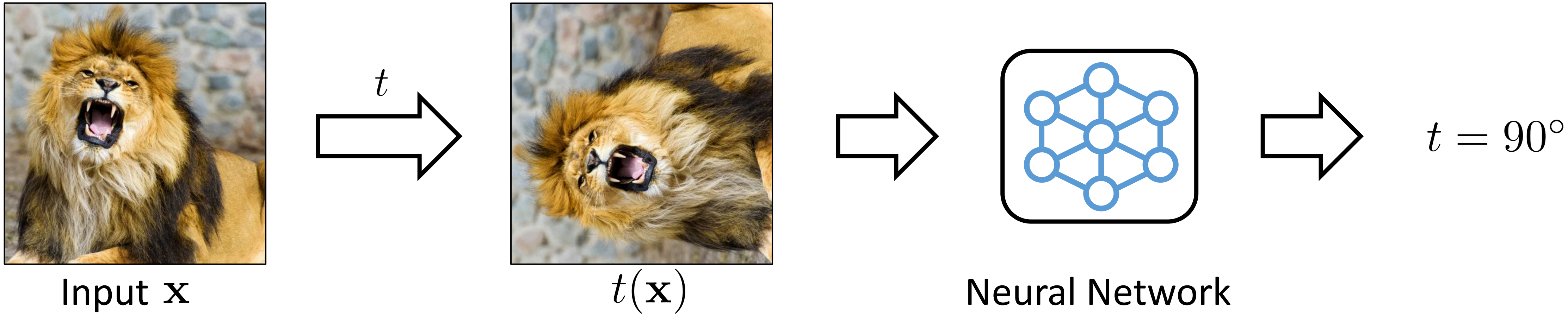
What is Self-supervised Learning?

Self-supervised learning approaches

1. Construct artificial labels, i.e., *self-supervision*, only using the input examples
2. Learn their representations via predicting the labels

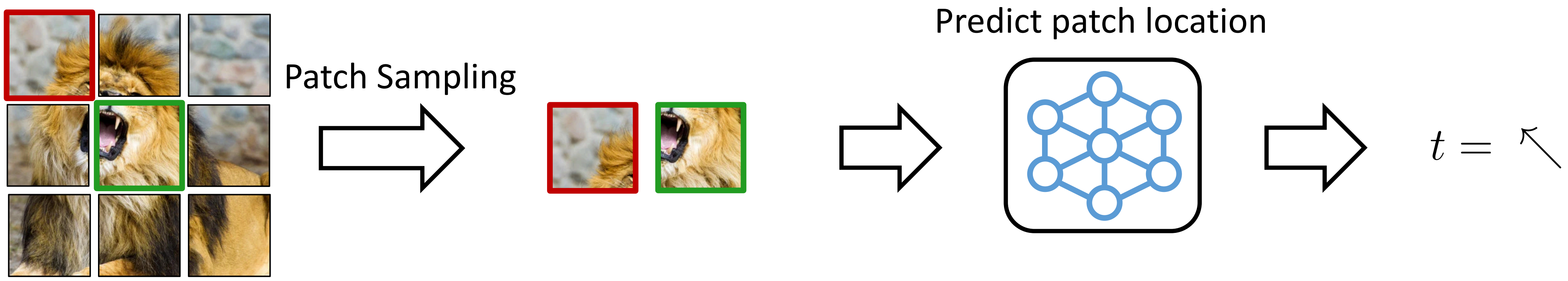
Transformation-based self-supervision

1. Apply a transformation $t \in \{t_1, \ldots, t_M\}$ into an input $\mathbf{x}$
2. Learn to predict the transformation t from observing only $t(\mathbf{x})$

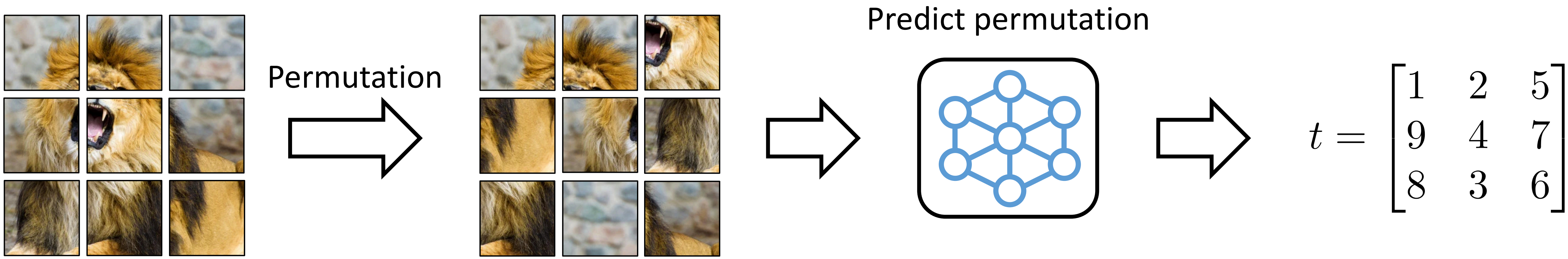


Examples of Self-supervision

- Relative Patch Location Prediction [Doersch et al., 2015]



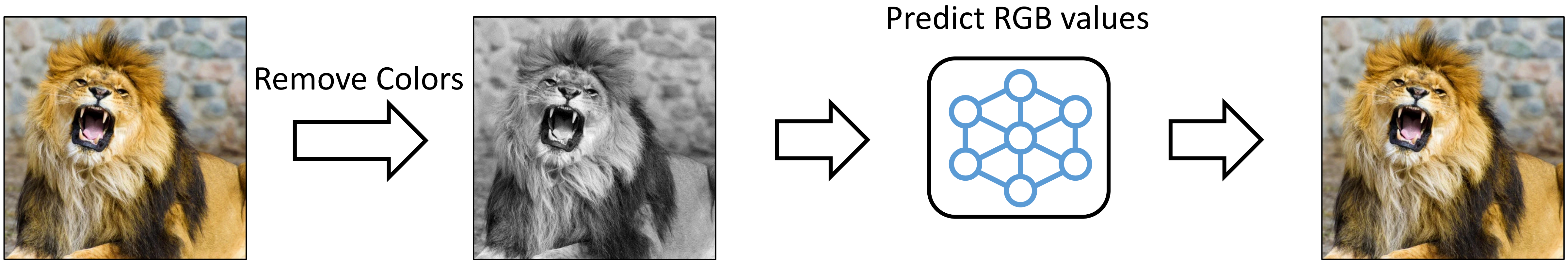
- Jigsaw Puzzle [Noroozi and Favaro, 2016]



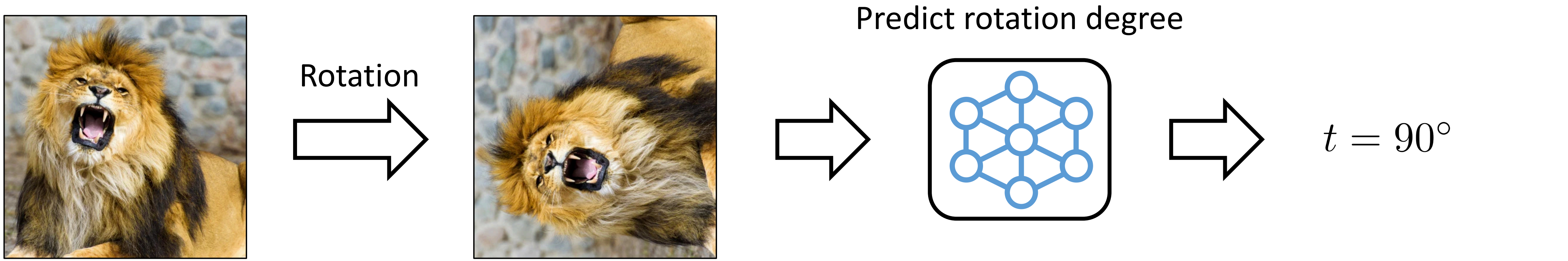
[Doersch et al., 2015] Unsupervised visual representation learning by context prediction, ICCV 2015
[Noroozi and Favaro, 2016] Unsupervised learning of visual representations by solving jigsaw puzzles, ECCV 2016

Examples of Self-supervision

- Colorization [Larsson et al., 2017]



