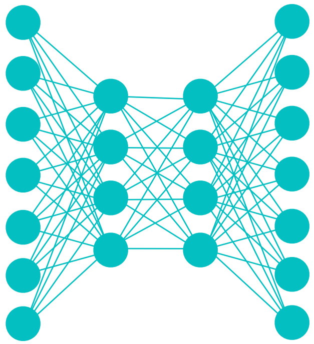
Lecture Notes for

Neural Networks and Machine Learning

Semi-supervised Loss Incorporation



Logistics and Agenda

- Logistics
  - Lab one due soon!!
- Agenda
  - Consistency Loss
  - Temporal Output Discrepancy
  - Student Paper Presentation
- Next Time
  - Multi-modal and Multi-Task
  - Multi-task demo and Town Hall
  - Finish Demos

Last Time

$$\min_{\mathbf{w}} \overbrace{\mathbf{E}_{\mathbf{x},y\in L}[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})]}^{\text{cross entropy}} + \lambda \overbrace{\mathscr{D}_{KL}\left(p_{\mathbf{w}}(y\,|\,\mathbf{x})\,||\,p_{\mathbf{w}}(y\,|\,\hat{\mathbf{x}})\right)}^{\text{consistency in augmentation}}$$

$$\mathscr{D}_{KL}(f\,||\,g) = -\sum f(x)\cdot\log\frac{g(x)}{f(x)}$$ definition of Kullback-Leibler (KL) Divergence

$$\mathscr{D}_{KL}(p(y\,|\,\mathbf{x})\,||\,p(y\,|\,\hat{\mathbf{x}})) = -\sum p(y\,|\,\mathbf{x})\cdot\log\frac{p(y\,|\,\hat{\mathbf{x}})}{p(y\,|\,\mathbf{x})} = -\sum p(y\,|\,\mathbf{x})\cdot\left(\log p(y\,|\,\hat{\mathbf{x}}) - \log p(y\,|\,\mathbf{x})\right)$$

$$= -\sum p(y\,|\,\mathbf{x})\cdot\log p(y\,|\,\hat{\mathbf{x}}) + \sum p(y\,|\,\mathbf{x})\cdot\log p(y\,|\,\mathbf{x})$$

$$= \mathbf{E}_{\mathbf{x}\in U,\hat{\mathbf{x}}\leftarrow q(\hat{\mathbf{x}}|\mathbf{x})}\left[-\log p(y\,|\,\hat{\mathbf{x}})\right] + \cancel{\mathbf{E}_{\mathbf{x}\in U}\left[\log p(y\,|\,\mathbf{x})\right]}_{\text{ignore}}$$

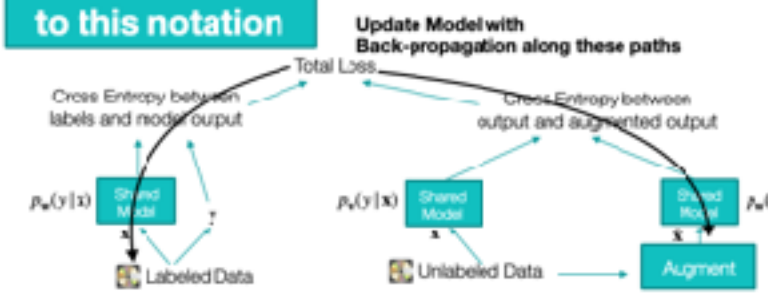
cross entropy of unsupervised labels after augmentation

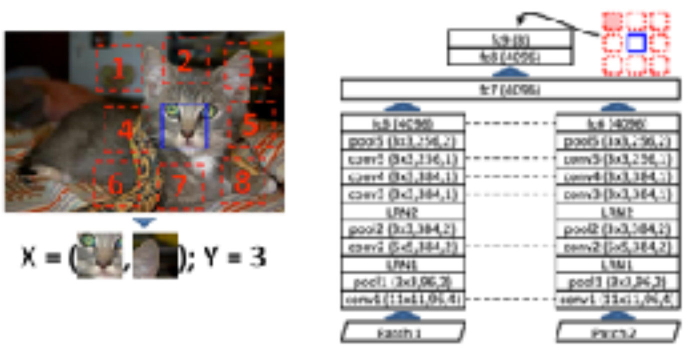
entropy of unsupervised labels ***constant***

