Vision AI

2022 arXiv Trends

2022-05

hutom

no.	Paper Title	Research group
1	Align before Fuse: Vision and Language Representation Learning with Momentum Distillation	Salesforce Research
2	Solving ImageNet: a Unified Scheme for Training any Backbone to Top Results	DAMO Academy, Alibaba Group
3	Self-Supervised Learning of Object Parts for Semantic Segmentation	1. Technical University of Munich 2. QUVA Lab University of Amsterdam

no.	Paper Title	Research group
4	FedILC: Weighted Geometric Mean and Invariant Gradient Covariance for Federated Learning on Non-IID Data	1. McGill University 2. Mila - Quebec AI Institute
5	MultiMAE: Multi-modal Multi-task Masked Autoencoders	Swiss Federal Institute of Technology Lausanne (EPFL)
6	DoubleMatch: Improving Semi-Supervised Learning with Self-Supervision	1. Saab AB, 2. Chalmers University of Technology

Align before Fuse: Vision and Language Representation Learning with Momentum Distillation

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh D. Gotmare
Shafiq Joty, Caiming Xiong, Steven C.H. Hoi
Salesforce Research
{junnan.li,rselvaraju,akhilesh.gotmare,sjoty,shoi}@salesforce.com

- Transformer 기반 Visual - Text multimodal learning
- 기존 transfoemr 기반의 multimodal encoder는 model이 visual, text tokens을 함께 학습하도록 한다.

  ⇒ visual word tokens이 unligned 되어있어 multimodal encoder 가 image-text interaction을 학습하는 것이 어렵다"
- 본 논문에서는 image-text representation을 합치기 전 align하기 위한 contrastive loss를 소개. ⇒ 그래서 본 논문의 제목도 Align before Fuse.
  - BBOX annotation, high-reslution image가 필요없다는 장점 (기존 VLP모델 region-based image features를 추출하는데, 사전학습된 object detector module을 사용하곤 함)
- 또한 Momentum distillation 으로 large noisy dataset (web dataset) 에서 발생할수 있는 문제를 해결하고 함 (image - text pair 가 다른수도 있고, 상당히 비슷할수도 있음)
  - momentm model에 의해 생성된 pseudo-targets으로 부터 self-training 하는 것
  - ALBEF에 대해 이론적인 분석 제공 ⇒ 실제 label distrubution 과 teacher의 pseudo label distribution간이 Mutual information을 높게 학습하다는 것과 동치임을 설명
  - 결국 Momentum distill을 통해 image-text pair를 학습시킬때 pair에 대한 다양한 view를 생성하는 것으로 해석가능. (data augmetnation ?)
- 2가지 noisy large dataset, 2가지 in-domain dataset 에서 5가지 task에 대한 실험
- SOTA 달성 및 github 공개
- 비교적 관련 VLP task의 논문성능 대비 여전히 좋은성능을 내고있음.

Align before Fuse: Vision and Language Representation Learning with Momentum Distillation

Vision and Language Pre-training (VLP)

Vision-and-Pre training (VLP) aims to learn multimodal representations from large-scale image-text paris that can improve downstream Vision-and-Language (V+L) tasks performance

- Image-text Retrieval
- Visual Alignmnet
- Visual Questions Anserwering (VQA)
- Natural language for Visual Reasoning (NLVR)
- Visual Grounding

Context

Most existing VLP Methods rely on pre-training object detectors to extract region based image features and employ a multimodal encoder to fuse the image features with word tokens,

This VLP framework suffers from several key limitations

1. Challenging for the multimodal encoder to model image and text iterations
2. Object detector is both annotation-expensive and compute-expensive
3. Existing pre-training objectives such as MLM may overfit to the noisy image-text pairs and degrade the model's generalization performance

Align before Fuse: Vision and Language Representation Learning with Momentum Distillation

ALBEF Model Architecture

- **image encoder - 12 layer ViT-B/16**
- **text encoder - first 6 layer of BERT_base**
- **multimodal encoder - last 6 layer of the BERT_base**

The full pre-training objective of ALBEF is:

$$\mathcal{L} = \mathcal{L}_{\text{itc}} + \mathcal{L}_{\text{mlm}} + \mathcal{L}_{\text{itm}} \quad (5)$$

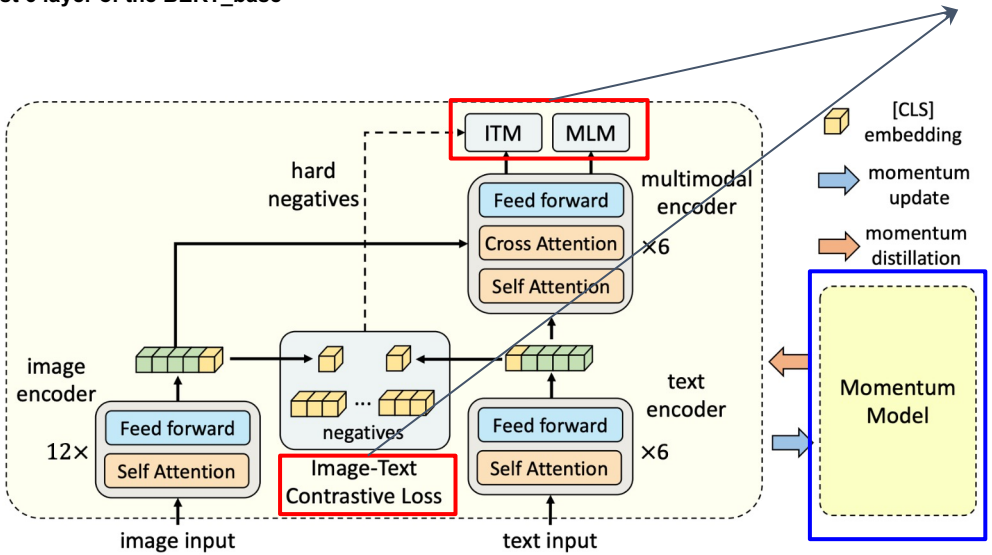


Figure 1: **Illustration of ALBEF.** It consists of an image encoder, a text encoder, and a multimodal encoder. We propose an image-text contrastive loss to align the unimodal representations of an image-text pair before fusion. An image-text matching loss (using in-batch hard negatives mined through contrastive similarity) and a masked-language-modeling loss are applied to learn multimodal interactions between image and text. In order to improve learning with noisy data, we generate pseudo-targets using the momentum model (a moving-average version of the base model) as additional supervision during training.

Align before Fuse: Vision and Language Representation Learning with Momentum Distillation

Image-Text Contrastive Learning to learn unimodal representations

- **Input Image is encoded into a sequence of embeddings**
- **Text is encoded into sequence of embeddings**
- **Similarity function (g 는 linear transformation)**
- **maintain two queues to store the most recent M image-text representations**

  from the momentum unimodal encoders:
- **Define**

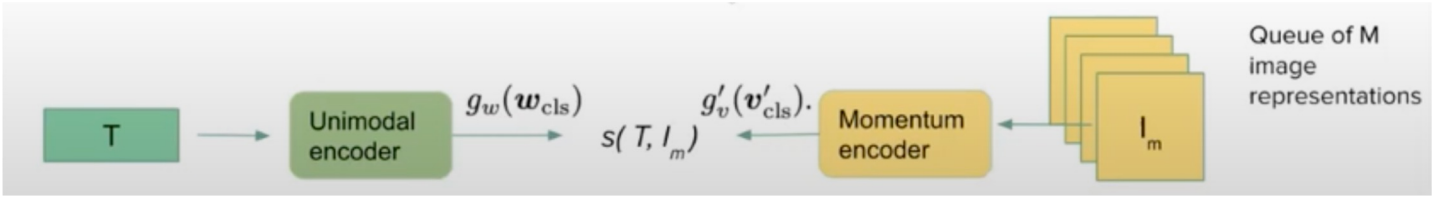
$$\{\boldsymbol{v}_{\text{cls}}, \boldsymbol{v}_1, ..., \boldsymbol{v}_N\}$$

$$\{\boldsymbol{w}_{\text{cls}}, \boldsymbol{w}_1, ..., \boldsymbol{w}_N\}$$

$$s = g_v(\boldsymbol{v}_{\text{cls}})^\top g_w(\boldsymbol{w}_{\text{cls}}),$$

$$g'_v(\boldsymbol{v}'_{\text{cls}}) \text{ and } g'_w(\boldsymbol{w}'_{\text{cls}}).$$

$$s(I,T) = g_v(\boldsymbol{v}_{\text{cls}})^\top g'_w(\boldsymbol{w}'_{\text{cls}}) \text{ and } s(T,I) = g_w(\boldsymbol{w}_{\text{cls}})^\top g'_v(\boldsymbol{v}'_{\text{cls}}).$$



Align before Fuse: Vision and Language Representation Learning with Momentum Distillation

ITC Loss

- **For each image and text, we calculate the softmax-normalized image-to-text and text-to-image similarity as:**

$$p_m^{\mathrm{i2t}}(I) = \frac{\exp(s(I, T_m)/\tau)}{\sum_{m=1}^{M} \exp(s(I, T_m)/\tau)}, \quad p_m^{\mathrm{t2i}}(T) = \frac{\exp(s(T, I_m)/\tau)}{\sum_{m=1}^{M} \exp(s(T, I_m)/\tau)} \tag{1}$$

of 1. The image-text contrastive loss is defined as the cross-entropy H between $\boldsymbol{p}$ and $\boldsymbol{y}$:

$$\mathcal{L}_{\mathrm{itc}} = \frac{1}{2}\mathbb{E}_{(I,T)\sim D}\left[\mathrm{H}(\boldsymbol{y}^{\mathrm{i2t}}(I), \boldsymbol{p}^{\mathrm{i2t}}(I)) + \mathrm{H}(\boldsymbol{y}^{\mathrm{t2i}}(T), \boldsymbol{p}^{\mathrm{t2i}}(T))\right] \tag{2}$$

Ground truth one-hot encoding

MLM Loss

- **Utilizes both image and contextual text to predict the masked workds**
- **Following BERT, we randomly mask out the input text tokens with 15% probability and replace them with a special tokan [MASK]**
- **MLM minimizes the cross-entropy loss:**

$$\mathcal{L}_{\mathrm{mlm}} = \mathbb{E}_{(I,\hat{T})\sim D}\mathrm{H}(\boldsymbol{y}^{\mathrm{msk}}, \boldsymbol{p}^{\mathrm{msk}}(I, \hat{T})) \tag{3}$$

One-hot GT vocabulary distribution

predicted prob for a masked token

Masked text

Align before Fuse: Vision and Language Representation Learning with Momentum Distillation

ITM Loss

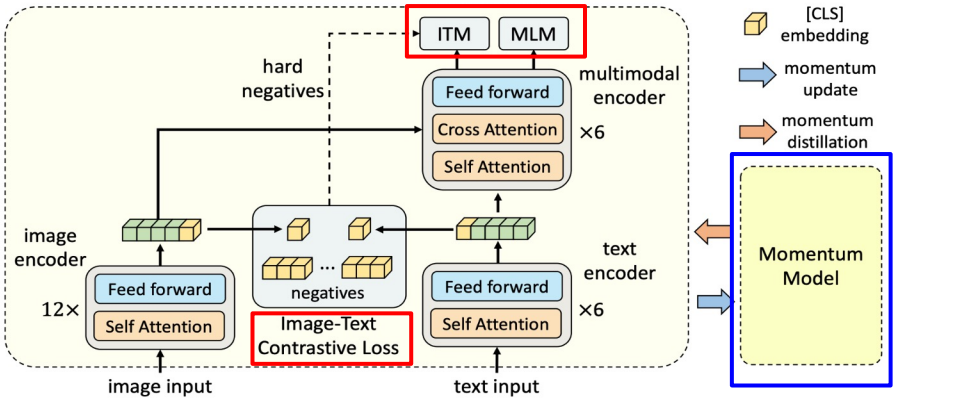
- **Predicts whether a pair of image and text is positive(Matched) or negative(UNmatched)**
- **Use the multimodal encoder's output embedding of the [CLS] token as the joint representation of the image-text pair**
- **Append a fully-connected layer followed by softmax to predict a two-class probability distribution p^itm**
- **ITM loss is:**

$$\mathcal{L}_{\text{itm}} = \mathbb{E}_{(I,T)\sim D}\mathrm{H}(\boldsymbol{y}^{\text{itm}}, \boldsymbol{p}^{\text{itm}}(I,T)) \tag{4}$$

2-d one-hot vector for ground-truth label

The full pre-training objective of ALBEF is:

$$\mathcal{L} = \mathcal{L}_{\text{itc}} + \mathcal{L}_{\text{mlm}} + \mathcal{L}_{\text{itm}} \tag{5}$$



Align before Fuse: Vision and Language Representation Learning with Momentum Distillation

Momentum Distillation

- a self-training method which learns from pseudo-targets produced by a momentum model,

  which is a continuosly-evolving teacher which consists of exponential-moving-average version of the unimodal and multimodal encoders.

$$\theta_{mo} \leftarrow m*\theta_{mo} + (1-m)*\theta$$

"polar bear in the [MASK]"

GT: wild
Top-5 pseudo-targets:
1. zoo
2. pool
3. water
4. pond
5. wild

"a man [MASK] along a road in front of nature in summer"

GT: standing
Top-5 pseudo-targets:
1. walks
2. walking
3. runs
4. running
5. goes

"a [MASK] waterfall in the deep woods"

GT: remote
Top-5 pseudo-targets:
1. small
2. beautiful
3. little
4. secret
5. secluded

GT: breakdown of the car on the road
Top-5 pseudo-targets:
1. young woman get out of the car near the road
2. a woman inspects her damaged car under a tree
3. a woman looking into a car after locking her keys inside
4. young woman with a broken car calling for help
5. breakdown of the car on the road

GT: the harbor a small village
Top-5 pseudo-targets:
1. the harbour with boats and houses
2. replica of the sailing ship in the harbour
3. ships in the harbor of the town
4. the harbor a small village
5. boats lined up alongside the geographical feature category in the village

Figure 2: Examples of the pseudo-targets for MLM (1st row) and ITC (2nd row). The pseudo-targets can capture visual concepts that are not described by the ground-truth text (*e.g.* "beautiful waterfall", "young woman").

Momentum Distillation for ITC

$$\mathcal{L}_{\text{itc}}^{\text{mod}} = (1-\alpha)\mathcal{L}_{\text{itc}} + \frac{\alpha}{2}\mathbb{E}_{(I,T)\sim D}\left[\text{KL}(\boldsymbol{q}^{\text{i2t}}(I) \parallel \boldsymbol{p}^{\text{i2t}}(I)) + \text{KL}(\boldsymbol{q}^{\text{t2i}}(T) \parallel \boldsymbol{p}^{\text{t2i}}(T))\right] \quad (6)$$

from momentum unimodal encoder

from unimodal encoder

Momentum Distillation for MLM

$$\mathcal{L}_{\text{mlm}}^{\text{mod}} = (1-\alpha)\mathcal{L}_{\text{mlm}} + \alpha\mathbb{E}_{(I,\hat{T})\sim D}\text{KL}(\boldsymbol{q}^{\text{msk}}(I,\hat{T}) \parallel \boldsymbol{p}^{\text{msk}}(I,\hat{T})) \quad (7)$$

D Additional Examples of Pseudo-targets

GT: this is real fast food
Top-5 pseudo-targets:
1. transform shredded chicken into decadent sandwiches
2. recipes up your game and make something other than just tacos this week
3. with rice chicken and vegetables this proves salad can be way more than a bed of lettuce
4. pork is roasted on site for these tacos
5. we used lean turkey instead of beef so we could stuff these babies with cheese

kitten playing with a [MASK]

GT: dog
Top-5 pseudo-targets:
1. toy
2. blanket
3. ball
4. mouse
5. bone

[MASK] clouds in the sky

GT: red
Top-5 pseudo-targets:
1. pink
2. colorful
3. sunset
4. red
5. dramatic

Align before Fuse: Vision and Language Representation Learning with Momentum Distillation

Pre training Data

- Two web datsets: Conceptual Captions, SBU Captions
- Two in-domain datasets: COCO and Visual Genome
- From the datasets above: total number of unique images is 4.0M. number of image-text pair is 5.1M
- To show the method is scalable, They also include the much noiser Conceptual 12M datasets. Increasing the total number of images to 14.1M

fine-tune ALBEF to downstream V+L task

Vision-and-Pre training (VLP) aims to learn multimodal representations from large-scale image-text paris that can improve downstream Vision-and-Language (V+L) tasks performance

- Image-text Retrieval
- Visual Alignmnet
- Visual Questions Anserwering (VQA)
- Natural language for Visual Reasoning (NLVR)
- Visual Grounding

- **Evaluation Proposed Methods**

Image-Text Retrieval contains two subtasks: image-to-text retrieval (TR) and text-to-image retrieval (IR). We evaluate ALBEF on the Flickr30K [49] and COCO benchmarks, and fine-tune the pre-trained model using the training samples from each dataset. For zero-shot retrieval on Flickr30K, we evaluate with the model fine-tuned on COCO. During fine-tuning, we jointly optimize the ITC loss (Equation 2) and the ITM loss (Equation 4). ITC learns an image-text scoring function based on similarity of unimodal features, whereas ITM models the fine-grained interaction between image and text to predict a matching score. Since the downstream datasets contain multiple texts for each image, we change the ground-truth label of ITC to consider multiple positives in the queue, where each positive has a ground-truth probability of $1/\#\text{positives}$. During inference, we first compute the feature similarity score s_{itc} for all image-text pairs. Then we take the top-k candidates and calculate their ITM score s_{itm} for ranking. Because k can be set to be very small, our inference speed is much faster than methods that require computing the ITM score for all image-text pairs [2, 3, 8].

- **Evaluation Proposed Methods**

#Pre-train Images	Training tasks	TR (flickr test)	IR (flickr test)	SNLI-VE (test)	NLVR^2 (test-P)	VQA (test-dev)
4M	MLM + ITM	93.96	88.55	77.06	77.51	71.40
	ITC + MLM + ITM	96.55	91.69	79.15	79.88	73.29
	ITC + MLM + ITM_{hard}	97.01	92.16	79.77	80.35	73.81
	ITC_{MoD} + MLM + ITM_{hard}	97.33	92.43	79.99	80.34	74.06
	Full (ITC_{MoD} + MLM_{MoD} + ITM_{hard})	97.47	92.58	80.12	80.44	74.42
	ALBEF (Full + $\text{MoD}_{\text{Downstream}}$)	97.83	92.65	80.30	80.50	74.54
14M	ALBEF	98.70	94.07	80.91	83.14	75.84

Table 1: Evaluation of the proposed methods on four downstream V+L tasks. For text-retrieval (TR) and image-retrieval (IR), we report the average of R@1, R@5 and R@10. ITC: image-text contrastive learning. MLM: masked language modeling. ITM_{hard}: image-text matching with contrastive hard negative mining. MoD: momentum distillation. $\text{MoD}_{\text{Downstream}}$: momentum distillation on downstream tasks.

Align before Fuse: Vision and Language Representation Learning with Momentum Distillation

- **Evaluation Proposed Methods (TR, IR)**

Method	# Pre-train Images	Flickr30K (1K test set)						MSCOCO (5K test set)					
		TR			IR			TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
UNITER	4M	87.3	98.0	99.2	75.6	94.1	96.8	65.7	88.6	93.8	52.9	79.9	88.0
VILLA	4M	87.9	97.5	98.8	76.3	94.2	96.8	-	-	-	-	-	-
OSCAR	4M	-	-	-	-	-	-	70.0	91.1	95.5	54.0	80.8	88.5
ALIGN	1.2B	95.3	99.8	100.0	84.9	97.4	98.6	77.0	93.5	96.9	59.9	83.3	89.8
ALBEF	4M	94.3	99.4	99.8	82.8	96.7	98.4	73.1	91.4	96.0	56.8	81.5	89.2
ALBEF	14M	**95.9**	**99.8**	**100.0**	**85.6**	**97.5**	**98.9**	**77.6**	**94.3**	**97.2**	**60.7**	**84.3**	**90.5**

Table 2: Fine-tuned image-text retrieval results on Flickr30K and COCO datasets.

