PAWS

Semi-Supervised Learning of Visual Features by Non-Parametrically Predicting View Assignments with Support Samples

Generative Models Reading Group

December 6, 2021

How do we best combine self-supervision with limited labeled data?

Common approach is to first pretrain then fine-tune on labeled data.

But when we do this the self-supervised schemes are less compute efficient than supervised learning + fine-tuning.

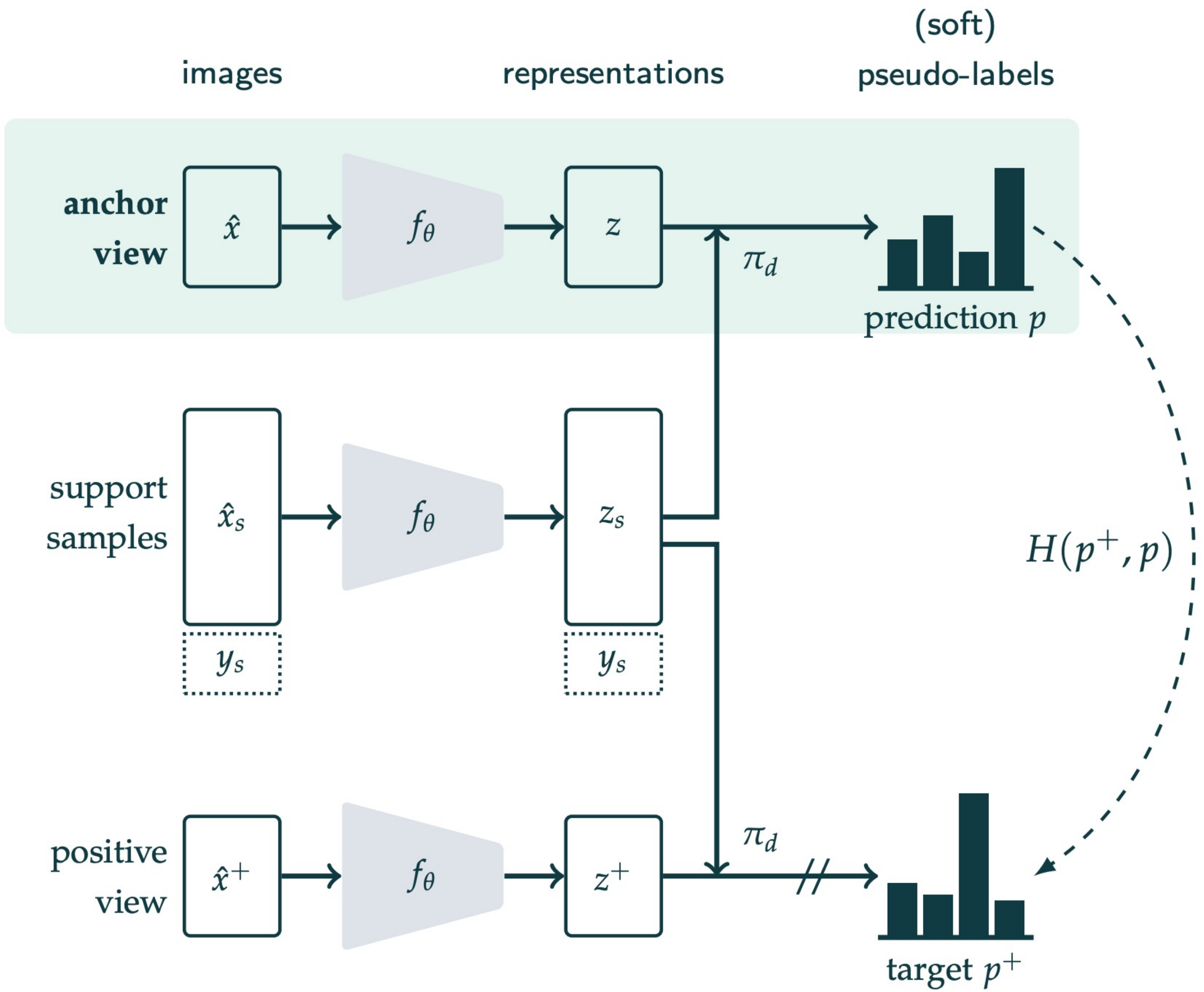
Instead use “pseudo-labels” as in SimCLR2 (use labeled examples to pseudo-label others)