Neural Network approximates $p(y|\mathbf{x})$ by $\mathbf{w}$
Use labeled data to minimize network

Sample new $\mathbf{x}$ from unlabeled pool with function q
function q is augmentation procedure
Minimize cross entropy of two models

Get accustomed to this notation





Unsupervised Visual Representation Learning by Context Prediction

Unsupervised Consistency Loss (review)

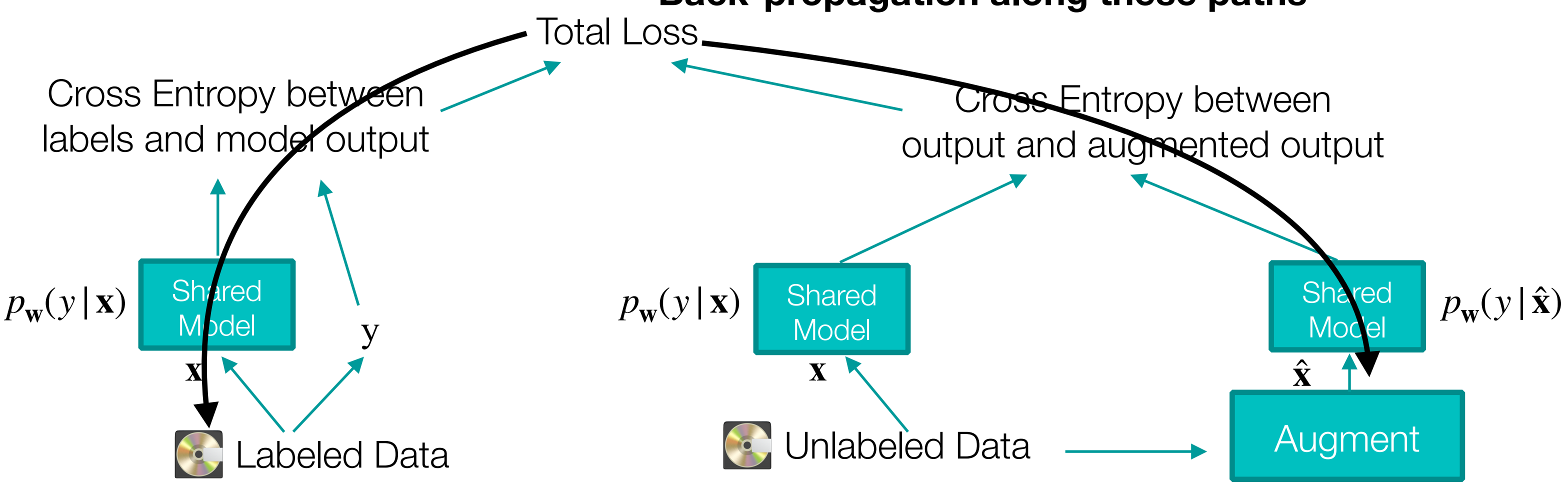
$$\min_{\mathbf{w}} \overbrace{\mathbf{E}_{\mathbf{x},y\in L}[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})]}^{\text{cross entropy}} + \lambda \overbrace{\mathcal{D}_{KL}\left(\underset{\textbf{no}\text{ back prop}}{p_{\mathbf{w}}(y\,|\,\mathbf{x})}\,||\,\underset{\textbf{yes}\text{ back prop}}{p_{\mathbf{w}}(y\,|\,\hat{\mathbf{x}})}\right)}^{\text{consistency in augmentation}}$$

Neural Network approximates $p(y|\mathbf{x})$ by $\mathbf{w}$
Use labeled data to minimize network

Sample new $\mathbf{x}$ from unlabeled pool with function q
function q is augmentation procedure
Minimize cross entropy of two models