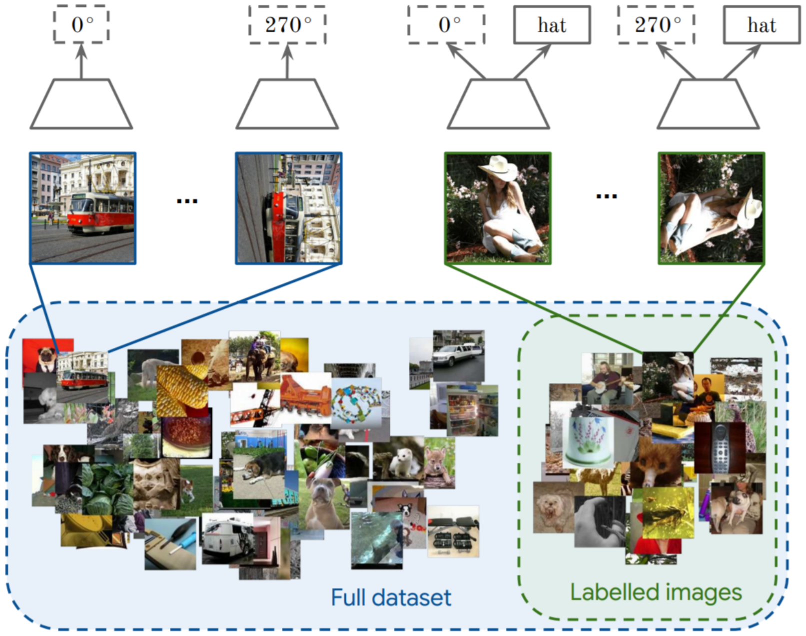
- Rotation [Gidaris et al., 2018]



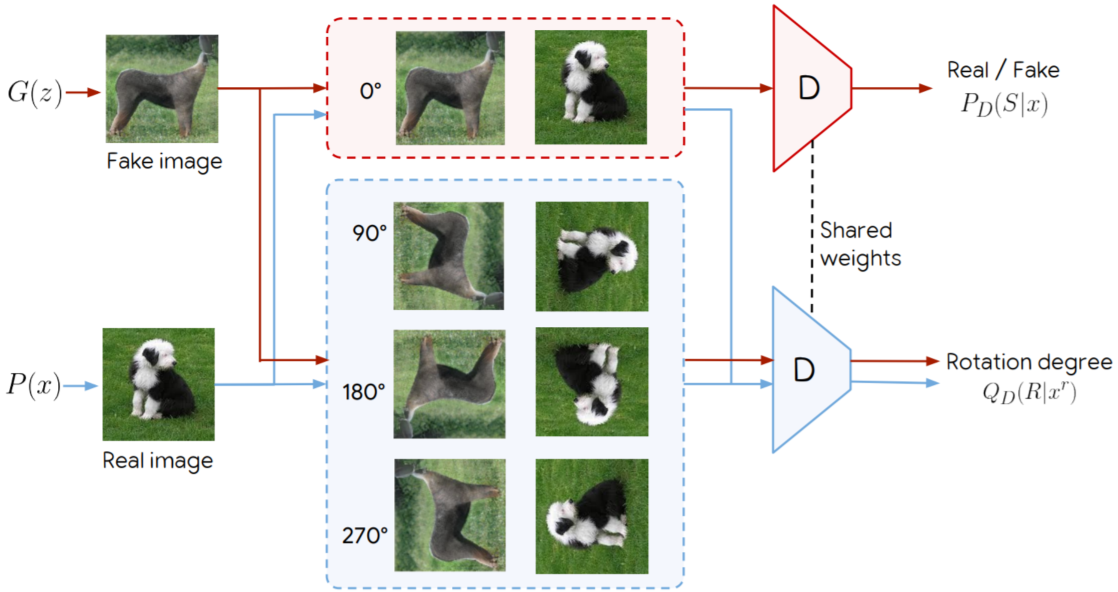
[Larsson et al., 2017] Colorization as a proxy task for visual understanding, CVPR 2017
[Gidaris et al., 2018] Unsupervised representation learning by predicting image rotations, ICLR 2018

Applications of Self-supervision

- Simplicity of transformation-based self-supervision encourages its wide applicability
  - Semi-supervised learning [Zhai et al., 2019; Berthelot et al., 2020]
  - Improving robustness [Hendrycks et al., 2019]
  - Training generative adversarial networks [Chen et al., 2019]



S4L [Zhai et al., 2019]



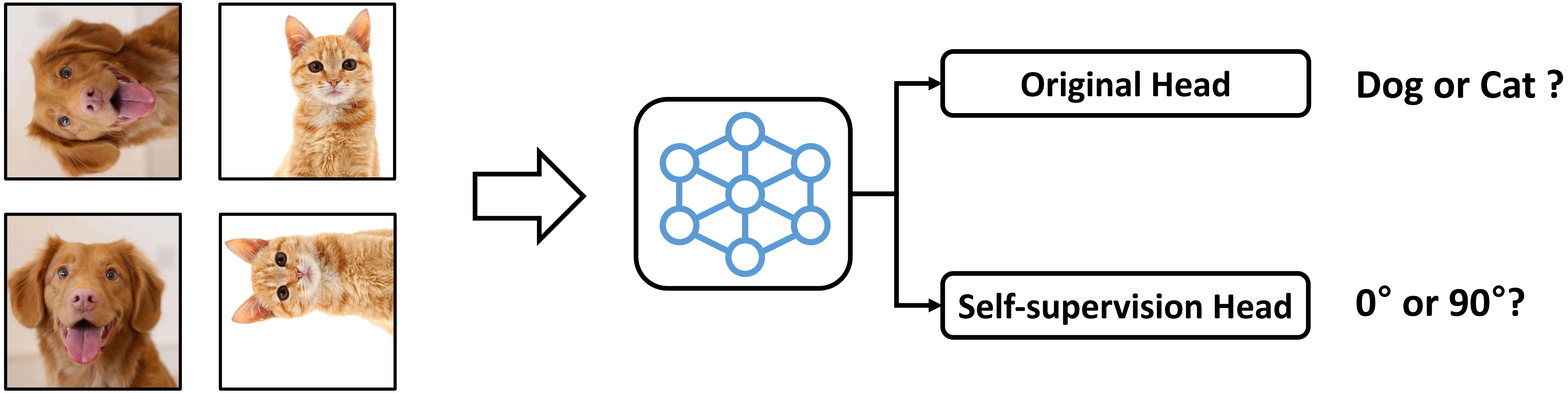
SSGAN [Chen et al., 2019]

[Zhai et al., 2019] S4L: Self-supervised semi-supervised learning
[Berthelot et al., 2020] Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring, ICLR 2020
[Hendrycks et al., 2019] Using self-supervised learning can improve model robustness and uncertainty, NeurIPS 2019
[Chen et al., 2019] Self-supervised gans via auxiliary rotation loss, CVPR 2019

Applications of Self-supervision

- Simplicity of transformation-based self-supervision encourages its wide applicability
  - Semi-supervised learning [Zhai et al., 2019; Berthelot et al., 2020]
  - Improving robustness [Hendrycks et al., 2019]
  - Training generative adversarial networks [Chen et al., 2019]

- The prior works maintain **two separate classifiers for original and self-supervised tasks**, and optimize their objectives simultaneously



Applications of Self-supervision

- Simplicity of transformation-based self-supervision encourages its wide applicability
  - Semi-supervised learning [Zhai et al., 2019; Berthelot et al., 2020]
  - Improving robustness [Hendrycks et al., 2019]
  - Training generative adversarial networks [Chen et al., 2019]

- The prior works maintain **two separate classifiers for original and self-supervised tasks**, and optimize their objectives simultaneously
  - This approach can be considered as multi-task learning

- This typically provides **no accuracy gain when working with fully-labeled datasets**

⇨ **Q)** How can we effectively utilize the **self-supervision** for **fully-supervised** classification tasks?

Outline

Self-supervised Learning

- What is self-supervised learning?
- Applications of self-supervision
- *Motivation:* How effectively utilize self-supervision in fully-supervised settings?