[**Contribution #1**] PAWs incorporates labels throughout training (rather than pretraining-> pseudo-labeling -> fine-tuning on labels

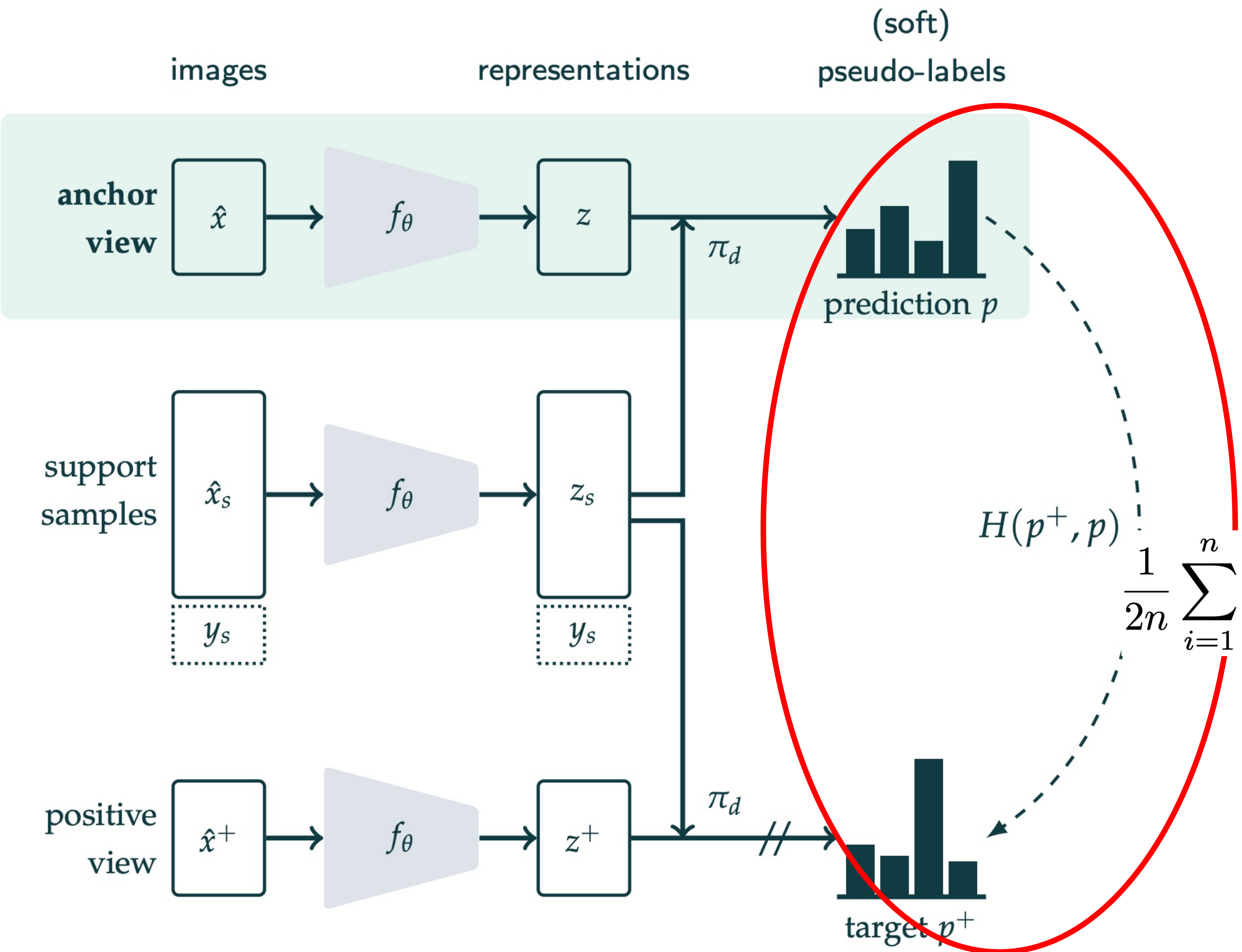
This extends BYOL and SWaV contrastive SSL setups to the semi-supervised setting (learning with limited labels).

[**Contribution #2**] Prediction sharpening/regularization instead of asymmetry in model architecture.