Get accustomed to this notation

Update Model with
Back-propagation along these paths



Unsupervised Data Augmentation (UDA) for Consistency Training, Xie et al., NeurIps 2019

$$\min_{\mathbf{w}} \overbrace{\mathbf{E}_{\mathbf{x},y\in L}[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})]}^{\text{cross entropy}} + \lambda \underbrace{\mathcal{D}_{KL}\left(p_{\mathbf{w}}(y\,|\,\mathbf{x})\,||\,p_{\mathbf{w}}(y\,|\,\hat{\mathbf{x}})\right)}^{\text{consistency in augmentation}}$$

$$E[g] = \sum p(g)\cdot g$$ definition of expected value

$$E[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})] = -\sum p(y)\cdot \log p_{\mathbf{w}}(y\,|\,\mathbf{x})$$ insert -log probability, log likelihood

$$NLL(y, p_{\mathbf{w}}(y\,|\,\mathbf{x})) = -\sum_{c} p(y=c)\cdot \log p_{\mathbf{w}}(y=c\,|\,\mathbf{x})$$ negative log likelihood

$$CE(f,g) = -\sum f(x)\cdot \log g(x)$$ cross entropy of two functions

$$CE(y, p_{\mathbf{w}}(y\,|\,\mathbf{x})) = -\sum_{c} (y=c)\cdot \log p_{\mathbf{w}}(y=c\,|\,\mathbf{x})$$ if y=c is a probability, these are same equation

```
cce = tf.keras.losses.CategoricalCrossentropy()
cce(y_true, y_pred)
```

$$\min_{\mathbf{w}} \overbrace{\mathbf{E}_{\mathbf{x},y\in L}[-\log p_{\mathbf{w}}(y\,|\,\mathbf{x})]}^{\text{cross entropy}} + \lambda \overbrace{\mathscr{D}_{KL}\left(p_{\mathbf{w}}(y\,|\,\mathbf{x})\,||\,p_{\mathbf{w}}(y\,|\,\hat{\mathbf{x}})\right)}^{\text{consistency in augmentation}}$$

$$\mathscr{D}_{KL}(f\,||\,g) = -\sum f(x)\cdot\log\frac{g(x)}{f(x)}$$ definition of Kullback-Leibler (KL) Divergence

$$\mathscr{D}_{KL}(p_{\mathbf{w}}(y\,|\,\mathbf{x})\,||\,p_{\mathbf{w}}(y\,|\,\hat{\mathbf{x}}))$$

$$\mathscr{D}_{KL}(p(y\,|\,\mathbf{x})\,||\,p(y\,|\,\hat{\mathbf{x}})) = -\sum p(y\,|\,\mathbf{x})\cdot\log\frac{p(y\,|\,\hat{\mathbf{x}})}{p(y\,|\,\mathbf{x})} = -\sum p(y\,|\,\mathbf{x})\cdot\left(\log p(y\,|\,\hat{\mathbf{x}}) - \log p(y\,|\,\mathbf{x})\right)$$

$$= -\sum p(y\,|\,\mathbf{x})\cdot\log p(y\,|\,\hat{\mathbf{x}}) + \sum p(y\,|\,\mathbf{x})\cdot\log p(y\,|\,\mathbf{x})$$

$$p(y\,|\,\mathbf{x}) \approx p(y)$$ if **x** is a very large subset of the entire domain and $p_{\mathbf{w}}$ is a good *variational* approximation

So this is lower bound!

$$= \mathbf{E}_{\mathbf{x}\in U,\hat{\mathbf{x}}\leftarrow q(\hat{\mathbf{x}}|\mathbf{x})}\left[-\log p(y\,|\,\hat{\mathbf{x}})\right] + \mathbf{E}_{\mathbf{x}\in U}\cancel{\left[\log p(y\,|\,\mathbf{x})\right]}$$ ignore

cross entropy of unsupervised labels after augmentation

entropy of unsupervised labels
cannot calculate, always > 0

```
cce = tf.keras.losses.CategoricalCrossentropy()
cce(y_pred, y_pred_augmented)
```

Aside:

- We have just seen two motivations:

Neural Network approximates $p(y|x)$ by $\mathbf{w}$
Use labeled data to minimize network