Self-supervised Label Augmentation (SLA)

- *Observation:* Learning invariance to transformations
- *Main idea:* Eliminating invariance via joint-label classifier
- ***Aggregation*** across all transformations & **Self-distillation** from aggregation

Experiments

- Standard fully-supervised / few-shot / imbalance settings

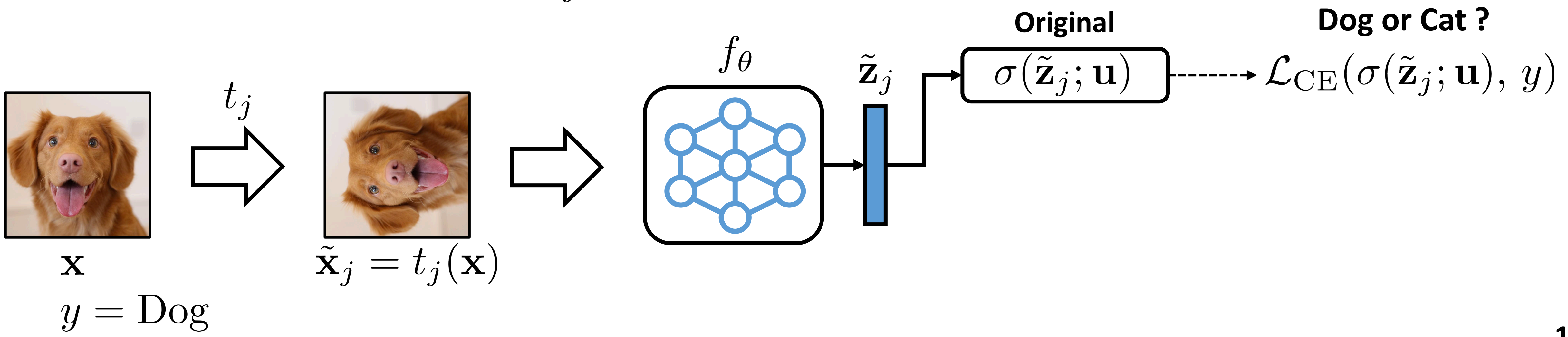
Data Augmentation with Transformations

- Notation
  - $\{t_1, \ldots, t_M\}$: Pre-defined transformations, e.g., rotation by 0°, 90°, 180°, 270°
  - $\tilde{\mathbf{z}}_j = f_\theta(t_j(\mathbf{x}))$: Penultimate feature of the modified input $\tilde{\mathbf{x}}_j = t_j(\mathbf{x})$
  - $\sigma_i(\tilde{\mathbf{z}}; \mathbf{u}) = \exp(\mathbf{u}_i^\top \tilde{\mathbf{z}}) / \sum_k \exp(\mathbf{u}_k^\top \tilde{\mathbf{z}})$: Softmax classifier with a weight matrix $\mathbf{u}$

- Data augmentation (DA) approach can be written as

Not depending on t_j

$$\mathcal{L}_{\mathrm{DA}}(\mathbf{x}, y) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}_{\mathrm{CE}}(\sigma(\tilde{\mathbf{z}}_j; \mathbf{u}), y)$$



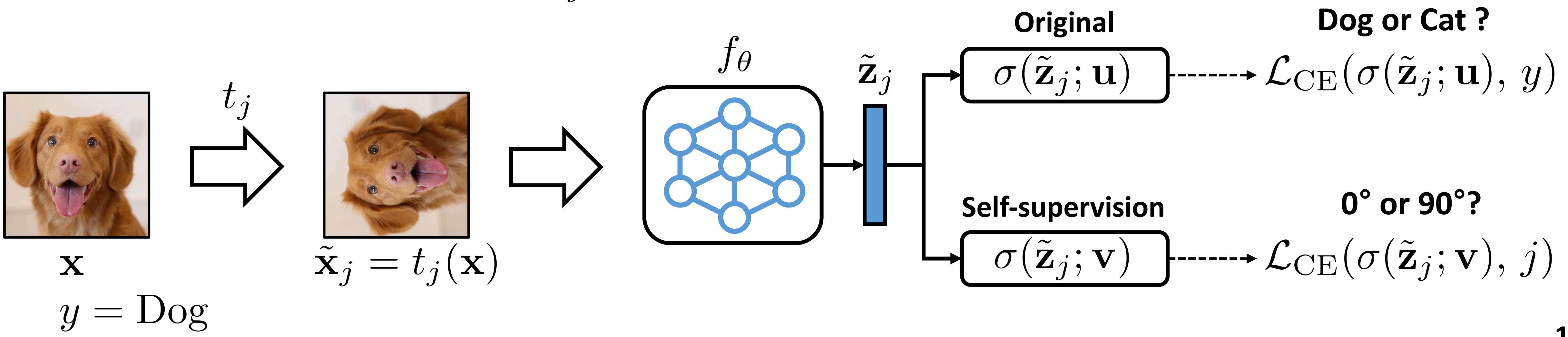
Multi-task Learning with Self-supervision

- Notation
  - $\{t_1, \ldots, t_M\}$: Pre-defined transformations, e.g., rotation by 0°, 90°, 180°, 270°
  - $\tilde{\mathbf{z}}_j = f_\theta(t_j(\mathbf{x}))$: Penultimate feature of the modified input $\tilde{\mathbf{x}}_j = t_j(\mathbf{x})$
  - $\sigma_i(\tilde{\mathbf{z}}; \mathbf{u}) = \exp(\mathbf{u}_i^\top \tilde{\mathbf{z}}) / \sum_k \exp(\mathbf{u}_k^\top \tilde{\mathbf{z}})$: Softmax classifier with a weight matrix $\mathbf{u}$

- Multi-task learning (MT) approach is formally written as

Depending on t_j

$$\mathcal{L}_{\mathrm{MT}}(\mathbf{x}, y) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}_{\mathrm{CE}}(\sigma(\tilde{\mathbf{z}}_j; \mathbf{u}),\ y) + \mathcal{L}_{\mathrm{CE}}(\sigma(\tilde{\mathbf{z}}_j; \mathbf{v}),\ j)$$



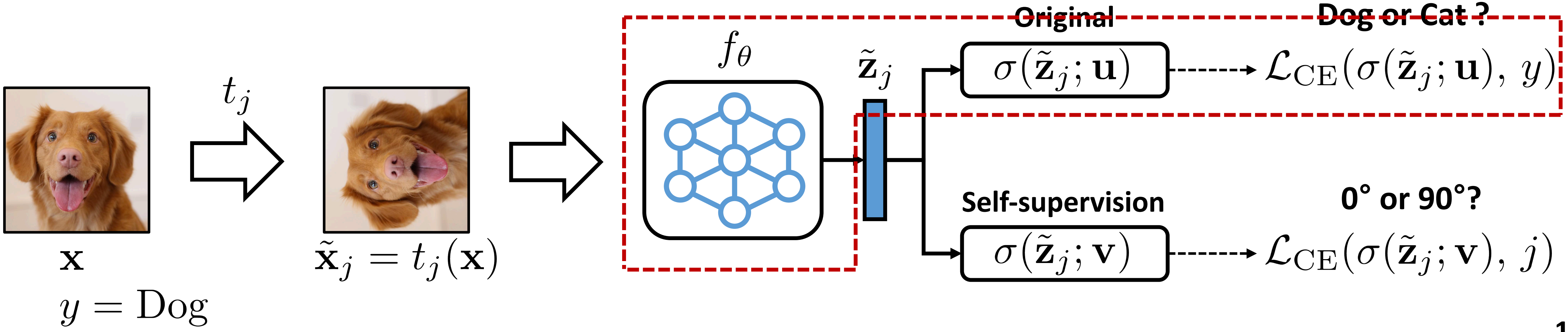
Multi-task Learning with Self-supervision

- Notation
  - $\{t_1, \ldots, t_M\}$: Pre-defined transformations, e.g., rotation by 0°, 90°, 180°, 270°
  - $\tilde{\mathbf{z}}_j = f_\theta(t_j(\mathbf{x}))$: Penultimate feature of the modified input $\tilde{\mathbf{x}}_j = t_j(\mathbf{x})$
  - $\sigma_i(\tilde{\mathbf{z}}; \mathbf{u}) = \exp(\mathbf{u}_i^\top \tilde{\mathbf{z}}) / \sum_k \exp(\mathbf{u}_k^\top \tilde{\mathbf{z}})$: Softmax classifier with a weight matrix $\mathbf{u}$

- Multi-task learning (MT) approach is formally written as

This enforces invariance to transformations ⇒ more difficult optimization

$$\mathcal{L}_{\mathrm{MT}}(\mathbf{x}, y) = \frac{1}{M} \sum_{j=1}^{M} \boxed{\mathcal{L}_{\mathrm{CE}}(\sigma(\tilde{\mathbf{z}}_j; \mathbf{u}), y)} + \mathcal{L}_{\mathrm{CE}}(\sigma(\tilde{\mathbf{z}}_j; \mathbf{v}), j)$$



Learning Invariance to Transformations

Learning discriminability from transformations	$\Rightarrow$	Self-supervised learning (SSL)
Learning invariance to transformations	$\Rightarrow$	Data augmentation (DA)

- Transformations for DA $\neq$ Transformations for SSL
  - Learning invariance to SSL transformations degrades performance
  - Ablation study:
    - We use 4 rotations with degrees of 0°, 90°, 180°, 270° for transformations $\{t_1, \ldots, t_M\}$
    - We train Baseline w/o rotation, Data Augmentation (DA), and Multi-task Learning (MT) objectives