Learning by Predicting View Assignments with Support Samples



Symmetric? Sharpen predictions and regularize



Sharpening:

$$[\rho(p_i)]_k := \frac{[p_i]_k^{1/T}}{\sum_{j=1}^{K} [p_i]_j^{1/T}}$$

$$k = 1, \ldots, K.$$

Regularization (use all classes in support):

$$\overline{p} := \frac{1}{2n} \sum_{i=1}^{n} \left(\rho(p_i) + \rho(p_i^+)\right)$$

aka mean entropy maximization

Full Loss:

$$\frac{1}{2n} \sum_{i=1}^{n} \left(H(\rho(p_i^+), p_i) + H(\rho(p_i), p_i^+)\right) - H(\overline{p})$$

	Top 1	
	1%	10%
With ME-MAX	63.8	73.9
Without ME-MAX	52.9	73.6

Guarantees against Trivial Solutions

Assumption 1 (Class Balanced Sampling)**.** Each mini-batch of labeled support samples contains an equal number of instances from each of the sampled classes.

Assumption 2 (Target Sharpening)**.** The target p^+ is sharpened, such that it is not equal to the uniform distribution.

Proposition 1 (Non-Collapsing Representations)**.** Suppose Assumptions 1 and 2 hold. If f_θ is such that the representations collapse, i.e., $z_i = z$ for all $z_i \in \mathcal{S}$, then $\|\nabla_\theta H(p^+, p)\| > 0$.

Proof. Since $z = z_i$ for all $z_i \in \mathcal{S}$, it holds that $d(z, z_i) = d(z, z_j)$ for all $z_i, z_j \in \mathcal{S}$. Therefore $p := \pi_d(z, \mathcal{S}) = {}^1/_n \sum_{(z_i, y_i)} y_i$, where y_i is the one-hot class label for the representation z_i. Let K denote the number of classes represented in the mini-batch of support samples. By Assumption 1, since the mini-batch of support samples contains an equal number of instances from each sampled class, it follows that there are ${}^n/_K$ instances for each of the K represented classes. Therefore, the prediction p further simplifies to $\frac{1}{n}\left(\mathbf{1_K}\frac{n}{K}\right) = \frac{1}{K}\mathbf{1_K}$, the uniform distribution over the K classes. However, by Assumption 2, the targets p^+ are sharpened such that they are not equal to the uniform distribution. Therefore, $p \neq p^+$, from which it follows that $\|\nabla H(p^+, p)\| > 0$. ■