Method	# Pre-train Images	Flickr30K (1K test set)					
		TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10
UNITER [2]	4M	83.6	95.7	97.7	68.7	89.2	93.9
CLIP [6]	400M	88.0	98.7	99.4	68.7	90.6	95.2
ALIGN [7]	1.2B	88.6	98.7	99.7	75.7	93.8	96.8
ALBEF	4M	90.5	98.8	99.7	76.8	93.7	96.7
ALBEF	14M	**94.1**	**99.5**	**99.7**	**82.8**	**96.3**	**98.1**

Table 3: Zero-shot image-text retrieval results on Flickr30K.

- **Evaluation Proposed Methods (VQA NLVR SNLI-VE)**

Method	VQA		NLVR^2		SNLI-VE	
	test-dev	test-std	dev	test-P	val	test
VisualBERT [13]	70.80	71.00	67.40	67.00	-	-
VL-BERT [10]	71.16	-	-	-	-	-
LXMERT [1]	72.42	72.54	74.90	74.50	-	-
12-in-1 [12]	73.15	-	-	78.87	-	76.95
UNITER [2]	72.70	72.91	77.18	77.85	78.59	78.28
VL-BART/T5 [54]	-	71.3	-	73.6	-	-
ViLT [21]	70.94	-	75.24	76.21	-	-
OSCAR [3]	73.16	73.44	78.07	78.36	-	-
VILLA [8]	73.59	73.67	78.39	79.30	79.47	79.03
ALBEF (4M)	74.54	74.70	80.24	80.50	80.14	80.30
ALBEF (14M)	**75.84**	**76.04**	**82.55**	**83.14**	**80.80**	**80.91**

Table 4: Comparison with state-of-the-art methods on downstream vision-language tasks.

Align before Fuse: Vision and Language Representation Learning with Momentum Distillation

- **Weakly-supervised Visual Grounding**

Method	Val	TestA	TestB
ARN [57]	32.78	34.35	32.13
CCL [58]	34.29	36.91	33.56
$\text{ALBEF}_{\text{itc}}$	51.58	60.09	40.19
$\text{ALBEF}_{\text{itm}}$	**58.46**	**65.89**	**46.25**

Table 5: Weakly-supervised visual grounding on RefCOCO+ [56] dataset.

Figure 4: Grad-CAM visualization on the cross-attention maps in the 3rd layer of the multimodal encoder.

Figure 5: Grad-CAM visualizations on the cross-attention maps of the multimodal encoder for the VQA model.

Figure 6: Grad-CAM visualizations on the cross-attention maps corresponding to individual words.

Solving ImageNet: a Unified Scheme for Training any Backbone to Top Results

Tal Ridnik, Hussam Lawen, Emanuel Ben-Baruch, Asaf Noy
DAMO Academy, Alibaba Group
tal.ridnik@alibaba-inc.com

https://arxiv.org/pdf/2204.03475.pdf

어떠한 아키텍처던 관계 없이 **하나의 unified 된 training scheme 을 가지고**, CNN, Transformer, NLP 등에 좋은 성능을 보이고자 함.

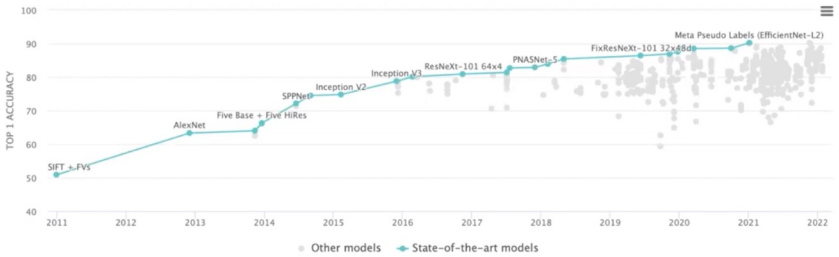
Solving ImageNet: a Unified Scheme for Training any Backbone to Top Results

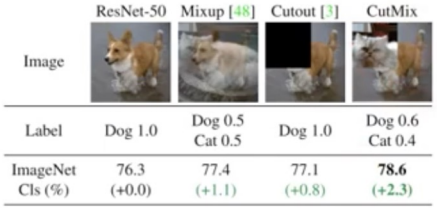
Previous works

- **Regularizations**
  - Stronger augmentations: AutoAugment, RandAugment (다양한 augmentation 방법)
  - **Image-based regularizations Cutout, Cutmix and Mixup**
    - **이미지에서 적용할 수 있는 새로운 augmentation**
  - Architecture regularizations like drop-path, drop-block
    - 아키텍처 특정한 부분은 weight 업데이트 안함.
  - Label-smoothing
  - Progressive image resizing during training
  - **Different train-test resolutions**
    - **train, test resolutions 을 다르게 보는 것.**

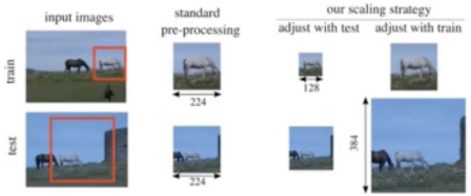
- **Training configuration**
  - More training epochs
  - Dedicated optimizer for large batch size (LAMB Optimizer), Scaling learning rate with batch size
  - Exponential-moving average (EMA) of model weights
  - Improved weights initializations
  - Decoupled weight decay (AdamW)

- **Architecture**





Yun, S. et al., CutMix: Regularization strategy to train strong classifiers with localizable features. ICCV 2019



Fixing the train-test resolution discrepancy. NeurIPS 2019

Solving ImageNet: a Unified Scheme for Training any Backbone to Top Results

Motivation - Architecture 와 관계없이 잘 작동하는 training scheme 제안 필요

- **Architecture 마다 맞춤형 training scheme 이 적용됨**
  - ResNet 계열 (TResNet, SEResNet, ResNet-D, …)
    - 일반적으로 다양한 training scheme 에 잘 작동함
    - (Ross Wightman et al., 2021) 에서 제안한 방법이 ResNet 계열을 학습시키는데 standard 가 되었다고 함.
  - Mobile-oriented models
    - Depth-wise convolutions 에 많이 의존
    - RMSProp optimizer, waterfall learning rate scheduling and EMA
  - Transformer- based, NLP-only models
    - Inductive bias 가 없어 훈련하기 어려움 -> longer training (1000 epochs), strong cutmix-mixup and drop-path regularizations, large weight-decay and repeated augmentations

- **어떤 한 모델에 대한 맞춤형 training scheme 은 다른 모델에 적용하면 성능이 낮아짐**
  - ResNet50을 위한 training scheme 을 EfficientNet v2 model 에 적용했을 때, 맞춤형 training scheme 을 적용했을 때 보다 3.3%의 성능 하락을 보임 (Mingxing Tan et al., PMLR, 2021)

Solving ImageNet: a Unified Scheme for Training any Backbone to Top Results

Unified training scheme for ImageNet without any hyper-parameter tuning or tailor-made tricks per model

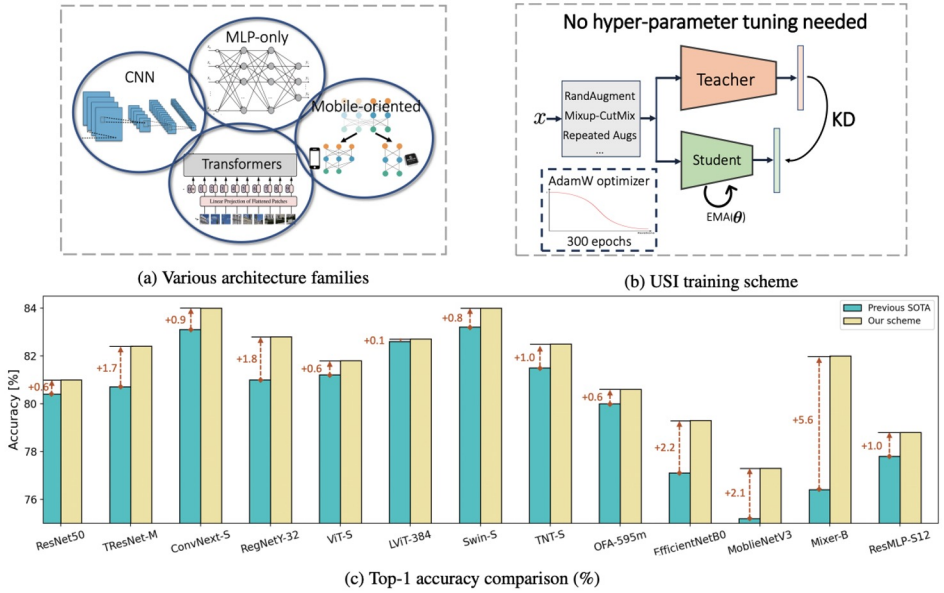


Figure 1: **Our unified training scheme for ImageNet, USI**. With USI, we can train any backbone to top results on ImageNet, without any hyper-parameter tuning or adjustments per architecture.

Solving ImageNet: a Unified Scheme for Training any Backbone to Top Results

Methods

- **Knowledge Distillation (KD) 적용**
  - ResNet50 의 image classification, DeIT, NAS 등 소개하는 previous work 에서 KD 가 성능 향상에 중요한 역할을 함.
  - **But, KD is not a common practice for ImageNet training.**
  - **그럼에도, KD 를 사용해야 하는 이유.**
    - **(b) Wing, airplane : Teacher network 는 image 가 완전히 mutually-exclusive 하지 않은 case 를 보완한다.**
    - **(c) Hem 55.5% 사람이 봐도 애매한데, 그 애매함을 teacher 의 classification 결과가 반영한다.**
    - **(d) Task 로 보면 틀린 답이지만, English setter 가 이미지에서 main object 라고 볼 수 있다.**

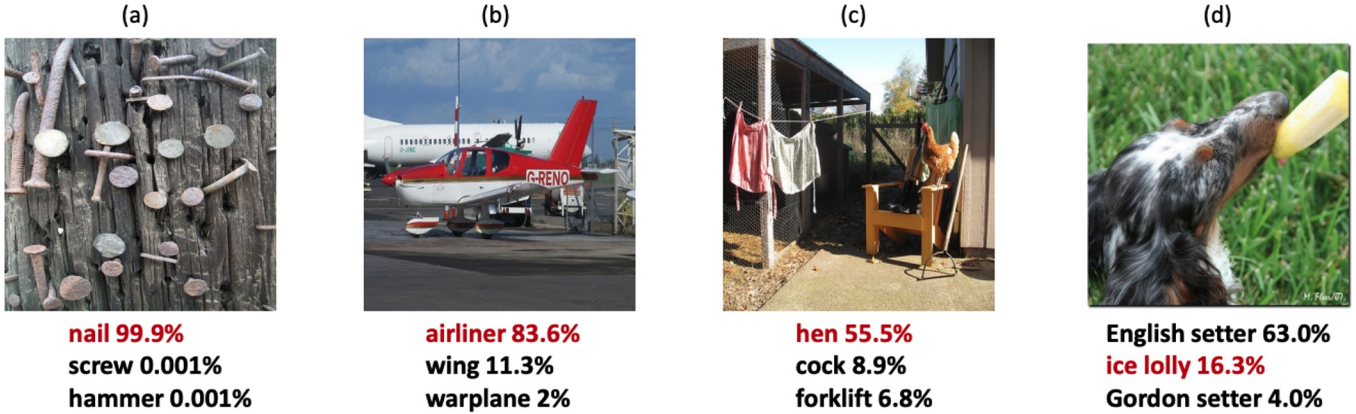
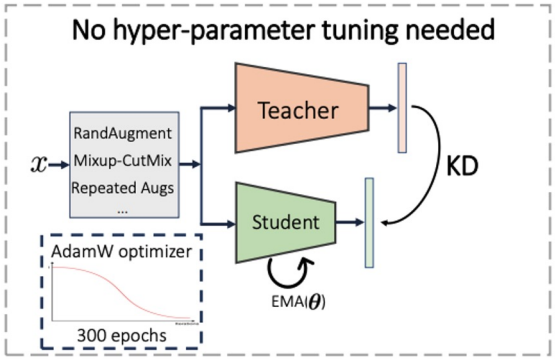


Figure 2: **Examples for teacher predictions**. ImageNet ground-truth labels are highlighted in red. Unlike the ground-truth, the teacher predictions account for similarities and correlations between classes, objects' saliency, pictures with several objects, and more. The teacher predictions would also better represent the content of images under strong augmentations.

Solving ImageNet: a Unified Scheme for Training any Backbone to Top Results

Methods

- **Knowledge Distillation (KD) 적용**
  - **그럼에도, KD 를 사용해야 하는 이유.**
    - (b) Wing, airplane : Teacher network 는 image 가 완전히 mutually-exclusive 하지 않은 case 를 보완한다.
    - (c) Hem 55.5% 사람이 봐도 애매한데, 그 애매함을 teacher 의 classification 결과가 반영한다.
    - (d) Task 로 보면 틀린 답이지만, English setter 가 이미지에서 main object 라고 볼 수 있다.
  - **즉, Teacher label 에 GT label 보다 더 많은 정보가 포함되어 있음 (class 간의 유사성과 상관관계)**
  - **Label error 을 보정할 수 있음. Label smoothing 을 따로 할 필요가 없음.**
  - **Lead to a more effective and robust optimization process, compared to training with hard-labels only.**
    - **hard label 만을 사용해서 process 하는 것보다, 좀 더 optimizational 한 결과 값을 포함할 수 있다.**
  - **KD 를 활용해 아키텍처가 달라고 동일한 training configuration 을 적용할 수 있도록 제안.**



(b) USI training scheme

Procedure	Value
Train resolution	224
Test resolution	224
Epochs	300
Optimizer	AdamW
Weight decay	2e-2
Learning rate	2e-3
LR decay	One-cycle policy
Mixup alpha	0.8
Cutmix alpha	1.0
Augmentations	Rand-augment (7/0.5)
Test crop ratio	0.95
Repeated Augs	3
Base loss	Cross entropy
KD loss	KL-divergence
KD temperature	1
α_{kd}	5
Teacher	TResNet-L
Batch size	512 to 3456

Table 1: **USI training configuration**. With USI, exactly the same training recipe is applied to any backbone, and no hyper-parameter tuning is needed.

Solving ImageNet: a Unified Scheme for Training any Backbone to Top Results

Experiments

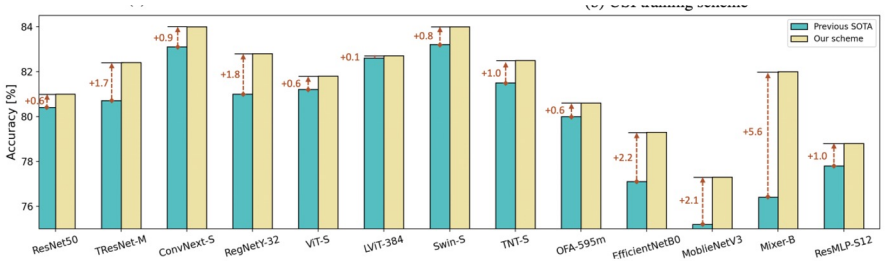
- USI 의 robustness 검증
- 제안한 training scheme (KD), loss function 이 잘 작동함을 확인
- 추가로 성능 향상할 수 있는 방법 제안
- Application: Speed-Accuracy comparison

USI 의 robustness 검증

- 모델들에 똑같이 USI 를 적용했을 때, tailor-made schemes 을 적용한 각 논문의 Top1 accuracy 보다 좋은 성능을 보임

Model Type	Model Name	USI Top1 Acc. [%]	Comparable Training Scheme			
			Top1 Acc. [%]	Epochs	KD	Additional Details
CNN	ResNet50	81.0	80.4 [42]	600	no	ResNet-strike-back, A1 config
			80.2 [47]	300	yes	Relabel-based KD
	TResNet-M	82.4	80.7 [30]	300	no	
	ConvNext-S	84.0	83.1 [25]	300	no	
	RegNetY-32	82.8	81.0 [28]	100	no	
Transformer	ViT-S	81.8	79.8 [8]	300	no	Original paper scheme
			81.2 [38]	1000	yes	DeiT scheme
	LeViT-384	82.7	82.6 [11]	1000	yes	DeiT scheme
	Swin-S	84.0	83.2 [24]	300	no	
	TNT-S	82.5	81.5 [12]	300	no	
Mobile-Oriented CNN	OFA-595m	80.6	80.0 [3]	255	yes	KD from super network
	EfficientnetB0	79.3	77.1 [34]	not stated	no	
	MobileNetV3	77.3	75.2 [17]	not stated	no	
MLP-Only	Mixer-B	82.0	76.4 [36]	255	no	
	ResMLP-S12	78.8	77.8 [37]	400	yes	

Table 2: Comparison of our proposed scheme, USI, to previous state-of-the-art results. Train and test resolution - 224.



(c) Top-1 accuracy comparison (%)

Solving ImageNet: a Unified Scheme for Training any Backbone to Top Results

USI 의 robustness 검증

- 모델들에 똑같이 USI 를 적용했을 때, tailor-made schemes 을 적용한 각 논문의 Top1 accuracy 보다 좋은 성능을 보임

Batch Size	Top1 Acc. [%]	Training speed [img/sec]
512	82.3	1100
1024	82.5	1900
2048	82.3	3000
2752	82.4	4100
3456	82.4	4300
3456		4900 (no-KD reference)

Table 3: Accuracy and training speed, for different batch sizes. Model tested - TResNet-M.

Student	Student Type	Teacher	Teacher Type	Top1 Acc. [%]
ResNet50	CNN	TResNet-L	CNN	81.0
		Volo-d1	Transformer	80.9
LeViT384	Transformer	TResNet-L	CNN	82.7
		Volo-d1	Transformer	82.7

Table 4: Testing different students with different teachers.

In Table 9 we test whether adding drop-path to our scheme, when training a Transformer-based model, would improve results.

Drop-path	Top1 Acc. [%]
0	82.7
0.1	82.6
0.2	82.5

Table 9: Accuracy for different values of drop-path regularization. Model tested - LeViT-384

Solving ImageNet: a Unified Scheme for Training any Backbone to Top Results

제안한 training scheme (KD), loss function 이 잘 작동함을 확인

- ImageNet Training 에서 KD 가 효과적임을 입증
- Vanilla softmax probabilities 를 사용하는 것이 좋음

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(\mathbf{p}^s(1), \mathbf{y}) + \alpha_{\text{kd}}\mathcal{L}_{\text{KL}}(\mathbf{p}^s(\tau), \mathbf{p}^t(\tau)), \qquad (2)$$

KD relative weight, α_{kd}	Top1 Acc. [%]
0 (no KD loss)	76.2
1	80.8
5	82.7
10	82.6
20	82.7
∞ (no CE loss)	82.7

Table 6: Accuracy for different KD relative weights. Model tested - LeViT-384

3.6.2 KD Temperature

In Table 7 we investigate the impact of KD Temperature (τ in Eq. 2) on the accuracy.

KD Temperature, τ	Top1 Acc. [%]
0.1	79.3
1	82.7
2	82.7
5	81.7
10	81.4

Table 7: Accuracy for different KD temperatures. Model tested - LeViT-384

Solving ImageNet: a Unified Scheme for Training any Backbone to Top Results

추가로 성능을 더 높일 수 있는 방법

- Epoch 에 관한 USI 의 default configuration 은 300 이지만, 더 긴 training epoch 으로 성능을 향상 시킬 수 있다.
- Augmentation 은 적용하는게 좋음.

