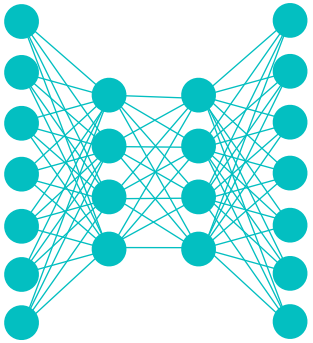
Lecture Notes for

Neural Networks and Machine Learning

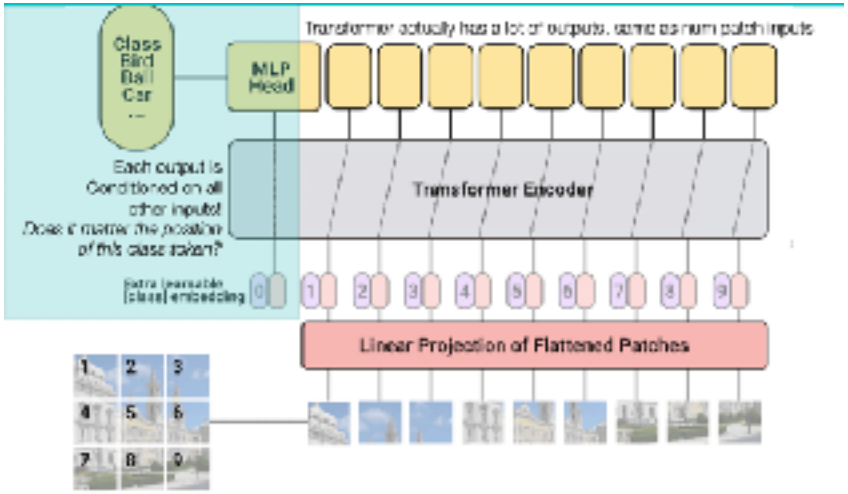
Self-Supervision, Multi-task, and Multi-Modal Learning



Logistics and Agenda

- Logistics
  - Grading update
- Agenda
  - Vision transformer and Town Hall
  - Student Paper Presentation
  - Consistency, Contrastive, and Triplet Loss
- Next Time
  - Multi-modal and Multi-Task

Last Time: Transformers

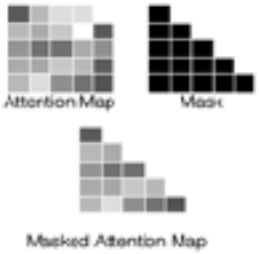


- Decoder only, text encoding happens in attention

$$\mathcal{L}_1(\mathcal{U}) = \sum_{i \in S} \log P(\underbrace{u_i}_{\text{curr}} \mid \overbrace{u_{i-k}, \ldots, u_{i-1}}^{\text{other words}}; \underbrace{\mathbf{W}}_{\text{params}})$$