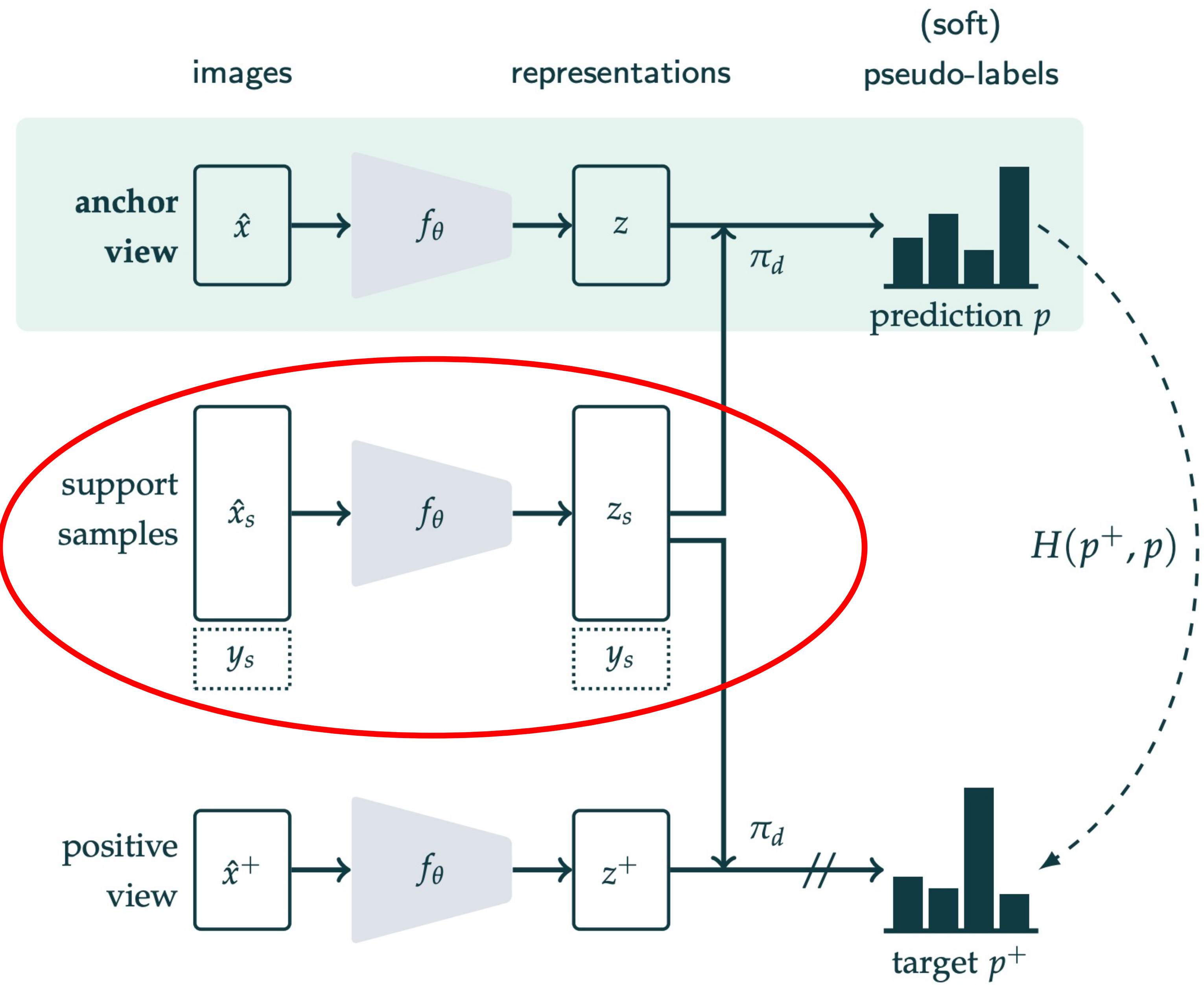
Idea: Collapsed representations have high entropy.

But we sharpen our predictions -> low entropy.

This means the collapsed representation is not a stationary point of our optimization.