Training epochs	Top1 Acc. [%]
100	80.0
200	81.9
300	82.7
600	83.0
1000	83.2

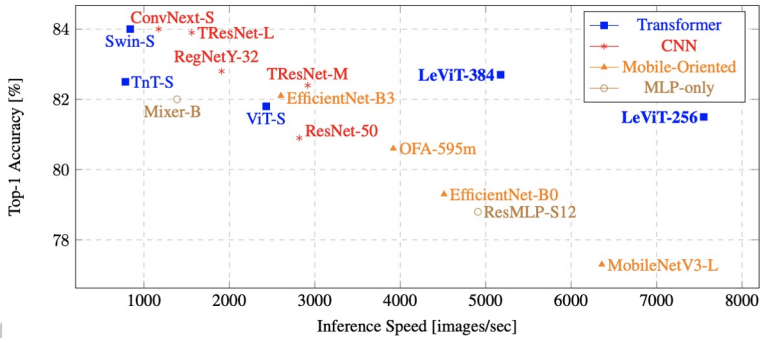
Table 5: Accuracy for different numbers of epochs. Model tested - LeViT-384.

Augmentation Type	Top1 Acc. [%]
None	82.0
Cutout	82.4
Mixup-Cutmix	82.7

Table 8: Accuracy for different augmentations. Model tested - LeViT-384

Speed-Accuracy comparison

- USI 를 활용해 모든 backbone 에 대해 동일한 하이퍼 파라미터를 적용했고, 이에 따라 재현성과 신뢰도가 높은 speed-accuracy trade-off 비교가 가능하다.



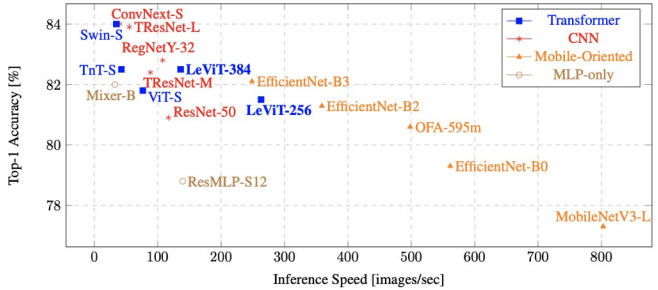


Figure 4: Speed-Accuracy comparison on an Intel Xeon Cascade Lake 2.5Ghz CPU, 16 cores.

CVPR2022
https://arxiv.org/pdf/2204.13101.pdf

Self-Supervised Learning of Object Parts for Semantic Segmentation

Adrian Ziegler
Technical University of Munich
adrian.ziegler@tum.de

Yuki M. Asano
QUVA Lab
University of Amsterdam
y.m.asano@uva.nl

This paper proposes **Vision Transformer's capability of attending to objects and combine it with a spatially dense clustering task for fine-tuning the spatial tokens.**

- Self-supervised learning:
  - strong image representation learning(**image-level learning**)을 가능하게 함
  - unsupervised **image segmentation**은 **spatially-diverse representations이 필요해서 예외**
    - **learning dense representations 어려움**
      unsupervised context에서 다양한 potential object categories에 해당하는 representations을 학습하도록 모델을 guide하는 방법이 명확하지 않기 때문에

- **해결책: object parts**에 대한 self-supervised learning
  - leverage the recently proposed **Vision Transformer's capability of attending to objects** (ViT 객체 관리 기능을 활용)
  - combine it with a **spatially dense clustering task** for fine-tuning the spatial tokens. (공간 토큰을 미세 조정하기 위해 공간적으로 밀집된 클러스터링 작업과 결합)

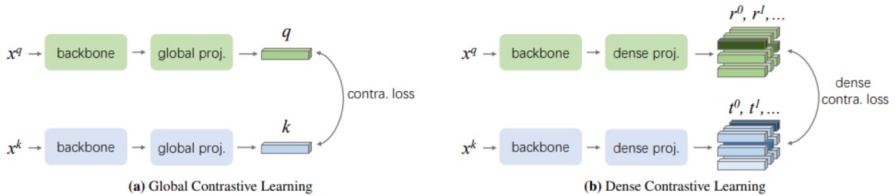
- 결과:
  - three semantic segmentation benchmarks에서 최첨단 기술을 3%-17% 능가 (= representations이 다양한 object definitions에서 다재다능)
  - extend this to fully unsupervised segmentation (테스트 시에도 레이블 정보의 사용을 완전히 자제함)
  - community detection 를 기반으로 발견된 object parts을 자동으로 병합하는 간단한 방법이 상당한 이득을 가져온다는 것 확인

** Spatial segmentation is a powerful tool that provides a concise representation of the high-dimensional data and **helps identify regions of interest (ROIs) for the downstream analysis.**

Dense Self supervised learning

이렇게 학습시킨 모델을 dense prediction이 필요한 task(semantic segmentation, object detection, instance segmentation)에 fine-tuning하면 gap이 존재한다. moco는 image classification을 기준으로 global represenation을 추출해 contrastive learning을 수행했기 때문.

이 gap을 채우기 위하여 dense contrastive learning을 도입한다.



(a) Global Contrastive Learning **(b)** Dense Contrastive Learning

Figure 2 – Conceptual illustration of two contrastive learning paradigms for representation learning. We use a pair of query and key for simpler illustration. The backbone can be any convolutional neural network. (a): The contrastive loss is computed between the single feature vectors outputted by the global projection head, at the level of global feature; (b): The dense contrastive loss is computed between the dense feature vectors outputted by the dense projection head, at the level of local feature. For both paradigms, the two branches can be the same encoder or different ones, *e.g.*, an encoder and its momentum-updated one.

head를 하나 더 추가하여 dense feature map을 추출하고 이 feature map의 vector 사이에 contrastive learning을 수행함.

head를 하나 더 추가하여 dense feature map을 추출하고 이 feature map의 vector 사이에 contrastive learning을 수행함.

$$\mathcal{L}_r = \frac{1}{S^2}\sum_s -\log \frac{\exp(r^s \cdot t_+^s/\tau)}{\exp(r^s \cdot t_+^s) + \sum_{t_-^s} \exp(r^s \cdot t_-^s/\tau)},$$

전체 loss는 global contrastive loss와 dense contrastive loss를 함께 사용한다.

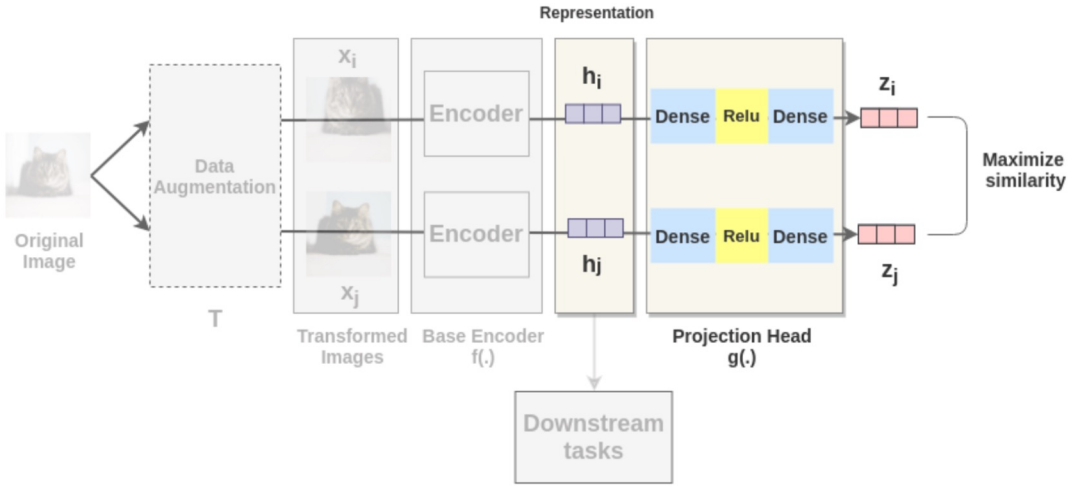
$$\mathcal{L} = (1-\lambda)\mathcal{L}_q + \lambda \mathcal{L}_r,$$

SIMCLR - projection head

3. Projection Head

The representations h_i and h_j of the two augmented images are then passed through a series of non-linear **Dense -> Relu -> Dense** layers to apply non-linear transformation and project it into a representation z_i and z_j. This is denoted by $g(.)$ in the paper and called projection head.

Projection Head Component



Self-Supervised Learning of Object Parts for Semantic Segmentation

Introduction

To tackle the lack of a principled object definition during training, many methods resort to defining object priors such as saliency and contour detectors to induce a notion of objectness into their pretext tasks, effectively rendering such methods semi-supervised and potentially not generalizeable.
Training할 object definition이 부족하니까, saliency 및 contour detectors와 같은 object priors를 정의 -> objectness에 대한 개념을 pretext tasks에 유도

- effectively rendering such methods semi-supervised
- potentially not generalizeable.

Supervised manner:

- great potential unifying architectures and scaling well with data into billions
- work for image-level tasks and dense tasks

Self Supervised manner:

- **self-supervised ViTs to localize objects with our dense loss to train spatial tokens for unsupervised segmentation.**
- **목표: to close this gap by self-supervisedly learning dense ViT models**

Fig1:
ViT:

- Supervised manner:
  - semantic segmentation 성능 증가
  - attention heads는 object의 localization에서 저조한 성능
- Self Supervised manner:
  - object extracting 탁월
  - spatial token embedding space 학습X

** object prior: feature map을 기준으로 aspect ratio와 scale에 따라 미리 계산된 박스
** pretext tasks: 일반적으로 self-supervised learning은 pretext tasks와 downstream으로 구성
** spatial token:

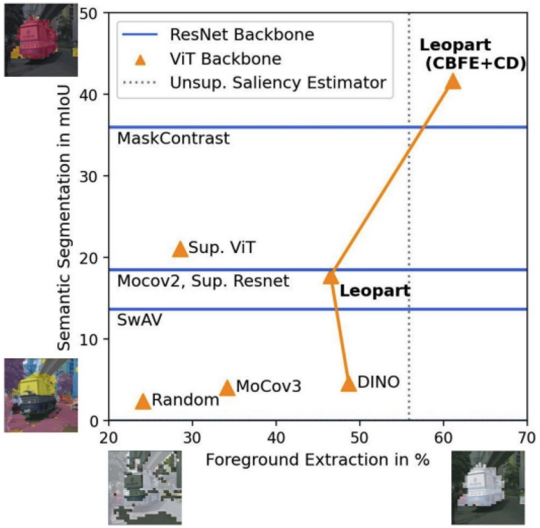


Figure 1. **ViTs and Resnets compared under foreground extraction and semantic segmentation.** We use Jaccard distance as a measure for foreground extraction. Starting from a DINO initialization, our method, Leopart, closes the performance gap between self-supervised ViTs and their supervised counterparts as well as Resnets. Leopart (CBFE+CD) further improves a ViT's object extraction capabilities and sets new state-of-the-art for fully unsupervised semantic segmentation.

Self-Supervised Learning of Object Parts for Semantic Segmentation

Methods

- 목표: object의 동일한 부분을 포함하는 image patch를 group화하는 embedding space 학습
  - object part representation은 데이터셋 간에 전달이 더 잘되어야
  - image patch level에서 image category learning을 하도록 pretext task
  - teacher-student network
  - use our loss to fine-tune pretrained neural networks.
  - this circumvents known cluster stability issues and clusters capturing low-level image features when applied to a patch-level

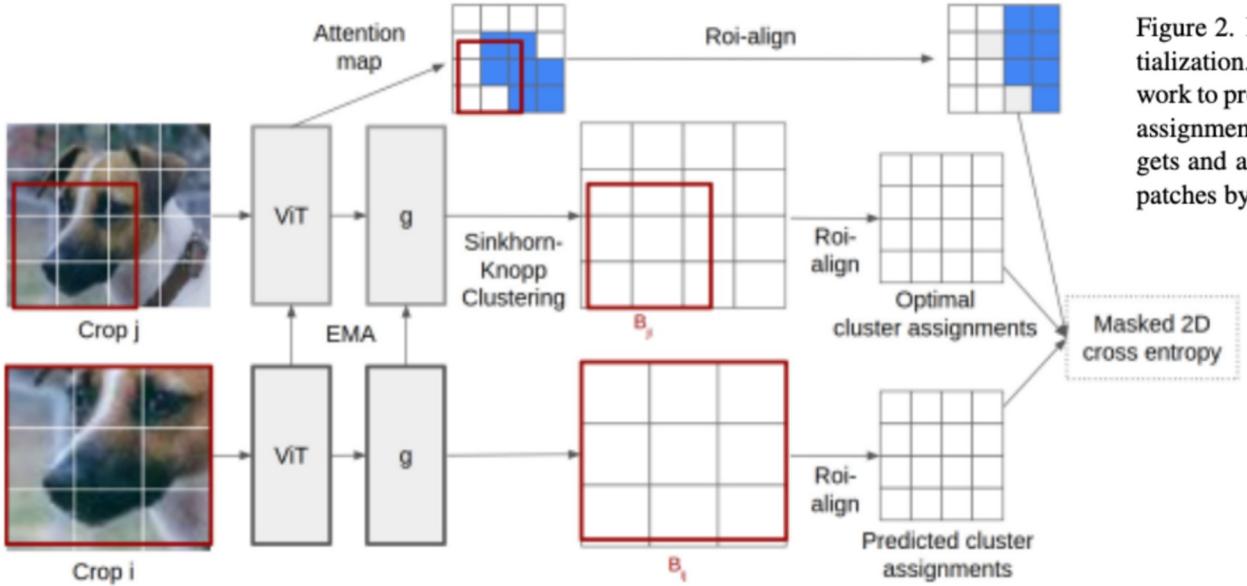
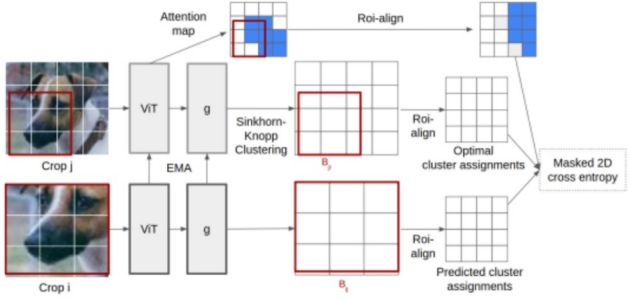


Figure 2. **Leopart training pipeline.** We start from a DINO initialization. We feed different crops to the student and teacher network to produce patch-level cluster predictions and optimal cluster assignments targets. This requires an alignment step of cluster targets and assignments. We further focus clustering on foreground patches by leveraging the ViT's attention map.

Self-Supervised Learning of Object Parts for Semantic Segmentation

Methods

- Fine-tunning loss for spatial tokens
  - Image Encoder
    - image flatten
    - seperate patches
  - **Leopart fine-tunning loss**



Leopart fine-tuning loss. To train the ViT's spatial tokens, we first randomly crop the image V-times into v_g global views and v_l local views. When sampling the views we compute their pairwise intersection in bounding box format and store it in a matrix B. We denote the transformed version of the image as $x_{t_j}, j \in \{1, \dots, V\}$. Then, we forward the spatial tokens through a MLP projection head g with a L2-normalization bottleneck to get spatial features for each crop: $g(f(x_{t_j})) = Z_{t_j} \in \mathbb{R}^{D \times N}$. To create prediction targets, we next find an optimal soft cluster assignment Q_{t_j} of all spatial token's feature vector Z_{t_j} to K prototype vectors $[c_1, \dots, c_K] = C \in \mathbb{R}^{D \times K}$. For that, we follow the online optimization objective of SwAV [5] that works on the whole image batch b. Q is optimized such that the similarity between all feature vectors in the batch and the prototypes is maximized, while at the same time being regularized towards assigning equal probability mass to each prototype vector. This can be cast to an optimal transport problem and is solved efficiently with the Sinkhorn-Knopp algorithm [1, 14]. Instead of optimizing over $|b|$ feature vectors, we instead optimize over $N \cdot |b|$ spatial feature vectors as we have N spatial tokens for each image. As our batchsizes are small, we utilize a small queue that keeps the past 8192 features, as is done in SwAV.

With the optimal cluster assignment of all image crops' spatial tokens $Q_{t_k} \in \mathbb{R}^{N \times K}$, we formulate a swapped prediction task:

$$L(x_{t_1}, ..., x_{t_V}) = \sum_{j=0}^{v_g} \sum_{i=0}^{V} \mathbb{1}_{k \neq j} l(x_{t_i}, x_{t_j}) \tag{1}$$

Here, l is the 2D cross entropy between the softmaxed and aligned cluster assignment predictions and the aligned optimal cluster assignments:

$$l(x_{t_i}, x_{t_j}) = H\left[(s_\tau(\alpha_{B_{j,i}}(g(\Phi(x_{t_i}))^T C), \alpha_{B_{ij}}(Q_{t_j})\right], \tag{2}$$

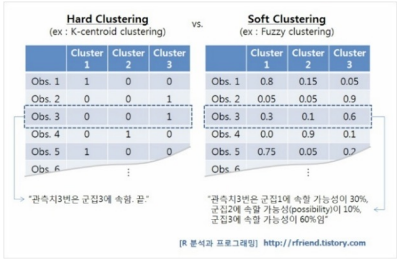
where H is cross-entropy and s_τ a softmax scaled by temperature τ. We use L to jointly minimize the prototypes C as well as the neural networks f and g. C is further L2-normalized after each gradient step such that $Z^T C$ directly computes the cosine similarity between spatial features and prototypes.

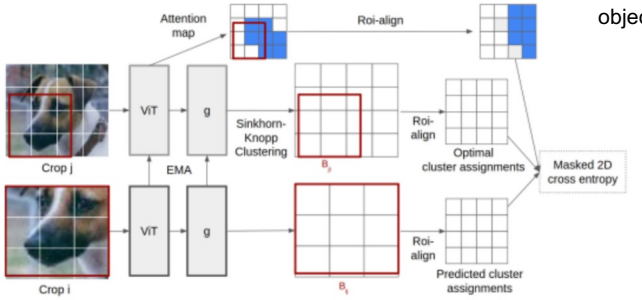
Since global crops capture the majority of an image, we solely use these to compute Q_{t_j}, as the spatial tokens can attend to global scene information such that the overall prediction target quality improves. Further, as local crops just cover parts of images and thus also parts of objects, using these produces cluster assignment predictions that effectively enable object-parts-to-object-category reasoning, an important ability for scene understanding.

Self-Supervised Learning of Object Parts for Semantic Segmentation

Methods

- Fine-tunning loss for spatial tokens
  - Image Encoder
    - image flatten
    - seperate patches
  - **Leopart fine-tunning loss**
    - **to train ViT spatial tokens:**
      - **randomly crop** the image V -times into vg global views and vl local views
      - **forward the spatial tokens** through a MLP projection head g with a L2-normalization bottleneck to get spatial features for each crop
      - To create prediction targets, **find an optimal soft cluster assignment Qtj** of all spatial token's feature vector Ztj to K prototype vectors
        - Q is optimized such that the similarity between all feature vectors in the batch and the prototypes is maximized, while at the same time being regularized towards assigning equal probability mass to each prototype vector.
        - This can be cast to an optimal transport problem and is solved efficiently with the **Sinkhorn- Knopp algorithm**
        - Instead of optimizing over |b| feature vectors, we optimize over N ·|b| **spatial feature vectors as we have N spatial tokens for each image.**
        - As our batch sizes are small, we **utilize a small queue that keeps the past 8192 features, as is done in SwAV.**
        - With the optimal cluster assignment of all image crops' spatial tokens Qt formulate a swapped prediction task
        - **2D cross entropy between the softmaxed and aligned cluster assignment predictions and the aligned optimal cluster assignments**
        - global crops
          - capture the majority of an image, we solely use these to compute Qtj , as the **spatial tokens can attend to global scene information** such that the overall prediction target quality improves.
        - local crops
          - cover parts of images and thus also parts of objects, using these produces cluster assignment predictions that effectively enable object-parts-to-object-category reasoning, an **important ability for scene understanding.**





**MLP Projection Head:
A small neural network, MLP with one hidden layer, is used to map the representations from the base encoder to 128-dimensional latent space where contrastive loss is applied.