Predict the next word from unlabeled dataset

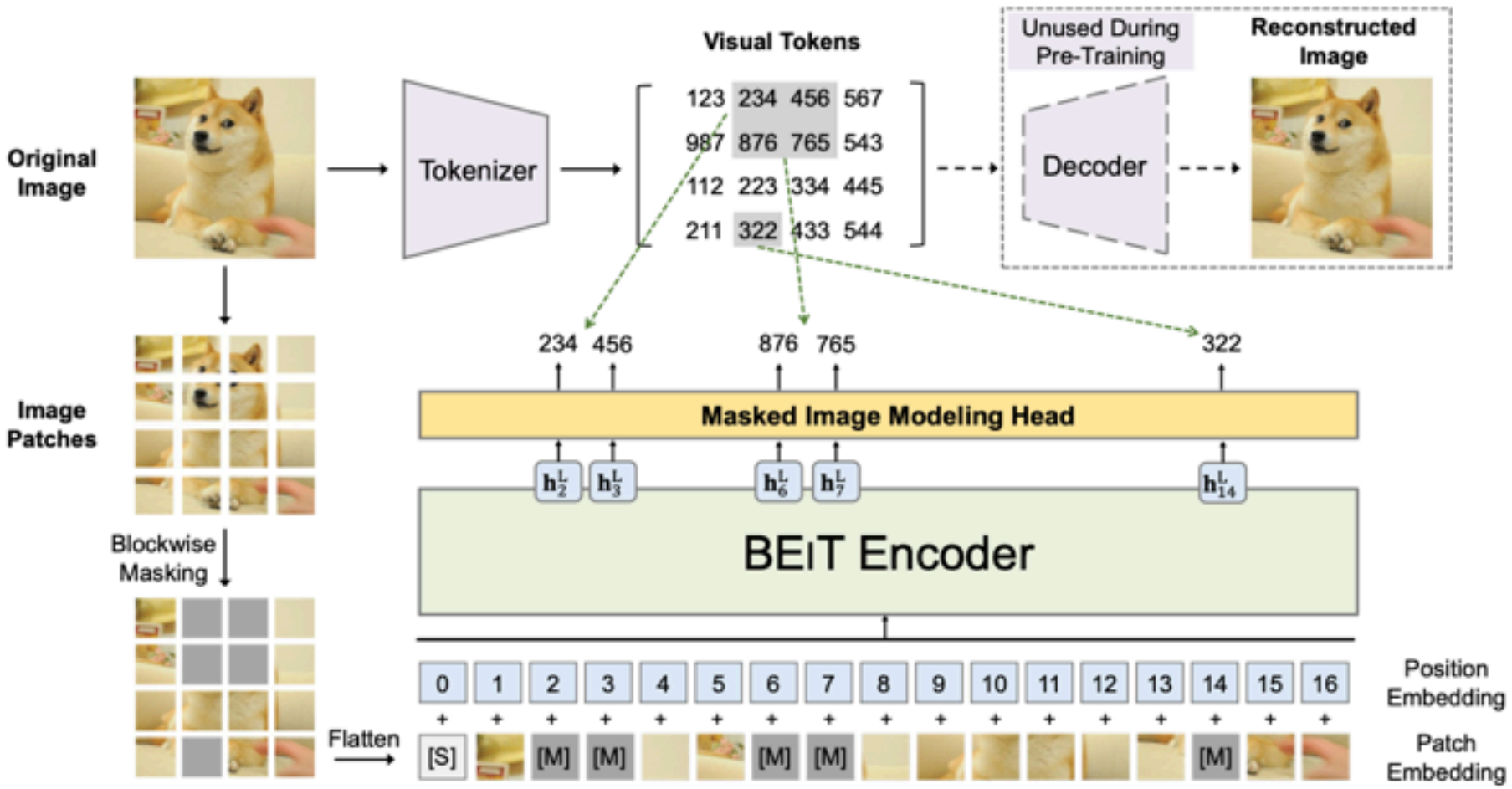
Also known as generative pre-training (GPT)



Vision Transformers Video

https://ai.googleblog.com/2020/12/transformers-for-image-recognition-at.html?m=1

Pre-training for ViT

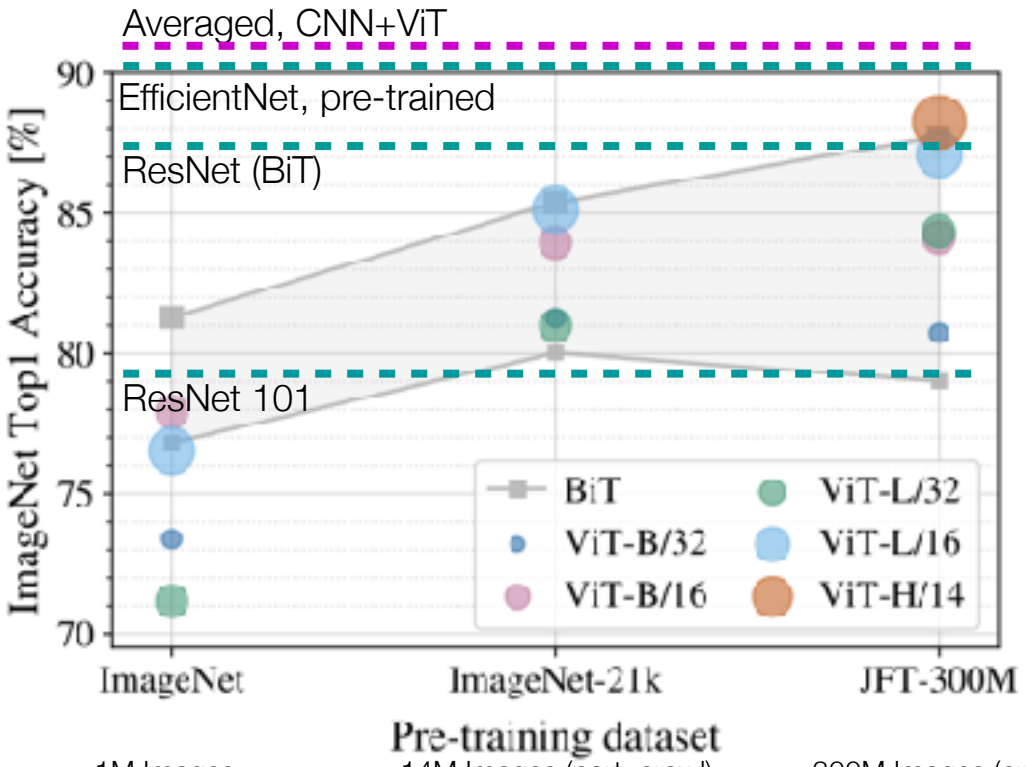


Downstream Fine Tuning: Image classification [CLS], semantic segmentation (patch output), and others…

https://www.fast.ai/2020/01/13/self_supervised/

Fine tuning: Do they work?

- Less than 14M images for pre-training? Do not use as a base model!
  - CNNs will be easier and very performant…

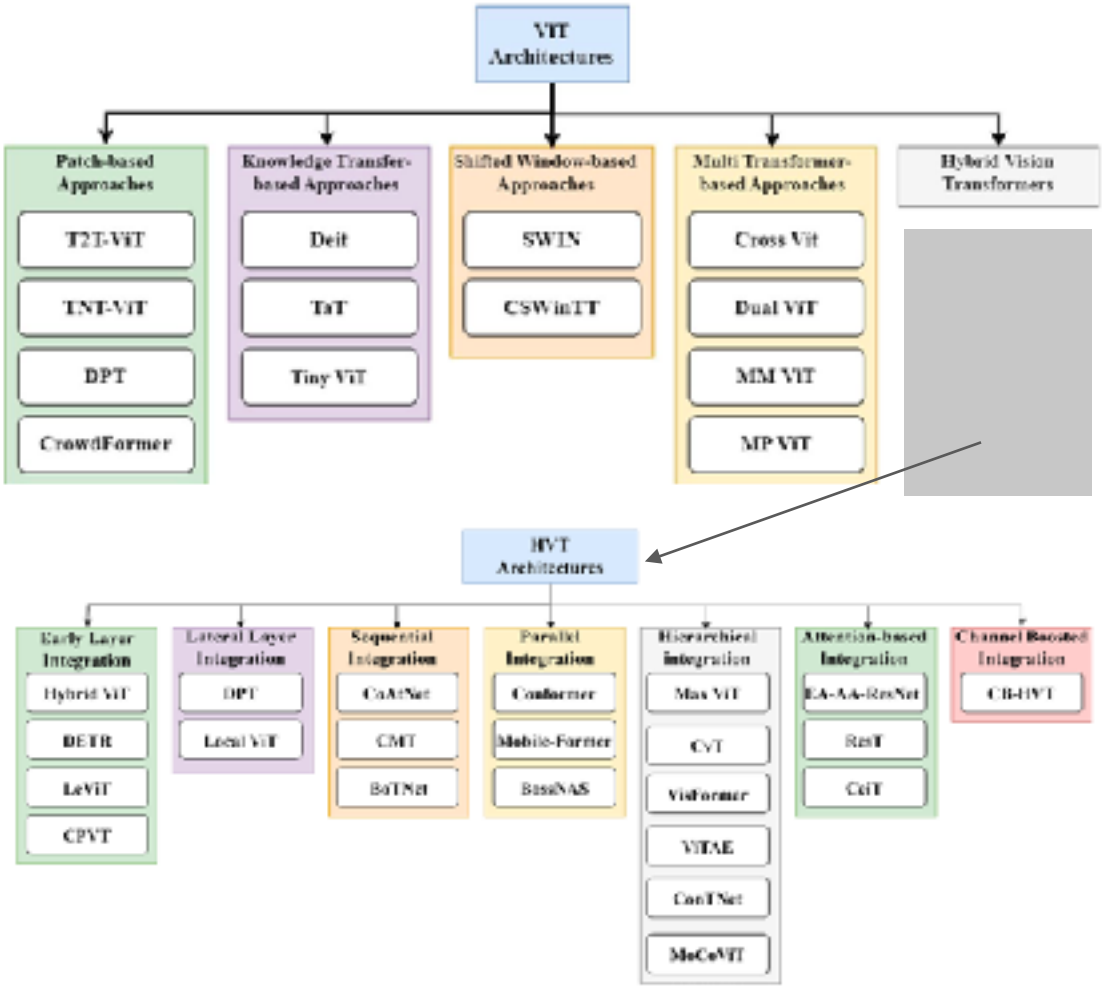


Transfer Learning From Huge ViT

	BeIT	ViT-H		Previous SOTA
ImageNet	89.5	88.55	←	88.5
ImageNet-ReaL		90.72	←	90.55
Cifar-10		99.50	←	99.37
Cifar-100	91.8	94.55	←	93.51
Pets		97.56	←	96.62
Flowers		99.68	←	99.63

Many Variants of the ViT

- ongoing area of research
- Input Patch Structure (overlap)
- Efficient Attention
- Cross Attention (mix of experts)
- Methods of SSL
- Hybrid Approaches
- Image/text generation
  - Generative images patching…



Kahn et al., A survey of the vision transformers and their CNN-transformer variants (2023), *Springer Nature* https://link.springer.com/article/10.1007/s10462-023-10595-0

Paper Presentation

Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction

Keyu Tian[1,2], Yi Jiang[2,*], Zehuan Yuan[2,*], Bingyue Peng[2], Liwei Wang[1,3,*]
[1]Center for Data Science, Peking University [2]Bytedance Inc.
[3]State Key Lab of General Artificial Intelligence, School of Intelligence Science and Technology, Peking University
keyutian@stu.pku.edu.cn, jiangyi.enjoy@bytedance.com, yuanzehuan@bytedance.com, bingyue.peng@bytedance.com, wanglw@pku.edu.cn

Try and explore our online demo at: https://var.vision
Codes and models: https://github.com/FoundationVision/VAR

Figure 1: **Generated samples from Visual AutoRegressive (VAR) transformers trained on ImageNet.** We show 512×512 samples (top), 256×256 samples (middle), and zero-shot image editing results (bottom).

Transformer and Transfer Learning Town Hall

Hugging Face Text transformers: https://huggingface.co/transformers/v3.3.1/pretrained_models.html
Hugging Face ViT: https://huggingface.co/docs/transformers/model_doc/vit
Keras text Transformers: https://keras.io/guides/keras_nlp/transformer_pretraining/
Keras ViT: https://github.com/faustomorales/vit-keras

Consistency Loss

I'm from Canada, but live in the States now.

It took me a while to get used to writing boolean variables with an "Is" prefix, instead of the "Eh" suffix that Canadians use when programming.

For example:

```
MyObj.IsVisible

MyObj.VisibleEh
```


Unsupervised Consistency Loss

$$\min_{\mathbf{w}} \mathbf{E}_{\mathbf{x},y \in L}[-\log p_{\mathbf{w}}(y \mid \mathbf{x})] + \lambda \quad \mathscr{D}_{KL}\left(p_{\mathbf{w}}(y \mid \mathbf{x}) \,||\, p_{\mathbf{w}}(y \mid \hat{\mathbf{x}})\right)$$

no back prop **yes** back prop

Neural Network approximates $p(y|\mathbf{x})$ by $\mathbf{w}$

Sometimes shown as $p(y \mid \mathbf{x}; \mathbf{W})$

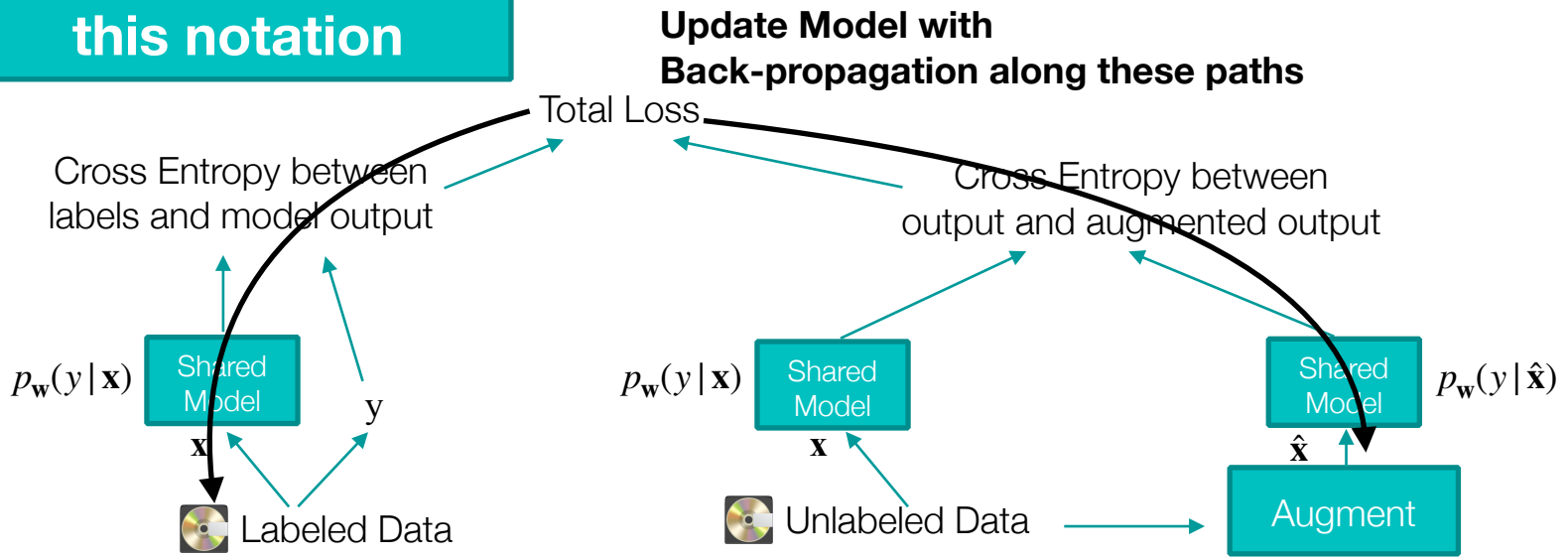
Use labeled data to minimize network

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

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

$$E[g] = \sum p(g) \cdot g$$ 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, discrete classes

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

$$CE(y, p_{\mathbf{w}}(y\,|\,\mathbf{x})) = -\sum_c p(y=c) \cdot \log p_{\mathbf{w}}(y=c\,|\,\mathbf{x})$$

if y=c is a probability, these are same
NLL is Cross Entropy

```
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

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

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

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

cross entropy of augmented and not augmented model

constant

global entropy of model

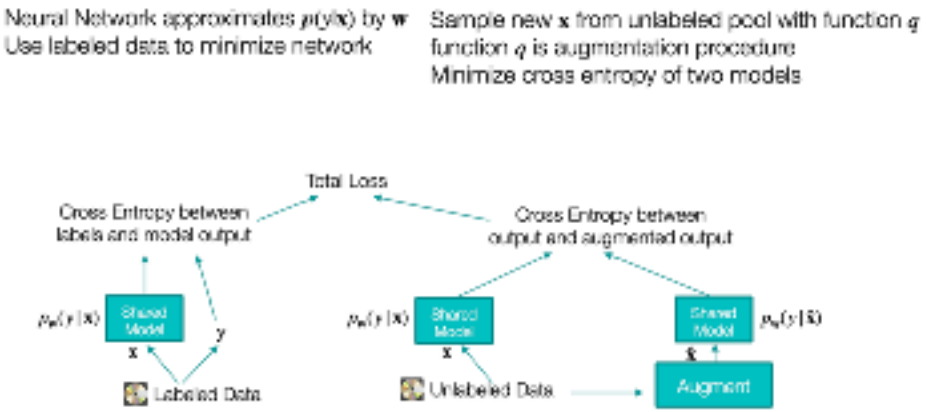
constant as $p(y\,|\,\mathbf{x})$ has no uncertainty

$$CE\left(p_{\mathbf{w}}(y\,|\,\mathbf{x}), p_{\mathbf{w}}(y\,|\,\hat{\mathbf{x}})\right) = -\sum_{c} p_{\mathbf{w}}(y=c\,|\,\mathbf{x})\cdot\log p_{\mathbf{w}}(y=c\,|\,\hat{\mathbf{x}})$$

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


Aside:

- We have just seen two motivations:



intuition of final model

keep labels consistent, any measure would be okay

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

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

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

cross entropy of augmented and not augmented model

constant

global entropy of model output

constant as $p(y\,|\,\mathbf{x})$ has no uncertainty

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

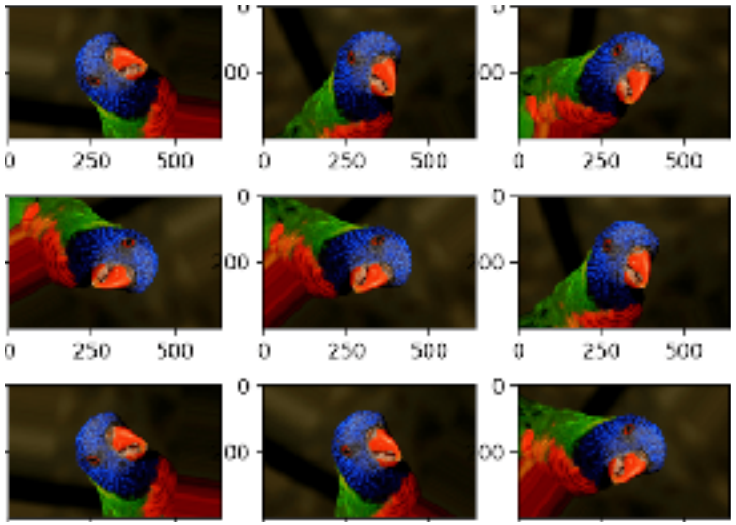
mathematics with strict assumptions

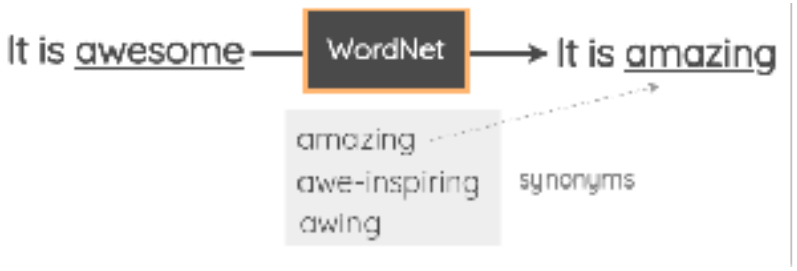
not necessarily intuitive, but provides guarantees

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

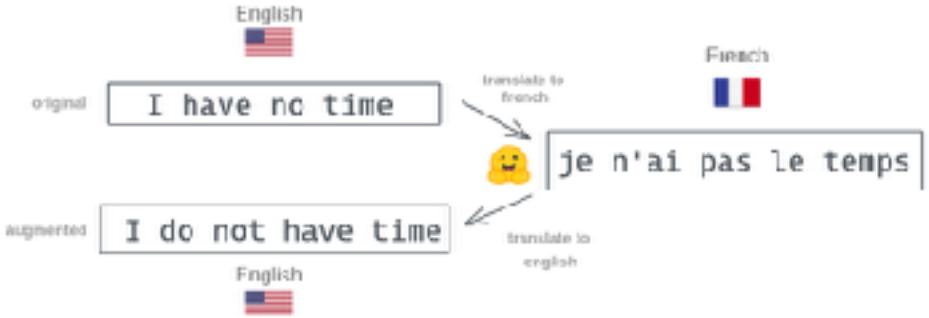
Augmentation with Consistency Loss

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





Synonym Replacement



Back Translation

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

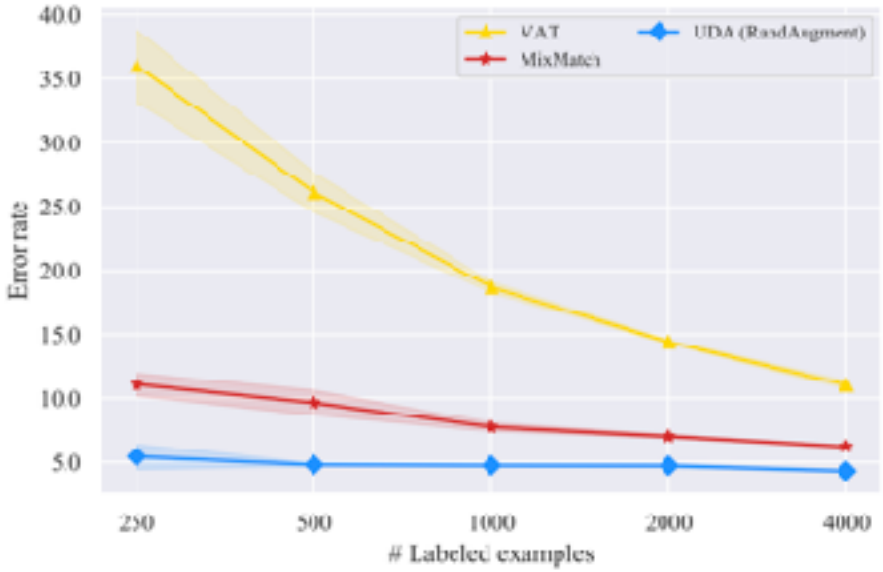
Unsupervised Consistency Loss

Augmentation (# Sup examples)	Sup (50k)	Semi-Sup (4k)
Crop & flip	5.36	10.94
Cutout	4.42	5.43
RandAugment	**4.23**	**4.32**

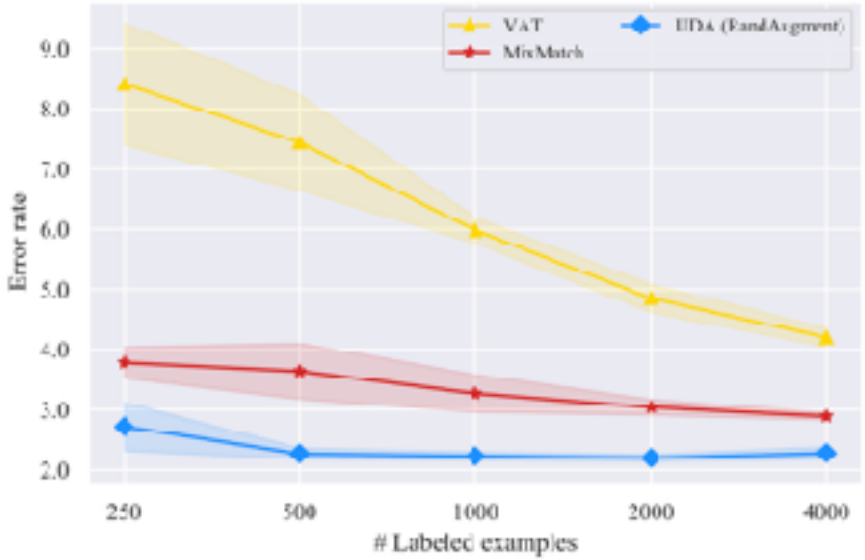
Table 1: Error rates on CIFAR-10.

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., Neurlps 2019

Unsupervised Consistency Loss

Semi-supervised setting							
Initialization	**UDA**	IMDb (20)	Yelp-2 (20)	Yelp-5 (2.5k)	Amazon-2 (20)	Amazon-5 (2.5k)	DBpedia (140)
Random	✗	43.27	40.25	50.80	45.39	55.70	41.14
	✓	25.23	8.33	41.35	16.16	44.19	7.24
$\text{BERT}_{\text{BASE}}$	✗	18.40	13.60	41.00	26.75	44.09	2.58
	✓	5.45	2.61	33.80	3.96	38.40	1.33
$\text{BERT}_{\text{LARGE}}$	✗	11.72	10.55	38.90	15.54	42.30	1.68
	✓	4.78	2.50	33.54	3.93	37.80	1.09
$\text{BERT}_{\text{FINETUNE}}$	✗	6.50	2.94	32.39	12.17	37.32	-
	✓	**4.20**	**2.05**	**32.08**	**3.50**	**37.12**	-

Contrastive Loss

Jürgen Schmidhuber
@SchmidhuberAI

DeepSeek [1] uses elements of the 2015 reinforcement learning prompt engineer [2] and its 2018 refinement [3] which collapses the RL machine and world model of [2] into a single net through the neural net distillation procedure of 1991 [4]: a distilled chain of thought system.

REFERENCES (easy to find on the web):

[1] #DeepSeekR1 (2025): Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv 2501.12948

[2] J. Schmidhuber (JS, 2015). On Learning to Think: Algorithmic Information Theory for Novel Combinations of Reinforcement Learning Controllers and Recurrent Neural World Models. arXiv 1210.0118. Sec. 5.3 describes the reinforcement learning (RL) prompt engineer which learns to actively and iteratively query its model for abstract reasoning and planning and decision making.

[3] JS (2018). One Big Net For Everything. arXiv 1802.08864. See also US11853886B2. This paper collapses the reinforcement learner and the world model of [2] (e.g., a foundation model) into a single network, using the neural network distillation procedure of 1991 [4]. Essentially what's now called an RL "Chain of Thought" system, where subsequent improvements are continually distilled into a single net. See also [5].

Dealing with Data Sparsity

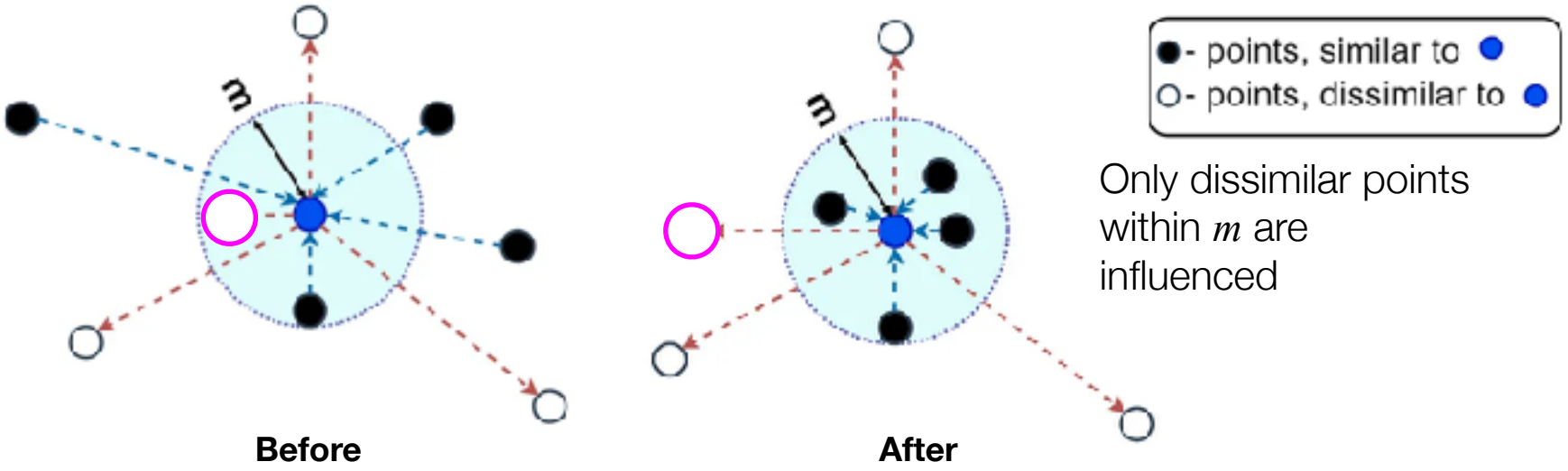
- How can we augment the number of gradient updates we have, without needing more data collection?
  - Can we leverage the existing examples in the dataset?
  - Can the process provide an exponential number of new examples for back prop?
  - Can it make latent space better behaved for classification?
- **Contrastive Loss:**
  - Use a metric to measure similarity of samples within latent space (e.g., cosine distance, euclidean, etc.)
  - Randomly sample from two or more classes
  - Push same classes together (within a threshold)
  - Push different classes apart (within a threshold)

Contrastive loss

Latent representations of $\mathbf{x}$, from model f

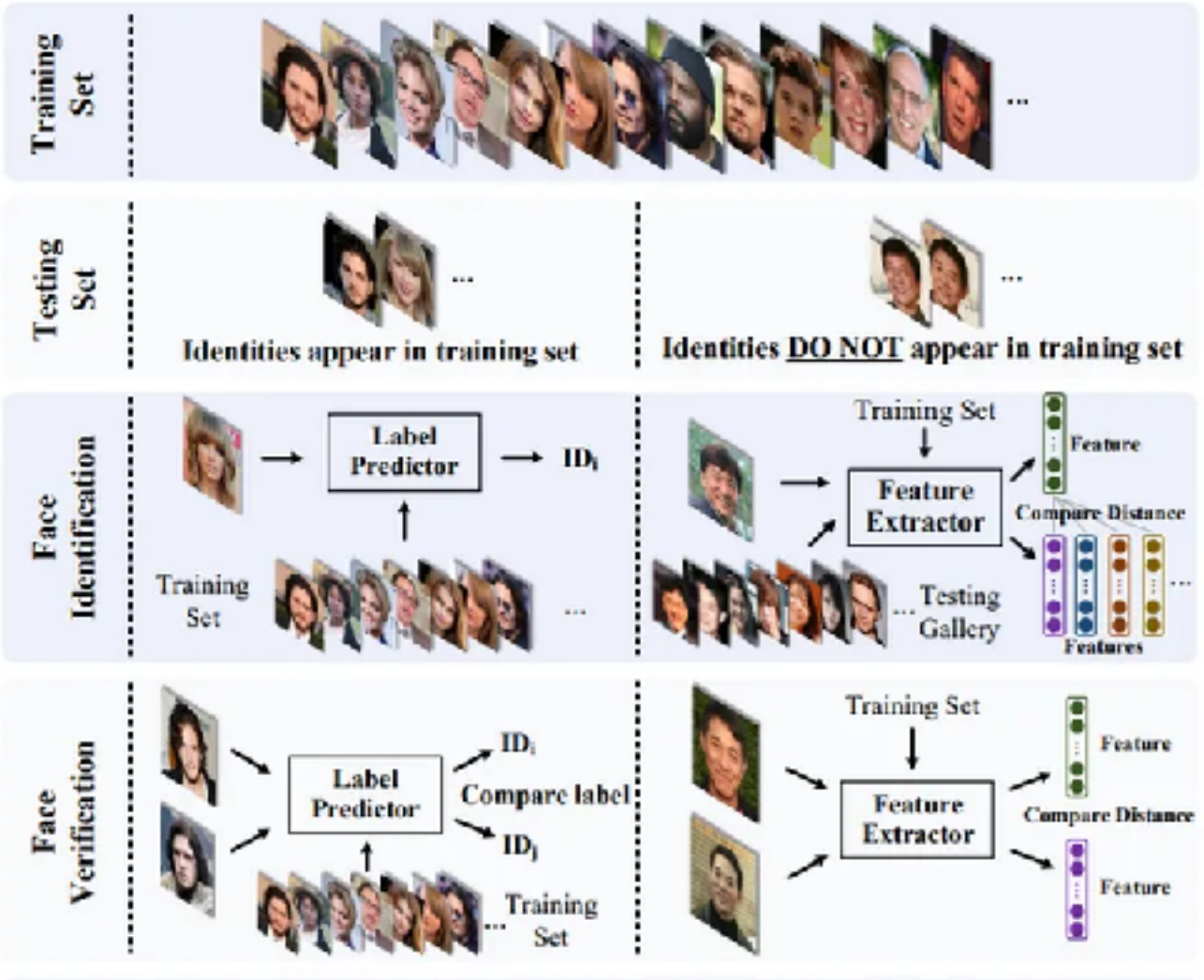
$$D(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \|f_\theta(\mathbf{x}^{(i)}) - f_\theta(\mathbf{x}^{(j)})\|_2$$

$$\mathscr{L}_c = \underbrace{\sum_{i,j \in S} D(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})}_{\text{similar}} + \underbrace{\sum_{i,j \in \hat{S}} \max\left(0, m - D(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})\right)}_{\text{not similar}}$$



https://medium.com/@maksym.bekuzarov/losses-explained-contrastive-loss-f8f57fe32246

Contrastive loss in Face Detect/Identify



Contrastive Loss great for authenticating without re-training

However, the pairs chosen for positive and negative samples tends to be very sensitive to sampling for goos performance.