Sample new $\mathbf{x}$ from unlabeled pool with functi
function q is augmentation procedure
Minimize cross entropy of two models

$$\mathcal{D}_{KL}(p_w(y|\mathbf{x})\,||\,p_w(y|\hat{\mathbf{x}}))$$

$$\mathcal{D}_{KL}(p(y|\mathbf{x})\,||\,p(y|\hat{\mathbf{x}})) = -\sum p(y|\mathbf{x})\cdot\log\frac{p(y|\hat{\mathbf{x}})}{p(y|\mathbf{x})} = -\sum p(y|\mathbf{x})\cdot(\log p(y|\hat{\mathbf{x}}) - \log p(y|\mathbf{x}))$$

$$= -\sum p(y|\mathbf{x})\cdot\log p(y|\hat{\mathbf{x}}) + \sum p(y|\mathbf{x})\cdot\log p(y|\mathbf{x})$$

$$p(y|\mathbf{x}) \approx p(y)$$ if x is a very large subset of the entire domain and p_w is a good variational approximation

So this is lower bound!

$$= \mathbf{E}_{\mathbf{x}\in U,\hat{\mathbf{x}}\leftarrow q(\hat{\mathbf{x}}|\mathbf{x})}\left[-\log p(y|\hat{\mathbf{x}})\right] + \cancel{\mathbf{E}_{\mathbf{x}\in U}\left[\log p(y|\mathbf{x})\right]}_{\text{ignore}}$$

cross entropy of unsupervised labels after augmentation

entropy of unsupervised labels **cannot calculate, always > 0**

intuition of final product

keep labels consistent, any measure would be okay

mathematics with heavy approximation

cross entropy is lower bound
for KL divergence, which is a nice measure

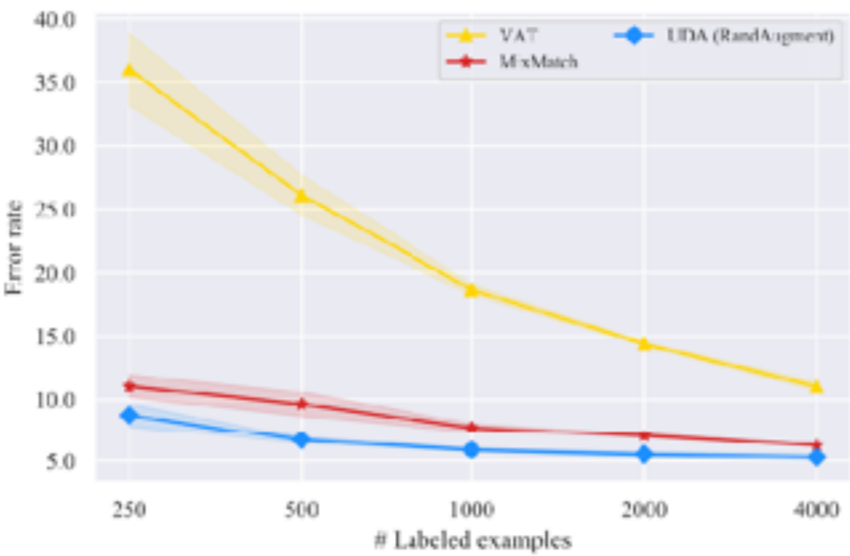
Unsupervised Consistency Loss

Augmentation (# Sup examples)	Sup (50k)	Semi-Sup (4k)
Crop & flip	5.36	16.17
Cutout	4.42	6.42
RandAugment	**4.23**	**5.29**

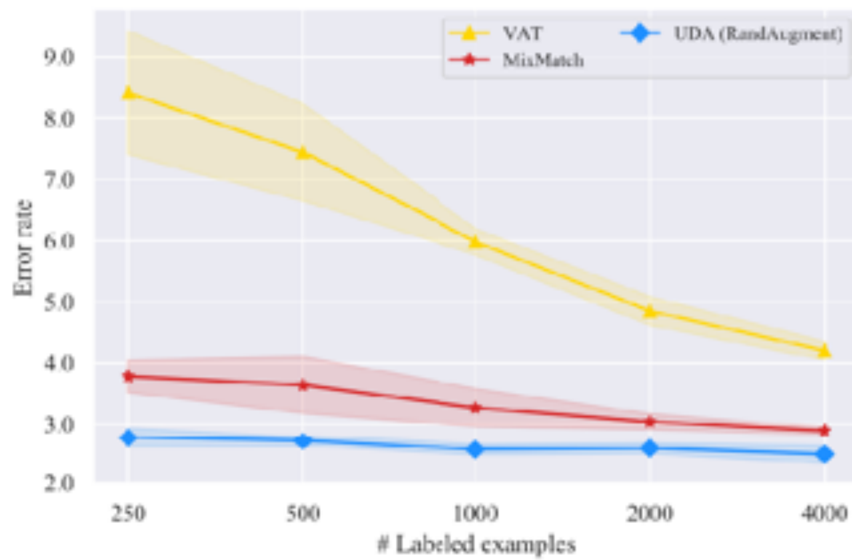
Table 1: Error rates on CIFAR-10.