Notation

$$\mathbf{z} = f_\theta(\mathbf{x}), \quad \tilde{\mathbf{z}}_j = f_\theta(t_j(\mathbf{x})),$$
$$\sigma_i(\mathbf{z}; \mathbf{u}) = \frac{\exp(\mathbf{u}_i^\top \mathbf{z})}{\sum_k \exp(\mathbf{u}_k^\top \mathbf{z})}$$

Baseline: $\mathcal{L}_{\text{Baseline}}(\mathbf{x}, y) = \mathcal{L}_{\text{CE}}(\sigma(\mathbf{z}; U),\ y)$

Data Augmentation: $\mathcal{L}_{\text{DA}}(\mathbf{x}, y) = \frac{1}{M}\sum_{j=1}^{M} \mathcal{L}_{\text{CE}}(\sigma(\tilde{\mathbf{z}}_j; U),\ y)$

Multi-task Learning: $\mathcal{L}_{\text{MT}}(\mathbf{x}, y) = \frac{1}{M}\sum_{j=1}^{M} \mathcal{L}_{\text{CE}}(\sigma(\tilde{\mathbf{z}}_j; U),\ y) + \mathcal{L}_{\text{CE}}(\sigma(\tilde{\mathbf{z}}_j; V),\ j)$

Learning Invariance to Transformations

Learning discriminability from transformations	$\Rightarrow$	Self-supervised learning (SSL)
Learning invariance to transformations	$\Rightarrow$	Data augmentation (DA)

- Transformations for DA $\neq$ Transformations for SSL
  - Learning invariance to SSL transformations degrades performance
  - Ablation study:
    - We use 4 rotations with degrees of 0°, 90°, 180°, 270° for transformations $\{t_1, \ldots, t_M\}$
    - We train Baseline w/o rotation, Data Augmentation (DA), and Multi-task Learning (MT) objectives
    - In CIFAR-10/100, tiny-ImageNet, learning invariance to rotations degrades classification performance

Dataset	Baseline	DA	MT
CIFAR10	92.39	90.44	90.79
CIFAR100	68.27	65.73	66.10
tiny-ImageNet	63.11	60.21	58.04

Learning invariance to rotations degrades performance!

Learning Invariance to Transformations

Learning discriminability from transformations	$\Rightarrow$	Self-supervised learning (SSL)
Learning invariance to transformations	$\Rightarrow$	Data augmentation (DA)

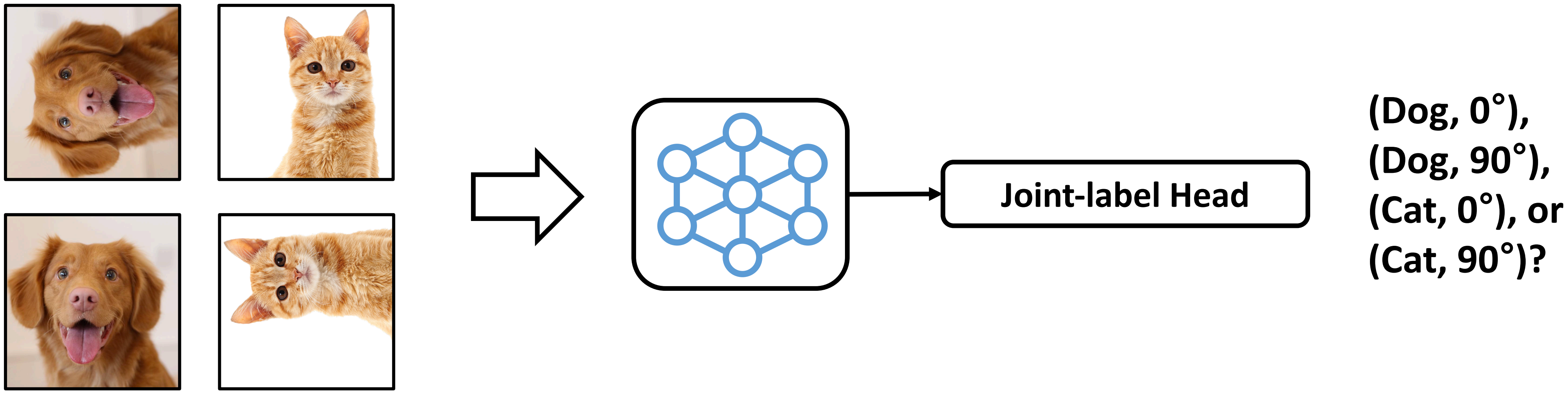
- Transformations for DA $\neq$ Transformations for SSL
  - Learning invariance to SSL transformations degrades performance
  - Ablation study:
    - We use 4 rotations with degrees of 0°, 90°, 180°, 270° for transformations $\{t_1, \ldots, t_M\}$
    - We train Baseline w/o rotation, Data Augmentation (DA), and Multi-task Learning (MT) objectives
    - In CIFAR-10/100, tiny-ImageNet, learning invariance to rotations degrades classification performance

- Similar findings in the prior work
  - AutoAugment [Cubuk et al., 2019] rotates images at most 30 degrees
  - SimCLR [Chen et al., 2020] with rotations (0°, 90°, 180°, 270°) fails to learn meaningful representations

[Cubuk et al., 2019] Autoaugment: Learning augmentation strategies from data, CVPR 2019
[Chen et al., 2020] A simple framework for contrastive learning of visual representations, 2020

Idea: Eliminating Invariance via Joint-label Classifier

- Our key idea is to **remove the unnecessary invariant property** of the classifier
  - Construct **joint-label distribution** of original and self-supervised labels
  - Use **one joint-label classifier** for the joint distribution



Idea: Eliminating Invariance via Joint-label Classifier

- Our key idea is to **remove the unnecessary invariant property** of the classifier
  - Construct **joint-label distribution** of original and self-supervised labels

$y \in \{1, 2, \dots, N\}$ Original labels

$j \in \{1, 2, \dots, M\}$ Self-supervised labels

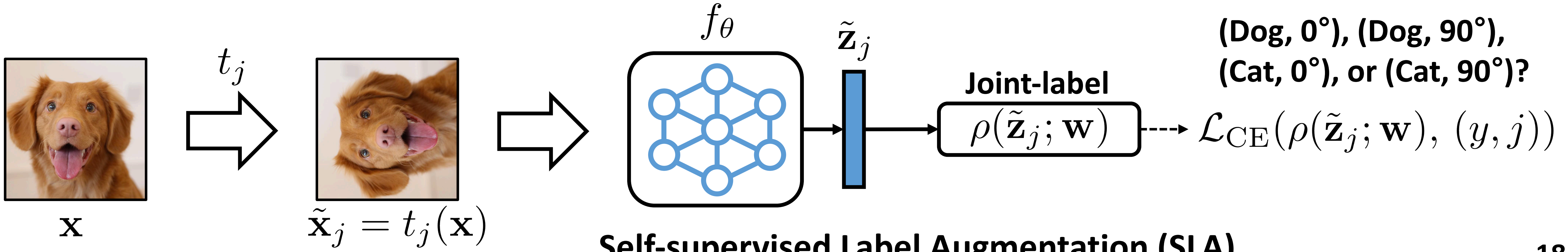
$\Rightarrow \quad (y, j) \in \{(1,1), (1,2), \dots, (N, M)\}$

    - For example, when considering 4 rotations and CIFAR-10, we have 40 joint-labels

  - Use **joint-label classifier** with a weight tensor $\mathbf{w}$ & joint-label cross-entropy loss

$$\rho_{ij}(\tilde{\mathbf{z}}; \mathbf{w}) = \frac{\exp(\mathbf{w}_{ij}^\top \tilde{\mathbf{z}})}{\sum_{k=1}^{N} \sum_{l=1}^{M} \exp(\mathbf{w}_{kl}^\top \tilde{\mathbf{z}})} \qquad \mathcal{L}_{\mathrm{CE}}(\rho(\tilde{\mathbf{z}}; \mathbf{w}),\ (y, j)) = -\log \rho_{yj}(\tilde{\mathbf{z}}; \mathbf{w})$$

    - It is equivalent to the single-label classifier with NM labels



Self-supervised Label Augmentation (SLA)

Idea: Eliminating Invariance via Joint-label Classifier

- Our key idea is to **remove the unnecessary invariant property** of the classifier
  - Construct **joint-label distribution** of original and self-supervised labels