This THM can be extended to transformations of y_i's i.e. label smoothing.

Draw support samples from labeled data



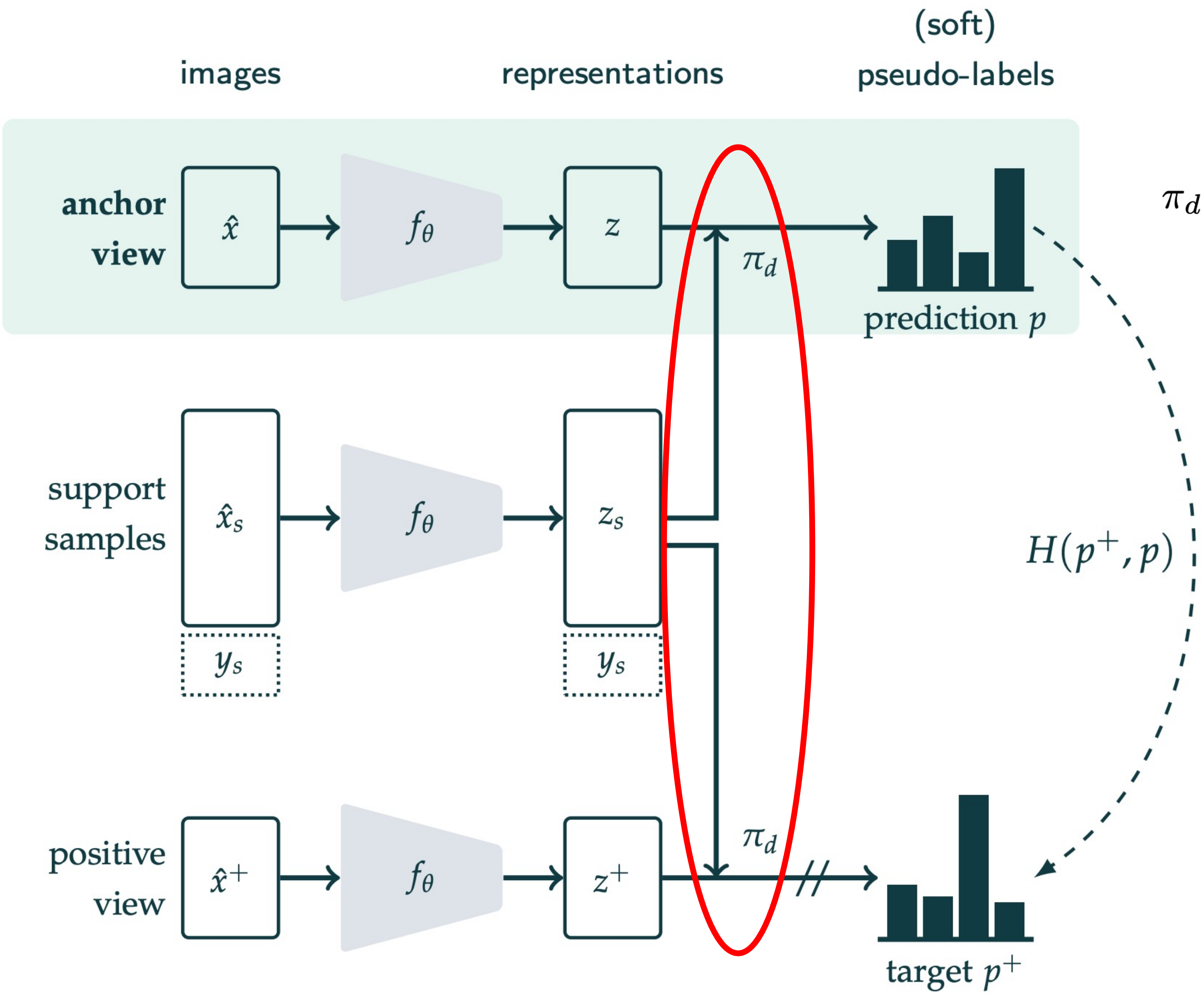
"Support samples" drawn from labeled data

More support samples -> better performance

More classes > more examples per class

Classes	Imgs. per Class	Top 1 1%	Top 1 10%
1000	16	–	74.5
1000	12	63.9	74.2
960	7	63.8	73.9
960	4	63.7	72.0
448	8	61.8	70.1

Semi-supervision by assignment to support samples



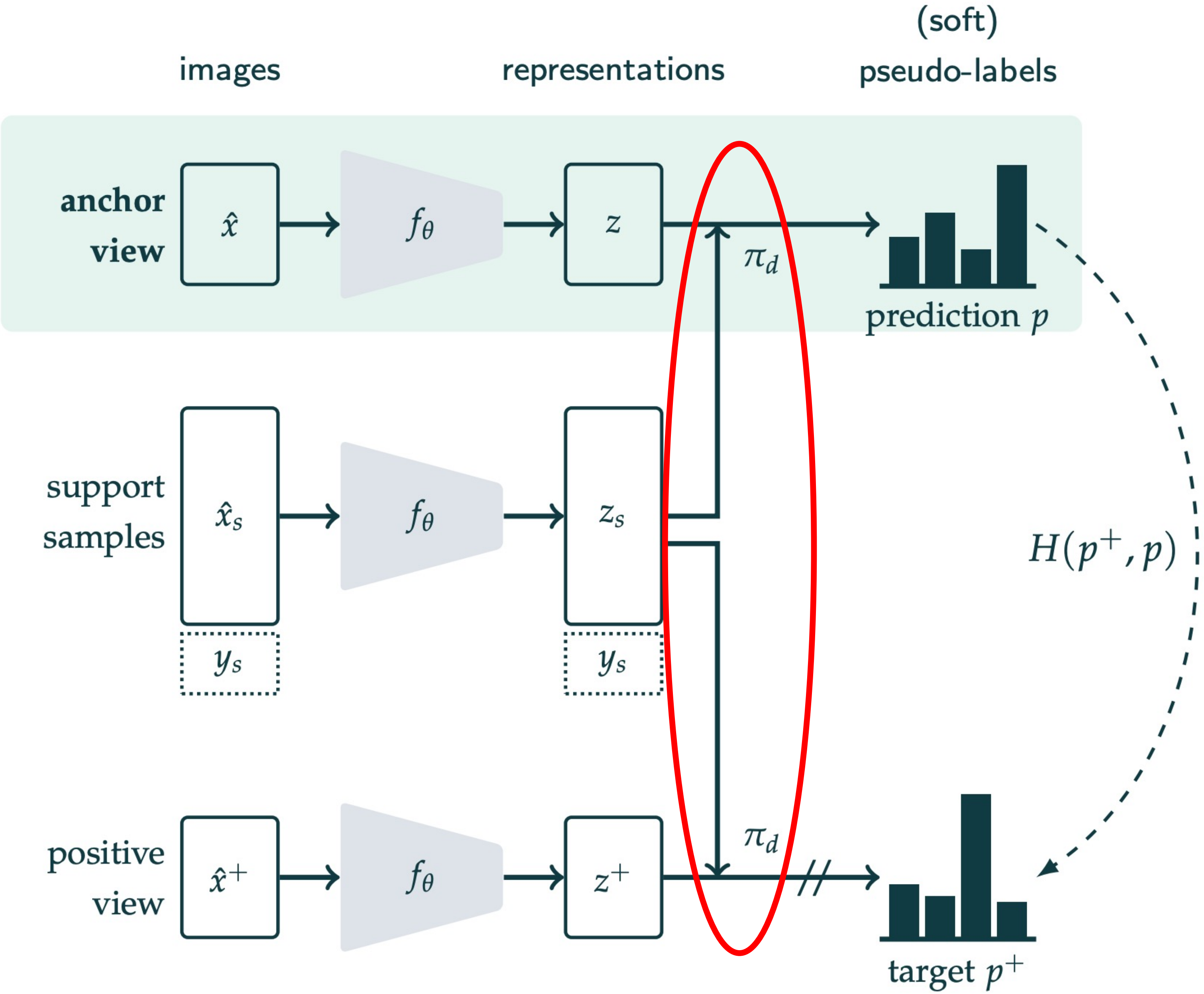
$$\pi_d(z_i, \mathbf{z}_\mathcal{S}) = \sum_{(z_{s\,j}, y_j) \in \mathbf{z}_\mathcal{S}} \left(\frac{d(z_i, z_{s\,j})}{\sum_{z_{s\,k} \in \mathbf{z}_\mathcal{S}} d(z_i, z_{s\,k})} \right) y_j$$

Soft nonparametric assignment

Similarity metric and predictions. In this work, we take the similarity metric $d(a, b)$ to be $\exp(a^T b / \|a\| \|b\| \tau)$, the exponential temperature-scaled cosine. For L2-normalized representations, the similarity classifier $\pi_d(\cdot, \cdot)$ can be concisely written as

$$p_i := \pi_d(z_i, \mathbf{z}_\mathcal{S}) = \sigma_\tau(z_i \mathbf{z}_\mathcal{S}^\top) \mathbf{y}_\mathcal{S},$$

PAWS as a Neural Architecture with External Memory



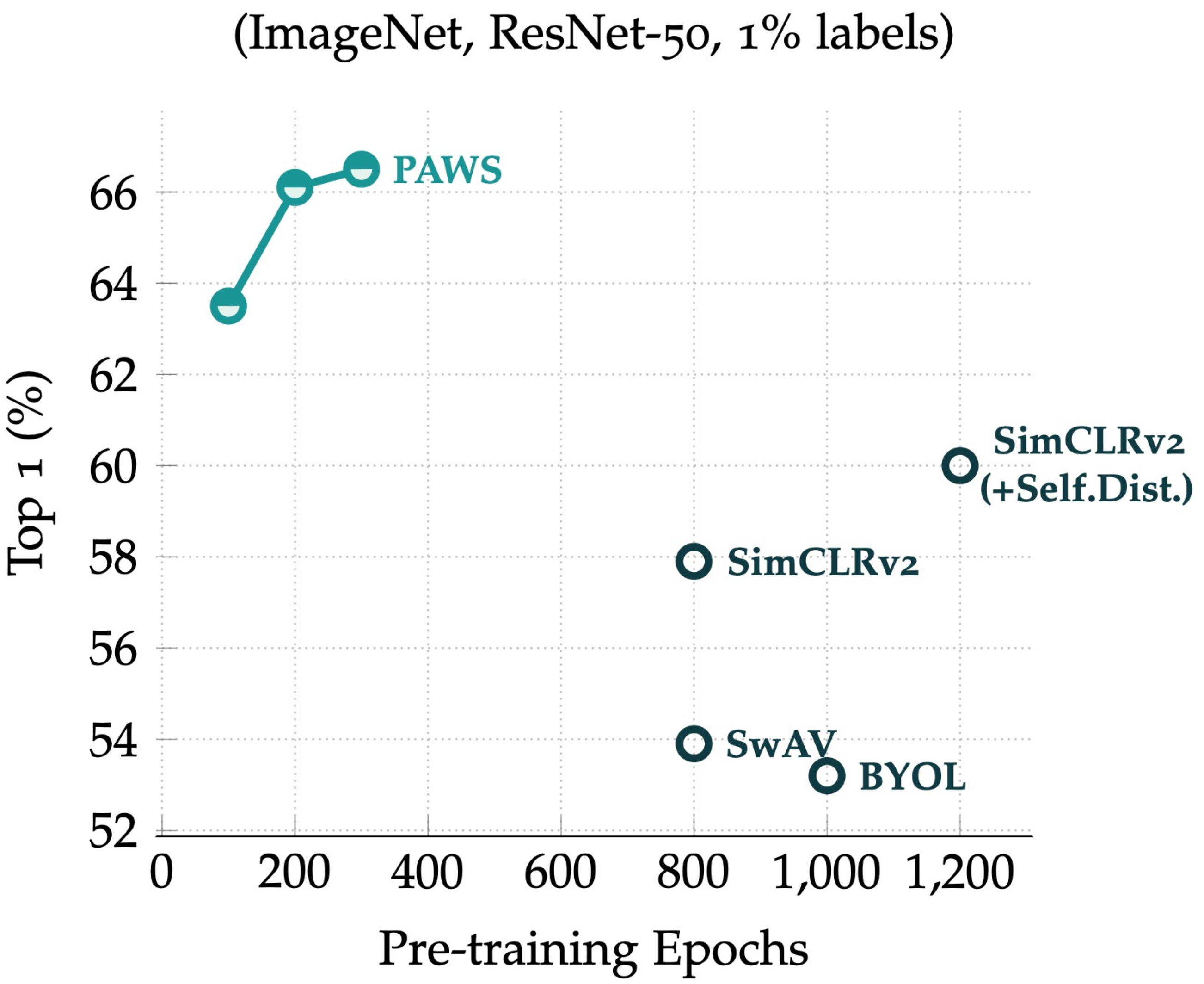
Support samples can be thought of as an external memory bank

Nonparametric classifier retrieves memory elements given a query vector

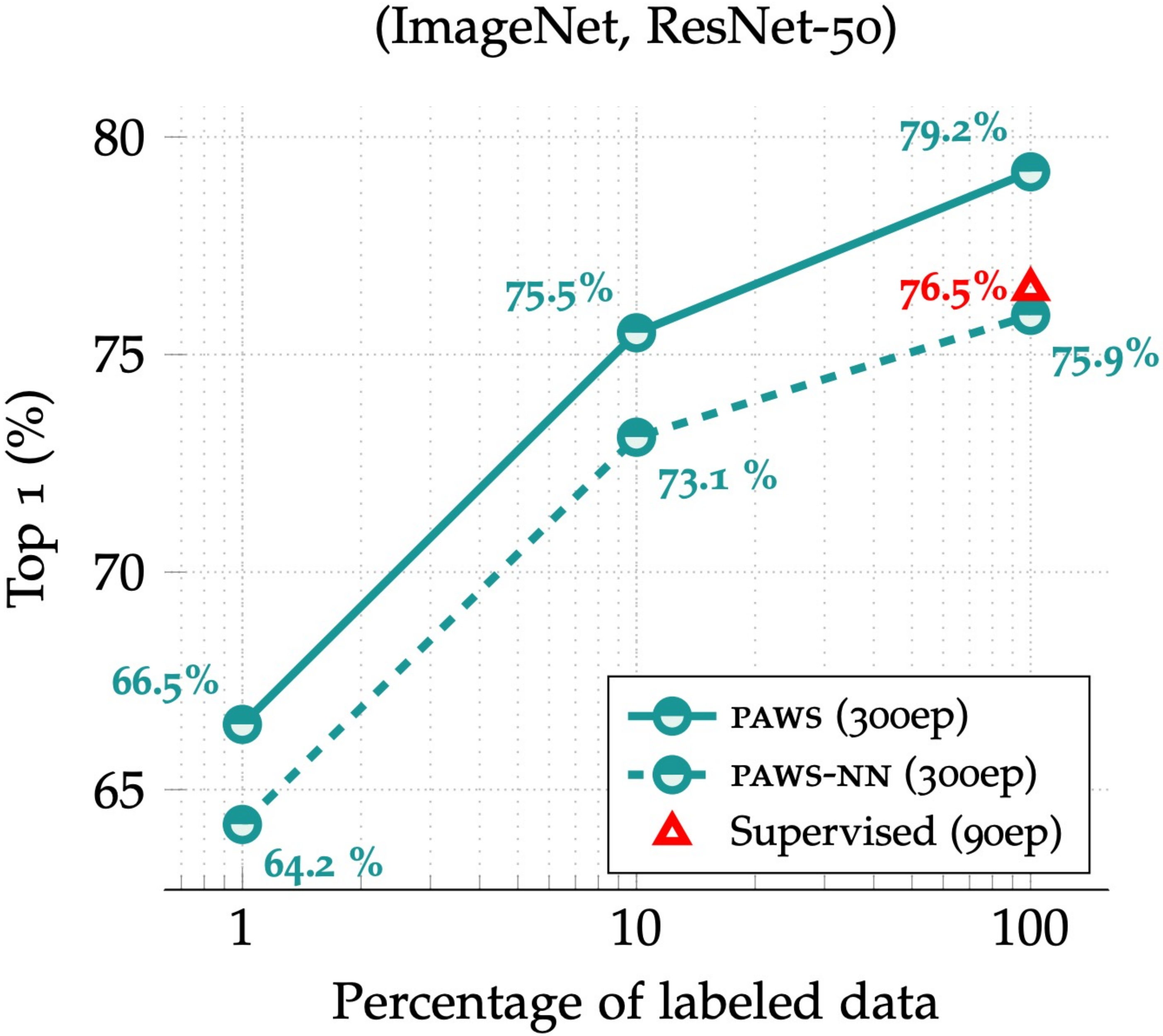
Optimization causes two views of the same image to query the same elements of the memory

Evaluations

Method	ResNet50 Epochs	Top 1 1%	Top 1 10%
Methods using label propagation:			
UDA [15]	800	–	68.1
FixMatch [11]	300	–	71.5
MPL [6]	*800	–	73.9
Methods using only representation learning:			
BYOL [4]	1000	53.2	68.8
SwAV [3]	800	53.9	70.2
SwAV+CT [40]	400	–	70.8
SimCLRv2 [1]	800	57.9	68.4
SimCLRv2 (+Self.Dist.) [1]	1200	60.0	70.5
PAWS	**100**	**63.8**	**73.9**
PAWS	**200**	**66.1**	**75.0**
PAWS	**300**	**66.5**	**75.5**
Non-parametric classification (no fine-tuning):			
PAWS-NN	100	61.5	71.0
PAWS-NN	200	63.2	71.9
PAWS-NN	300	64.2	73.1



Evaluations: Comparison to Supervised



TPU goes…

1/6th the compute of SWaV, better performance.

By reducing the number of pre-training epochs, PAWS can obtain significant computational savings compared to other approaches. We illustrate this observation by comparing PAWS training time on 64 NVIDIA V100-16G GPUs to the self-supervised SwAV method trained on identical hardware [3]. Pre-training with SwAV for 800 epochs requires 49.6 hours, while pre-trianing with PAWS for 100 epochs only requires 8.2 hours, and results in a +9.9% improvement in top-1 accuracy in the 1% label setting, and a +3.7% improvement in top-1 accuracy in the 10% label setting. In contrast to SimCLRv2 and BYOL, the PAWS method does not use an additional momentum encoder or a memory buffer, and thereby avoids this added computational and memory overhead, but may also benefit (in terms of final model accuracy) by incorporating such innovations.

No need for longer training.

		Top-1	
Architecture	**Epochs**	**1%**	**10%**
ResNet-50	100	63.8	73.9
ResNet-50	200	66.1	75.0
ResNet-50	300	66.5	75.5
ResNet-50 (2×)	100	68.2	77.0
ResNet-50 (2×)	200	69.6	77.8
ResNet-50 (2×)	300	69.6	77.7