Augmentation (# Sup examples)	Sup (650k)	Semi-sup (2.5k)
✗	38.36	50.80
Switchout	37.24	43.38
Back-translation	**36.71**	**41.35**

Table 2: Error rate on Yelp-5.



(a) CIFAR-10



(b) SVHN

Unsupervised Data Augmentation (UDA) for Consistency Training, Xie et al., NeurIps 2019

Unsupervised Consistency Loss

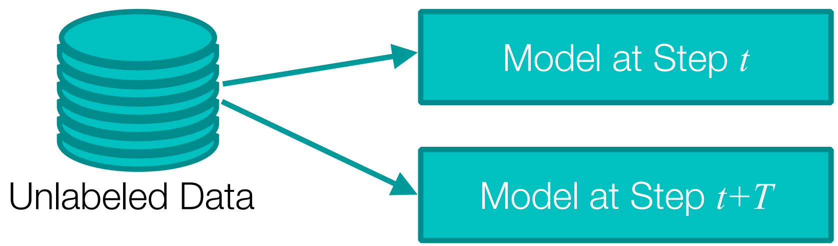
Method	Model	# Param	CIFAR-10 (4k)	SVHN (1k)
Π-Model (Laine & Aila, 2016)	Conv-Large	3.1M	12.36 ± 0.31	4.82 ± 0.17
Mean Teacher (Tarvainen & Valpola, 2017)	Conv-Large	3.1M	12.31 ± 0.28	3.95 ± 0.19
VAT + EntMin (Miyato et al., 2018)	Conv-Large	3.1M	10.55 ± 0.05	3.86 ± 0.11
SNTG (Luo et al., 2018)	Conv-Large	3.1M	10.93 ± 0.14	3.86 ± 0.27
VAdD (Park et al., 2018)	Conv-Large	3.1M	11.32 ± 0.11	4.16 ± 0.08
Fast-SWA (Athiwaratkun et al., 2018)	Conv-Large	3.1M	9.05	-
ICT (Verma et al., 2019)	Conv-Large	3.1M	7.29 ± 0.02	3.89 ± 0.04
Pseudo-Label (Lee, 2013)	WRN-28-2	1.5M	16.21 ± 0.11	7.62 ± 0.29
LGA + VAT (Jackson & Schulman, 2019)	WRN-28-2	1.5M	12.06 ± 0.19	6.58 ± 0.36
mixmixup (Hataya & Nakayama, 2019)	WRN-28-2	1.5M	10	-
ICT (Verma et al., 2019)	WRN-28-2	1.5M	7.66 ± 0.17	3.53 ± 0.07
MixMatch (Berthelot et al., 2019)	WRN-28-2	1.5M	6.24 ± 0.06	2.89 ± 0.06

Methods	SSL	10%	100%
ResNet-50 w. RandAugment	✗	55.09 / 77.26 58.84 / 80.56	77.28 / 93.73 78.43 / 94.37
UDA (RandAugment)	✓	**68.78 / 88.80**	**79.05 / 94.49**

Table 5: Top-1 / top-5 accuracy on ImageNet with 10% and 100% of the labeled set. We use image size 224 and 331 for the 10% and 100% experiments respectively.

Other Measures of Consistency: TOD

- Main idea: use unsupervised labels to prevent overfitting
- Temporal Output Discrepancy (TOD) (Huang et al., ICCV21)



$p_{\mathbf{w}_t}(y \mid \hat{\mathbf{x}})$

$p_{\mathbf{w}_{t+T}}(y \mid \hat{\mathbf{x}})$

Do not change too much between updates!