$y \in \{1, 2, \ldots, N\}$ Original labels

$j \in \{1, 2, \ldots, M\}$ Self-supervised labels

$\Rightarrow$ $(y, j) \in \{(1,1), (1,2), \ldots, (N, M)\}$

    - For example, when considering 4 rotations and CIFAR-10, we have 40 joint-labels

  - Use **joint-label classifier** with a weight tensor $\mathbf{w}$ & joint-label cross-entropy loss

$$\rho_{ij}(\tilde{\mathbf{z}}; \mathbf{w}) = \frac{\exp(\mathbf{w}_{ij}^\top \tilde{\mathbf{z}})}{\sum_{k=1}^{N} \sum_{l=1}^{M} \exp(\mathbf{w}_{kl}^\top \tilde{\mathbf{z}})}$$

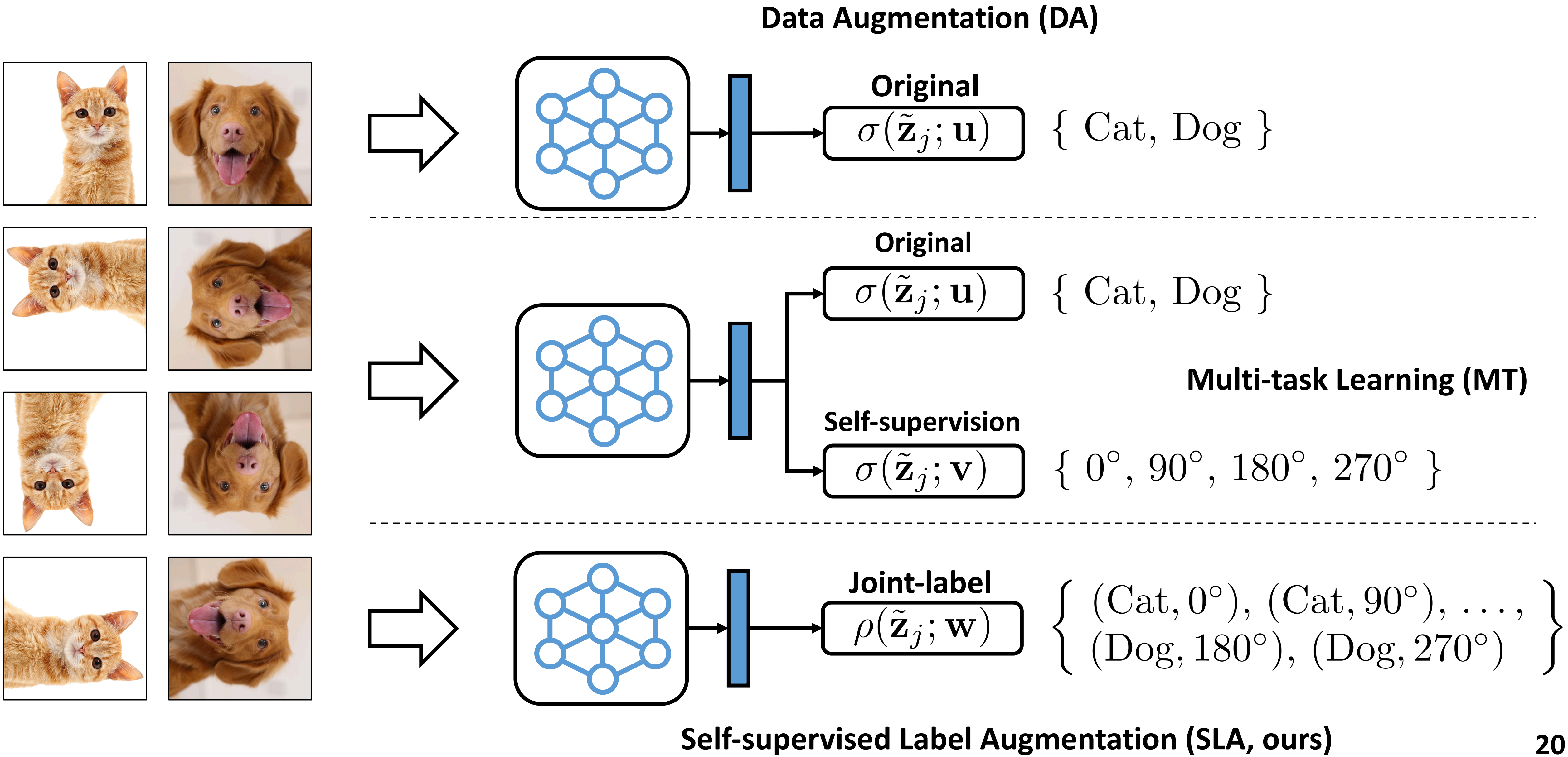
$$\mathcal{L}_{\mathrm{CE}}(\rho(\tilde{\mathbf{z}}; \mathbf{w}),\ (y, j)) = -\log \rho_{yj}(\tilde{\mathbf{z}}; \mathbf{w})$$

    - It is equivalent to the single-label classifier with NM labels

  - The objective is as follows:

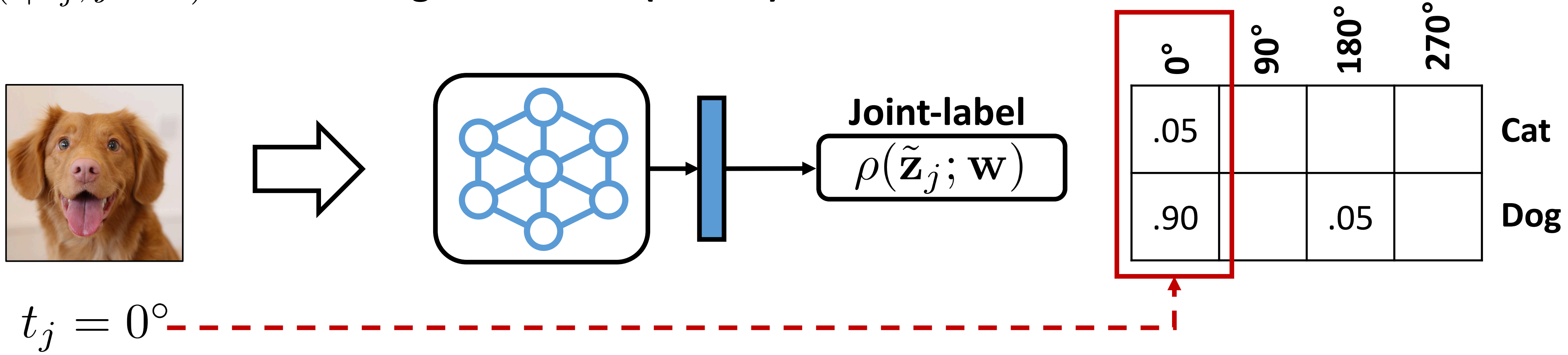
$$\mathcal{L}_{\mathrm{SLA}}(\mathbf{x}, y) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}_{\mathrm{CE}}(\rho(\tilde{\mathbf{z}}_j; \mathbf{w}),\ (y, j))$$

Comparison between DA, MT, and SLA



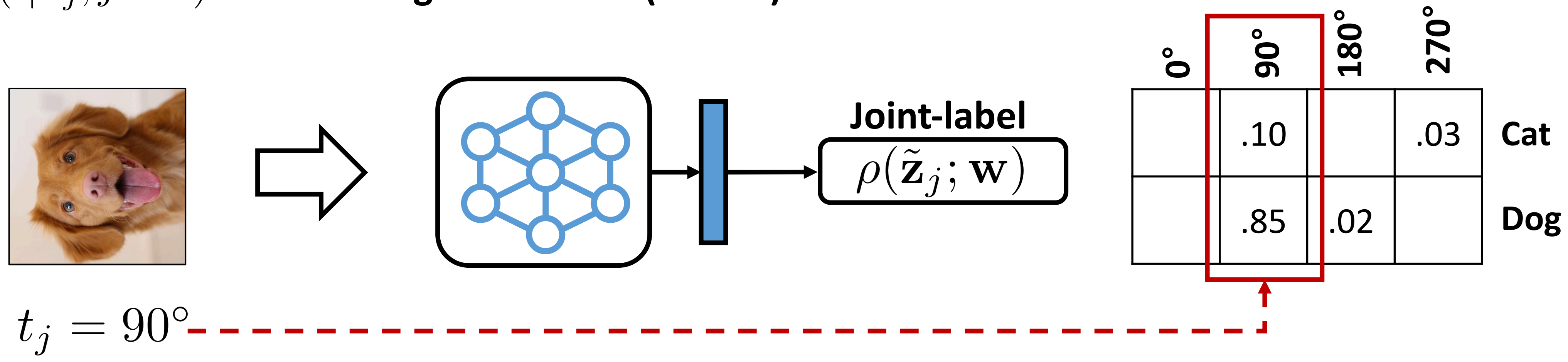
Aggregation across Transformations

- In the test phase, we do not need to consider all NM joint-labels
  - We make a prediction using the conditional probability $P(i|\tilde{\mathbf{x}}_j, j) = \exp(\mathbf{w}_{ij}^\top \tilde{\mathbf{z}}_j) / \sum_{k=1}^{N} \exp(\mathbf{w}_{kj}^\top \tilde{\mathbf{z}}_j)$
  - $P(i|\tilde{\mathbf{x}}_j, j=1)$ denotes **Single Inference (SLA+SI)**



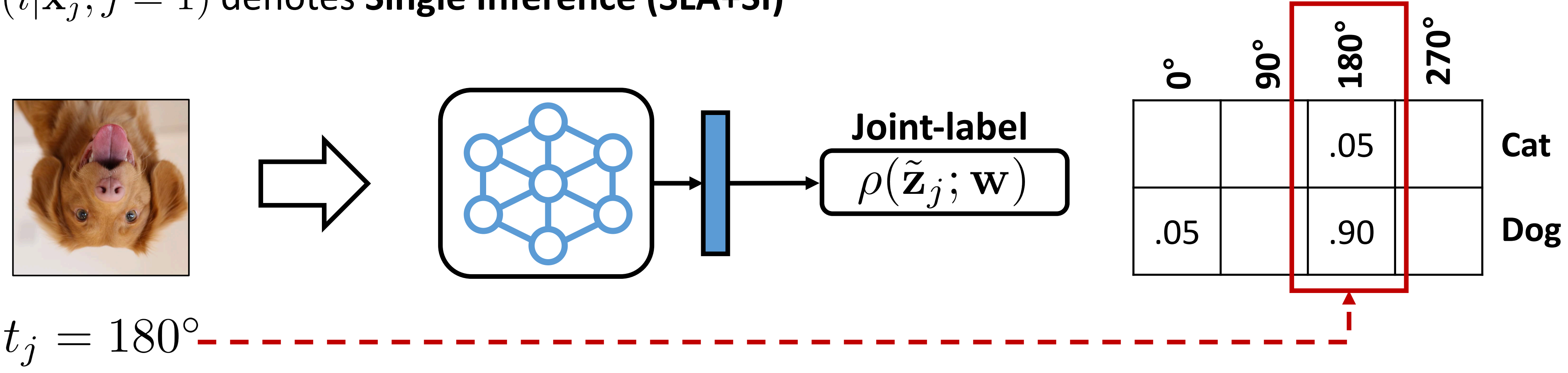
Aggregation across Transformations

- For inference, we do not need to consider all NM joint-labels
  - We make a prediction using the conditional probability $P(i|\tilde{\mathbf{x}}_j, j) = \exp(\mathbf{w}_{ij}^\top \tilde{\mathbf{z}}_j) / \sum_{k=1}^{N} \exp(\mathbf{w}_{kj}^\top \tilde{\mathbf{z}}_j)$
  - $P(i|\tilde{\mathbf{x}}_j, j=1)$ denotes **Single Inference (SLA+SI)**



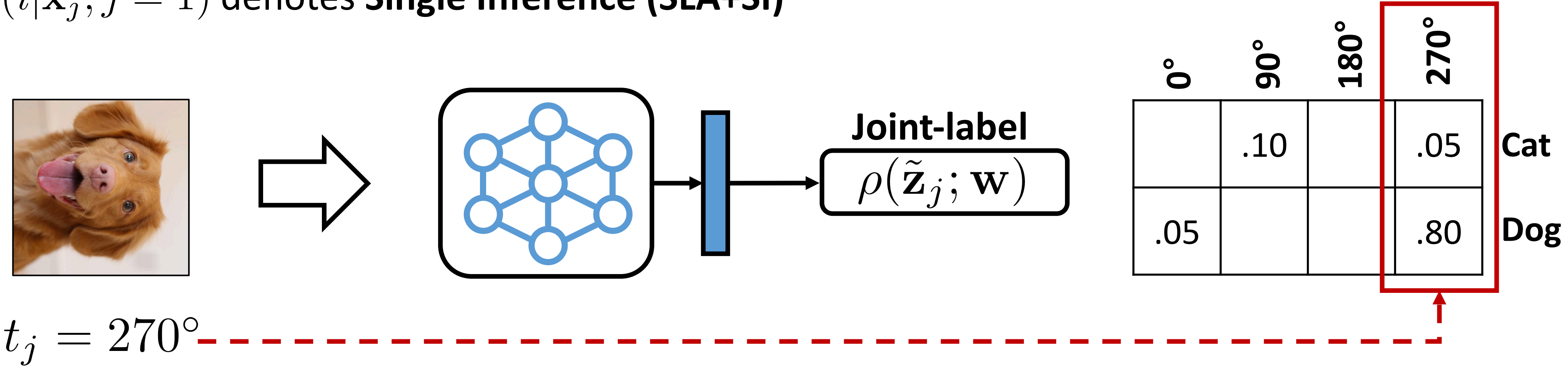
Aggregation across Transformations

- For inference, we do not need to consider all NM joint-labels
  - We make a prediction using the conditional probability $P(i|\tilde{\mathbf{x}}_j, j) = \exp(\mathbf{w}_{ij}^\top \tilde{\mathbf{z}}_j) / \sum_{k=1}^{N} \exp(\mathbf{w}_{kj}^\top \tilde{\mathbf{z}}_j)$
  - $P(i|\tilde{\mathbf{x}}_j, j=1)$ denotes **Single Inference (SLA+SI)**



Aggregation across Transformations

- For inference, we do not need to consider all NM joint-labels
  - We make a prediction using the conditional probability $P(i|\tilde{\mathbf{x}}_j, j) = \exp(\mathbf{w}_{ij}^\top \tilde{\mathbf{z}}_j) / \sum_{k=1}^{N} \exp(\mathbf{w}_{kj}^\top \tilde{\mathbf{z}}_j)$
  - $P(i|\tilde{\mathbf{x}}_j, j=1)$ denotes **Single Inference (SLA+SI)**



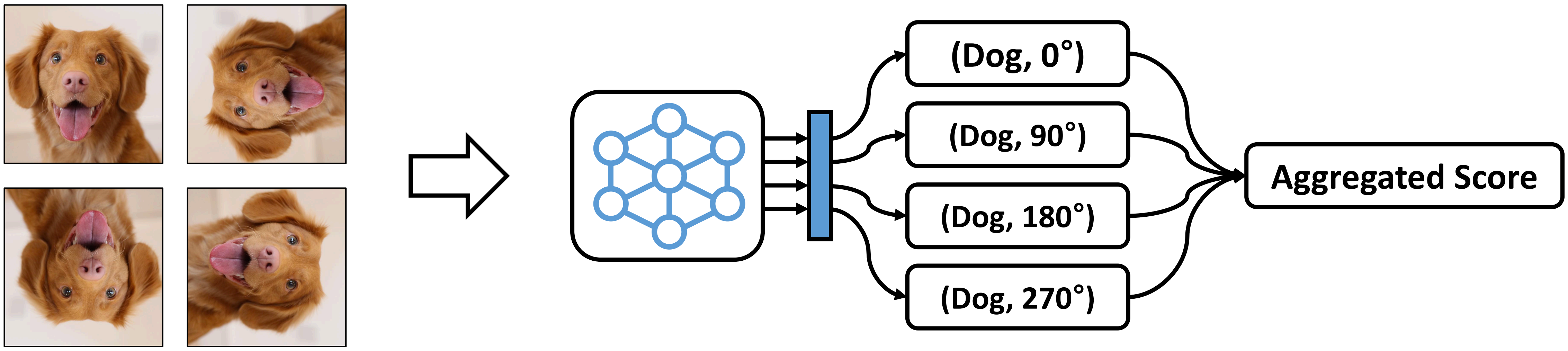
Aggregation across Transformations

- For inference, we do not need to consider all NM joint-labels
  - We make a prediction using the conditional probability $P(i|\tilde{\mathbf{x}}_j, j) = \exp(\mathbf{w}_{ij}^\top \tilde{\mathbf{z}}_j) / \sum_{k=1}^{N} \exp(\mathbf{w}_{kj}^\top \tilde{\mathbf{z}}_j)$
  - $P(i|\tilde{\mathbf{x}}_j, j=1)$ denotes **Single Inference (SI)**

- For all transformations $\{t_j\}$, we **aggregate** the corresponding conditional probabilities

$$P_{\text{aggregated}}(i|\mathbf{x}) = \frac{\exp(s_i)}{\sum_{k=1}^{N} \exp(s_k)} \quad \text{where} \quad s_i = \frac{1}{M} \sum_{j=1}^{M} \mathbf{w}_{ij}^\top \tilde{\mathbf{z}}_j$$

  - $P_{\text{aggregated}}(i|\mathbf{x})$ denotes **Aggregated Inference (SLA+AG)**



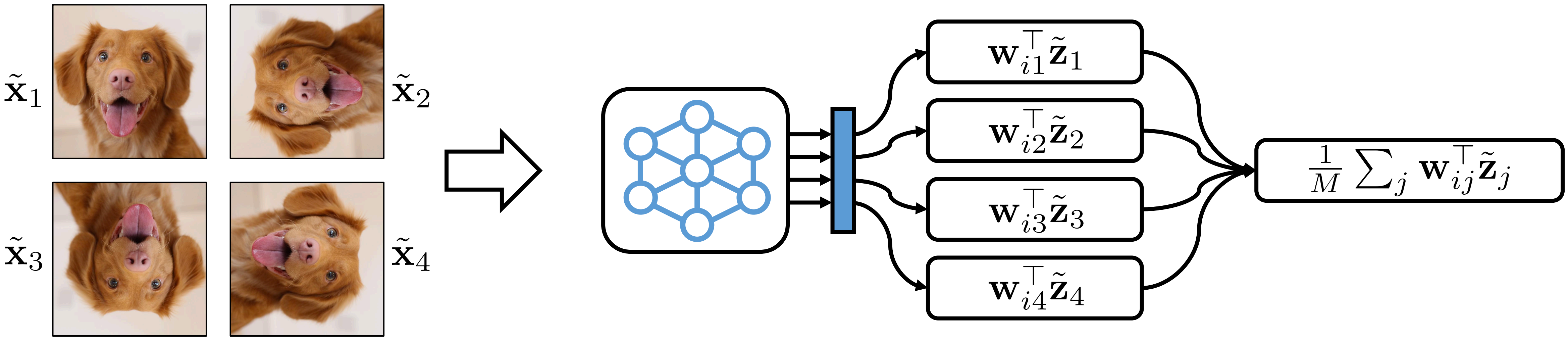
Aggregation across Transformations

- For inference, we do not need to consider all NM joint-labels
  - We make a prediction using the conditional probability $P(i|\tilde{\mathbf{x}}_j, j) = \exp(\mathbf{w}_{ij}^\top \tilde{\mathbf{z}}_j) / \sum_{k=1}^{N} \exp(\mathbf{w}_{kj}^\top \tilde{\mathbf{z}}_j)$
  - $P(i|\tilde{\mathbf{x}}_j, j=1)$ denotes **Single Inference (SI)**

- For all transformations $\{t_j\}$, we **aggregate** the corresponding conditional probabilities

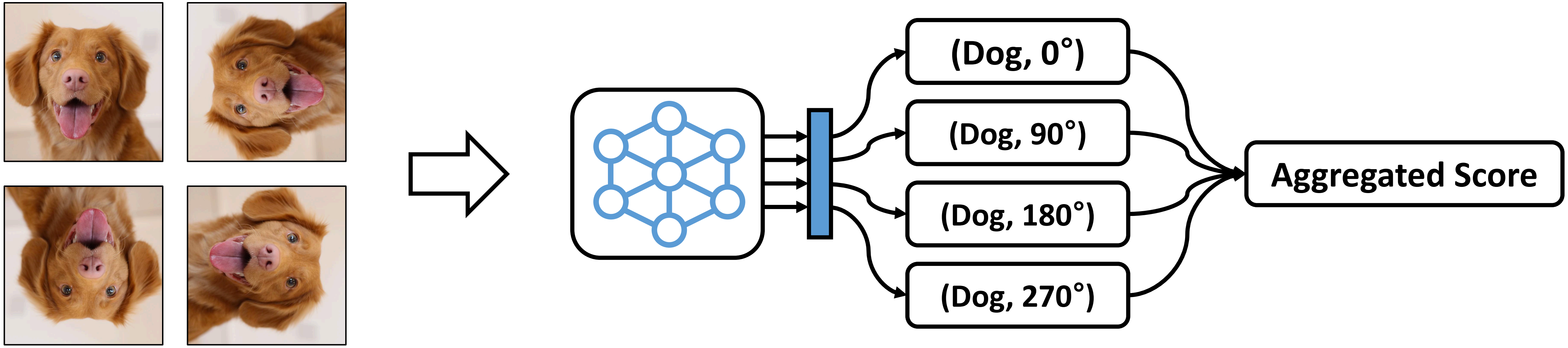
$$P_{\text{aggregated}}(i|\mathbf{x}) = \frac{\exp(s_i)}{\sum_{k=1}^{N} \exp(s_k)} \quad \text{where} \quad s_i = \frac{1}{M} \sum_{j=1}^{M} \mathbf{w}_{ij}^\top \tilde{\mathbf{z}}_j$$

  - $P_{\text{aggregated}}(i|\mathbf{x})$ denotes **Aggregated Inference (SLA+AG)**

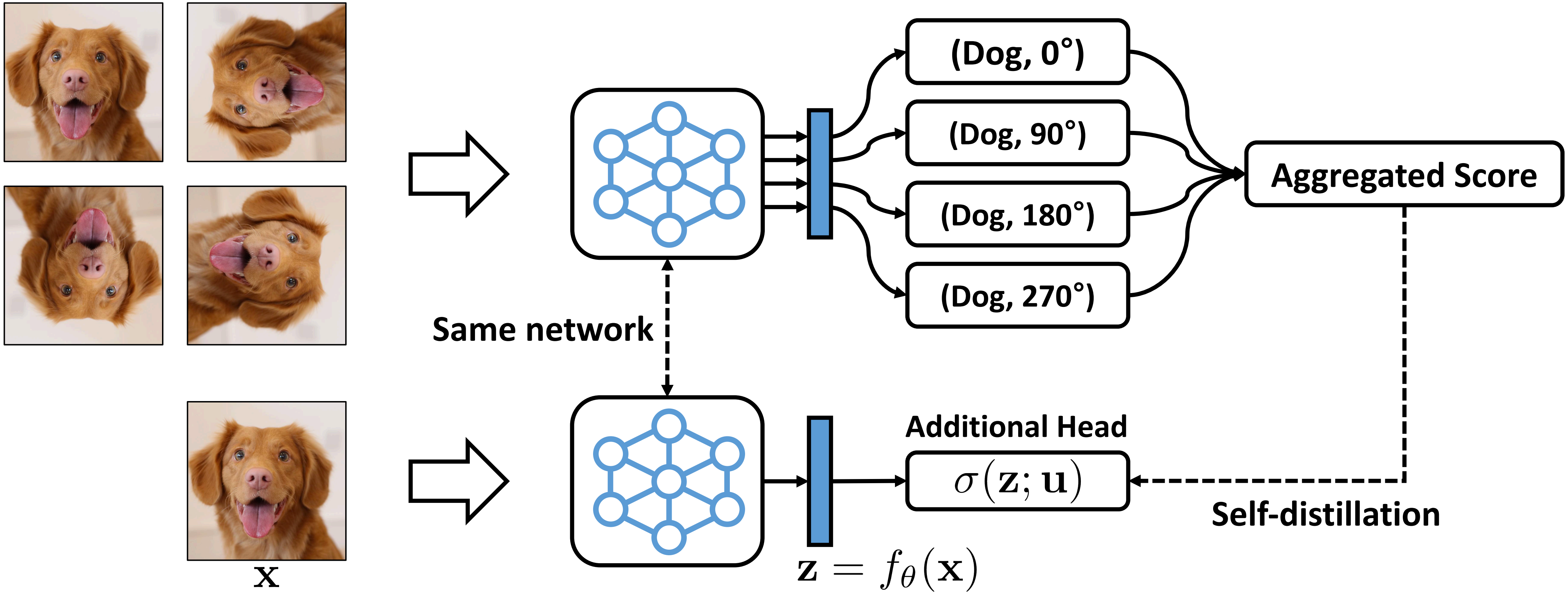


Self-distillation from Aggregation

- The aggregation scheme $P_{\text{aggregated}}(i|\mathbf{x})$ improves accuracy significantly
  - Note that this requires only a single model, but acts as an ensemble
  - Surprisingly, it achieves comparable performance with the ensemble of multiple independent models



Self-distillation from Aggregation



- We propose a self-distillation scheme for further improvements

$$\mathcal{L}_{\mathrm{SLA+SD}}(\mathbf{x}, y) = \mathcal{L}_{\mathrm{SLA}}(\mathbf{x}, y) + \underbrace{D_{\mathrm{KL}}(P_{\mathrm{aggregated}} \| \sigma(\mathbf{z};\mathbf{u}))}_{\text{Distillation term}} + \underbrace{\mathcal{L}_{\mathrm{CE}}(\sigma(\mathbf{z};\mathbf{u}),\ y)}_{\text{Classification term}}$$

  - $\sigma(\mathbf{z};\mathbf{u})$ denotes **Self-Distillation (SLA+SD)**

Outline

Self-supervised Learning

- What is self-supervised learning?
- Applications of self-supervision
- *Motivation:* How effectively utilize self-supervision in fully-supervised settings?

Self-supervised Label Augmentation (SLA)

- *Observation:* Learning invariance to transformations
- *Main idea:* Eliminating invariance via joint-label classifier
- **Aggregation** across all transformations & **Self-distillation** from aggregation

Experiments

- Standard fully-supervised / few-shot / imbalance settings

Experiments

- Transformations
  - **Rotation** (M=4)

0° 90° 180° 270°

  - **Color permutation** (M=6)

RGB RBG GRB GBR BRG BGR

- Classification tasks
  - **Standard classification:** CIFAR-10/100, CUB200, MIT67, Stanford Dogs, tiny-ImageNet
  - **Few-shot classification:** mini-ImageNet, CIFAR-FS, FC100
  - **Imbalance classification:** CIFAR-10/100

Standard Classification

- Self-supervised label augmentation (SLA) improves classification accuracy by large margin

Dataset	Baseline	Rotation SLA+SD	Rotation SLA+AG	Color Permutation SLA+SD	Color Permutation SLA+AG
CIFAR10	92.39	93.26 (+0.94%)	94.50 (+2.28%)	91.51 (-0.95%)	92.51 (+0.13%)
CIFAR100	68.27	71.85 (+5.24%)	74.14 (+8.60%)	68.33 (+0.09%)	69.14 (+1.27%)
CUB200	54.24	62.54 (+15.3%)	64.41 (+18.8%)	60.95 (+12.4%)	61.10 (+12.6%)
MIT67	54.75	63.54 (+16.1%)	64.85 (+18.4%)	60.03 (+9.64%)	59.99 (+9.57%)
Stanford Dogs	60.62	66.55 (+9.78%)	68.70 (+13.3%)	65.92 (+8.74%)	67.03 (+10.6%)
tiny-ImageNet	63.11	65.53 (+3.83%)	66.95 (+6.08%)	63.98 (+1.38%)	64.15 (+1.65%)

  - Using **rotation** as label augmentation improves classification accuracy on **all datasets**
  - Using **color permutation** provides meaningful gains on **fine-grained datasets**
  - Our aggregation scheme (SLA+AG) competes with independent ensemble (IE) of multiple models

Dataset	Single Model Baseline	Single Model SLA+AG	4 Models IE	4 Models IE + SLA+AG
CIFAR10	92.39	**94.50**	94.36	**95.10**
CIFAR100	68.27	**74.14**	74.82	**76.40**
tiny-ImageNet	63.11	**66.95**	68.18	**69.01**

Standard Classification

- Self-supervised label augmentation (SLA) improves classification accuracy by large margin

Dataset	Baseline	Rotation SLA+SD	Rotation SLA+AG	Color Permutation SLA+SD	Color Permutation SLA+AG
CIFAR10	92.39	93.26 (+0.94%)	94.50 (+2.28%)	91.51 (-0.95%)	92.51 (+0.13%)
CIFAR100	68.27	71.85 (+5.24%)	74.14 (+8.60%)	68.33 (+0.09%)	69.14 (+1.27%)
CUB200	54.24	62.54 (+15.3%)	64.41 (+18.8%)	60.95 (+12.4%)	61.10 (+12.6%)
MIT67	54.75	63.54 (+16.1%)	64.85 (+18.4%)	60.03 (+9.64%)	59.99 (+9.57%)
Stanford Dogs	60.62	66.55 (+9.78%)	68.70 (+13.3%)	65.92 (+8.74%)	67.03 (+10.6%)
tiny-ImageNet	63.11	65.53 (+3.83%)	66.95 (+6.08%)	63.98 (+1.38%)	64.15 (+1.65%)

  - Using **rotation** as label augmentation improves classification accuracy on **all datasets**
  - Using **color permutation** provides meaningful gains on **fine-grained datasets**
  - Our aggregation scheme (SLA+AG) competes with independent ensemble (IE) of multiple models
  - Furthermore, our SLA can be combined with existing augmentation techniques
    - Cutout, AutoAugment, CutMix

	CIFAR10	CIFAR100
WRN-40-2	5.24	25.63
+ Cutout	4.33	23.87
+ Cutout + **SLA+SD** (ours)	3.36	20.42
+ AutoAugment	3.70	21.44
+ AutoAugment + **SLA+SD** (ours)	**2.95**	**18.87**
PyramidNet200	3.85	16.45
+ Mixup	3.09	15.63
+ CutMix	2.88	14.47
+ CutMix + **SLA+SD** (ours)	**1.80**	**12.24**

Various Classification Scenarios

- Few-shot setting

Method	mini-ImageNet		CIFAR-FS		FC100	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MAML† (Finn et al., 2017)	48.70±1.84	63.11±0.92	58.9±1.9	71.5±1.0	-	-
R2D2† (Bertinetto et al., 2019)	-	-	65.3±0.2	79.4±0.1	-	-
RelationNet† (Sung et al., 2018)	50.44±0.82	65.32±0.70	55.0±1.0	69.3±0.8	-	-
SNAIL (Mishra et al., 2018)	55.71±0.99	68.88±0.92	-	-	-	-
TADAM (Oreshkin et al., 2018)	58.50±0.30	76.70±0.30	-	-	40.1±0.4	56.1±0.4
LEO‡ (Rusu et al., 2019)	61.76±0.08	77.59±0.12	-	-	-	-
MetaOptNet-SVM (Lee et al., 2019)	62.64±0.61	78.63±0.46	72.0±0.7	84.2±0.5	41.1±0.6	55.5±0.6
ProtoNet (Snell et al., 2017)	59.25±0.64	75.60±0.48	72.2±0.7	83.5±0.5	37.5±0.6	52.5±0.6
ProtoNet + **SLA+AG** (ours)	62.22±0.69	77.78±0.51	**74.6±0.7**	**86.8±0.5**	40.0±0.6	55.7±0.6
MetaOptNet-RR (Lee et al., 2019)	61.41±0.61	77.88±0.46	72.6±0.7	84.3±0.5	40.5±0.6	55.3±0.6
MetaOptNet-RR + **SLA+AG** (ours)	**62.93±0.63**	**79.63±0.47**	73.5±0.7	86.7±0.5	**42.2±0.6**	**59.2±0.5**

- Imbalanced setting

	Imbalanced CIFAR10		Imbalanced CIFAR100	
Imbalance Ratio ($N_{\text{max}}/N_{\text{min}}$)	100	10	100	10
Baseline	70.36	86.39	38.32	55.70
Baseline + **SLA+SD** (ours)	74.61 (+6.04%)	89.55 (+3.66%)	43.42 (+13.3%)	60.79 (+9.14%)
CB-RW (Cui et al., 2019)	72.37	86.54	33.99	57.12
CB-RW + **SLA+SD** (ours)	77.02 (+6.43%)	89.50 (+3.42%)	37.50 (+10.3%)	**61.00** (+6.79%)
LDAM-DRW (Cao et al., 2019)	77.03	88.16	42.04	58.71
LDAM-DRW + **SLA+SD** (ours)	**80.24** (+4.17%)	**89.58** (+1.61%)	**45.53** (+8.30%)	59.89 (+1.67%)

These show that SLA can be easily combined with existing approaches in various classification tasks!

Conclusion

- We consider **self-supervision** in **full-supervised** settings for improving classification accuracy
- We propose **Self-supervised Label Augmentation (SLA)** which augments the label space using self-supervised transformations
    - We propose additional techniques, aggregation and self-distillation
- We demonstrate the **wide applicability** and **compatibility** of SLA in various classification scenarios including few-shot and imbalanced settings
- We believe that the simplicity and effectiveness of SLA could bring in many interesting directions for future research
    - Using aggregation scheme for constructing pseudo labels in semi-supervised learning
    - Applying SLA to the contrastive learning frameworks, e.g., SimCLR [Chen et al., 2020]

[Chen et al., 2020] A simple framework for contrastive learning of visual representations, 2020

Thank you for listening!

hankook.lee @ kaist.ac.kr