**forward propagation:
뉴럴 네트워크 모델의 입력층부터 출력층까지 순서대로 변수들을 계산하고 저장

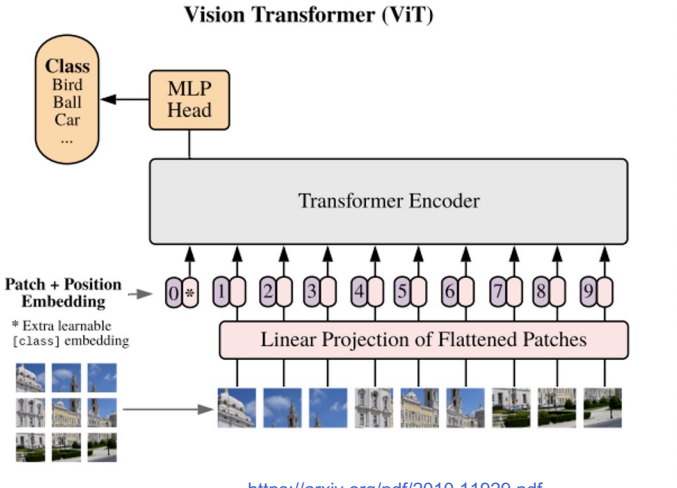
**back propagation:
뉴럴 네트워크의 파라미터들에 대한 그래디언트(gradient)를 계산

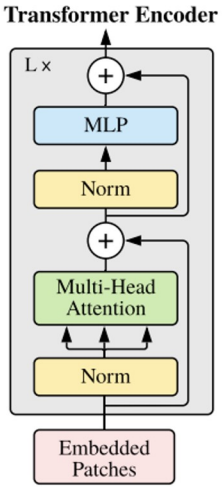
Self-Supervised Learning of Object Parts for Semantic Segmentation

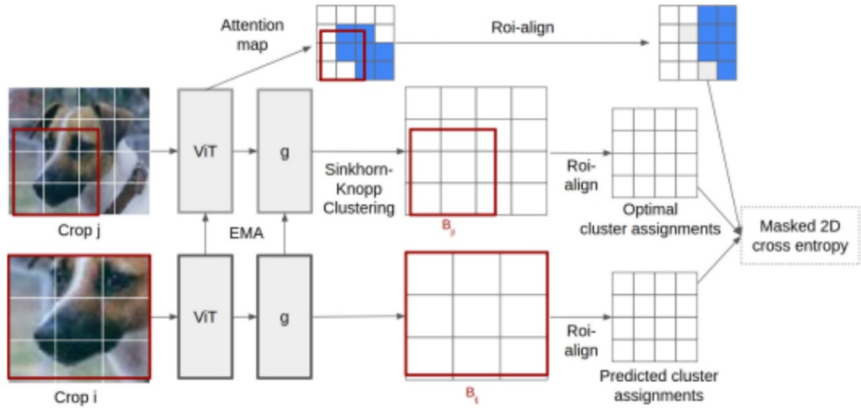
Methods

- Fine-tunning loss for spatial tokens
  - Alignment
    - use RoI-Align that produces features with a fixed and compatible output size.
  - Foreground focused clustering
    - leverage the **ViT's CLS token attention maps** of each of its attention heads.
    - To create a foreground clustering mask that can be used during training
      - average the attention heads to one map
      - apply a Gaussian filter for smoothing.
      - obtain a **binary mask by thresholding the map** to keep 60% of the mass
      - use alignment operator to **align the global crop's attention to the intersection with crop j.**
      - The resulting mask is then applied as 0-1 weighting to the 2D cross entropy loss
      - Note that we extract the attention maps and spatial tokens with the same forward pass, thus not impacting training speed.

**Region of Interest Align, or RoIAlign:
an operation for extracting a small feature map from each RoI in detection and segmentation based tasks.







https://arxiv.org/pdf/2010.11929.pdf

Self-Supervised Learning of Object Parts for Semantic Segmentation

Methods

- Fully unsupervised semantic segmentation
  Its constituent parts work directly in the learned spatial token embedding space and leverage simple K-means clustering.

  - Cluster-based Foreground Extraction (CBFE)
    - **clusters** in our learned embedding space **correspond to object parts,** we should be able to **extract foreground objects** by assigning each cluster id to foreground object (fg) or background (bg): $\Theta : \{1, \ldots, K\} \rightarrow \{fg, bg\}$
    - at evaluation time, we construct Θ without supervision, **by using ViT's merged attention maps as a noisy foreground hint.**
    - Similar to how we process the attention maps to focus our clustering pretext on foreground,
      - **average the attention heads,**
      - **apply Gaussian filtering with a 7x7 kernel size**
      - **keep 60% of the mass to obtain a binary mask.**
    - Using train data,
      - rank all clusters by pixel-wise precision
      - find a good threshold for classifying a cluster as foreground.
    - This gives us Θ that we apply to the patch-level clustering to get a foreground mask.

  - Overclustering with community detection (CD)
    - propose a novel overclustering method that requires no additional supervision at all.
    - **The key idea:**
      - **clusters correspond to object parts, and that a set of object parts frequently co- occur together.**
      - **Thus, local co-occurrence of clusters in an image should provide a hint about an object's constituent parts.**
    - we are the first to work with object parts and no labels and employ a novel network science method to discover objects.
    - we construct an undirected and weighted co- occurence network G = (V,E,w), with vi,i ∈ {1,...,K} corresponding to each cluster.
    - We use a localized co- occurrence variant that regards the 8-neighborhood up to a pixel distance d.
    - Then, we calculate the conditional co- occurrence probability P (vj |vi) for clusters i and j over all images D.
    - With the co-occurrence probabilities at hand, we define w(ei,j) = min(P(vj|vi),P(vj|vi)).
    - This asymmetric edge weight definition is motivated by the fact that parts need not be mutually predictive:
      - For instance, a car wind- shield might co-occur significantly with sky but presence of a sky is not predictive for a car windshield.

**If you load too much DNA,
clusters will be too close together (over-clustering),
resulting in poor image resolution and analysis problems.

Self-Supervised Learning of Object Parts for Semantic Segmentation

Experiments

- Setup
  - evaluation protocols
    - discard the projection head used during training.
    - Instead directly evaluate the ViT's spatial tokens.
    - use linear classifier
      - fine-tune a 1x1 convolutional layer on top of the frozen spatial token or the pre- GAP layer4 features,
    - use overclustering.
      - run K-Means on all spatial tokens of a given dataset.
      - then group cluster to ground-truth classes by greedily matching by pixel-wise precision
      - run Hungarian matching [40] on the merged cluster maps to make our evaluation metric permutation-invariant
      - always report overclustering results averaged over five different seeds.
  - model training
    - train a ViT-Small
    - with patch size 16
    - start training from DINO weights [6].
    - All models were trained for 50 epochs
    - using batches of size 32
    - on 2 GPUs.
  - datasets
    - We train our model on ImageNet-100, comprising 100 randomly sampled ImageNet classes [55], COCO [42] and Pascal VOC (PVOC)
    - When finetuning on COCO-Stuff and COCO-Thing, we use a 10% split of the training sets.
    - Evaluation results are computed on the full COCO validation data for COCO-Stuff and COCOThing and PVOC12 val.

Experiments

		Num. clusters		
mask	LC	100	300	500
all	67.4	37.9	44.6	47.8
bg	64.7	28.1	39.0	41.4
fg	**67.8**	**38.2**	**47.2**	**50.7**

(a) Focusing clustering on foreground (fg) helps.

		Num. clusters		
crops	LC	100	300	500
[2]	66.1	33.0	42.5	45.0
[2,2]	67.7	37.8	45.4	49.3
[2,4]	**67.8**	**38.2**	**47.2**	**50.7**

(b) Local crops boost performance.

		Num. clusters		
tchr	LC	100	300	500
✗	67.6	34.6	44.3	47.9
✓	**67.8**	**38.2**	**47.2**	**50.7**

(c) Using an EMA teacher helps.

		Num. clusters		
protos	LC	100	300	500
100	67.7	36.8	45.4	49.2
300	**67.8**	**38.2**	**47.2**	**50.7**
500	67.4	35.8	44.8	49.1

(d) 300 prototypes work well.

Table 1. **Ablations** of different design decisions for Leopart.

Dataset	size	LC	K=500	K=300	K=100
IN-100	126k	67.8	50.7	47.2	38.2
COCO	118k	**69.1**	**53.0**	**49.9**	**44.3**
PASCAL	10k	64.5	50.7	47.8	38.2

Table 2. **Training data study for Leopart**. We use the best performing model config from Table 1 and train on different datasets.

Self-Supervised Learning of Object Parts for Semantic Segmentation

Experiments

- Transfer learning
  - how well our dense representations, once learned, generalize to other datasets
  - 일부에서는 추가 데이터 세트와 감독을 사용하더라도 세 데이터 세트 모두에서 자체 감독 이전 작업을 큰 폭으로 능가한다.
  - **PVOC12에서 선형 평가에서는 17% 이상, 오버클러스터링에서는 5% 이상 state-of-the-art 능가한다.**
  - **COCO-Things 및 COCO-Stuff에서 선형 분류기를 > 5% 및 > 3% 개선하고 오버클러스터를 각각 > 8% 및 > 10% 향상시킨다.**
  - these gains are not due to the DINO initialisation nor due to ViTs per-se as the starting DINO model performs on par with other instance-level self-supervised methods that use ResNets like SwAV. In fact, DINO's embedding space exhibits inferior semantic structure in comparison to MoCo-v2 and SwAV as can be seen from the overclustering results on PVOC12 (-18%) and COCO-Things (-12%). Our method is also on par with the performance of a supervised ViT even though it was trained on a >10x times larger full ImageNet (IN-21k) dataset [51]. When fine-tuning on COCO instead of IN-100, we see further improvements on all datasets.
  - **The results confirm that it is desirable to learn object parts representations, as they work well under different object definitions,** as evidenced by strong performances across datasets.

- Table 6: evaluate Leopart by fine-tuning a full FCN on top of frozen features.
  - Again, we outperform all prior works, including **DenseCL**, the current state-of-the-art.
  - DenseCL은 선형 레이어가 아닌 FCN을 미세 조정할 때 20% 이상의 성능 향상을 보이는 반면, 미세 조정을 통한 성능 향상은 약 2%로 상대적으로 낮다
  - We hypothesize that this behaviour is **because our learned embedding space is already close to maximally informative for semantic segmentation under linear transformations.**
  - In contrast, **DenseCL's embedding space alone is less informative in itself and requires a more powerful non-linear transformation.**
  - We push state-of-the-art even further by fine-tuning a larger ViT-Base with patch size 8 (ViT-B/8) improving FCN performance by around 5%.

Method	Train	PVOC12 LC	PVOC12 K=500	COCO-Things LC	COCO-Things K=500	COCO-Stuff LC	COCO-Stuff K=500
Sup. ViT	IN + IN21	68.1	55.1	65.2	50.9	49.0	35.1
Sup. ResNet	IN	53.8	36.5	57.8	44.2	44.4	30.8
instance-level:							
MoCo-v2 [29]	IN	45.0†	39.1	47.5	36.2	32.6	28.3
DINO [6]	IN	50.6	17.4	50.6	23.5	47.7	32.1
SwAV [5]	IN	50.7†	35.7	56.7	37.3	46.0	33.1
pixel/patch-level:							
IIC [37]	PVOC	28.0†	-	-	-	-	-
MaskContrast [57]	IN+PVOC	49.2	45.4	47.5	37.0	32.0	25.6
DenseCL [59]	IN	49.0	43.6	53.0	41.0	40.9	30.3
Leopart	IN	68.0	50.5	62.5	49.2	51.2	**43.8**
Leopart	IN+CC	**69.3**	**53.3**	**67.6**	**55.9**	**53.5**	43.6

Table 3. **Transfer learning for semantic segmentation results.** Best results are in **bold** and second best are underlined. 'IN', 'IN21', 'CC' and 'PVOC' indicate training on ImageNet, ImageNet21k, CoCo and Pascal *trainaug* respectively. † indicates result taken from [57].

Method	Train	PVOC12 FCN
SegSort [35]	CC+PVOC	36.2†
Hier. Grouping [64]	CC+PVOC	48.8†
DINO [6]	IN	60.6
Hier. Grouping [64]	IN	64.7†
MoCo-v2 [29]	CC	64.5†
MoCo-v2 [29]	IN	67.5†
DenseCL [59]	CC	67.5†
DenseCL [59]	IN	69.4†
Leopart	IN	70.1
Leopart	IN+CC	**71.4**
Leopart (ViT-B/8)	IN+CC	**76.3**

Table 6. **FCN transfer learning results**. We follow the same notation as in Table 3. Note that Hierarchical Grouping and Segsort fine-tune a larger ASPP decoder. † indicates result taken from [59, 64]

Self-Supervised Learning of Object Parts for Semantic Segmentation

**Test Time Augmentation:
Train데이터 이미지에 다양한 Aug기법들을 적용하여 data create를 하는 것과 달리 테스트 이미지 데이터에 Augmentation을 적용하는 것

Experiments

- Fully unsupervised semantic segmentation
  - 표 3:
    - 학습된 임베딩 공간의 구조에만 의존하며 test-time label information를 사용하지 않는다.
    - 즉, 최종 클러스터 수는 실측 결과와 같아야 한다.
    - 각 토큰에 대한 클러스터 할당을 얻기 위해 공간 토큰의 간단한 K-mean 클러스터링으로 시작한다.
  - 표 4:
    - PVOC12 val을 기반으로 하고 임의의 데이터 세트(이 경우 COCO)를 기반으로 self supervised train
    - NAT은 가장 가까운 경쟁사인 MaskContrast를 4% 이상 능가
    - MaskConstrast와 마찬가지로, fg 토큰만 클러스터링하지만,
      - pretrained unsupervised saliency estimator 대신 임베딩 공간 클러스터링을 사용하여 클러스터 기반 전경 추출(CBFE)을 수행한다.
      - 이미지당 feature representation을 평균화하는 대신 (CD)가 있는 새로운 비지도 오버클러스터링 방법을 사용하여 한 이미지에서 여러 개체 카테고리를 감지할 수 있다.

Method	Train	PVOC12		COCO-Things		COCO-Stuff	
		LC	K=500	LC	K=500	LC	K=500
Sup. ViT	IN + IN21	68.1	55.1	65.2	50.9	49.0	35.1
Sup. ResNet	IN	53.8	36.5	57.8	44.2	44.4	30.8
instance-level:							
MoCo-v2 [29]	IN	45.0†	39.1	47.5	36.2	32.6	28.3
DINO [6]	IN	50.6	17.4	50.6	23.5	47.7	32.1
SwAV [5]	IN	50.7†	35.7	56.7	37.3	46.0	33.1
pixel/patch-level:							
IIC [37]	PVOC	28.0†	-	-	-	-	-
MaskContrast [57]	IN+PVOC	49.2	45.4	47.5	37.0	32.0	25.6
DenseCL [59]	IN	49.0	43.6	53.0	41.0	40.9	30.3
Leopart	IN	68.0	50.5	62.5	49.2	51.2	**43.8**
Leopart	IN+CC	**69.3**	**53.3**	**67.6**	**55.9**	**53.5**	43.6

Table 3. **Transfer learning for semantic segmentation results.** Best results are in **bold** and second best are underlined. 'IN', 'IN21', 'CC' and 'PVOC' indicate training on ImageNet, ImageNet21k, CoCo and Pascal *trainaug* respectively. † indicates result taken from [57].

Method	mIoU
Sup. ResNet	18.5
Sup. ViT	21.1
DINO [6]	4.6
SwAV [29]	13.7
MoCo-v2 [29]	18.5
MaskContrast [57]	35.0†
Leopart (CBFE+CD)	**41.7**

Table 4. **Unsupervised semantic segmentation results.** We outperform other state-of-the-art methods by a large margin. † indicates result taken from [57].

Self-Supervised Learning of Object Parts for Semantic Segmentation

Experiments

- Fully unsupervised semantic segmentation
  - performance gain study
    - Figure 3: gradual visual improvement of the segmentations.
      - DINO segmentations show no correspondence to object categories, whereas the segmentations obtained by Leopart assign the same colors to the bus in the first and third image of the top row as an example.
      - our segmentations do not correspond well with PVOC's object definitions, as we oversegment background.
        - To further improve this, we extract foreground resulting in the segmentation maps shown in Figure 3c.
        - The segmentation focuses on the foreground and object categories start to emerge more visibly.
        - However, some objects are still oversegemented such as busses and cats.
        - Thus, we run our proposed community detection algorithm to do fully unsupervised overclustering, resulting in the segmentations shown in Figure 3d.

(a) DINO

(b) + Leopart

(c) + CBFE

(d) + CD

Self-Supervised Learning of Object Parts for Semantic Segmentation

Experiments

- Fully unsupervised semantic segmentation
  - performance gain study
    - CBFE (Cluster-based Foreground Extraction)
      - This is remarkable as we can only improve the attention map if the foreground clusters also segment the foreground correctly where the noisy foreground hint from DINO's attention is wrong.
      - While the attention masks only mark the most discriminative regions they fail to capture the foreground object's shape (Fig. 4(a))
      - Our cluster masks, however, alleviate this providing a crisp foreground object segmentation (Fig. 4(b)).
      - With the foreground masks extracted, we can specify K-Means to run only on foreground spatial tokens. This further improves our fully unsupervised segmentation performance by > 17%, as shown in Table 5.

Method	Jacc. (%)	B-F1 [13] (%)
DINO attention [6]	48.7	36.5
Unsup. saliency [57]	55.9	40.8
Leopart IN CBFE	58.6	**42.1**
Leopart CC CBFE	**59.6**	40.7

Table 7. **Foreground extraction results on PVOC val.** Our method improves over DINO atenttions with respect to Jaccard distance and Boundary F1 score and shows performance on par with a dedicated unsupervised saliency estimator.

(a) Attention Masks (b) Cluster masks

Figure 4. **DINO Attention masks vs. Leopart Cluster masks.**

	mIoU
K=150	48.8
DINO	4.6
+ Leopart	18.9 (+14.3%)
+ CBFE	36.6 (+17.7%)
+ CD	41.7 (+5.1%)

Table 5. **Component contributions.** We show the gains that each individual component brings for PVOC segmentation and K=21.

Self-Supervised Learning of Object Parts for Semantic Segmentation

Experiments

- Fully unsupervised semantic segmentation
  - performance gain study
    - CD (Overclustering with community detection)
      - overclustering yields benefits in terms of performance
      - but requires additional supervision for merging clusters during evaluation.
      - construct a network based on cluster co-occurrences and run community detection (CD)
      - 표 5 :
        - 성능을 5% 이상 향상시킬 수 있고
        - K = 150으로 감독된 오버클러스터의 상한에 가깝게 할 수 있다
      - Figure 5:
        - 레오파트는 자전거 바퀴와 자동차 바퀴를 따로 발견
          낮은 수준의 정보에 얽매이지 않는 높은 수준의 의미론적 클러스터를 학습할 수 있음
        - 인간의 머리카락과 사람의 얼굴과 같이 의미적으로 유사한 클러스터도
          같은 공동체의 일부이며 결과 네트워크에서 가깝다는 것을 관찰할 수 있다.
        - 상호 연결된 서로 다른 커뮤니티의 일부인 개 주둥이와 캐터어(catear)에 대해 표시된
          연결된 구성 요소 내에서 점진적인 의미적 전환을 관찰할 수 있다.

	mIoU
K=150	48.8
DINO	4.6
+ Leopart	18.9 (+14.3%)
+ CBFE	36.6 (+17.7%)
+ CD	41.7 (+5.1%)

Table 5. **Component contributions.** We show the gains that each individual component brings for PVOC segmentation and K=21.

Figure 5. **Communities found in our cluster co-occurrence network constructed through self-supervision.** Each node corresponds to a cluster in our learnt embedding space. The nodes are colored by community membership.

Self-Supervised Learning of Object Parts for Semantic Segmentation

Discussion

- limitations
  - learn on a pixel but on a patch level
    - segmentation maps are limited in their resolution and detection capabilities.
    - method will fail when fine-grained pixel-level segmentation is required or very small objects covering less than an image patch are supposed to be segmented.
  - unsupervised overclustering method does a hard assignment of clusters to communities.
    - This has the limitation that object parts which occur in several objects are assigned to the wrong object category when they appear in a specific context.

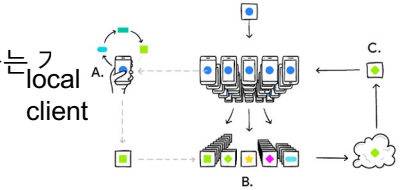
- potential negative societal impact
  - Self-supervised semantic segmentation는 segmentation 결과에 대한 엄격한 모니터링은 필수
  - 모니터링 부족은 자율 주행 및 가상 현실과 같은 영역에 잠재적인 부정적인 영향을 미칠 수 있다.

- conclusion
  - propose a dense clustering pretext task for the spatial tokens of a ViT that learns a semantically richer embedding space in contrast to other self-supervised ViTs.
  - 서로 다른 object definitions와 granularities을 특징으로 하는 PVOC, COCO-Stuff 및 COCOTHING semantic segmentation 벤치마크에 대한 최첨단 기술을 개선함에 따라 이 formulation이 유리함을 확인할 수 있다.
  - embedding space can also be directly used for fully unsupervised segmentation,
  - showing that objects can be defined as co-occurring object parts.

FedILC: Weighted Geometric Mean and Invariant Gradient Covariance for **Federated Learning** on **Non-IID**

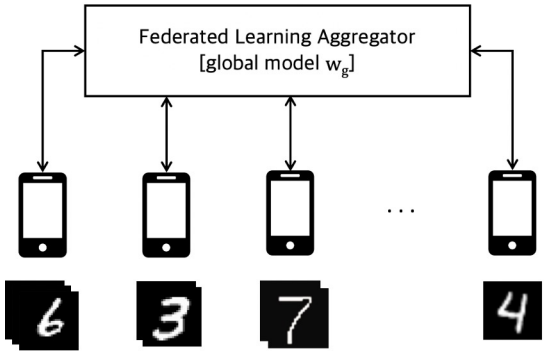
Federated Learning (FL; 연합 학습)

- 다수의 로컬 클라이언트와 하나의 중앙 서버가 협력하여, 데이터가 탈중앙화된 상황에서 글로벌 모델을 학습하는 기
  ⇒ 최종 목표: 데이터는 공유하지 않으면서 모든 데이터셋을 학습한 글로벌 모델을 만드는 것
  ⇒ 방법: 중앙 서버에서 로컬 업데이트를 받아 글로벌 모델을 수정



- 특징
  - **데이터 프라이버시 향상**
    - e.g. 병원의 임상 데이터와 같은 환자 개인정보가 보호되어야 하는 상황에서 데이터 유출 없이 학습 가능
  - **커뮤니케이션 효율성**
    - e.g. 수 만개의 로컬 디바이스의 데이터를 모두 중앙 서버로 전송하게 되면 네트워크 트래픽과 스토리지 비용 증가하는데, FL 을 사용하면 로컬 모델의 업데이트 정보만을 주고 받으므로 커뮤니케이션 비용이 상당히 줄어듦

- **Non-IID (Independent and Identically Distributed)**
  - FL 에서는 학습하는 데이터가 Non-IID 이기 때문에 글로벌하게 최적화된 모델을 만드는데 어려움이 있음
  - Non-IID 데이터
    - 클라이언트가 가지고 있는 각 데이터가 독립 + 동일한 확률 분포를 가지고 있지 않
    - e.g. 대한민국에 거주하고 있는 우리 스마트폰에는 동양인의 사진이 많을 것이고, 다양한 국가에서 FL 을 수행하면 각각의 로컬 업데이트 값의 차이가 크게 생
      e.g. 각 로컬 디바이스는 MNIST 데이터셋의 특정 클래스 데이터만을 가지고 있다
      이런 상황에서 Universal 한 MNIST Classifier 를 만들 수 있을까? NO!



FedILC: Weighted Geometric Mean and Invariant Gradient Covariance for Federated Learning on Non-IID Data

FedILC: Weighted Geometric Mean and Invariant Gradient Covariance for Federated Learning on Non-IID Data

Mike He Zhu
McGill University
Mila - Quebec AI Institute
he.zhu@mila.quebec

Léna Néhale Ezzine
Mila - Quebec AI Institute
lena-nehale.ezzine@mila.quebec

Dianbo Liu
Mila - Quebec AI Institute
dianbo.liu@mila.quebec

Yoshua Bengio
Université de Montréal
Mila - Quebec AI Institute
yoshua.bengio@mila.quebec

Subjects: **Machine Learning (cs.LG)**; Artificial Intelligence (cs.AI); Distributed, Parallel, and Cluster Computing (cs.DC)
Cite as: arXiv:2205.09305 **[cs.LG]**
(or arXiv:2205.09305v1 **[cs.LG]** for this version)
https://doi.org/10.48550/arXiv.2205.09305

https://arxiv.org/pdf/2205.09305.pdf
https://github.com/mikemikezhu/FedILC

Federated Learning (FL; 연합 학습) 은 domain shift problems 을 가짐.

- where the learning models are unable to generalize to unseen domains whose data distribution is non-i.i.d. with respect to the training domains.

이 문제를 해결하기 위해 Federated Invariant Learning Consistency (FedILC) approach 제안

- which leverages the gradient covariance and the geometric mean of Hessians to capture both inter-silo and intra-silo consistencies of environments

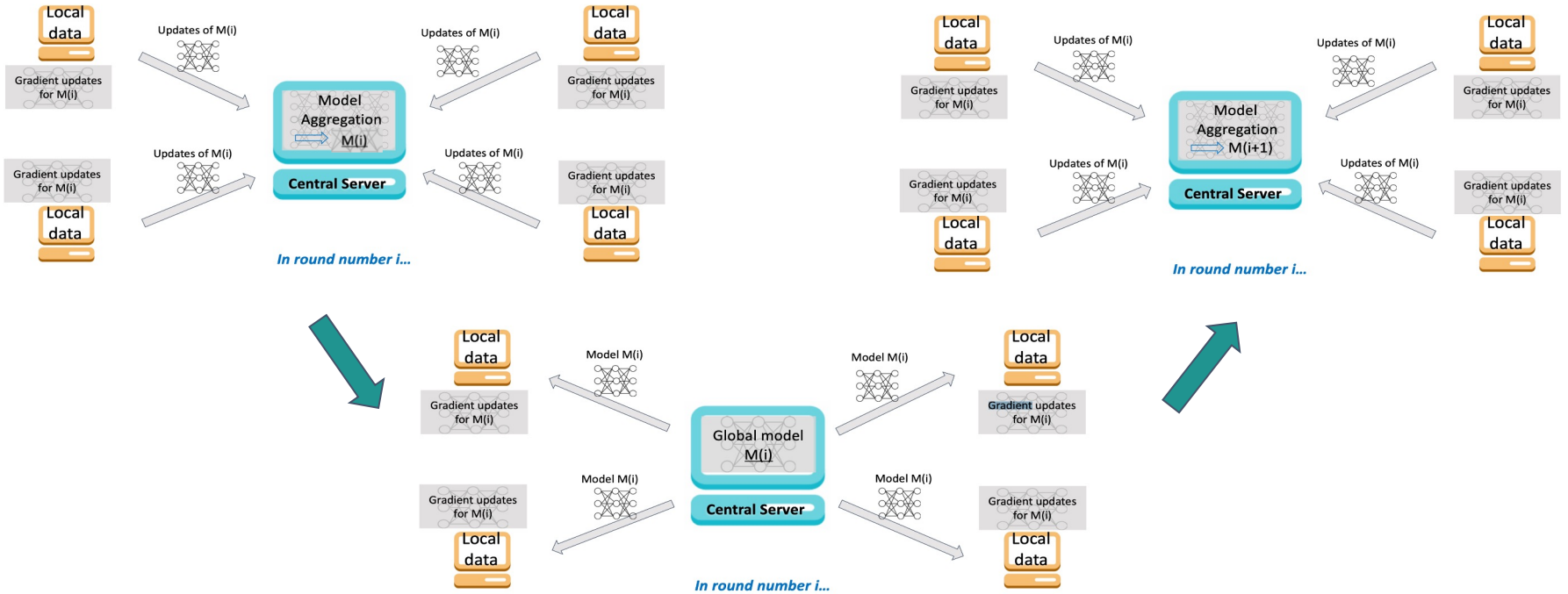
결과

- Benchmark 와 real-world 데이터셋 실험은 conventional baselines 과 유사한 federated learning 알고리즘을 능가
- 이는 다양한 분야 (Medical healthcare, computer vision , IoT) 에 적용 가능함.

FedILC: Weighted Geometric Mean and Invariant Gradient Covariance for Federated Learning on Non-IID Data

Methods

1. Weighted Geometric Mean
2. Fishr+Inter-Geo: Fishr and Inter-silo Weighted Geometric Mean
3. Fishr+Intra-Geo: Fishr and Intra-silo Weighted Geometric Mean



Weighted Geometric Mean

- Parascandolo 는 geometric averaging of Hessians 을 제안했고, 이는 어느 정도 좋은 성능을 보임. 그러나 모든 신호가 consistent 한 환경 속에서만 잘 작동함. Inconsistent signs 에서 계산되는 경우는 거의 없음.
- 본 논문에서는 inconsistent signs 의 geometric mean 을 편리하게 계산하기 위한 **weighted geometric mean** 방법을 제안함.

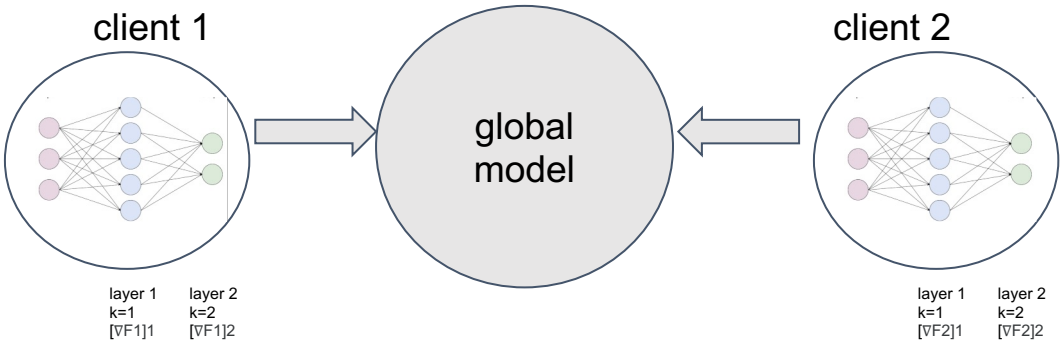
Client model

Arithmetic mean of G

$$arithmean(G) = \frac{\mathcal{E}^+}{\mathcal{E}} \frac{\sum_{e \in \mathcal{E}^+} |G_e|}{\mathcal{E}^+} - \frac{\mathcal{E}^-}{\mathcal{E}} \frac{\sum_{e \in \mathcal{E}^-} |G_e|}{\mathcal{E}^-}$$

Weighted geometric mean of G

$$weightedgeo(G) = \frac{\mathcal{E}^+}{\mathcal{E}} \left(\prod_{e \in \mathcal{E}^+} |G_e| \right)^{\frac{1}{\mathcal{E}^+}} - \frac{\mathcal{E}^-}{\mathcal{E}} \left(\prod_{e \in \mathcal{E}^-} |G_e| \right)^{\frac{1}{\mathcal{E}^-}}$$



$[\nabla F\varepsilon]k = G\varepsilon$

G의 평균 = { [∇F1]1,[∇F2]1,[∇F3]1,[∇F4]1,... }

Weighted Geometric Mean

- Parascandolo 는 geometric averaging of Hessians 을 제안했고, 이는 어느 정도 좋은 성능을 보임. 그러나 모든 신호가 consistent 한 환경 속에서만 잘 작동함. Inconsistent signs 에서 계산되는 경우는 거의 없음.
- 본 논문에서는 inconsistent signs 의 geometric mean 을 편리하게 계산하기 위한 **weighted geometric mean** 방법을 제안함.

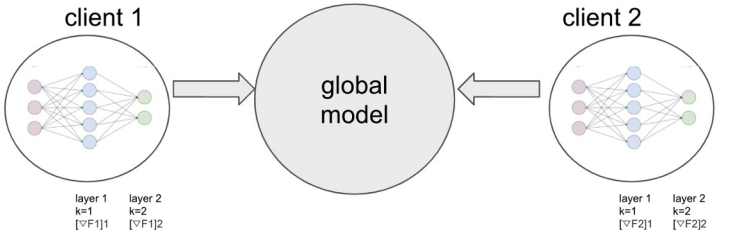
Client model

Arithmetic mean of G

$$arithmean(G) = \frac{\mathcal{E}^+}{\mathcal{E}} \frac{\sum_{e \in \mathcal{E}^+} |G_e|}{\mathcal{E}^+} - \frac{\mathcal{E}^-}{\mathcal{E}} \frac{\sum_{e \in \mathcal{E}^-} |G_e|}{\mathcal{E}^-}$$

Weighted geometric mean of G

$$weightedgeo(G) = \frac{\mathcal{E}^+}{\mathcal{E}} \left(\prod_{e \in \mathcal{E}^+} |G_e| \right)^{\frac{1}{\mathcal{E}^+}} - \frac{\mathcal{E}^-}{\mathcal{E}} \left(\prod_{e \in \mathcal{E}^-} |G_e| \right)^{\frac{1}{\mathcal{E}^-}}$$



[▽Fε]k = Gε

G의 평균 = { [▽F1]1,[▽F2]1,[▽F3]1,[▽F4]1,... }

Global model

$$[\nabla \hat{} F]_k = \frac{\mathcal{E}^+}{\mathcal{E}} \left(\prod_{e \in \mathcal{E}^+} |[\nabla F_e]_k| \right)^{\frac{1}{\mathcal{E}^+}} - \frac{\mathcal{E}^-}{\mathcal{E}} \left(\prod_{e \in \mathcal{E}^-} |[\nabla F_e]_k| \right)^{\frac{1}{\mathcal{E}^-}}$$

Then we update the global model in the federated networks based on the calculated geometric mean of gradients: $w \leftarrow w - \eta \nabla \hat{} F$, where $\nabla \hat{} F = \{\nabla \hat{} F_k, k \in \{0, .., n-1\}\}$. The implementation of weighted geometric mean implementation is further illustrated at `https://colab.research.google.com/drive/17y6ZuwiRE3iHVhFxaXz1b9eSZP7x250m?usp=sharing`.

FedILC: Weighted Geometric Mean and Invariant Gradient Covariance for Federated Learning on Non-IID Data

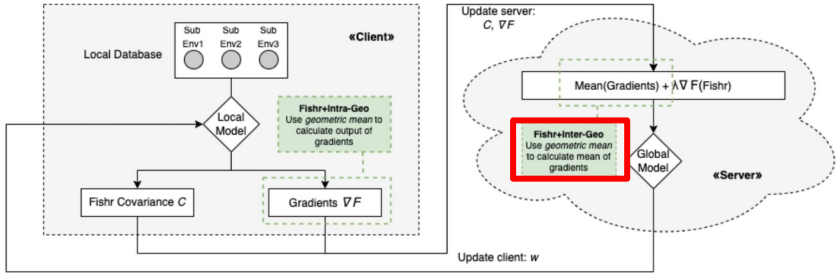


Figure 1: Server-client communication of Fishr+Inter-Geo and Fishr+Intra-Geo in the federated learning settings.

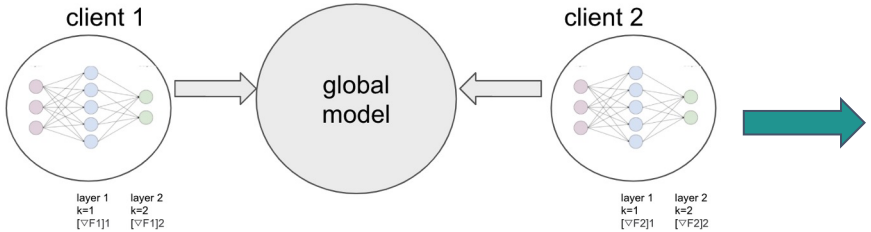
Fishr : Another gradient-based regularization method for domain generalization. Fishr 는 학습 동안의 domain 수를 알고 있다는 가정이 있다.
본 논문에서도 Fishr 를 federated learning setting 으로 사용하고, federated clients (ε) 을 알고 있다고 가정한다.

Fishr+Inter-Geo: Fishr and Inter-silo Weighted Geometric Mean

- Fishr 를 federated 방식으로 사용하기 위해서, gradient covariance matrix 를 정의한다.

$$C_e = \frac{1}{n_e}\sum_{i=1}^{n_e}\left(\nabla F_e^i - \nabla\bar{F}_e\right)\left(\nabla F_e^i - \nabla\bar{F}_e\right)^\top$$

$$L_{Fishr} = \frac{1}{\mathcal{E}}\sum_{e\in\mathcal{E}}\|C_e - \bar{C}\|^2$$



Client 1: [∇F1]1,[∇F1]2,[∇F1]3, … + ∇F1 mean
Client 2: [∇F2]1,[∇F2]2,[∇F2]3, … + ∇F2 mean

=> Client covariance: C1, C2, …
=> L fisfr

[∇Fε]k = Gε

G의 평균 = { [∇F1]1,[∇F2]1,[∇F3]1,[∇F4]1,... }

FedILC: Weighted Geometric Mean and Invariant Gradient Covariance for Federated Learning on Non-IID Data

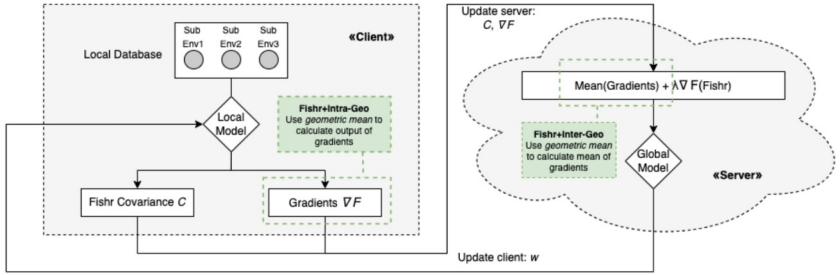


Figure 1: Server-client communication of Fishr+Inter-Geo and Fishr+Intra-Geo in the federated learning settings.

Fishr : Another gradient-based regularization method for domain generalization. Fishr 는 학습 동안의 domain 수를 알고 있다는 가정이 있다.
본 논문에서도 Fishr 를 federated learning setting 으로 사용하고, federated clients (ε) 을 알고 있다고 가정한다.

Fishr+Inter-Geo: Fishr and Inter-silo Weighted Geometric Mean

- Fishr 를 federated 방식으로 사용하기 위해서, gradient covariance matrix 를 정의한다.

$$C_e = \frac{1}{n_e} \sum_{i=1}^{n_e} \left(\nabla F_e^i - \nabla \bar{F}_e \right) \left(\nabla F_e^i - \nabla \bar{F}_e \right)^\top$$

Then we compute the Fishr loss on the centralized server, where the mean covariance matrix is $\bar{C} = \frac{1}{\mathcal{E}} \sum_{e \in \mathcal{E}} C_e$:

$$L_{Fishr} = \frac{1}{\mathcal{E}} \sum_{e \in \mathcal{E}} \| C_e - \bar{C} \|^2$$

$$w_{t+1} = w_t - \eta \left(\nabla \hat{} F_t + \lambda \nabla L_{Fishr_t} \right)$$

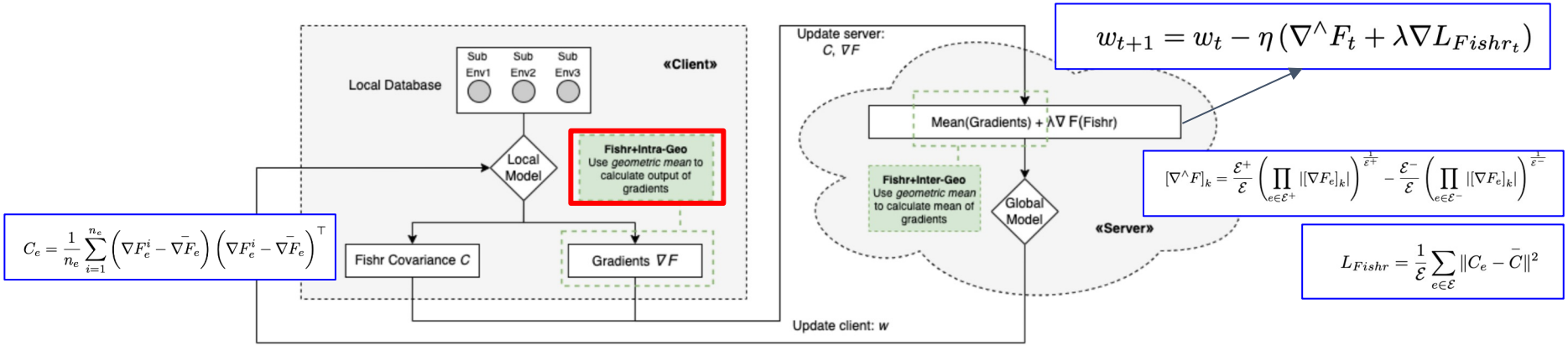


Figure 1: Server-client communication of Fishr+Inter-Geo and Fishr+Intra-Geo in the federated learning settings.

Fishr+Intra-Geo: Fishr and Intra-silo Weighted Geometric Mean

- 이론적으로 하나의 client 의 데이터는 통계적으로 균질하지만, **실제 데이터는 하나의 클라이언트 내부에서도 non-i.i.d 하다.**
  e.g. 하나의 병원 (하나의 클라이언트)에서 clinical data 은 환자마다 다르다. 이렇게 client 까지 고려한 method 는 이전 연구에는 없었다.

- **이를 위해 intra-silo weighted geometric mean 을 수행하는데, 이는 학습 데이터 각각을 independent environment 로 간주한다.**

- Compute gradient covariance in the same federated client.
- Each client collaboratively uploads the computed gradients and covariance information to the orchestration of the central server to update the global model's parameters.

Theoretical Analysis

- Inconsistency Score
  - Inconsistency Score 는 client A 와 client B 가 same curvature in Hessians 를 가질 때 작다.
    즉, client A 와 client B 의 distribution 이 유사할 수록 작음.
  - 제안한 methods (including both Fishr+Inter-Geo, Fishr+Intra-Geo) 는 Inconsistency Score I 를 성공적으로 감소시켰다.
    즉, 다른 환경의 consistencies 를 촉진한다.

$$I\left(\theta^{*}\right)=max\left\{\underset{|L_A(\theta)-L_A(\theta^{*})|\leq\epsilon}{max}\left|L_B\left(\theta\right)-L_A\left(\theta^{*}\right)\right|,\underset{|L_B(\theta)-L_B(\theta^{*})\leq\epsilon}{max}\left|L_A\left(\theta\right)-L_B\left(\theta^{*}\right)\right|\right\}$$

Experiments

- Dataset
  - Color-MNIST
  - Rotated-CiFAR10
  - real-world eICU

- Compare performance of six different methodologies
  - Baseline methods (4)
  - Proposed methods (2)

Baseline methods:

- FedSGD: the baseline federated learning algorithm which performs the arithmetic mean of the uploaded clients' gradients.
- FedCurv: the state-of-the-art baseline federated learning algorithm which integrates with vanilla Fishr regularization.
- Geometric: the federated learning algorithm which performs the weighted geometric mean of the uploaded clients' gradients.
- Fishr+Intra-Arith: Fishr and intra-silo arithmetic mean algorithm for comparison purpose.

Proposed methods:

- Fishr+Inter-Geo: Fishr and inter-silo weighted geometric mean algorithm.
- Fishr+Intra-Geo: Fishr and intra-silo weighted geometric mean algorithm.

Experiments

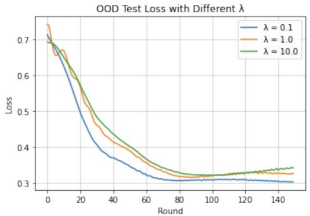


Figure 2: The OOD test loss with different Fishr regularizer λ on the eICU dataset.

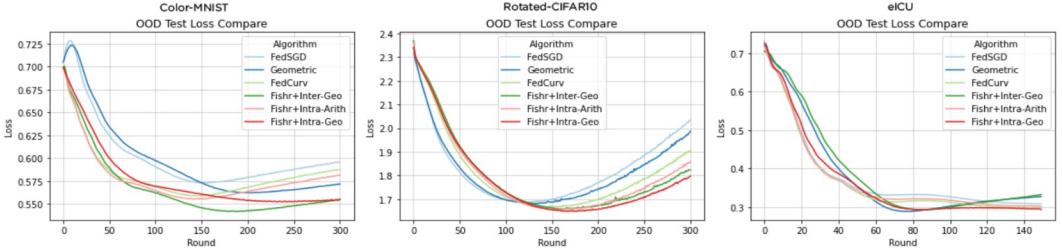


Figure 3: The OOD test loss comparison among different algorithms on the Color-MNIST, Rotated-CIFAR10, and eICU dataset.

Table 2: The OOD test loss comparison among different algorithms on the Color-MNIST, Rotated-CIFAR10, and eICU dataset.

	Color-MNIST	Rotated-CIFAR10	eICU
FedSGD	0.566±0.009	1.681±0.010	0.302±0.017
Geometric	0.562±0.006	1.681±0.006	**0.289±0.009**
FedCurv	0.559±0.003	1.666±0.024	0.298±0.009
Fishr+Inter-Geo	**0.542±0.006**	1.658±0.004	0.293±0.009
Fishr+Intra-Arith	0.555±0.004	1.655±0.007	0.302±0.009
Fishr+Intra-Geo	0.552±0.005	**1.648±0.006**	**0.289±0.008**

Table 3: The variance of the accuracy distribution and KL-divergence between the normalized accuracy vector and the uniform distribution (which can be directly translated to the entropy of accuracy) on the Rotated-CIFAR10 dataset when the OOD test loss is lowest.

	Variance (x1000)	Entropy (x10)
FedSGD	0.809±0.251	21.948±0.007
FedCurv	**0.606±0.093**	21.955±0.003
Fishr+Inter-Geo	0.687±0.234	21.953±0.007
Fishr+Intra-Geo	0.611±0.120	**21.956±0.003**

Table 4: The AUCROC and AUCPR comparison among different algorithms on the Color-MNIST, Rotated-CIFAR10, and eICU dataset when the OOD test loss is lowest.

(a) AUCROC

	Color-MNIST	eICU
FedSGD	0.780±0.001	0.556±0.004
Geometric	0.781±0.003	0.575±0.004
FedCurv	0.780±0.003	0.556±0.005
Fishr+Inter-Geo	**0.789±0.002**	**0.577±0.005**
Fishr+Intra-Arith	0.782±0.005	0.553±0.002
Fishr+Intra-Geo	0.778±0.002	0.567±0.004

(b) AUCPR

	Color-MNIST	eICU
FedSGD	0.837±0.002	0.209±0.010
Geometric	0.838±0.004	0.214±0.008
FedCurv	0.838±0.004	0.217±0.018
Fishr+Inter-Geo	**0.845±0.002**	0.218±0.009
Fishr+Intra-Arith	0.839±0.004	0.216±0.016
Fishr+Intra-Geo	0.836±0.001	**0.221±0.013**

MultiMAE: Multi-modal Multi task Masked Autoencoders

CVPR 2022
https://arxiv.org/pdf/2204.01678.pdf

MultiMAE: Multi-modal Multi-task Masked Autoencoders

Roman Bachmann* David Mizrahi* Andrei Atanov Amir Zamir
Swiss Federal Institute of Technology Lausanne (EPFL)

https://multimae.epfl.ch

- Pre-training strategy “Multi-modal Multi-task Masked Autoencoders” (MultiMAE).
  1. optionally accept additional modalites information in the input (Multi-modal)
  2. training objective includes predicting multiple outputs (Multi-task)

- 이미지 path와 input modalities의 masking을 사용하여 MultiMAE training
  ⇒ cross-modality predictive coding

- Multiple transfer tasks (Image classification, semantic segmentation, depth estimation)
- dataset (ImageNet, ADE20K, Taskonomy, Hypersim, NYCv2)
- classification, semantic segmentation, dense regression tasks
- Demo: https://multimae.epfl.ch/ / Git: https://github.com/EPFL-VILAB/MultiMAE

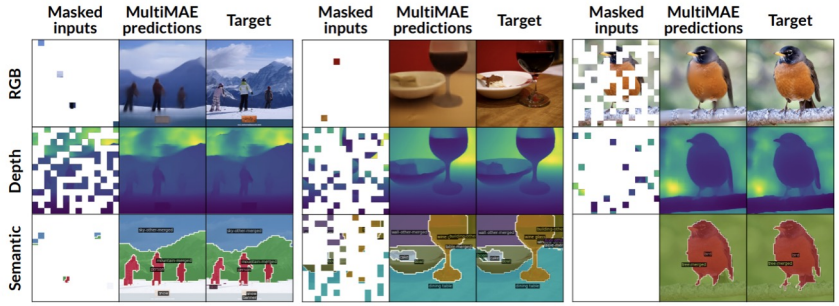


Figure 1. **MultiMAE pre-training objective.** We randomly select 1/6 of all 16×16 image patches from multiple modalities and learn to reconstruct the remaining 5/6 masked patches from them. The figure shows validation examples from ImageNet, where masked inputs (left), predictions (middle), and non-masked images (right) for RGB (top), depth (middle), and semantic segmentation (bottom) are provided. Since we do not compute a loss on non-masked patches, we overlay the input patches on the predictions.

MultiMAE: Multi-modal Multi task Masked Autoencoders

CVPR 2022
https://arxiv.org/pdf/2204.01678.pdf

- **Related Work**

1. **Masked Image Prediction**

    a. Masked Auto Encoder (MAE)

1. **Multi-modal learning**

    a. more flexible architecture and perform masked autoencoding to learn cross-modal predictive coding among optional inputs

1. **Multi-task learning**

    a. multiple output domains from a single input

    b. A common approach for multi-task learning is to use a single encoder to learn a shared representation followed by multiple task-specific decoders

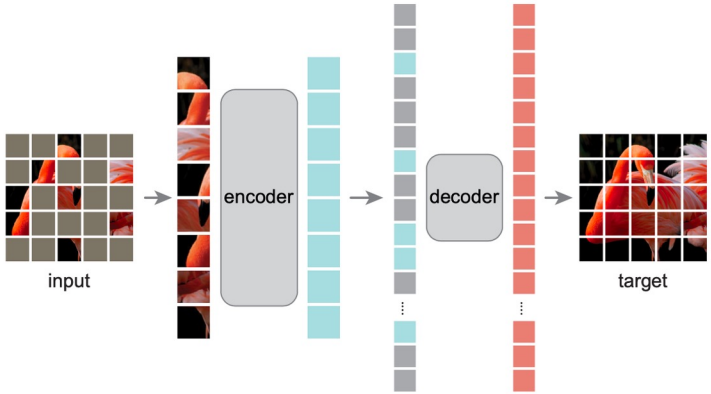
    c. Input and Output along with masking

MultiMAE: Multi-modal Multi task Masked Autoencoders

- **Related Work | Masked Auto Encoder (MAE)**

1. ‘MAE’ 왈, GPT, BERT와 같이 NLP에서는 masked autoencoding이 좋은 성능을 보여주는데 Vision은 그렇지 못함
   1. language와 vision의 architectural gap
   2. Information density is different between language and vision. ⇒ vision의 경우 spatial redundancy하므로 주변 픽셀로 부터 여럽지 않게 정보를 얻어 prediction할수 있음.
   3. decoder 부분에서 language와 vision의 역할이 다르다. ⇒ letent representation으로 부터 픽셀값 (low sementic information) 에 대한 semantic level을 결정할 수 있도록 vision decoder design이 잘 이루어져야 함.

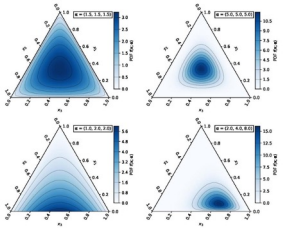
1. Approach
   a. Autoencoder의 컨셉으로 접근 (encoder ⇒ latent vector ⇒ decoder(reconstruct) 구조)
   b. asymmetric 디자인, (encoder 에는 masking 되지 않은 토큰만 사용해서 decoder보다 모델을 가
   c. Reconstruction target = masked 패치의 픽셀값 예측
      i. MAE loss function, 원본과 이미지와의 픽셀값 차이를 쉽게 구할 수 있는 MSE를 사용
      ii. 이때 masked patch 영역에 대해서만 loss calculation



1. Pseudo Implem.
   a. 먼저 이미지를 서로 겹치지 않게 패치로 나누어 tokinization (linear projection + embedding)
   b. token ⇒ random shuffle ⇒ masking ratio (75%) 패치 masking
   c. unmasked patch만 encoding
   d. encoded patch + mask token 다시 원래 위치로 복귀
   e. Reconstruction.

MultiMAE: Multi-modal Multi task Masked Autoencoders

Random sampling
(symmetric-Dirichlet distribution Sampling step ⇒ alpha)
total 96 ea - (⅓ of each modalites tokens)



three of multiple modalities

low computational complexity
(only encode no masked token)

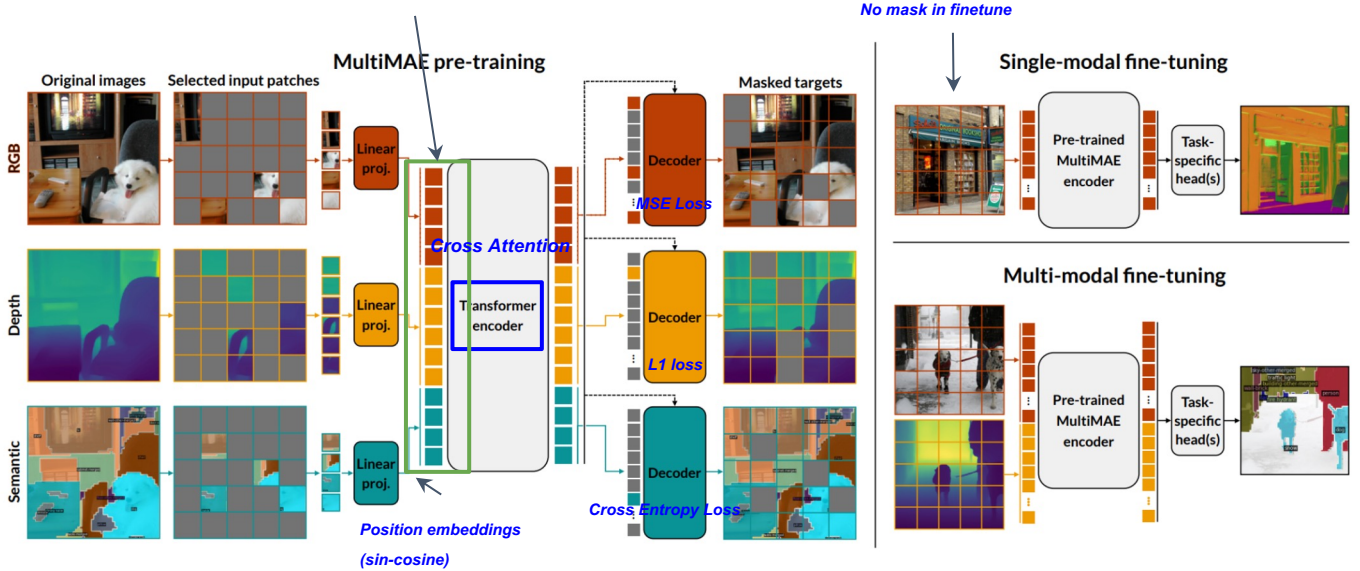


Figure 2. **(Left) MultiMAE pre-training**: A small subset of randomly sampled patches from multiple modalities (e.g., RGB, depth, and semantic segmentation) is linearly projected to tokens with a fixed dimension and encoded using a Transformer. Task-specific decoders reconstruct the masked-out patches by first performing a cross-attention step from queries to the encoded tokens, followed by a shallow Transformer. The queries consist of mask tokens (in gray), with the task-specific encoded tokens added at their respective positions. **(Right) Fine-tuning**: By pre-training on multiple modalities, MultiMAE lends itself to fine-tuning on single-modal and multi-modal downstream tasks. No masking is performed at transfer time.

Suppl.

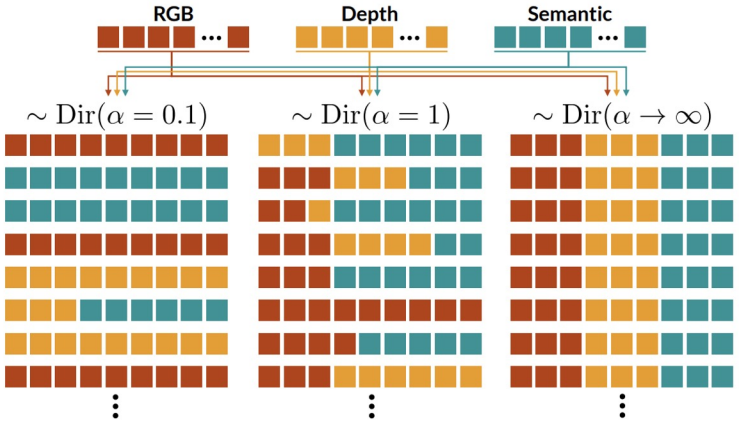


Figure 8. **Multi-modal mask sampling**: We sample the proportion of tokens per modality using a symmetric Dirichlet distribution $\mathrm{Dir}(\alpha)$ with concentration parameter α. We illustrate here the sampling behavior for different choices of α values when selecting nine tokens from three modalities. Each row represents one sample of tokens. With small α, most tokens will be sampled from single modalities, while large α values result in equal representation of each modality. Setting $\alpha = 1$ is equivalent to sampling uniformly over the support and results in a more diverse sampling behavior.

α	ImageNet-1K [23]	ADE20K [102]
0.2	82.7	44.6
0.5	82.5	44.8
1.0	82.8	**45.1**
∞	**82.9**	42.9

Table 10. **Comparison of mask sampling strategies.** We report RGB-only transfers to ImageNet-1K [23] classification and ADE20K [102] semantic segmentation using MultiMAEs pre-trained with different Dirichlet concentration parameter α. All models were trained for 400 epochs and do *not* use the additional per-patch-standardized RGB decoder (see Sec. 3.4). By $\alpha = \infty$ we denote always sampling an equal number of visible tokens for each tasks.

MultiMAE: Multi-modal Multi task Masked Autoencoders

- **details setup**

1. **Models - ViT-B 16 - 400/1600 epoch pretrained per experimental setup**
2. **Pseudo labeled multi-task**
    a. DPT-Hybrid (trained on Omnidata)
    b. Mask2Former (trained Swin-S backbone on COCO)
3. **three of downstream tasks**
    a. classification (ImageNet 1k)
    b. semantic segmentation (ADE 20K, NYUv2)
    c. dense regression tasks (NYUv2)
        i. how our models transfer to geometric tasks
        ⇒ such as surface normals, depth and reshading, as well as tasks extracted from RGB images, such as keypoint or edge detection

- **Experiments**

1. transfer results for the case where the only available input modality is RGB (Sec. 4.2).
2. MultiMAE can significantly improve downstream performance if other modalities like depth are either available as ground truth (sensor),
    or can be cheaply pseudo labeled (Sec. 4.3).
3. We follow up with an ablation on the influence of pre-training tasks on the downstream performance (Sec. 4.4),
4. and finally we visually demonstrate that MultiMAE integrates and exchanges information across modalities (Sec. 4.5).

4.2 Transfers with RGB-only | 4.3 Transfers with multiple modalities

Method	IN-1K (C)	ADE20K (S)	Hypersim (S)	NYUv2 (S)	NYUv2 (D)
Supervised [81]	81.8	45.8	33.9	50.1	80.7
DINO [12]	83.1	44.6	32.5	47.9	81.3
MoCo-v3 [17]	82.8	43.7	31.7	46.6	80.9
MAE [35]	**83.3**	**46.2**	36.5	50.8	85.1
MultiMAE	**83.3**	**46.2**	**37.0**	**52.0**	**86.4**

Table 1. **Fine-tuning with RGB-only.** We report the top-1 accuracy (↑) on ImageNet-1K (IN-1K) [23] classification (C), mIoU (↑) on ADE20K [102] , Hypersim [68] , and NYUv2 [73] semantic segmentation (S), as well as δ_1 accuracy (↑) on NYUv2 depth (D). Text in **bold** and underline indicates the first and second-best results, respectively. All methods are pre-trained on ImageNet-1K (with pseudo labels for MultiMAE).

	Hypersim (S)			NYUv2 (S)		
Method	RGB	D	RGB-D	RGB	D	RGB-D
MAE	36.5	32.5	36.9	50.8	23.4	49.3
MultiMAE	**37.0**	**38.5**	**47.6**	**52.0**	**41.4**	**56.0**

Table 2. **Fine-tuning with RGB and ground truth depth.** We report semantic segmentation transfer results from combinations of RGB and depth, measured in mIoU (↑). MultiMAE can effectively leverage additional modalities such as depth, while MAE cannot. Text in gray indicates a modality that the model was not pre-trained on.

	ADE20K (S)					Hypersim (S)					NYUv2 (S)				
Method	RGB	pD	RGB-pD	RGB-pS	RGB-pD-pS	RGB	pD	RGB-pD	RGB-pS	RGB-pD-pS	RGB	pD	RGB-pD	RGB-pS	RGB-pD-pS
MAE	**46.2**	20.0	46.3	**46.2**	46.3	36.5	21.0	36.9	37.7	37.3	50.1	23.8	49.1	50.1	49.3
MultiMAE	**46.2**	**34.4**	**46.8**	45.7	**47.1**	**37.0**	**30.6**	**37.9**	**38.4**	**40.1**	**52.0**	**39.9**	**53.6**	**53.5**	**54.0**

Table 3. **Fine-tuning with RGB and pseudo labels.** Semantic segmentation transfer results using *pseudo labeled* depth and semantic segmentation maps, measured in mIoU (↑). MultiMAE benefits much more than MAE from pseudo labeled modalities as input. Text in gray indicates a modality that the model was not pre-trained on.

4.4. Influence of pre-training task choices and masking on transfer performance

"To summarize, the results in this section show that **using all modalities to pre-train a MultiMAE results in a more generalist model** that does well at transferring to a range of downstream tasks."

Method	IN-1K (C)	NYUv2 (S)	NYUv2 (D)	Taskonomy (D)
MAE (D2)	83.0	44.0	81.3	3.8
RGB-D	82.8	45.8	83.3	2.1
RGB-S	**83.2**	**51.6**	**85.5**	2.6
RGB-D-S	83.0	50.6	85.4	**1.5**

(a) **Impact of additional modalities.** Transfer results of several MultiMAE models pre-trained on different input modalities / target tasks, compared against MAE (single-modal baseline). D2 = MAE pre-trained with a decoder of depth 2 and width 256, comparable in size to the decoders of MultiMAE

Method	IN-1K (C)	NYUv2 (S)	NYUv2 (D)	Taskonomy (D)
RGB→D	82.7	44.0	**87.1**	1.6
RGB→S	82.5	46.8	82.9	4.0
RGB→D-S	82.8	48.6	84.6	2.9
MultiMAE	**83.0**	**50.6**	85.4	**1.5**

(b) **Comparison to non-masked pre-training**. We compare standard single-task and multi-task baselines pre-trained using *non-masked* RGB inputs against the RGB-D-S MultiMAE. The RGB→D-S model is conceptually similar to MuST using depth and semantic segmentation as target tasks.

Table 4. **Ablation experiments.** We study the impact of additional modalities in Table 4a, and compare MultiMAE to non-masked pre-training in Table 4b. All models are pre-trained for 400 epochs. We report the top-1 accuracy (↑) on ImageNet-1K (IN-1K) [23] classification (C), mIoU (↑) on NYUv2 [73] semantic segmentation (S), δ_1 accuracy (↑) on NYUv2 depth (D) and avg. rank ↓ on Taskonomy [99]. While some specialized pre-trained models perform better at certain downstream tasks, they perform poorly at others. MultiMAE pre-trained with RGB, depth and semantic segmentation is a more generalist model that does well at transferring to a range of downstream tasks.

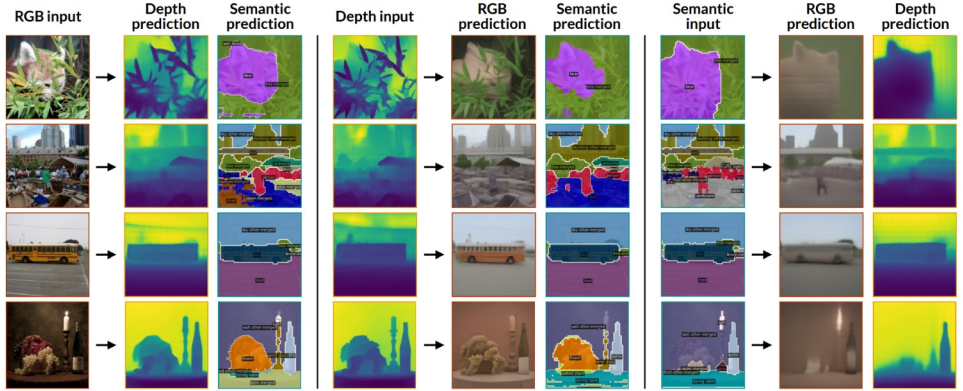


Figure 4. **Single-modal predictions.** We visualize MultiMAE cross-modal predictions on ImageNet-1K validation images. Only a single, full modality is used as input. The predictions remain plausible despite the absence of input patches from other modalities.

4.5. Cross-modal exchange of information

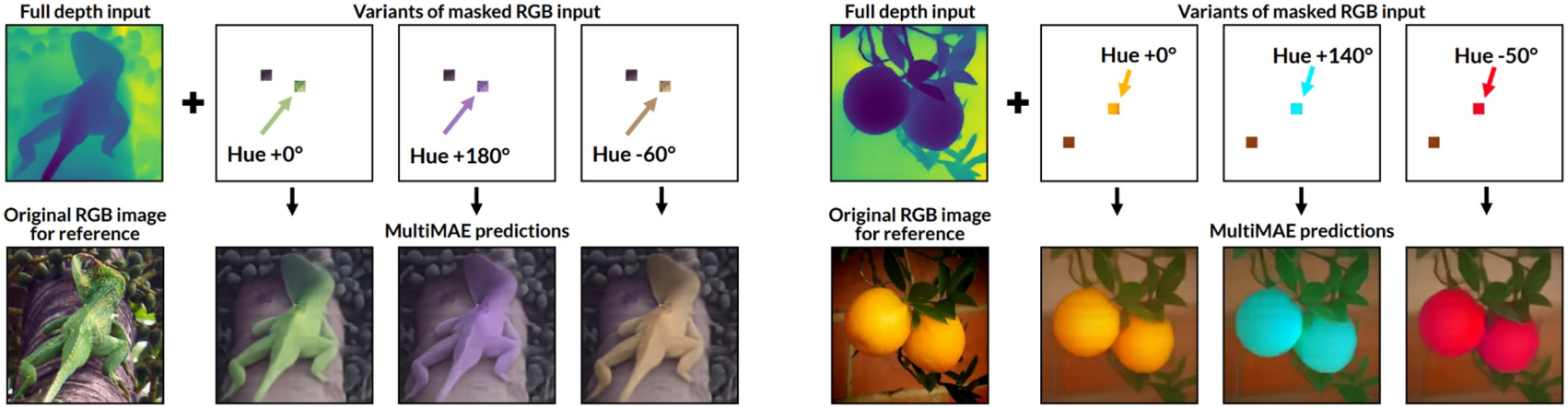


Figure 5. **Demonstration of cross-modal interaction.** The input is the full depth, only two RGB patches, and *no* semantic segmentation. By editing the hue of a single input patch, the color of the lizard (left) and oranges (right) changes, while keeping the background constant. More interactive examples are available on our website.

4.5. Cross-modal exchange of information

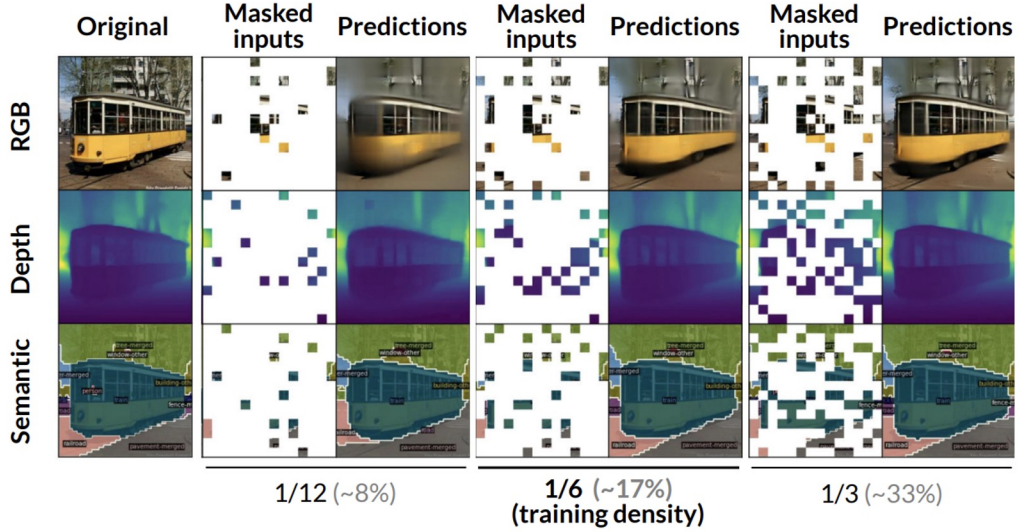


Figure 6. **MultiMAE predictions for a varying number of visible patches.** The predictions are plausible even when given half the number of patches seen during pre-training, and the reconstruction quality improves as the number of visible patches increases. An interactive visualization is available on our website.

DoubleMatch: Improving Semi-Supervised Learning with Self-Supervision

Erik Wallin[1,2], Lennart Svensson[2], Fredrik Kahl[2], Lars Hammarstrand[2]
[1]Saab AB, [2]Chalmers University of Technology
{walline,lennart.svensson,fredrik.kahl,lars.hammarstrand}@chalmers.se

ICPR2022
paper:https://arxiv.org/pdf/2205.05575
code: https://github.com/walline/doublematch

semi- supervised learning (SSL) is a family of methods:

- which in addition to a labeled training set,
- also use a sizable collection of unlabeled data for fitting a model.
- Most of the recent successful SSL methods are **based on pseudo-labeling approaches**: letting confident model predictions act as training labels.

semi supervised training에서 **labeled** training set을 **without label**로 수정

- == self supervised learning
- class predict (X), feature representation learn (O)
- **unlabeld data 완전히 활용**
- semi supervised learning보다 나을지도…?

This paper proposes **new SSL algorithm, DoubleMatch,**

- combines the **pseudo-labeling technique with a self-supervised loss,**
- enabling the model to utilize all unlabeled data in the training process.

This paper show that this inefficient and incomplete use of unlabeled data

- unnecessarily long training times
- for some datasets, reductions in classification accuracy.

DoubleMatch: Improving Semi-Supervised Learning with Self-Supervision

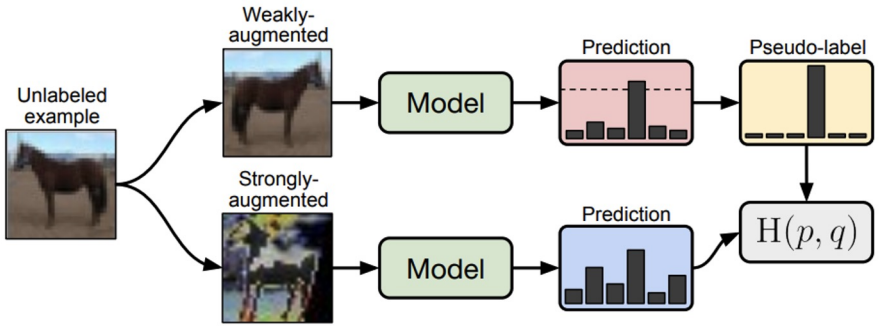
UDA(Unsupervised Data Augmentation) and **FixMatch**:
recently gained wide recognition because of their simple yet powerful frameworks
for combining consistency regularization and pseudo-labeling in semi-supervised learning.

- **impressive classification accuracy of 94.93%** on CIFAR-10 [8] using only 250 labeled images for training
- **drawback of only enforcing consistency regularization on unlabeled data** with confident model predictions while harder data samples are essentially discarded.

=> model **only uses a smaller part of unlabeled data during training**

This paper show that this inefficient and incomplete use of unlabeled data

- unnecessarily long training times
- for some datasets, reductions in classification accuracy.



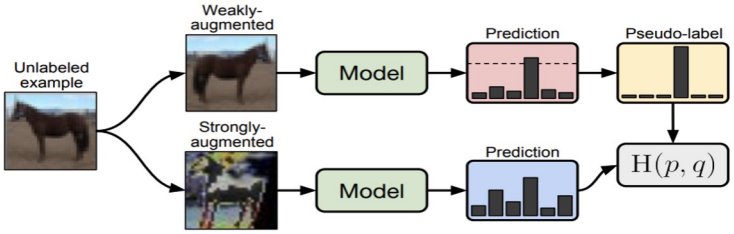
FixMatch: https://arxiv.org/ftp/arxiv/papers/2001/2001.07685.pdf

DoubleMatch: Improving Semi-Supervised Learning with Self-Supervision

DoubleMatch: extension of **FixMatch**:
how DoubleMatch unifies the typical setups for semi- and self- supervised learning
by operating both in the label and feature space for unlabeled data.

- **In the label space:**
  - as with FixMatch for confident data,
  - the model is evaluated based on predicted class distributions (*pseudo-label loss*).
- **in the feature space:**
  - the model is assessed
    based on the similarity of the predicted feature representations
    (*self-supervised loss*).

we suggest adding a self-supervised feature loss to the FixMatch framework
by enforcing **consistency regularization on *all* unlabeled data in the feature space.**
**consistency regularization: enforced on feature vectors from penultimate layer of classification network



FixMatch:
https://arxiv.org/ftp/arxiv/papers/2001/2001.07685.pdf

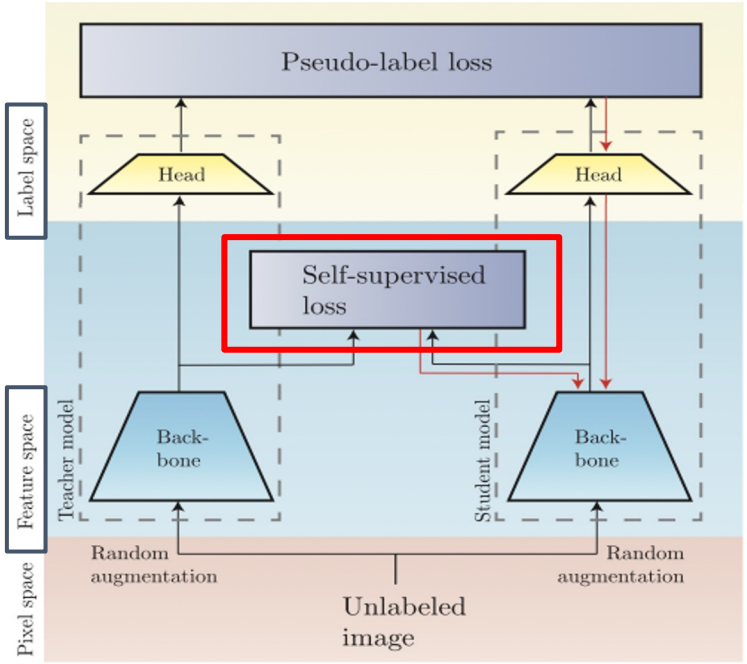


Fig. 1. A unification of semi- and self-supervised frameworks. While many existing methods for semi-supervised learning operate in the label space, DoubleMatch operates in both the label and feature spaces for a more efficient use of unlabeled data. Red arrows mark the flow of gradients.

DoubleMatch: Improving Semi-Supervised Learning with Self-Supervision

DoubleMatch: extension of **FixMatch**:

Many existing methods for semi-supervised learning have two central terms in their loss functions:

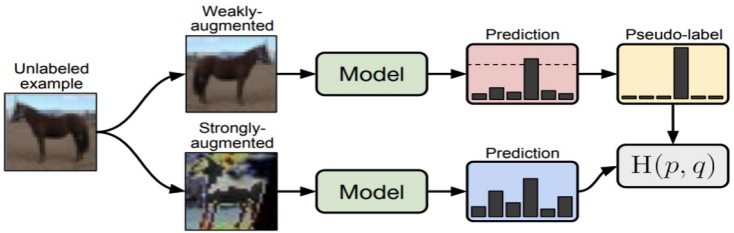
- **a supervised loss term for labeled data**
- **a pseudo-label loss term for unlabeled data**
- a third, **self-supervised loss term**, to fully utilize unlabeled data for faster training and increased accuracy.

This term acts as a natural extension with minimal computational overhead to methods that utilize consistency regularization through data augmentation.

$$l = l_l + l_p + w_s l_s,$$

supervised loss in FixMatch is the standard cross-entropy loss given by

$$l_l = \frac{1}{B}\sum_{i=1}^{B} H(y_i, p_i),$$



FixMatch:
https://arxiv.org/ftp/arxiv/papers/2001/2001.07685.pdf

supervised loss is supplemented with a pseudo-label loss on unlabeled data:

$$l_p = \frac{1}{\mu B}\sum_{i=1}^{\mu B} \mathbb{1}\{\max(w_i) > \tau\} H(\text{argmax}(w_i), q_i). \qquad l_s = -\frac{1}{\mu B}\sum_{i=1}^{\mu B} \frac{h(v_i)\cdot z_i}{\|h(v_i)\|\|z_i\|} = -\frac{1}{\mu B}\sum_{i=1}^{\mu B} \cos(h(v_i), z_i).$$

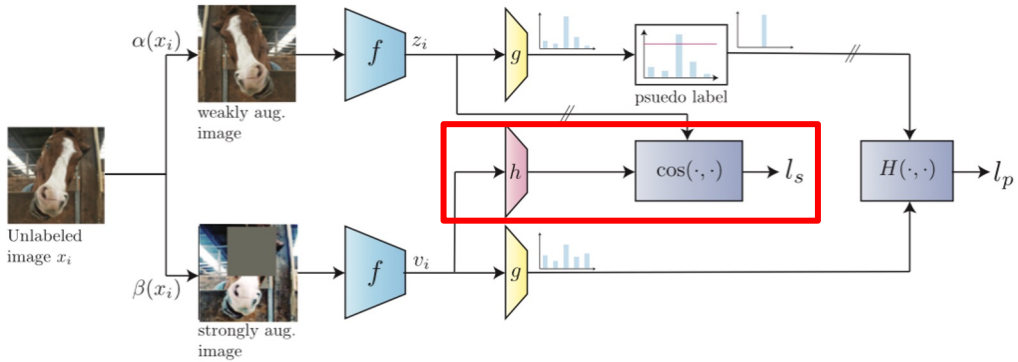


Fig. 2. Graph showing loss evaluation for unlabeled images. Double slash marks a stop-gradient operation.

$$l = l_l + l_p + w_s l_s,$$

$$l_l = \frac{1}{B}\sum_{i=1}^{B} H(y_i, p_i),$$

$$l_p = \frac{1}{\mu B}\sum_{i=1}^{\mu B} \mathbb{1}\{\max(w_i) > \tau\} H(\text{argmax}(w_i), q_i).$$

$$l_s = -\frac{1}{\mu B}\sum_{i=1}^{\mu B} \frac{h(v_i)\cdot z_i}{\|h(v_i)\|\|z_i\|} = -\frac{1}{\mu B}\sum_{i=1}^{\mu B} \cos(h(v_i), z_i).$$

τ:sets the confidence threshold for assigning pseudo-labels.
wi :the predictions for a *weak*
qi : the predictions for a *strong* augmentation

μ determines the ratio of labeled to unlabeled data in a batch

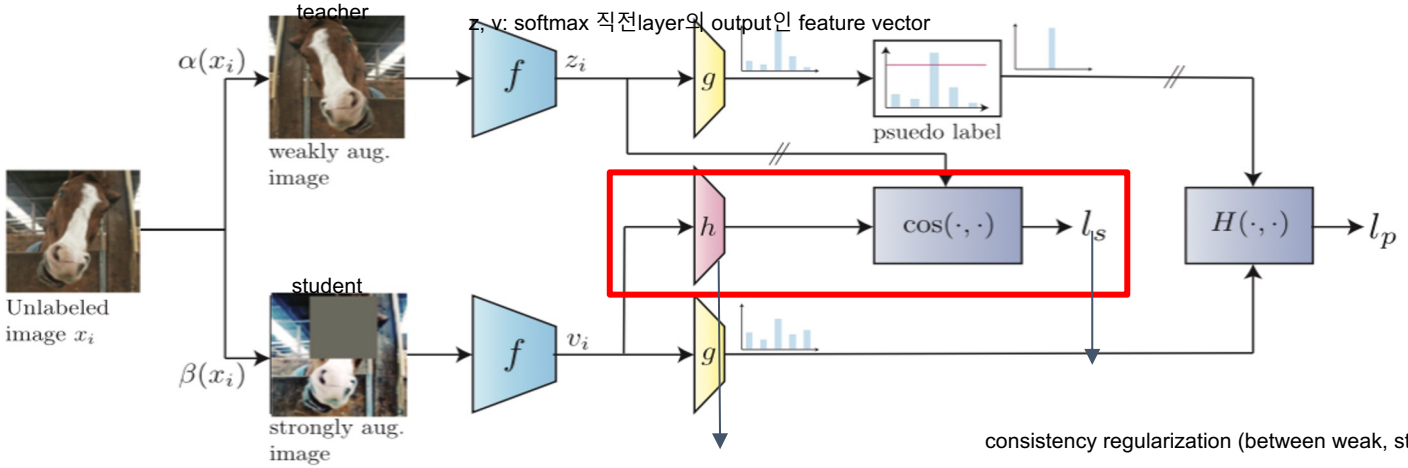


Fig. 2. Graph showing loss evaluation for unlabeled images. Double slash marks a stop-gradient operation.

DoubleMatch: Improving Semi-Supervised Learning with Self-Supervision

DoubleMatch: extension of **FixMatch**:

A. Data augmentation

- the weak augmentation is a **horizontal flip** with probability 0.5 followed by a random translation with maximum distance 0.125 of the image height.
- the strong augmentation stacks the weak augmentation, **CTA [16], and Cutout [18].**

B. Optimizer and regularization

For optimization, we stay close to the FixMatch settings and use SGD with Nesterov momentum [40]. The learning rate, η, is set to follow a cosine scheme [41] given by

$$\eta = \eta_0 \cos\left(\gamma \frac{\pi k}{2K}\right) \tag{5}$$

where η_0 is the initial learning rate, k is the current training step and K is the total number of training steps. We define one training step as one gradient update in the SGD optimization. The decay rate is determined by the hyperparameter $\gamma \in (0, 1)$.

Contrary to FixMatch which uses a constant γ, we suggest tuning γ for different datasets in order to minimize overfitting.

Finally, we add weight-decay regularization to the loss as

$$l_w = w_d \frac{1}{2}\left(\|\theta_f\|^2 + \|\theta_g\|^2 + \|\theta_h\|^2\right) \tag{6}$$

where θ_f, θ_g and θ_h are the vectors of parameters in the backbone, prediction head and projection head, respectively, and w_d is a hyperparameter controlling the weight of this regularization term. The weight-decay is identical to FixMatch with the exception that DoubleMatch has the additional parameters from the projection head, θ_h.

Algorithm 1 DoubleMatch algorithm

Require: Strong augmentation β, weak augmentation α, labeled batch $\{(x_1, y_1), \cdots, (x_B, y_B)\}$, unlabeled batch $\{\tilde{x}_1, \cdots, \tilde{x}_{\mu B}\}$, unsupervised loss weight w_s, weight decay parameter w_d, confidence threshold τ, backbone model f, prediction layer g, projection layer h

▷ Cross-entropy loss for (weakly augmented) labeled data
for $i = 1, \cdots, B$ **do**
  $p_i = g \circ f(\alpha(x_i))$
end for
$l_l = \frac{1}{B} \sum_{i=1}^{B} H(y_i, p_i)$

▷ Predictions on unlabeled data
for $i = 1, \cdots, \mu B$ **do**
  $z_i = f(\alpha(\tilde{x}_i))$ ▷ Weak augmentation
  $v_i = f(\beta(\tilde{x}_i))$ ▷ Strong augmentation
  $q_i = g(v_i)$
  $w_i = \texttt{stopgrad}(g(z_i))$
end for
▷ Self-supervised cosine similarity
$l_s = -\frac{1}{\mu B} \sum_{i=1}^{\mu B} \cos(h(v_i), \texttt{stopgrad}(z_i))$
▷ Cross-entropy with pseudo-labels
$l_p = \frac{1}{\mu B} \sum_{i=1}^{\mu B} \mathbb{1}\{\max(w_i) > \tau\} H(\text{argmax}(w_i), q_i)$

return $l_l + l_p + w_s l_s + w_d \frac{1}{2}\left(\|\theta_f\|^2 + \|\theta_g\|^2 + \|\theta_h\|^2\right)$

DoubleMatch: Improving Semi-Supervised Learning with Self-Supervision

Experiments

A. Classification results

- *CIFAR-10 and SVHN:*
  - both consist of color images of size 32×32.
  - Both datasets contain ten classes where CIFAR-10 has classes such as dog, horse and ship while the classes in SVHN are the ten digits.
  - img - set
    - CIFAR-10 has a test set of 10,000 images and a training set of 50,000 images.
    - SVHN has 26,032 images for testing and 73,257 for training.
  - For these datasets we use a **Wide ResNet- 28-2** [44] **with 1.5M parameters.**
  - This architecture makes the dimension of our feature vectors d = 128.
  - Even though we obtain competitive results on many of the splits, we do **perform worse than SOTA, especially in the very-low label regime**

The results are shown in Table I where we present mean and standard deviation of the error rate on the test set using five different data folds.

TABLE I

ERROR RATES ON DIFFERENT DATASETS USING DIFFERENT SIZES FOR THE LABELED TRAINING SET. DOUBLEMATCH ACHIEVES STATE-OF-THE-ART RESULTS ON MANY COMBINATIONS.

	CIFAR-10			CIFAR-100			SVHN			STL-10
Method	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels	40 labels	250 labels	1000 labels	1000 labels
MixMatch [7], [24]	47.54±11.50	11.05±0.86	6.42±0.10	67.61±1.32	39.94±0.37	28.31±0.33	42.55±14.53	3.98±0.23	3.50±0.28	10.41±0.61
UDA [6], [7]	29.05±5.93	8.82±1.08	4.88±0.18	59.28±0.88	33.13±0.22	24.50±0.25	52.63±20.51	5.69±2.76	2.46±0.24	7.66±0.56
ReMixMatch [7], [16]	19.10±9.64	5.44±0.05	4.72±0.13	44.28±2.06	27.43±0.31	23.03±0.56	3.34±0.20	2.92±0.48	2.65±0.08	5.23±0.45
FixMatch (CTA) [7]	11.39±3.35	5.07±0.33	4.31±0.15	49.95±3.01	28.64±0.24	23.18±0.11	7.65±7.65	2.64±0.64	2.36±0.19	5.17±0.63
DP-SSL [32]	**6.54±0.98**	**4.78±0.26**	4.23±0.20	43.17±1.29	28.00±0.79	22.24±0.31	**2.98±0.86**	**2.16±0.36**	**1.99±0.18**	4.97±0.42
Dash (CTA) [31]	9.16±4.31	**4.78±0.12**	**4.13±0.06**	44.83±1.36	27.85±0.19	22.77±0.21	3.14±1.60	2.38±0.29	2.14±0.09	**3.96±0.25**
DoubleMatch (last 20)	14.02±5.71	5.72±0.51	4.83±0.17	**42.61±1.15**	**27.48±0.19**	**21.69±0.26**	16.50±13.73	2.58±0.53	2.25±0.09	4.46±0.20
DoubleMatch (min)	13.59±5.60	5.56±0.42	4.65±0.17	**41.83±1.22**	**27.07±0.26**	**21.22±0.17**	15.37±11.81	2.37±0.35	2.10±0.07	4.35±0.20

Experiments

A. Classification results

- *CIFAR-100:*
  - CIFAR-100 is similar to CIFAR-10 in that it consists of color images of size 32 × 32 with training and test sets of size 50,000 and 10,000 respectively.
  - CIFAR-100 contains 100 classes, making it a much more challenging classification problem.
  - For this dataset, we use the **Wide ResNet-28-8** architecture **with 24M parameters**, resulting in d = 512.
  - On this dataset, we **achieve SOTA results across all splits,**
  - not only beating FixMatch and ReMixMatch, but also the more recent methods DP-SSL and Dash.
- *STL-10:*
  - STL-10 comprises color images of size 96 × 96 belonging to ten classes.
  - It has a labeled training set of 5,000 images and a unlabeled training set of 100,000 images.
  - we use a **Wide ResNet-37-2 with 6M parameters**, making d = 256.
  - On this dataset, we **achieve a very competitive error rate**, surpassed only by the result reported by Dash.

The results are shown in Table I where we present mean and standard deviation of the error rate on the test set using five different data folds.

TABLE I

ERROR RATES ON DIFFERENT DATASETS USING DIFFERENT SIZES FOR THE LABELED TRAINING SET. DOUBLEMATCH ACHIEVES STATE-OF-THE-ART RESULTS ON MANY COMBINATIONS.

	CIFAR-10			CIFAR-100			SVHN			STL-10
Method	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels	40 labels	250 labels	1000 labels	1000 labels
MixMatch [7], [24]	47.54±11.50	11.05±0.86	6.42±0.10	67.61±1.32	39.94±0.37	28.31±0.33	42.55±14.53	3.98±0.23	3.50±0.28	10.41±0.61
UDA [6], [7]	29.05±5.93	8.82±1.08	4.88±0.18	59.28±0.88	33.13±0.22	24.50±0.25	52.63±20.51	5.69±2.76	2.46±0.24	7.66±0.56
ReMixMatch [7], [16]	19.10±9.64	5.44±0.05	4.72±0.13	44.28±2.06	27.43±0.31	23.03±0.56	3.34±0.20	2.92±0.48	2.65±0.08	5.23±0.45
FixMatch (CTA) [7]	11.39±3.35	5.07±0.33	4.31±0.15	49.95±3.01	28.64±0.24	23.18±0.11	7.65±7.65	2.64±0.64	2.36±0.19	5.17±0.63
DP-SSL [32]	**6.54±0.98**	**4.78±0.26**	4.23±0.20	43.17±1.29	28.00±0.79	22.24±0.31	**2.98±0.86**	**2.16±0.36**	**1.99±0.18**	4.97±0.42
Dash (CTA) [31]	9.16±4.31	**4.78±0.12**	**4.13±0.06**	44.83±1.36	27.85±0.19	22.77±0.21	3.14±1.60	2.38±0.29	2.14±0.09	**3.96±0.25**
DoubleMatch (last 20)	14.02±5.71	5.72±0.51	4.83±0.17	**42.61±1.15**	**27.48±0.19**	**21.69±0.26**	16.50±13.73	2.58±0.53	2.25±0.09	4.46±0.20
DoubleMatch (min)	13.59±5.60	5.56±0.42	4.65±0.17	**41.83±1.22**	**27.07±0.26**	**21.22±0.17**	15.37±11.81	2.37±0.35	2.10±0.07	4.35±0.20

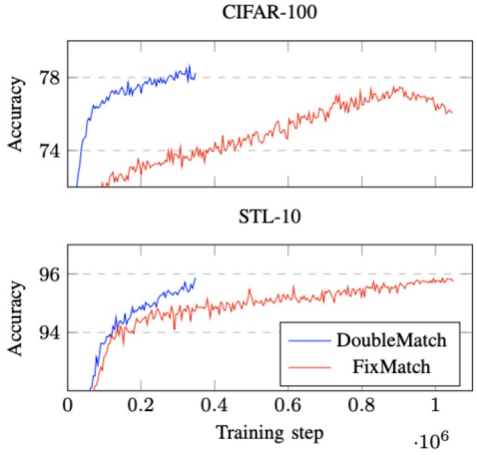
Experiments

B. Training speed

Running **FixMatch** for its full training duration on CIFAR- 100 using a single A100 GPU takes 5 days of wall-time.
DoubleMatch reduces these long training times by more efficiently making use of unlabeled data through the added self-supervised loss.
While the methods we use as comparison in Table I run for little more than 1M training steps, we run DoubleMatch for only roughly a third of that.
표 I에서 비교로 사용하는 방법은 **1M 이상의 training steps를 실행하는 반면, DoubleMatch는 그 약 1/3만 실행**

A clear illustration of our increase in training speed is seen in Fig. 3

- compare DoubleMatch to FixMatch during training runs on CIFAR-100 with 10,000 labeled training data
- compare DoubleMatch to FixMatch during training runs on STL-10 with 1,000 labeled data.



C. Discussion

high performance on CIFAR-100 (new SOTA) and STL-10.
-> hypothesis: the self-supervised loss contributes to the biggest improvement when it is difficult to reach high classification accuracies on the unlabeled training set.
레이블이 지정되지 않은 훈련 세트에서 높은 분류 정확도에 도달하기 어려울 때 self-supervised loss이 가장 큰 개선에 기여

unlabeled set 에 대한 낮은 정확도:
1) CIFAR-100의 경우와 같이 분류 문제의 어려움
2) STL-10의 경우와 같이 레이블링된 훈련 세트보다 더 넓은 분포에서 오는 레이블링되지 않은 데이터
<-> CIFAR-10 및 SVHN의 경우 일반적으로 레이블이 지정되지 않은 훈련 세트에서 높은 정확도에 도달할 수 있으므로 의사 레이블의 품질이 매우 높아져 self-supervised loss가 덜 중요할 수 있다.

DoubleMatch: Improving Semi-Supervised Learning with Self-Supervision

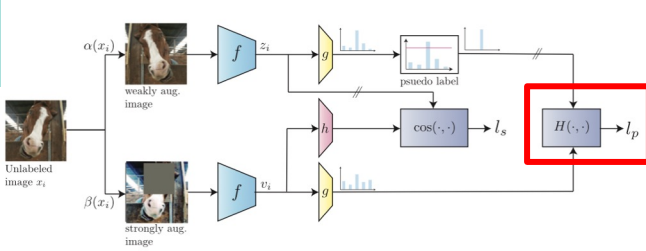


Fig. 2. Graph showing loss evaluation for unlabeled images. Double slash marks a stop-gradient operation.

Ablation

첫번째, self-supervised loss의 코사인 유사성이 다른 유사성 함수로 대체되는 실험 결과를 보임
두번째, 레이블링된 훈련 세트의 다양한 크기에 대한 pseudo-labeling loss의 중요성을 고려

1) **self-supervised loss의 코사인 유사성이 다른 유사성 함수로 대체**
10,000개의 레이블을 사용하여 CIFAR-100의 다양한 손실 함수를 평가
training runs:
   1) MSE
   2) Softmax λ = 1
   3) Softmax λ = 0.1

The self-supervised weight, ws, is re-tuned for every loss function.

=> 다른 함수와 비교할 때 코사인 유사성을 사용하여 상당히 낮은 error rate에 도달
=> 실제로 DoubleMatch에 대한 올바른 선택

2) **레이블링된 훈련 세트의 다양한 크기에 대한 pseudo-labeling loss의 중요성**
CIFAR-100에서 수행
pseudo-labeling loss가 있는 DoubleMatch와 pseudo-labeling loss가 없는 DoubleMatch 간의 acc차이 평가
The weight for the self-supervised loss, ws, is re-tuned for each split after removing lp.

DoubleMatch는 여전히 400개의 레이블로 CIFAR-100에 대한 SOTA 결과에 도달
-> 40개의 레이블로 구성된 CIFAR-10 및 SVHN에 대한 좋지 않은 결과
-> **self-supervised loss가 낮은 레이블 체제에서 자체적으로 잘 수행되지 않음을 나타냄**
개선된 pseudo-label selections이 낮은 라벨 체제에서 더 중요한 것으로 보인다는 것을 나타냄.
=> 충분한 레이블이 있으면 pseudo-label loss 성능 손실이 거의,전혀 없는 self-supervised loss로 대체가능

TABLE II
EVALUATIONS OF DIFFERENT FUNCTIONS FOR THE SELF-SUPERVISED LOSS IN DOUBLEMATCH ON CIFAR-100 WITH 10,000 LABELS.

Loss function	Error rate	w_s
Cosine	$21.22_{\pm 0.17}$	10.0
MSE	23.91	0.25
Softmax ($\lambda = 1$)	23.23	1.0
Softmax ($\lambda = 0.1$)	23.57	0.5

TABLE III
REDUCTION IN TEST ACCURACY ON CIFAR-100 BY REMOVING THE PSEUDO-LABEL LOSS FROM DOUBLEMATCH FOR DIFFERENT SIZES OF THE LABELED TRAINING SET.

	Nr. of labeled training data			
	400	1,000	2,500	10,000
Reduction	8.46	3.81	2.20	0.39

Conclusion

This paper shows that using a self-supervised loss in semi- supervised learning can lead

- **to reduced training times**
- **to increased test accuracies for multiple datasets.**

we present

- **new SOTA results on CIFAR-100** using different sizes of the labeled training set while using fewer training steps than existing methods.
- **enough labeled training data, the pseudo-labeling loss can be removed with no performance reduction in the presence of a self-supervised loss.**
  충분한 labeled training data 있고, self-supervised loss 존재하는 경우 -> 성능 저하 없이 pseudo-labeling loss을 제거할 수 있다

End of the Document