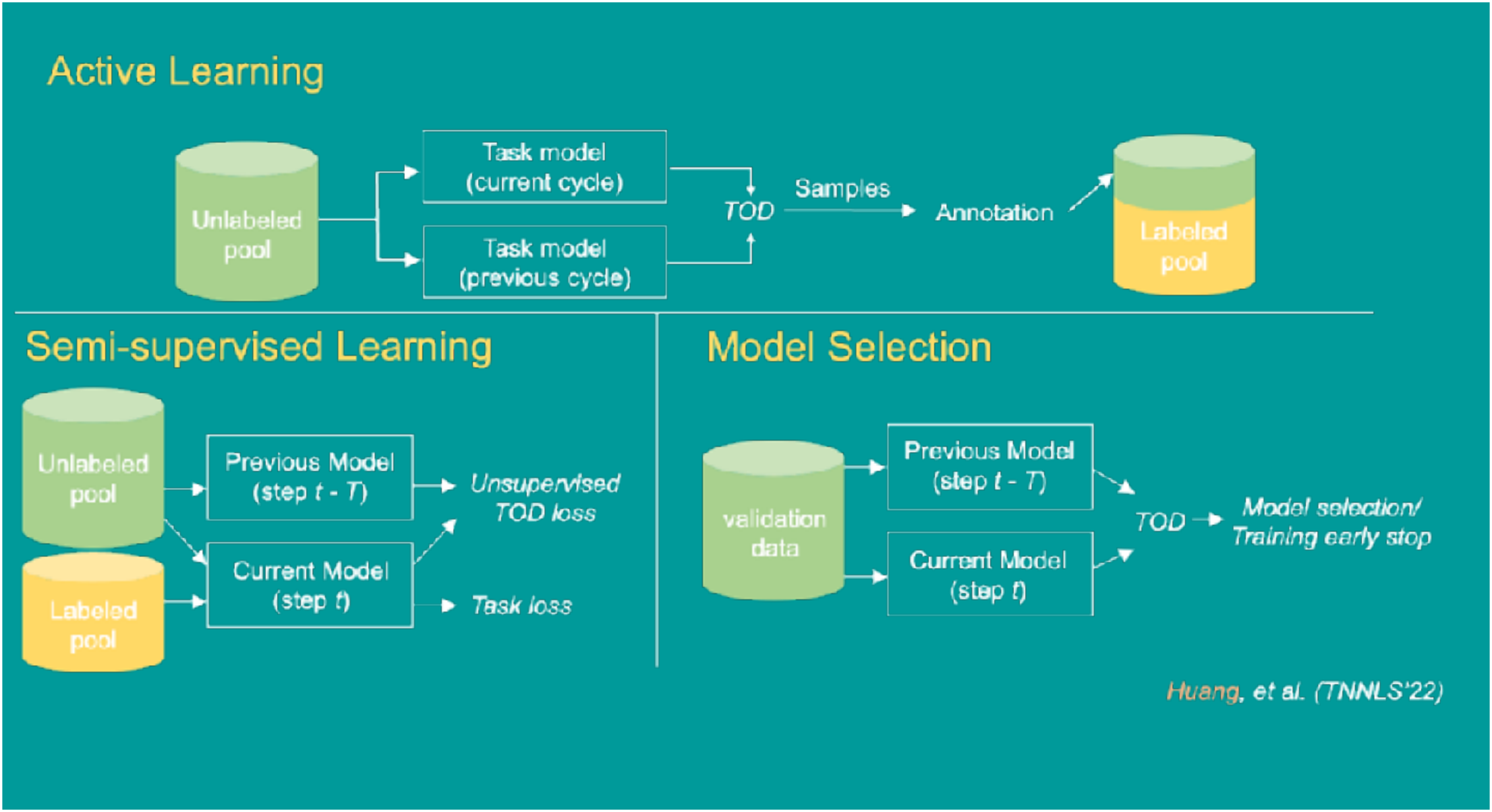
- Discrepancy

$$\left\| p_{\mathbf{w}_{t+T}}(y \mid \hat{\mathbf{x}}) - p_{\mathbf{w}_t}(y \mid \hat{\mathbf{x}}) \right\|$$

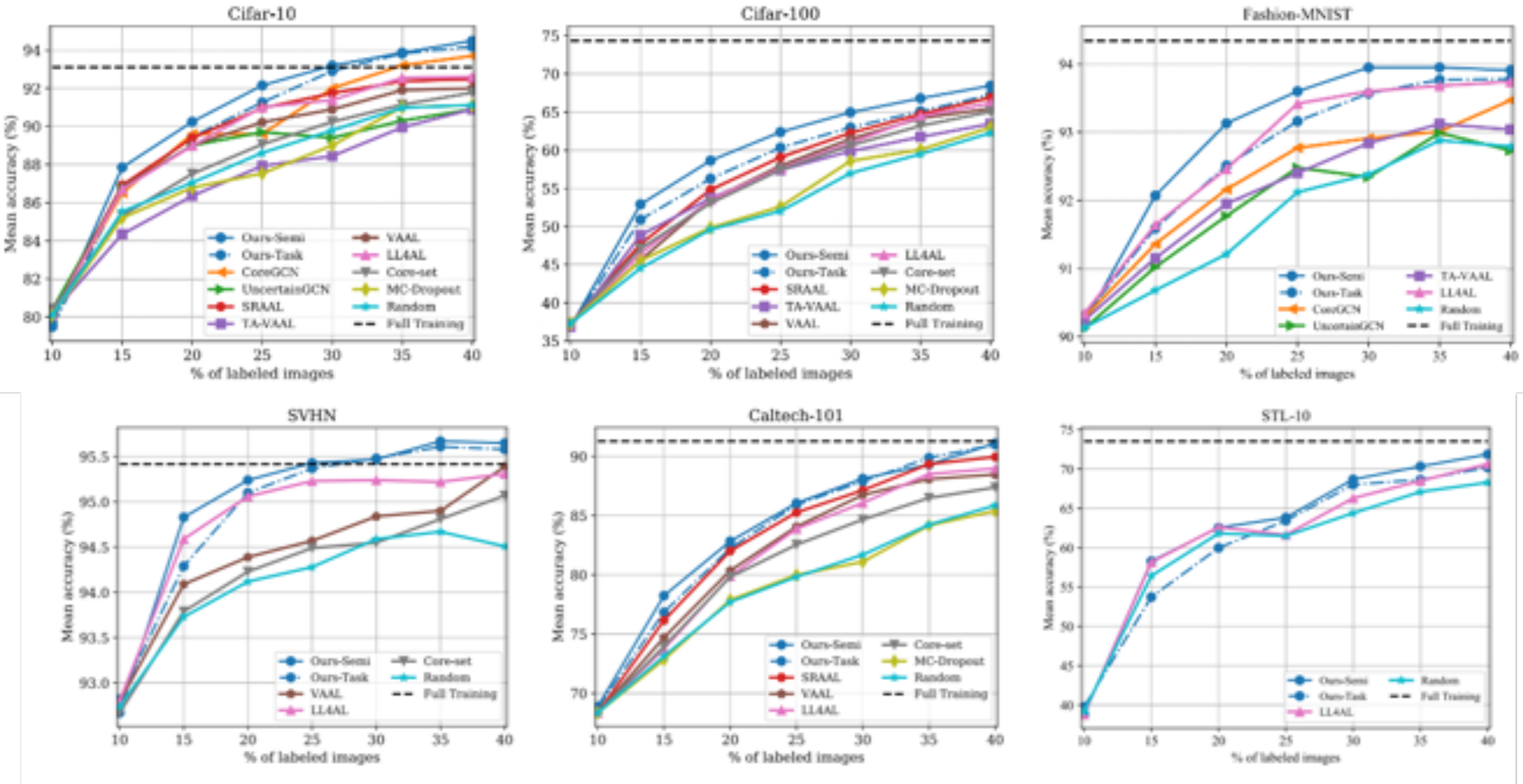
under certain conditions, this is a valid Wasserstein distance

$$\min_{\mathbf{w}} \underbrace{\mathbf{E}_{\mathbf{x},y\in L}[-\log p_{\mathbf{w}_{t+T}}(y \mid \mathbf{x})]}_{\text{cross entropy}} + \underbrace{\lambda \cdot \mathbf{E}_{\hat{\mathbf{x}}\in U}\left[\left\| p_{\mathbf{w}_{t+T}}(y \mid \hat{\mathbf{x}}) - p_{\mathbf{w}_t}(y \mid \hat{\mathbf{x}}) \right\| \right]}_{\text{discrepancy}}$$

Using Temporal Discrepancy



Active Learning with TOD



Paper Presentation: GPT-3

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI