Lecture 18: Transfer Learning

Speaker: Sara Beery

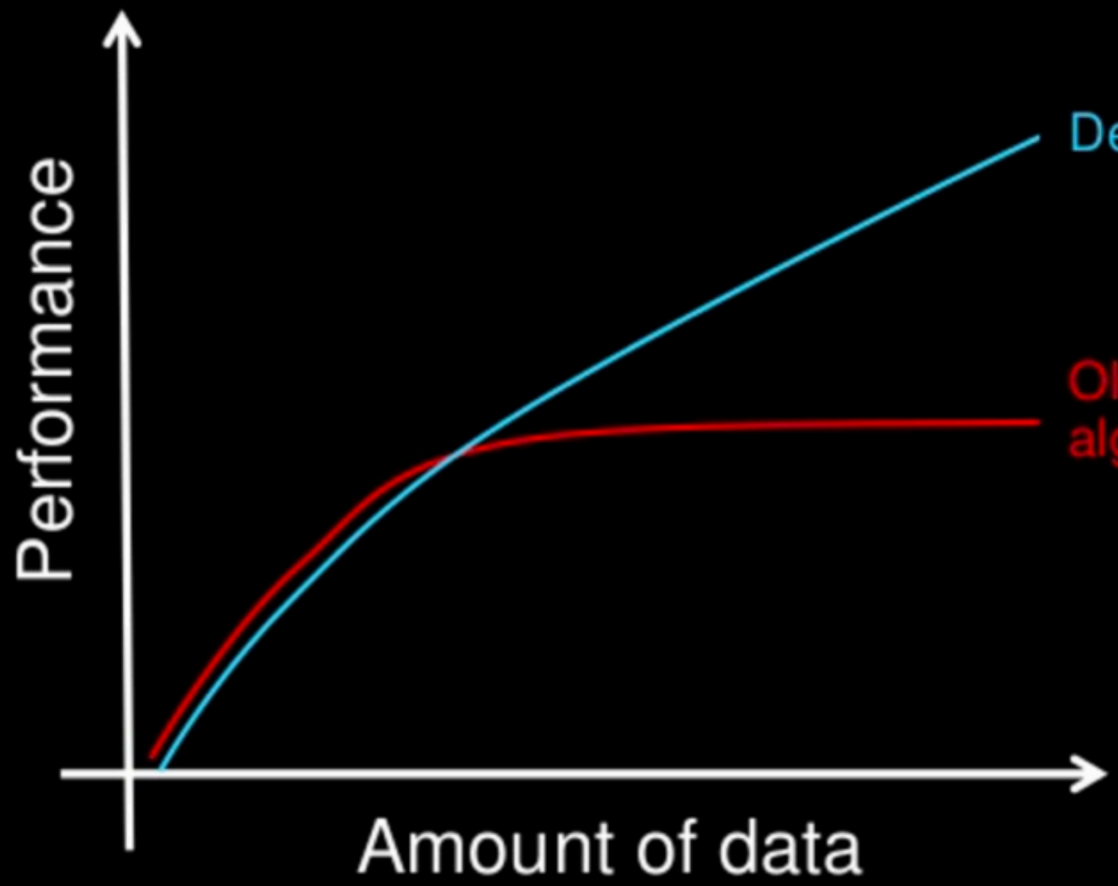


Transfer Learning

idea: want to try to learn even if you don't have very much data.

- Learning with little data
- Transferring knowledge about the mapping (finetuning and domain adaptation)
- Transferring knowledge about the outputs (distillation)
- Transferring knowledge about the inputs (generative models as data)
- Learning to learn quickly (meta-learning)

Why deep learning



data ↑ → Performance ↑

Model의 Capacity까지 data를 넣을수록 좋음.

Issue)

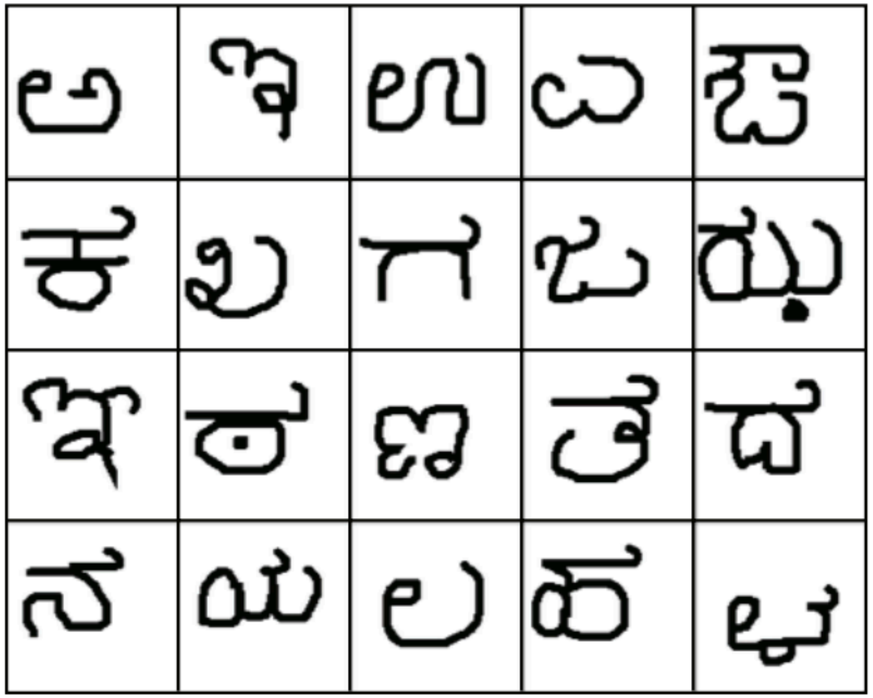
insufficient #of data.

How do data science techniques scale with amount of data?

[Slide credit: Andrew Ng]

Few-shot Learning





This is a "dax". (개상의 기호)

Which of the below symbols are also daxes?

[Lake, Salakhutdinov, Tenenbaum, 2015]

Few-shot Learning



Which of these is an example of the same concept as the item in the box?

Segway 2 wheels

[Lake, Salakhutdinov, Tenenbaum, 2015]

"Deep learning"



Human learning

various prior knowledge



Representations

Models

Skills

How can we give deep nets prior knowledge?

How can we give deep nets prior knowledge?

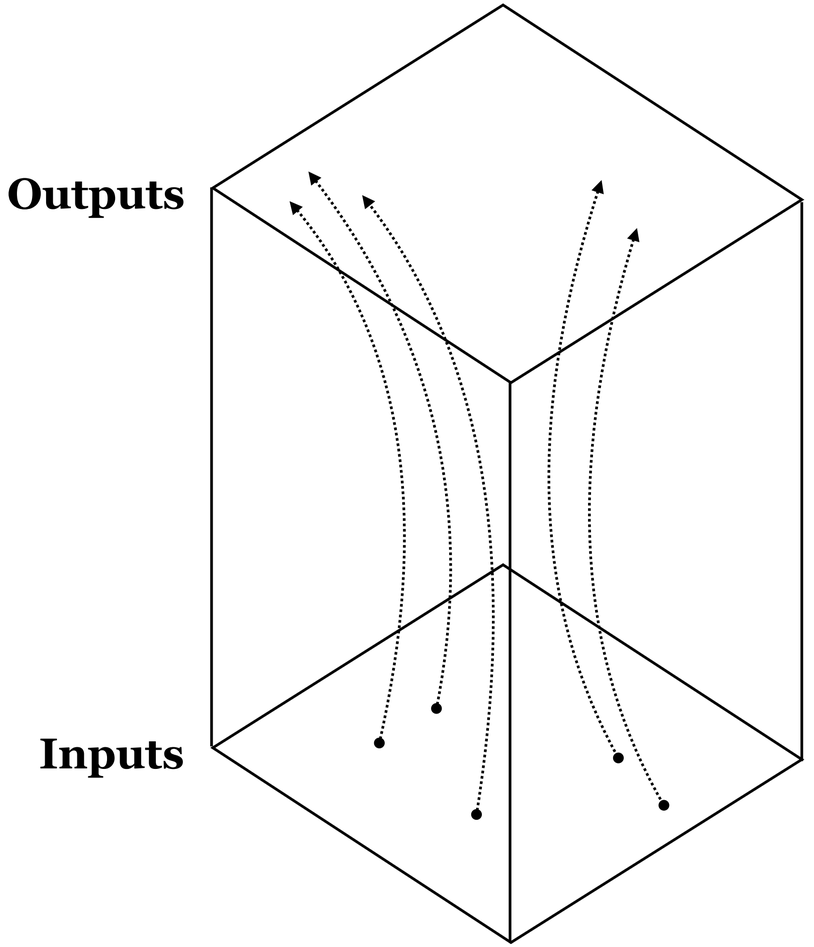
Just "pretrain" them on prior tasks!



Then transfer knowledge from the pretrained nets to solve novel tasks.

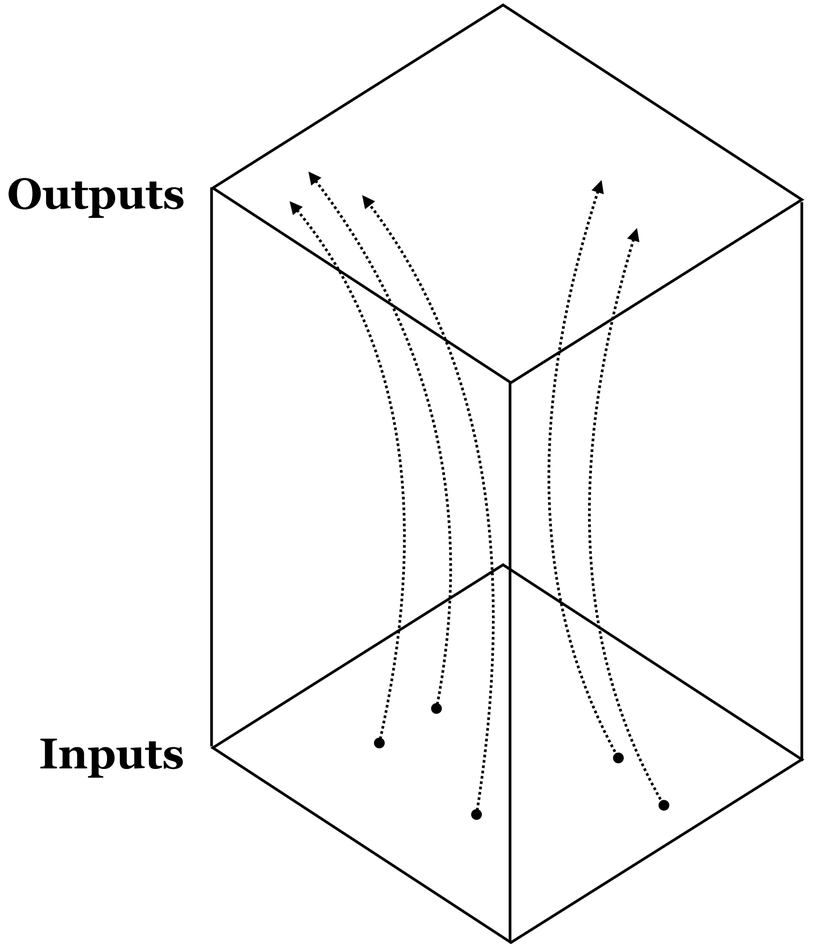
One view: the ***point*** of deep learning is to enable problem solving with little data

1. Knowledge about the mapping
2. Knowledge about the outputs
3. Knowledge about the inputs
4. Learning to learn fast



Transfer Learning

1. **Knowledge about the mapping**
2. Knowledge about the outputs
3. Knowledge about the inputs
4. Learning to learn fast

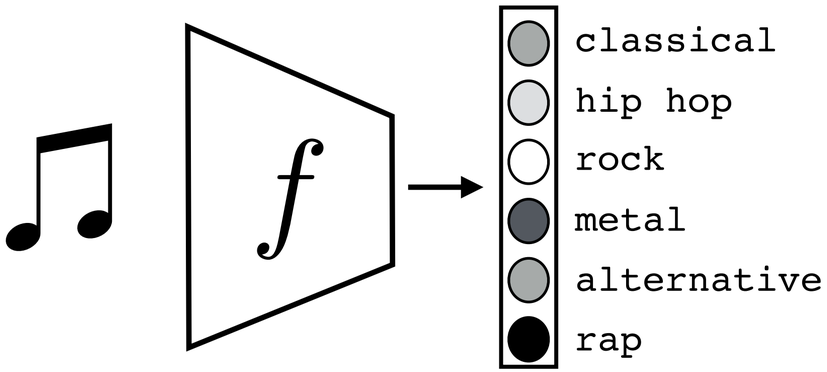


Transfer Learning

"Post-training"

Pretraining

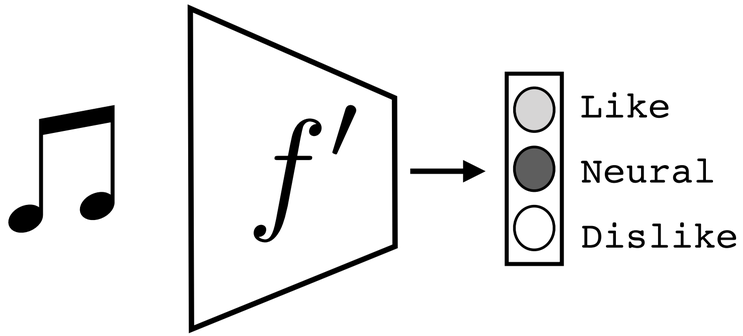
Genre recognition



A lot of data

~~Adapting~~

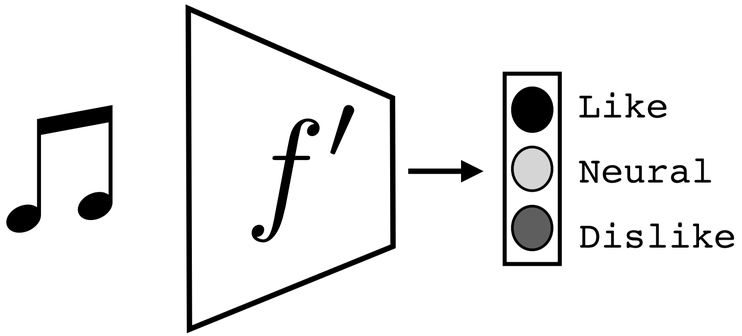
Preference prediction



A little data

Testing

Preference prediction



Finetuning

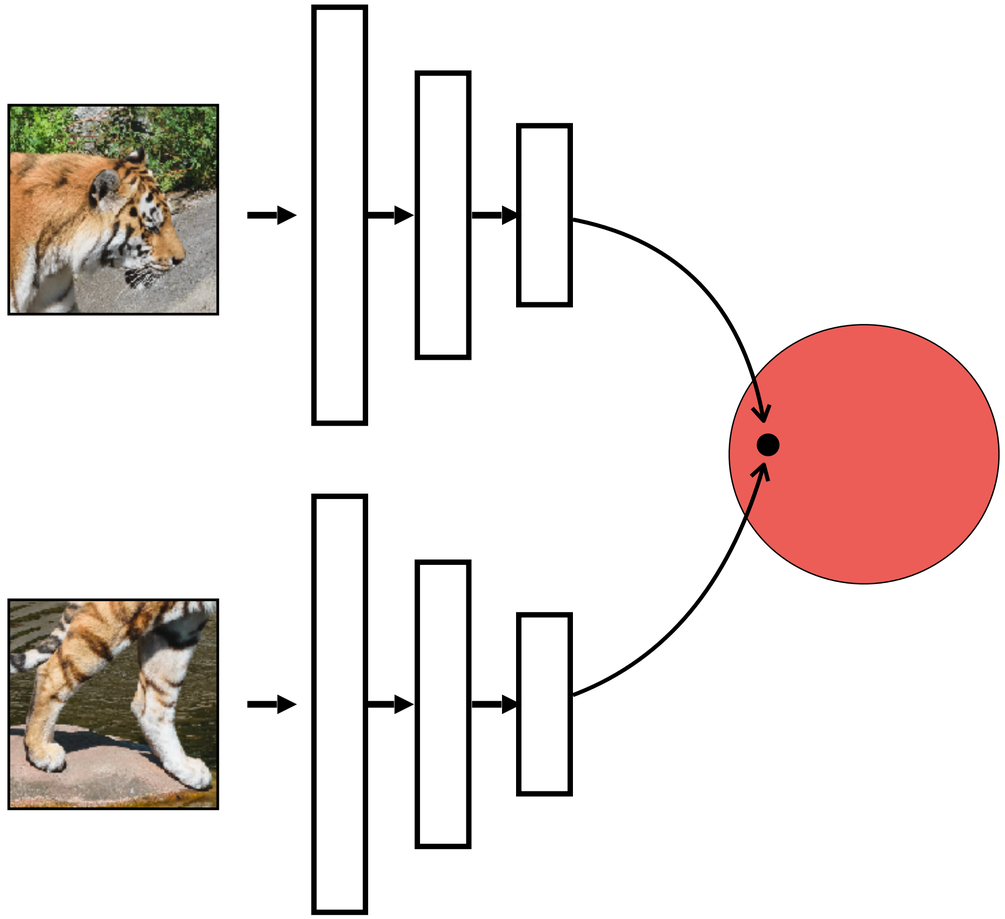
- Pretrain a network on task A, resulting in parameters **W** and **b**
- Initialize a second network with some or all of **W** and **b**
- Train the second network on task B, resulting in parameters **W'** and **b'**

The "learned representation" is the *encoder* (its weights and biases) so that's what we transfer

Contrastive pre-training

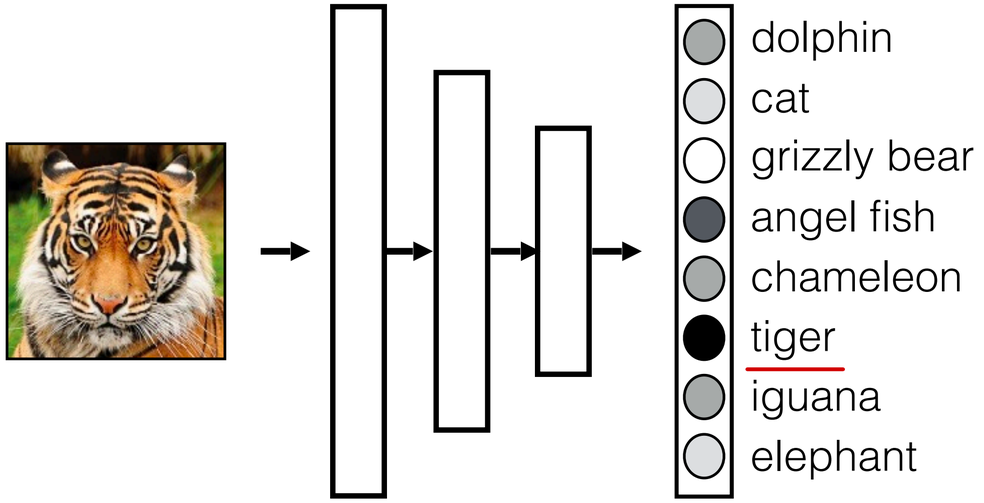
Pretraining

Self-supervised contrastive learning



Target

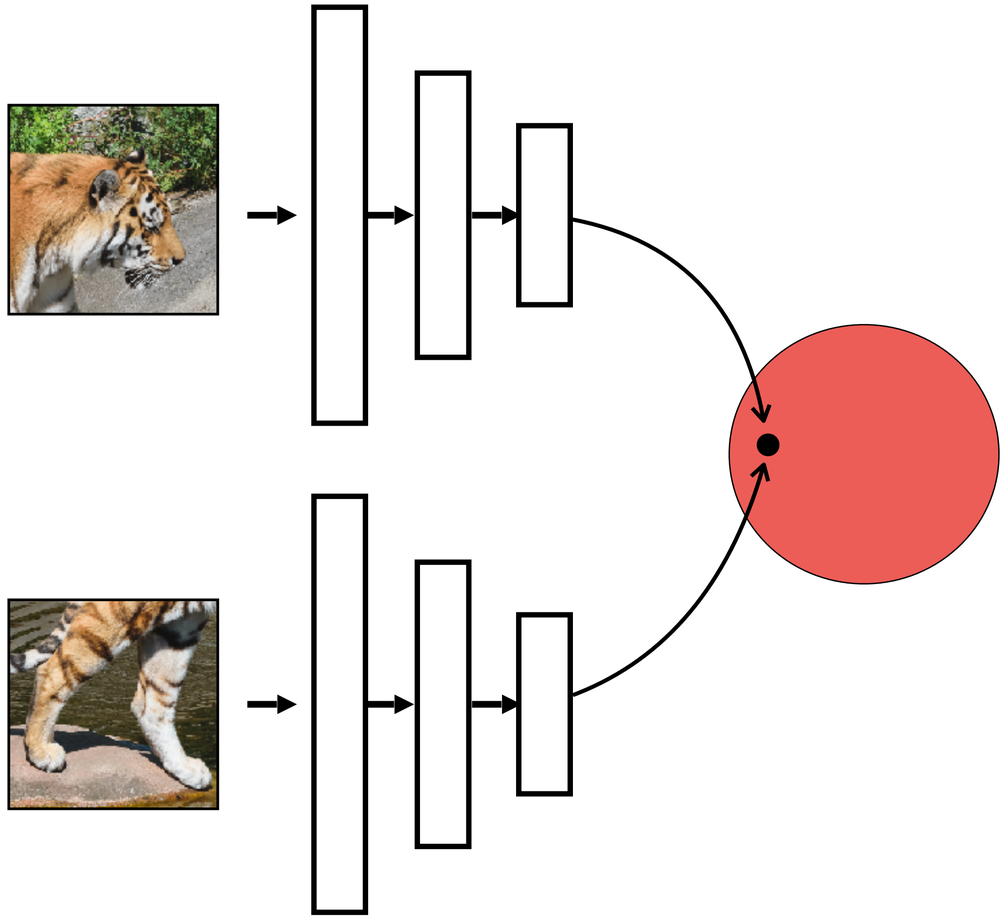
New recognition task



Contrastive pre-training

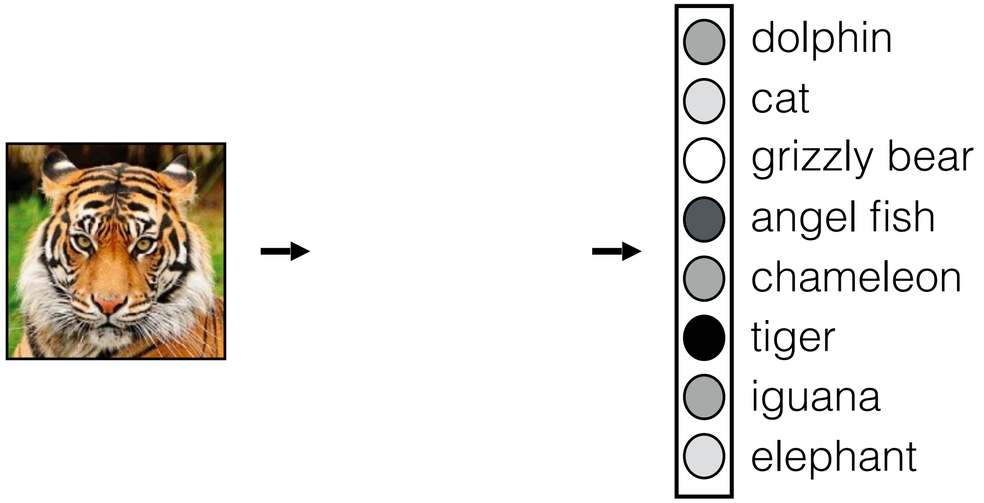
Pretraining

Self-supervised contrastive learning



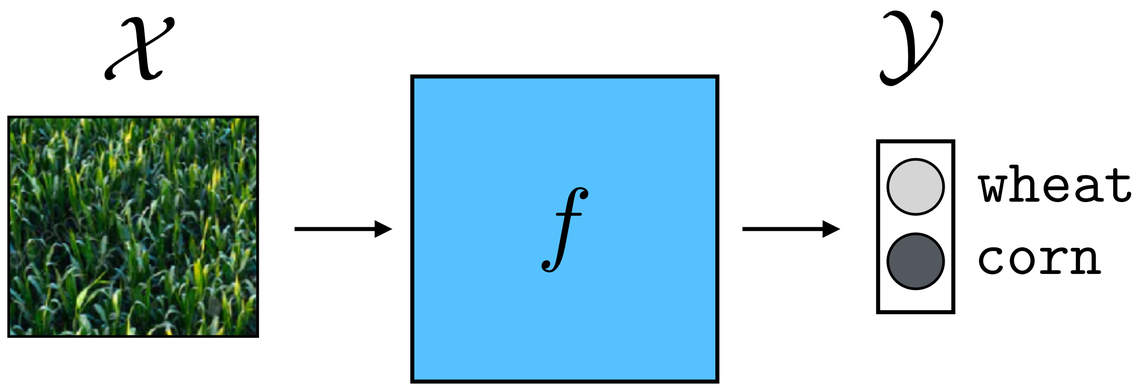
Finetuning

New recognition task

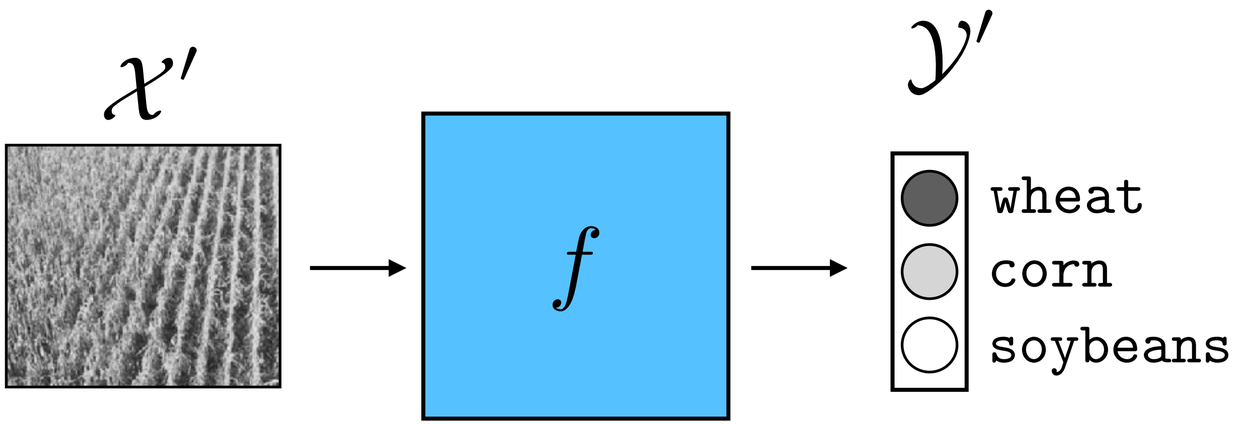


What if the input/output dimensions don't match?

Pretraining

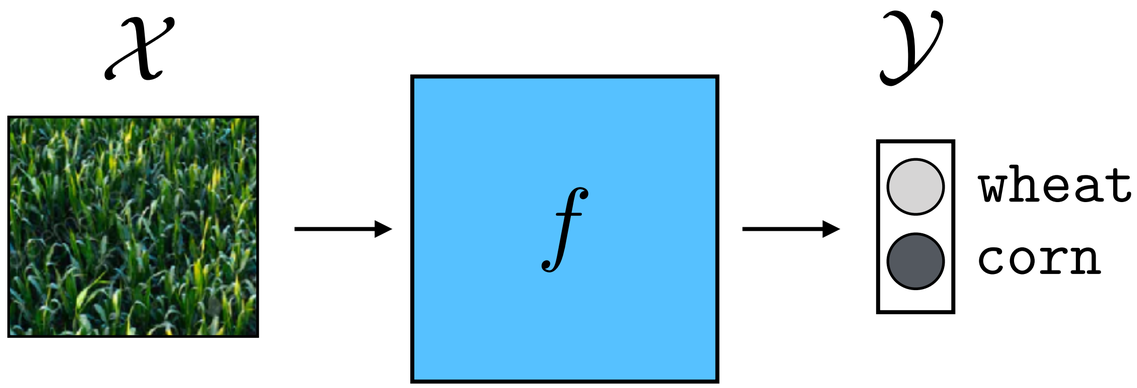


Finetuning

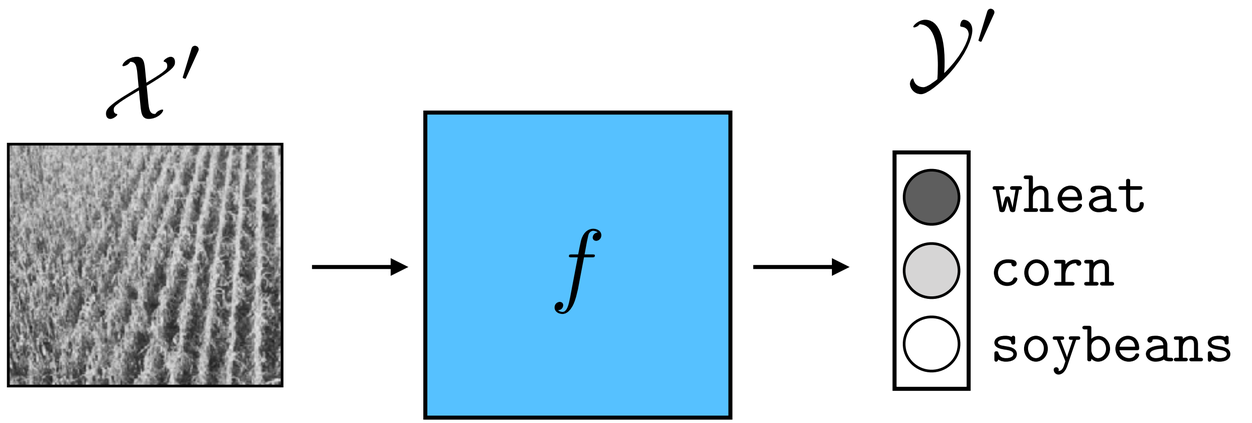


What if the input/output dimensions don't match?

Pretraining



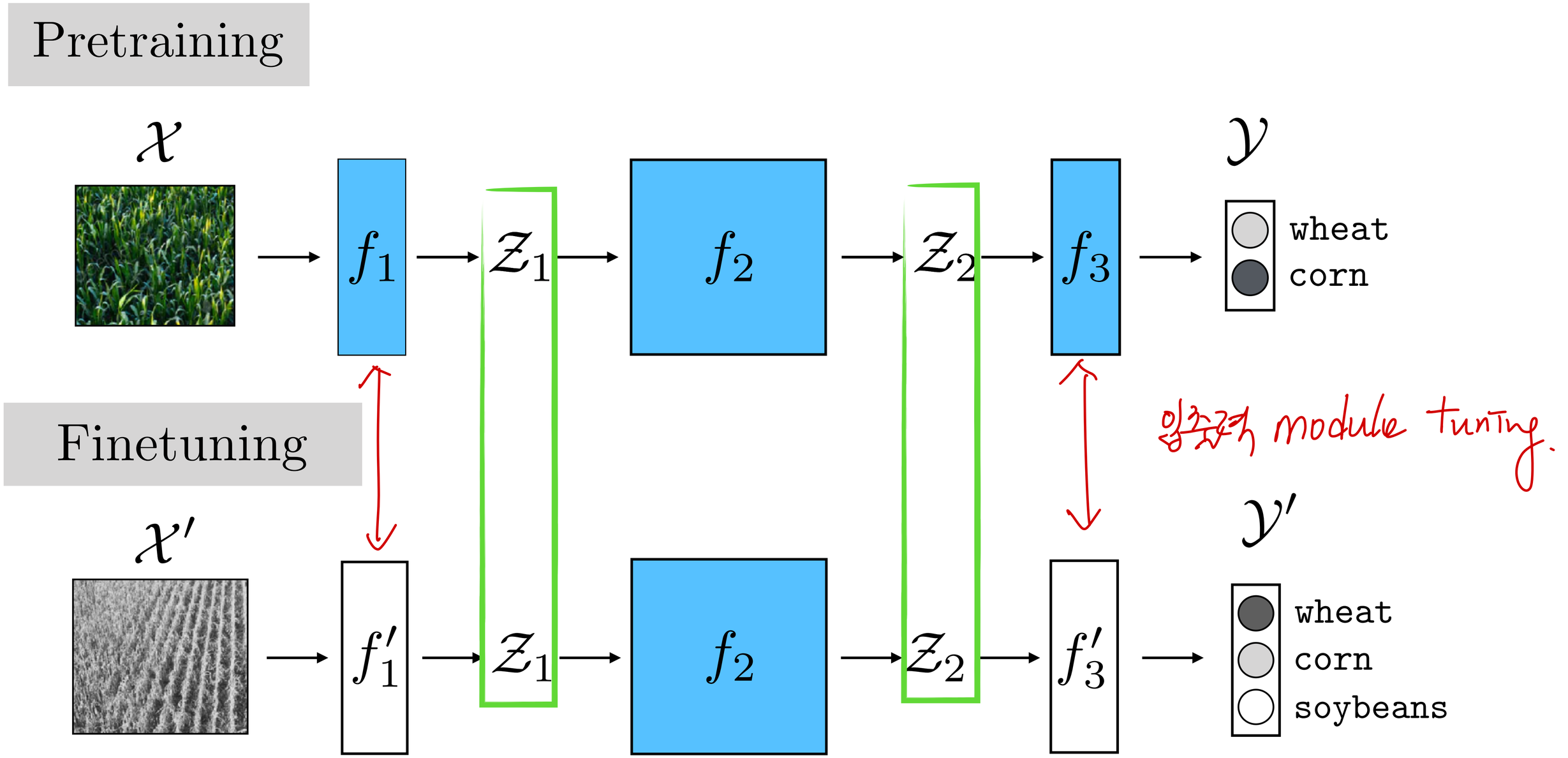
Finetuning



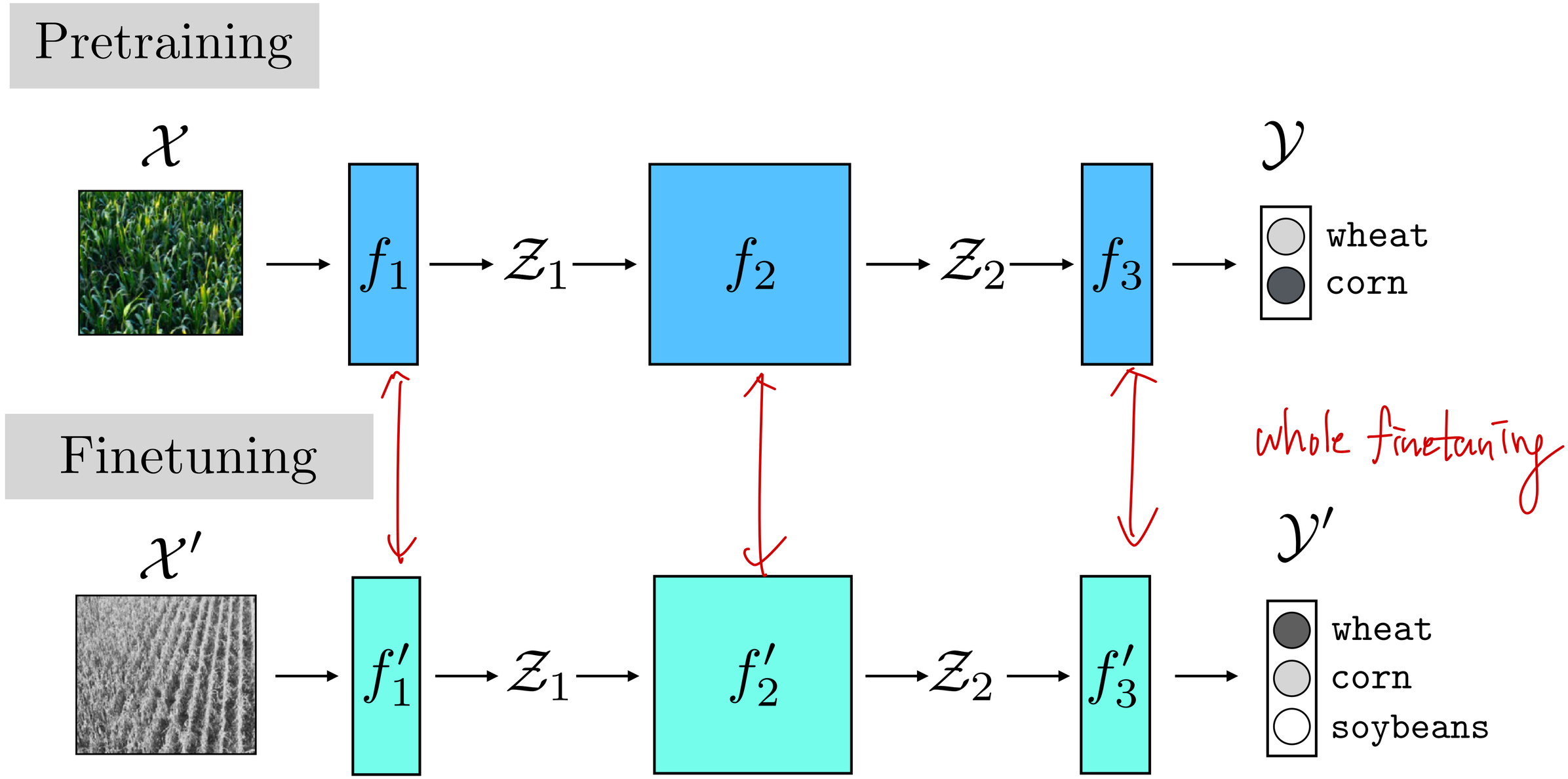
$$\mathcal{X}' \neq \mathcal{X}$$

$$\mathcal{Y}' \neq \mathcal{Y}$$

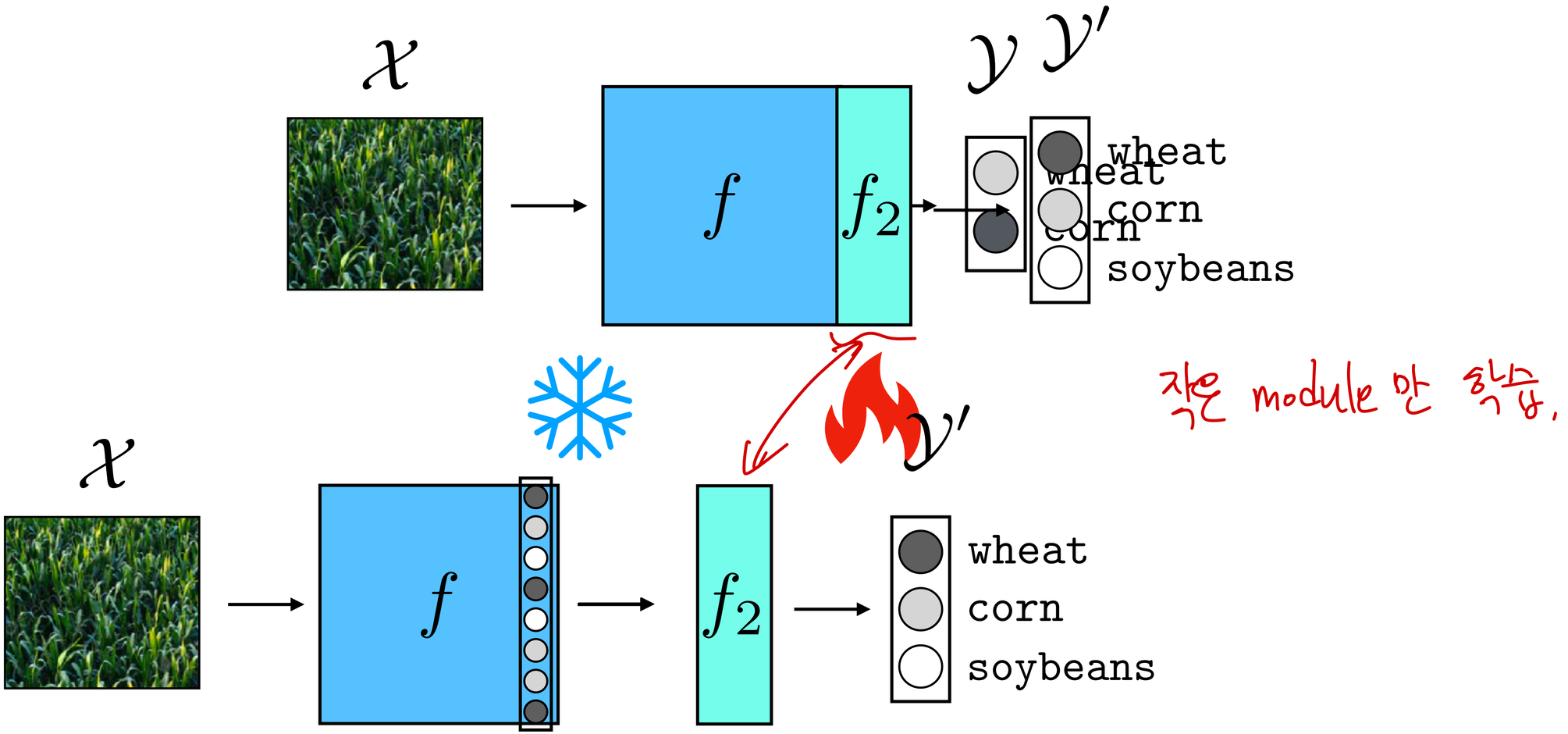
What if the input/output dimensions don't match?



What if the input/output dimensions don't match?



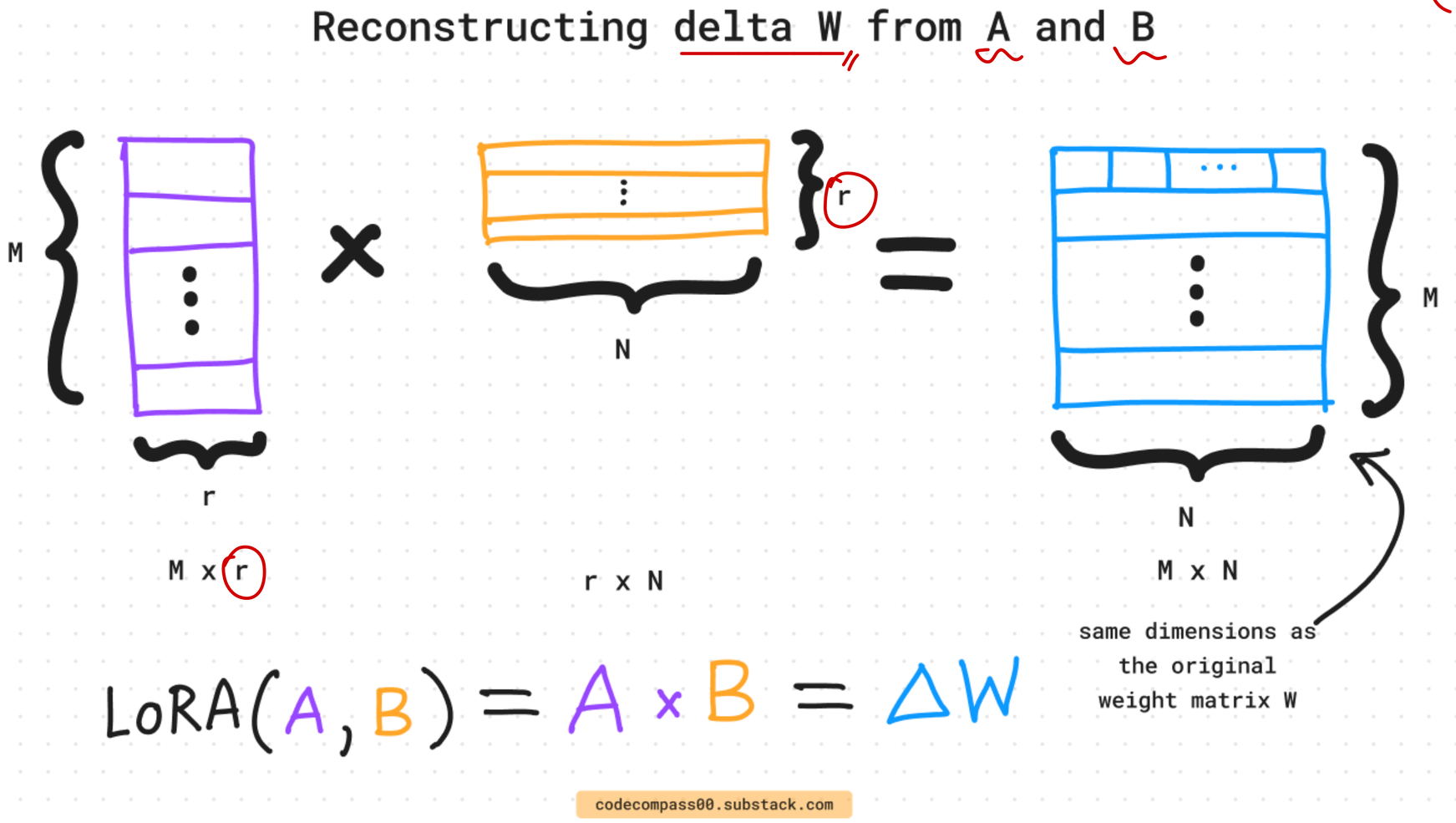
Efficient finetuning



"Linear probing" - if f2 is a linear layer

Low-rank Adaptation (LoRA)

parameter의 수는
r (low-rank)에 비례해 증가한다.

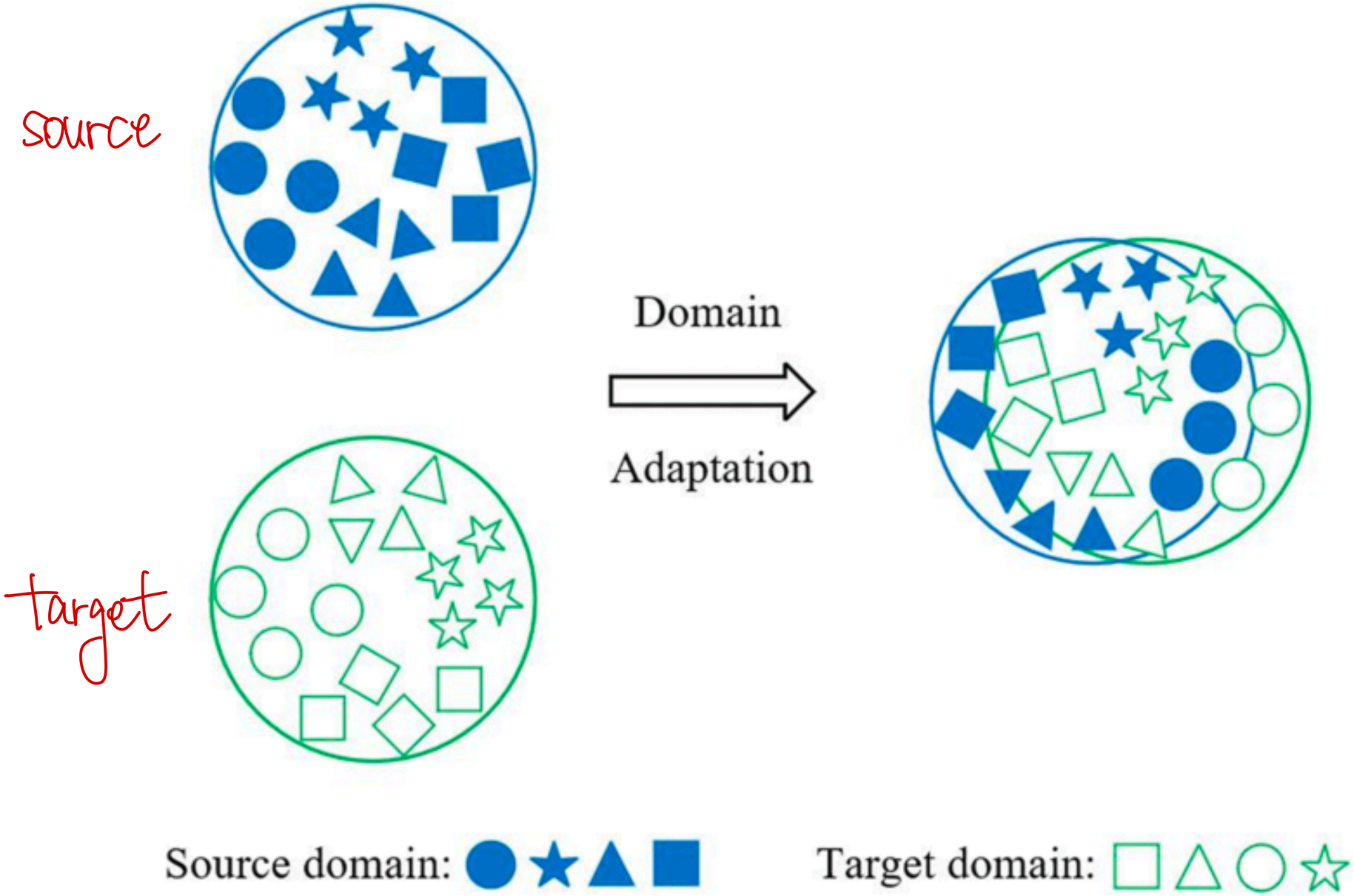


https://arxiv.org/pdf/2106.09685

Avoiding overfitting on a small amount of data for a new task while still enabling specialization can be a balancing act. A few methods:

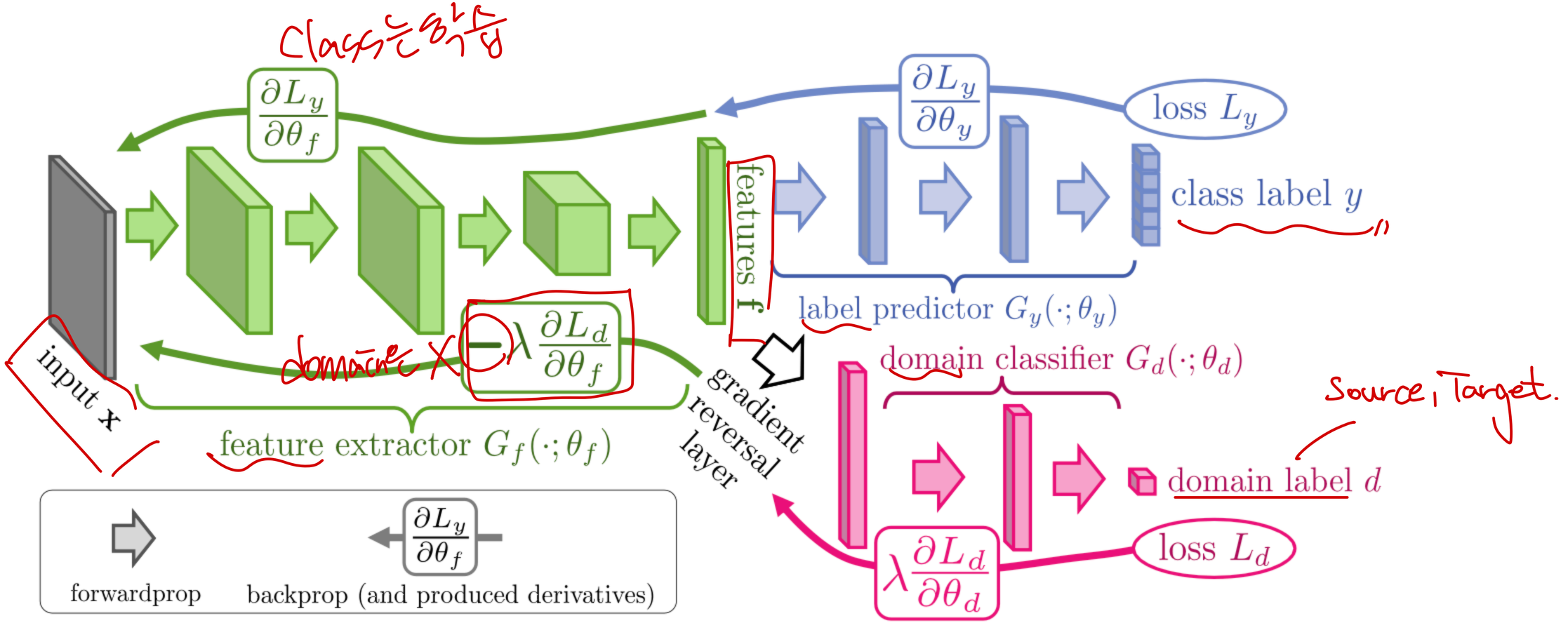
- Limit what can be updated (ie parameter freezing/linear probes)
- Use a small learning rate
- Early stopping
- Continue to train on the original data as well

All assume that you have a good, representative validation set for your task (not always possible when you may see distribution shift)



https://link.springer.com/article/10.1007/s10489-022-03709-8

Domain Adversarial Neural Networks (DANN)

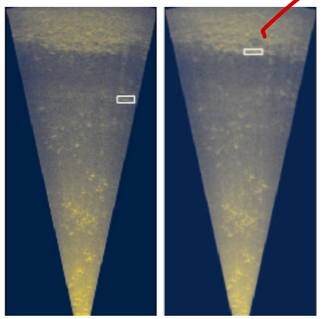


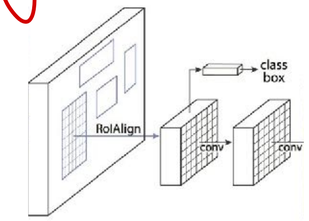
https://arxiv.org/abs/1505.07818

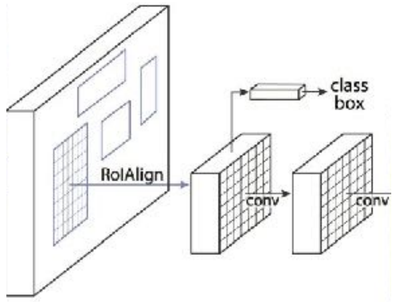
Unsupervised Domain Adaptation (UDA)

target Data

Unlabeled target

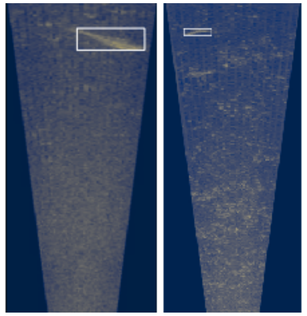


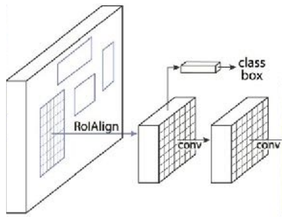




UDA model

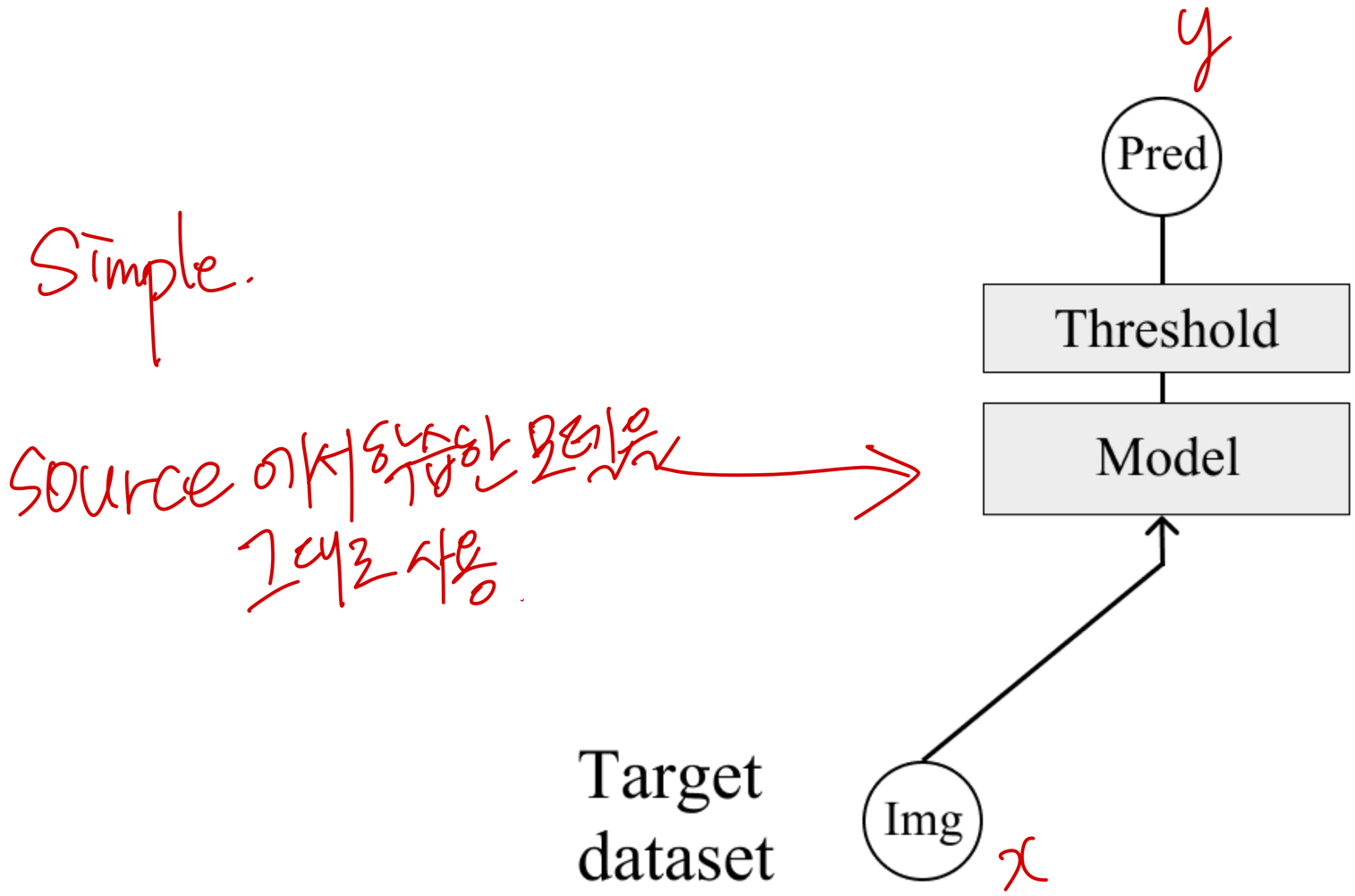
Labeled source

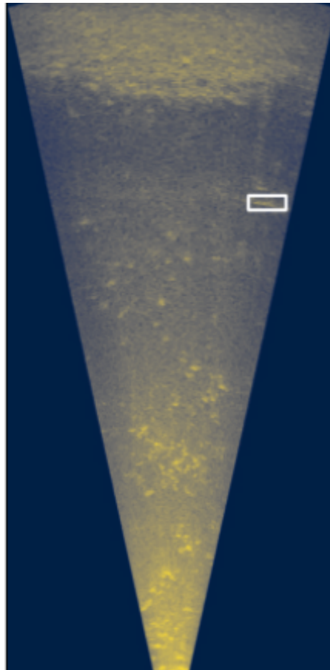


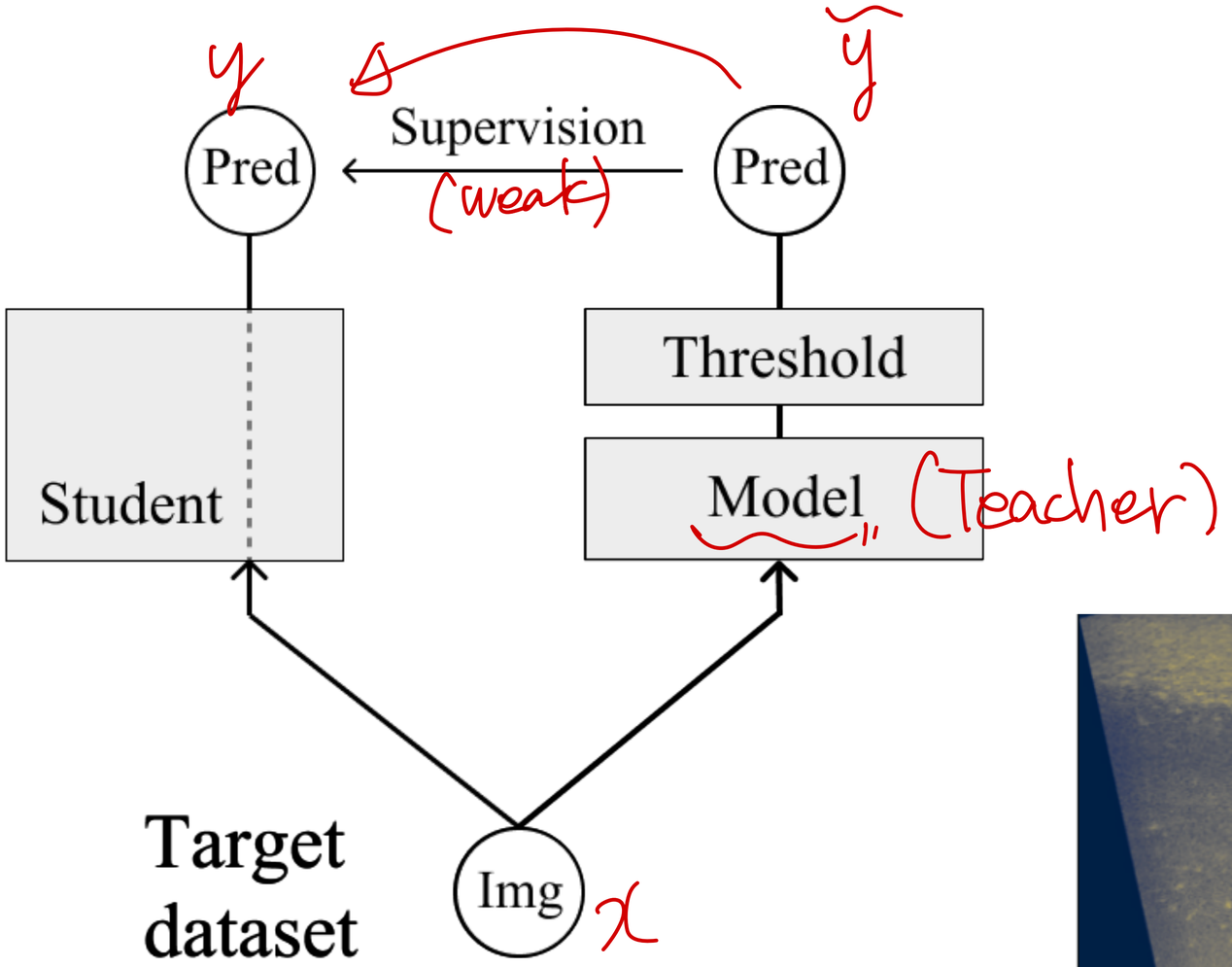


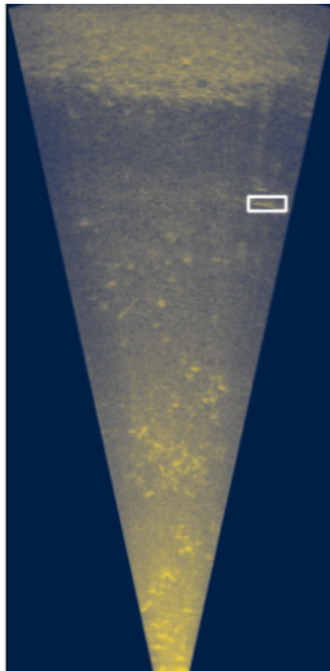
Simple.

Source 에서 학습한 모델을
그대로 사용.





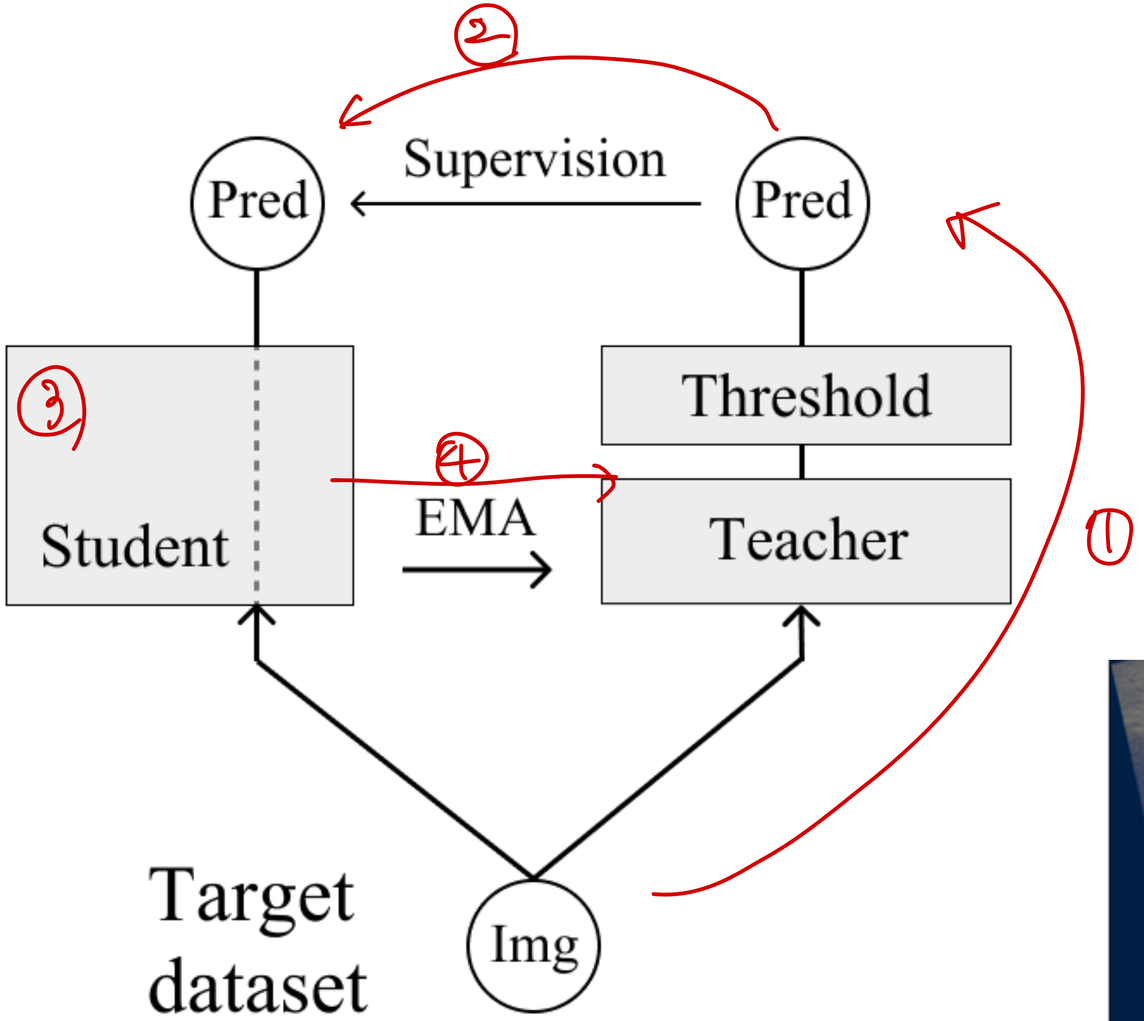

Pred
Supervision
(weak)
Pred
Threshold
Model
(Teacher)
Student
Target
dataset
Img

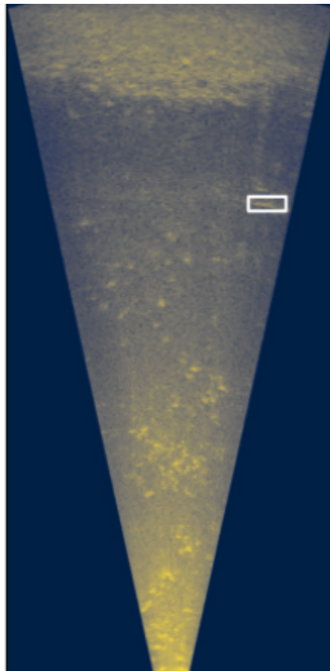


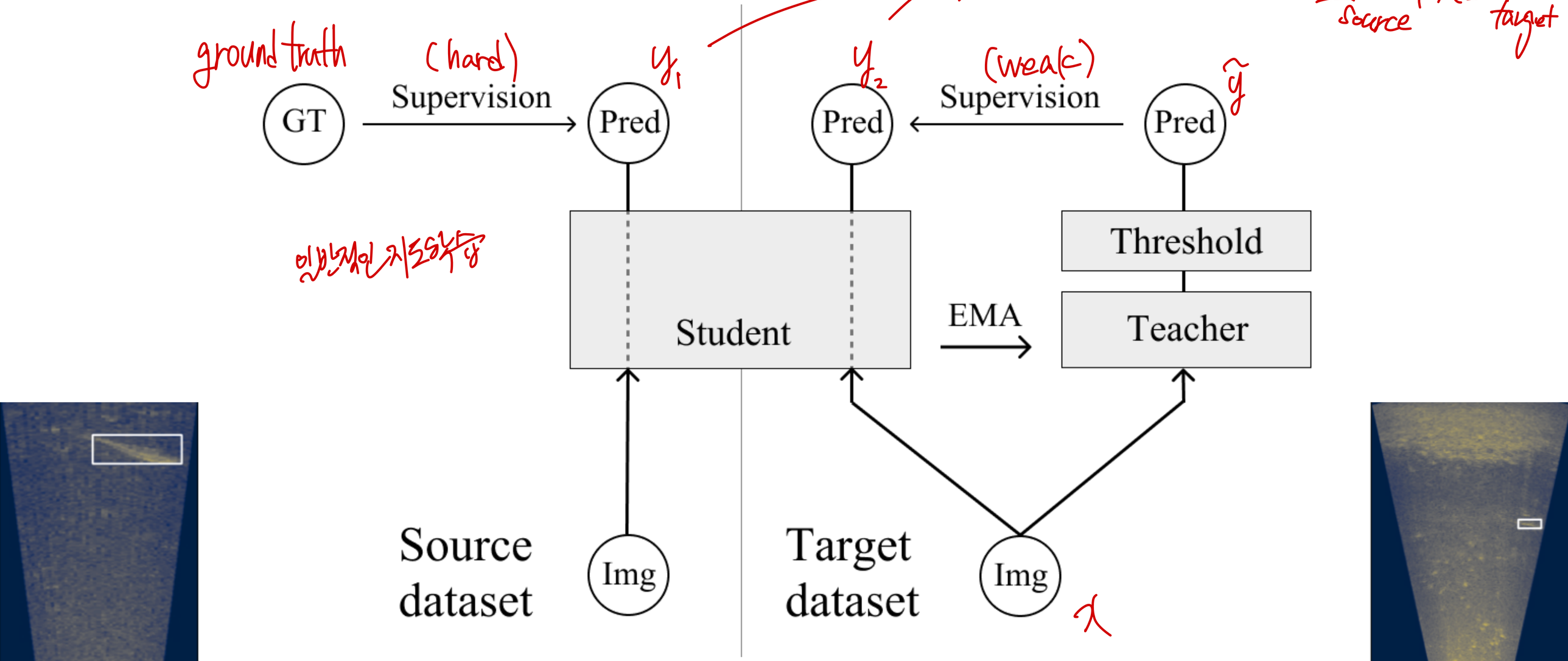
EMA: exponential moving Avg.

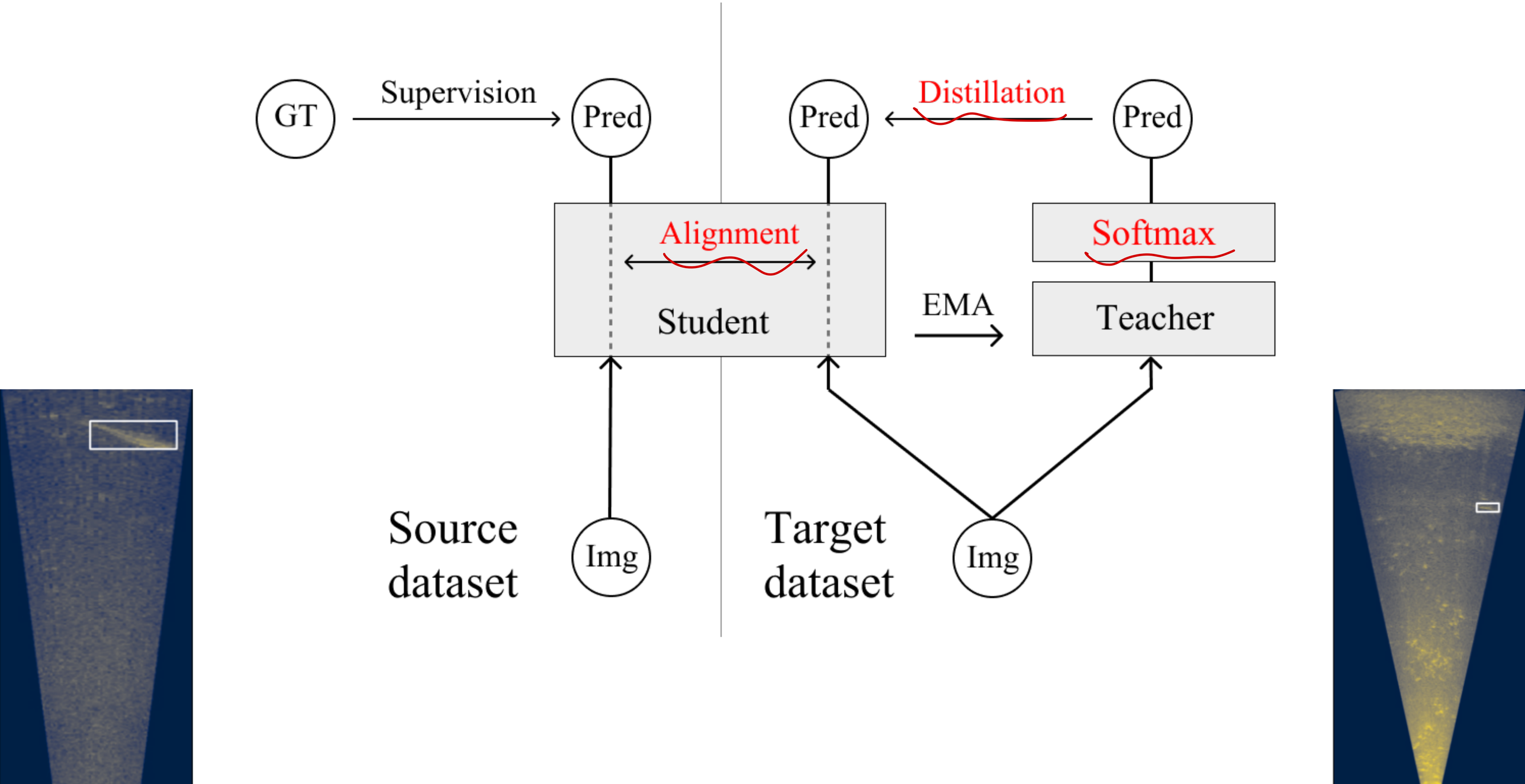
Student model의 weight가

teacher의 점차적인 영향.

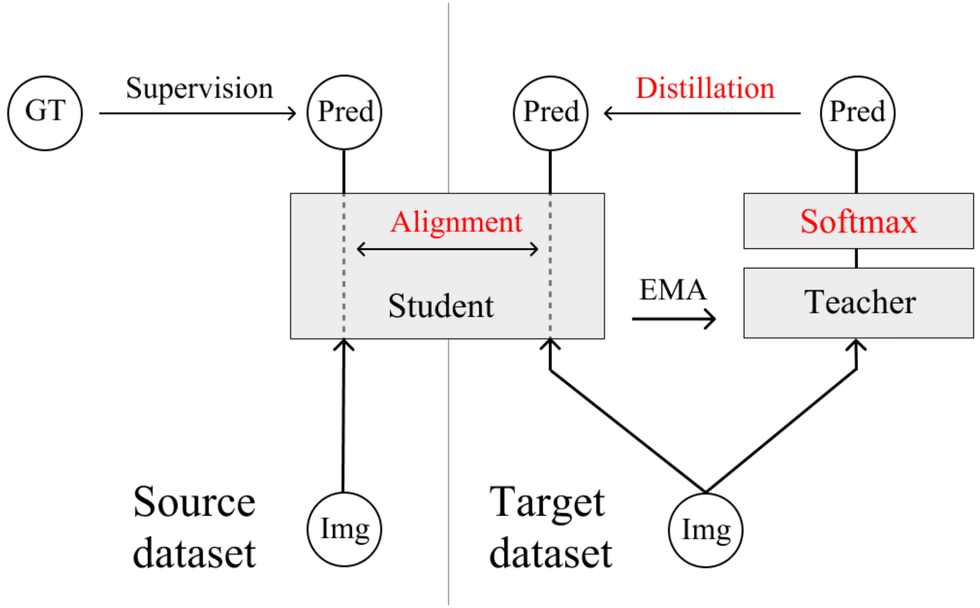


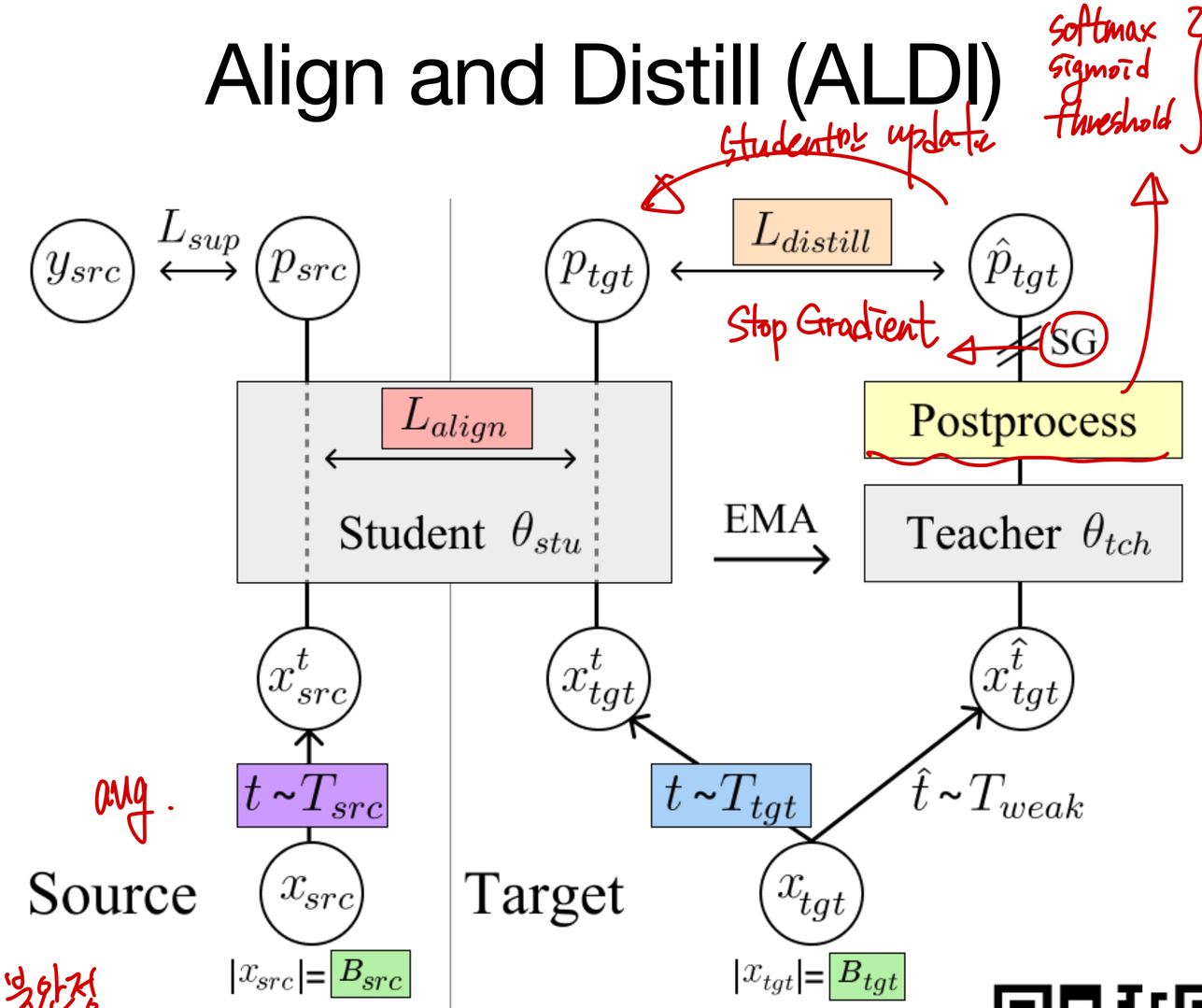



GT
Supervision
Pred
Pred
Supervision
Pred
Threshold
Teacher
EMA
Student
Source dataset
Target dataset
Img
Img


GT
Supervision
Pred
Pred
Distillation
Pred
Alignment
Softmax
Student
EMA
Teacher
Source dataset
Img
Target dataset
Img

Align and Distill (ALDI)



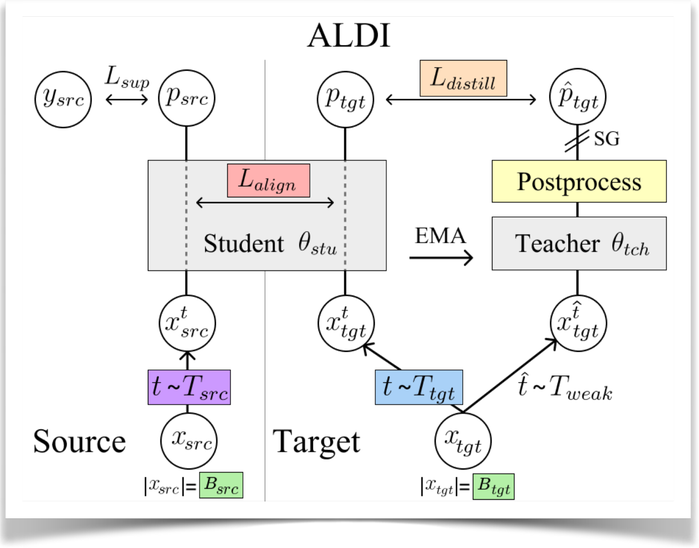


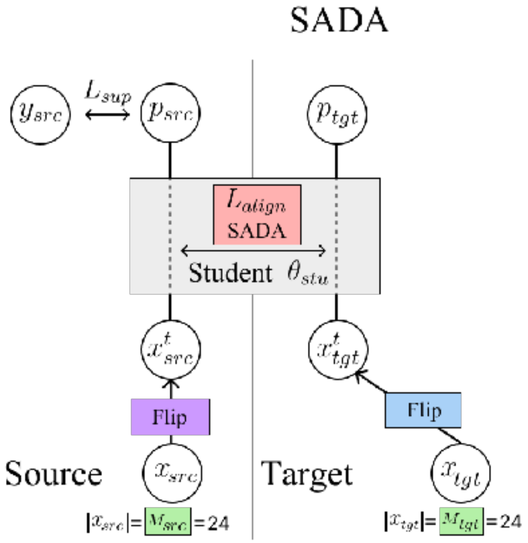
Why SG?

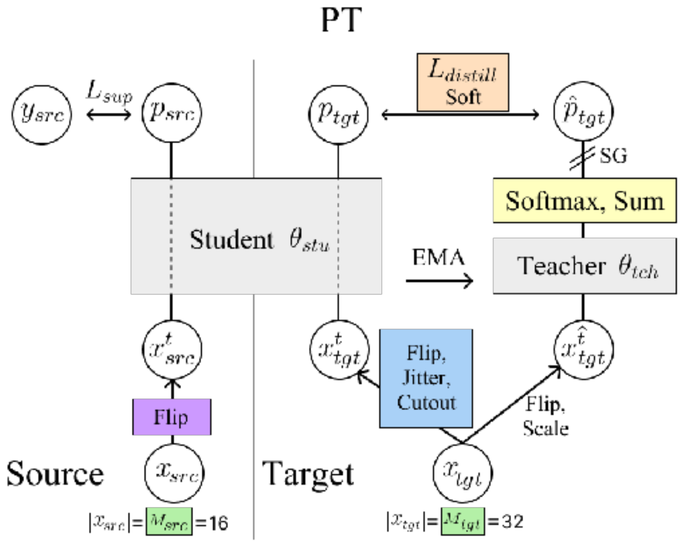
Student와 teacher가 같은 loss로 학습 시 loss 발산함

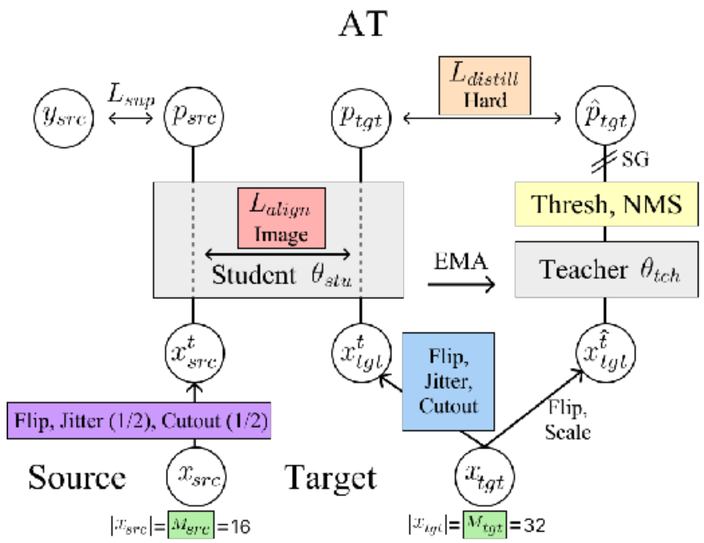
→ 대신 EMA로 학습

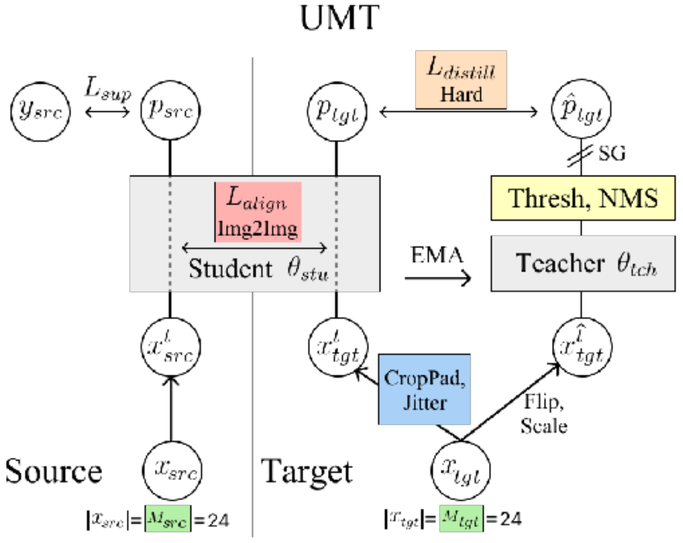
aldi-daod.github.io

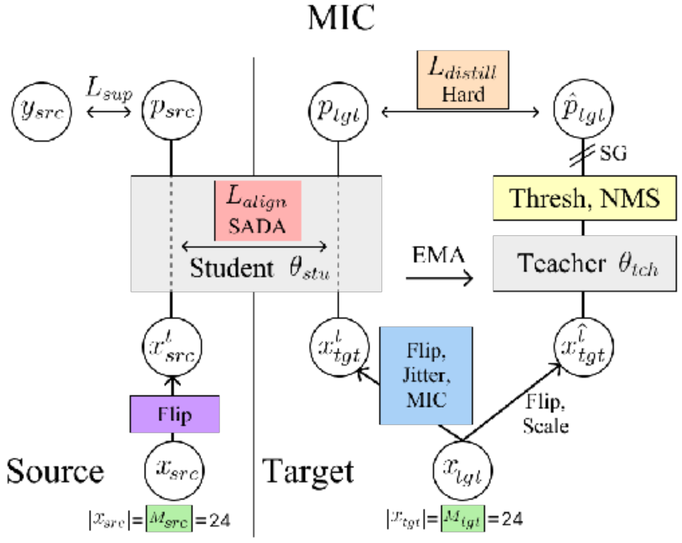

ALDI
L_{sup}
$L_{distill}$
SG
L_{align}
Postprocess
Student θ_{stu}
EMA
Teacher θ_{tch}
$t \sim T_{src}$
$t \sim T_{tgt}$
$\hat{t} \sim T_{weak}$
Source
Target
$|x_{src}|=B_{src}$
$|x_{tgt}|=B_{tgt}$


SADA
L_{sup}
L_{align}
SADA
Student θ_{stu}
Flip
Flip
Source
Target
$|x_{src}|=M_{src}=24$
$|x_{tgt}|=M_{tgt}=24$

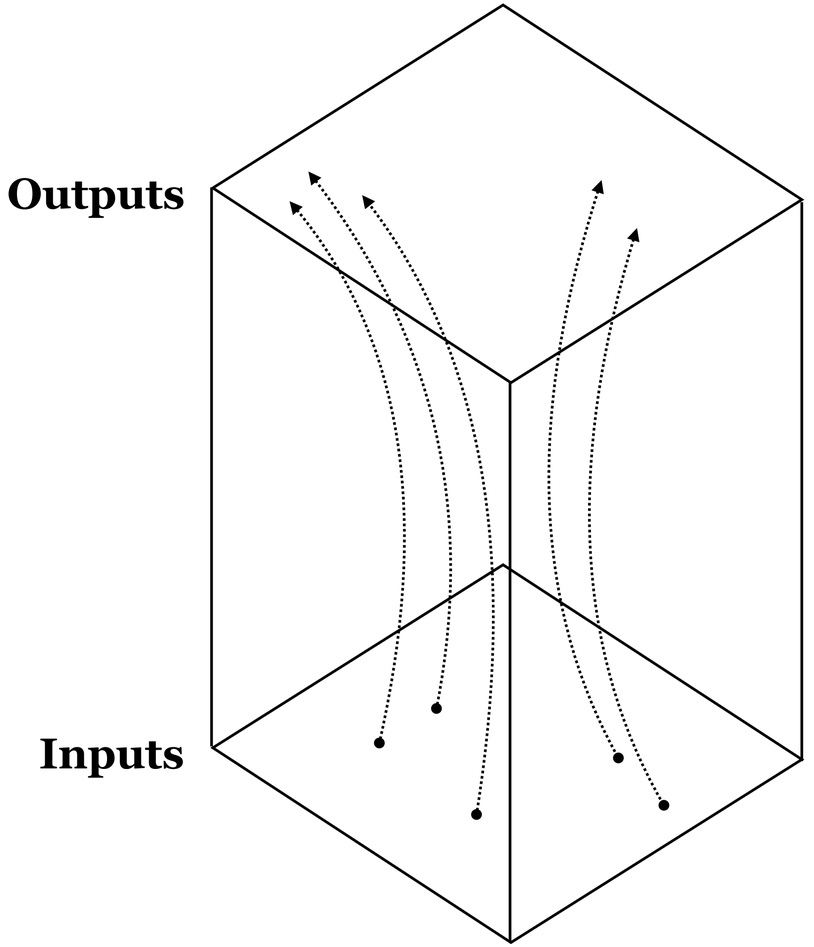

PT
L_{sup}
$L_{distill}$
Soft
SG
Softmax, Sum
Student θ_{stu}
EMA
Teacher θ_{tch}
Flip
Flip, Jitter, Cutout
Flip, Scale
Source
Target
$|x_{src}|=M_{src}=16$
$|x_{tgt}|=M_{tgt}=32$


AT
L_{sup}
$L_{distill}$
Hard
SG
L_{align}
Image
Thresh, NMS
Student θ_{stu}
EMA
Teacher θ_{tch}
Flip, Jitter, Cutout
Flip, Jitter (1/2), Cutout (1/2)
Flip, Scale
Source
Target
$|x_{src}|=M_{src}=16$
$|x_{tgt}|=M_{tgt}=32$


UMT
L_{sup}
$L_{distill}$
Hard
SG
L_{align}
Img2Img
Thresh, NMS
Student θ_{stu}
EMA
Teacher θ_{tch}
CropPad, Jitter
Flip, Scale
Source
Target
$|x_{src}|=M_{src}=24$
$|x_{tgt}|=M_{tgt}=24$

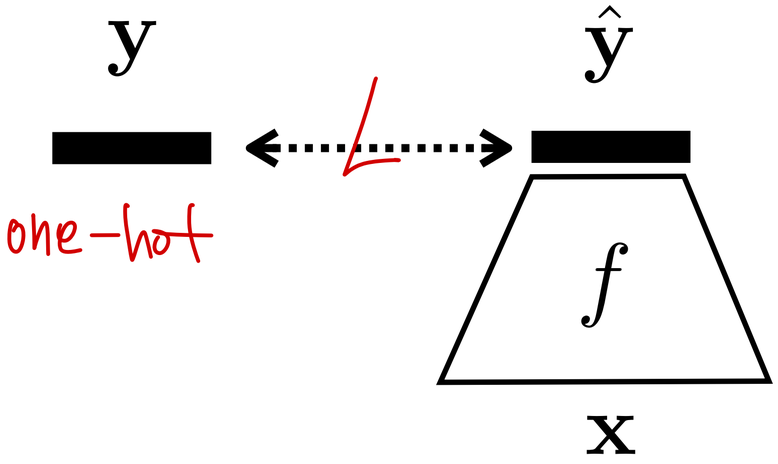

MIC
L_{sup}
$L_{distill}$
Hard
SG
L_{align}
SADA
Thresh, NMS
Student θ_{stu}
EMA
Teacher θ_{tch}
Flip
Flip, Jitter, MIC
Flip, Scale
Source
Target
$|x_{src}|=M_{src}=24$
$|x_{tgt}|=M_{tgt}=24$

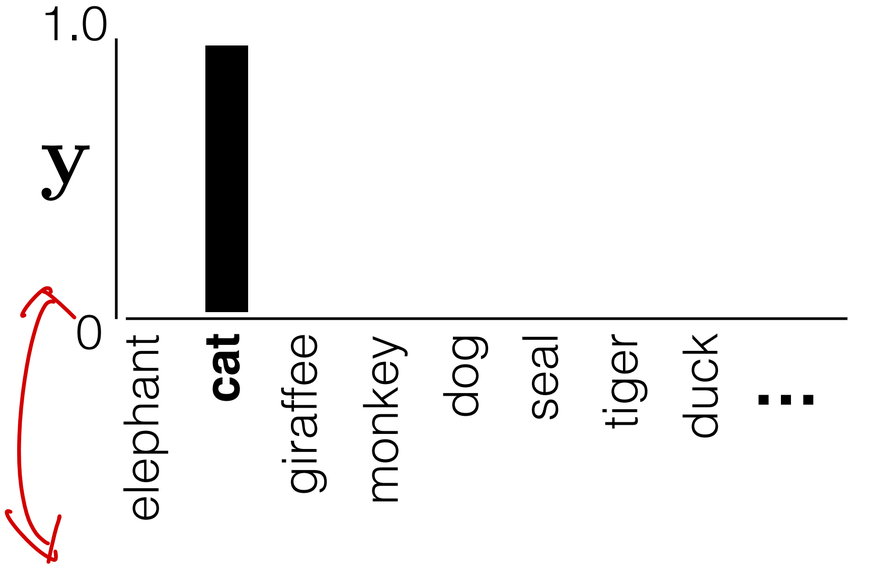
1. Knowledge about the mapping
2. **Knowledge about the outputs**
3. Knowledge about the inputs
4. Learning to learn fast

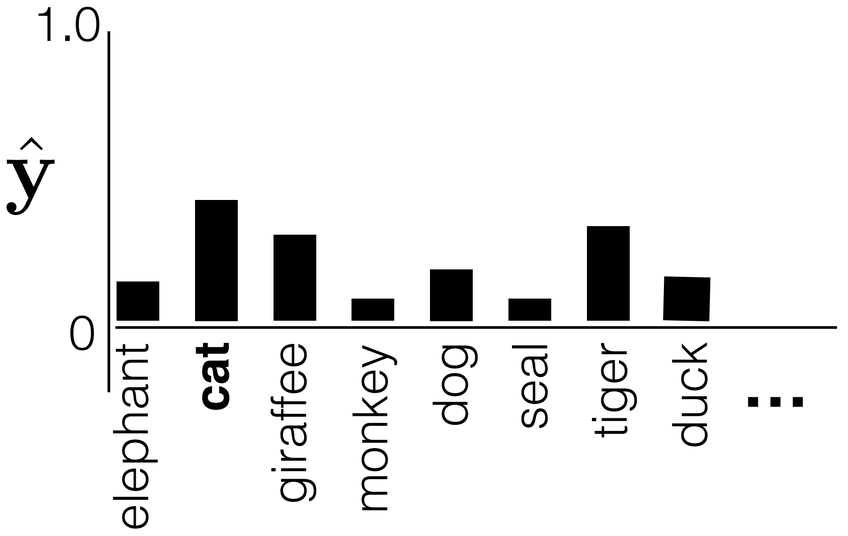


Transfer Learning

Recall: training a classifier with cross-entropy







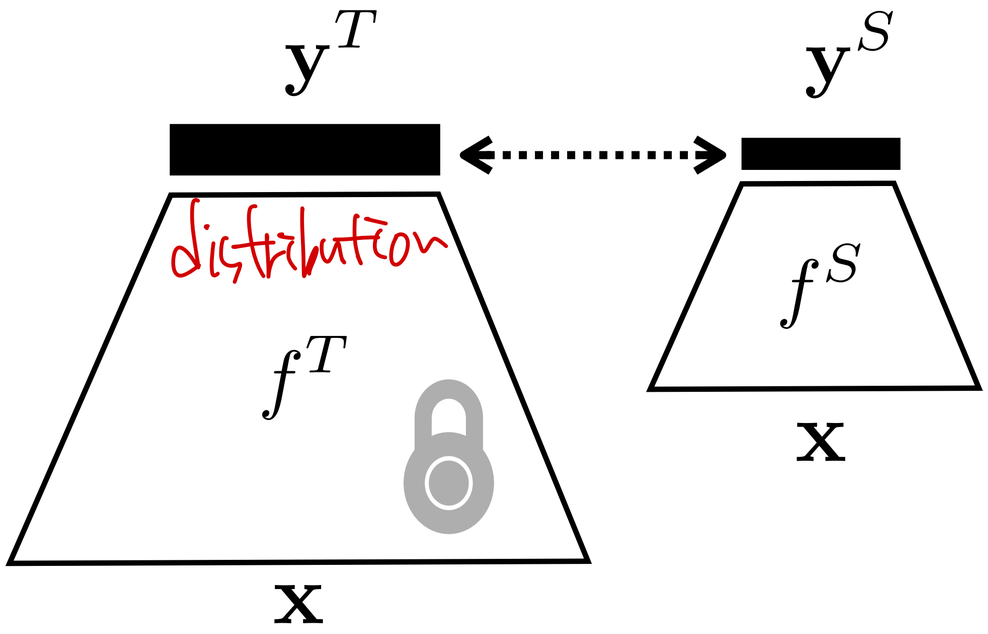
"Teacher" "Student"

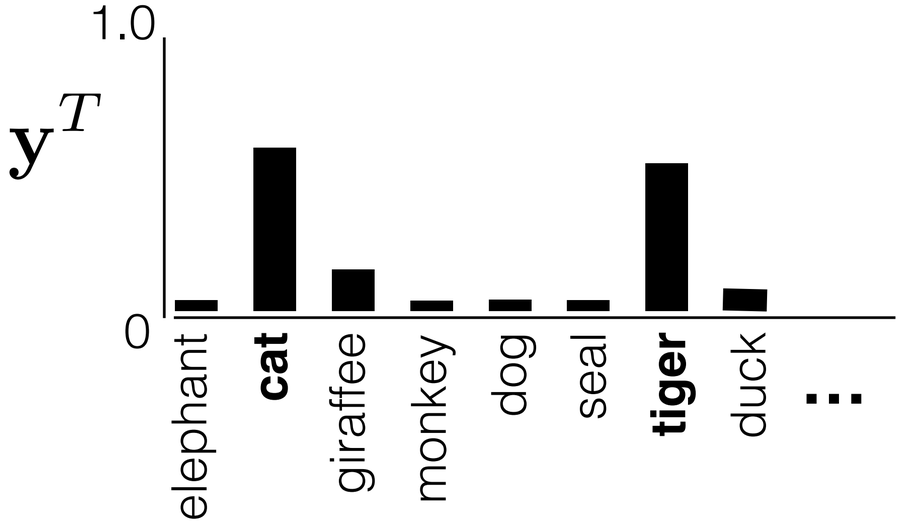
$$L(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_i \mathbf{y}_i \log \hat{\mathbf{y}}_i$$

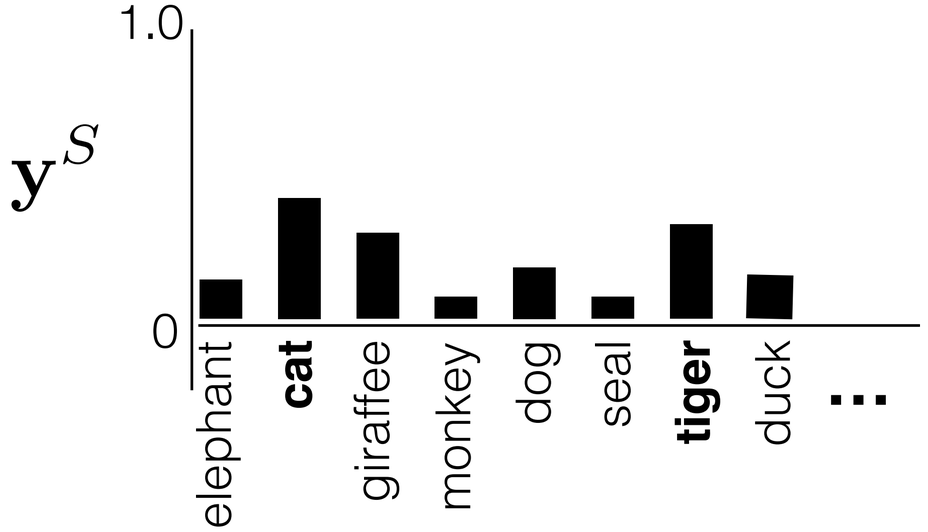
Knowledge Distillation [Hinton, Vinyals, Dean, 2015]

정답 대신 teacher model.

one-hot → distribution as input.







$$L(\mathbf{y}^S, \mathbf{y}^T) = -\sum_i \mathbf{y}_i^T \log \mathbf{y}_i^S$$

Knowledge Distillation [Hinton, Vinyals, Dean, 2015]

- The "label" you get for each input is more informative than just the ground truth class
- The "label" (**soft target**) says not just "this is a dog" but "this is a dog that looks a bit like a cat and very different from a bear" (so maybe it's a small dog).

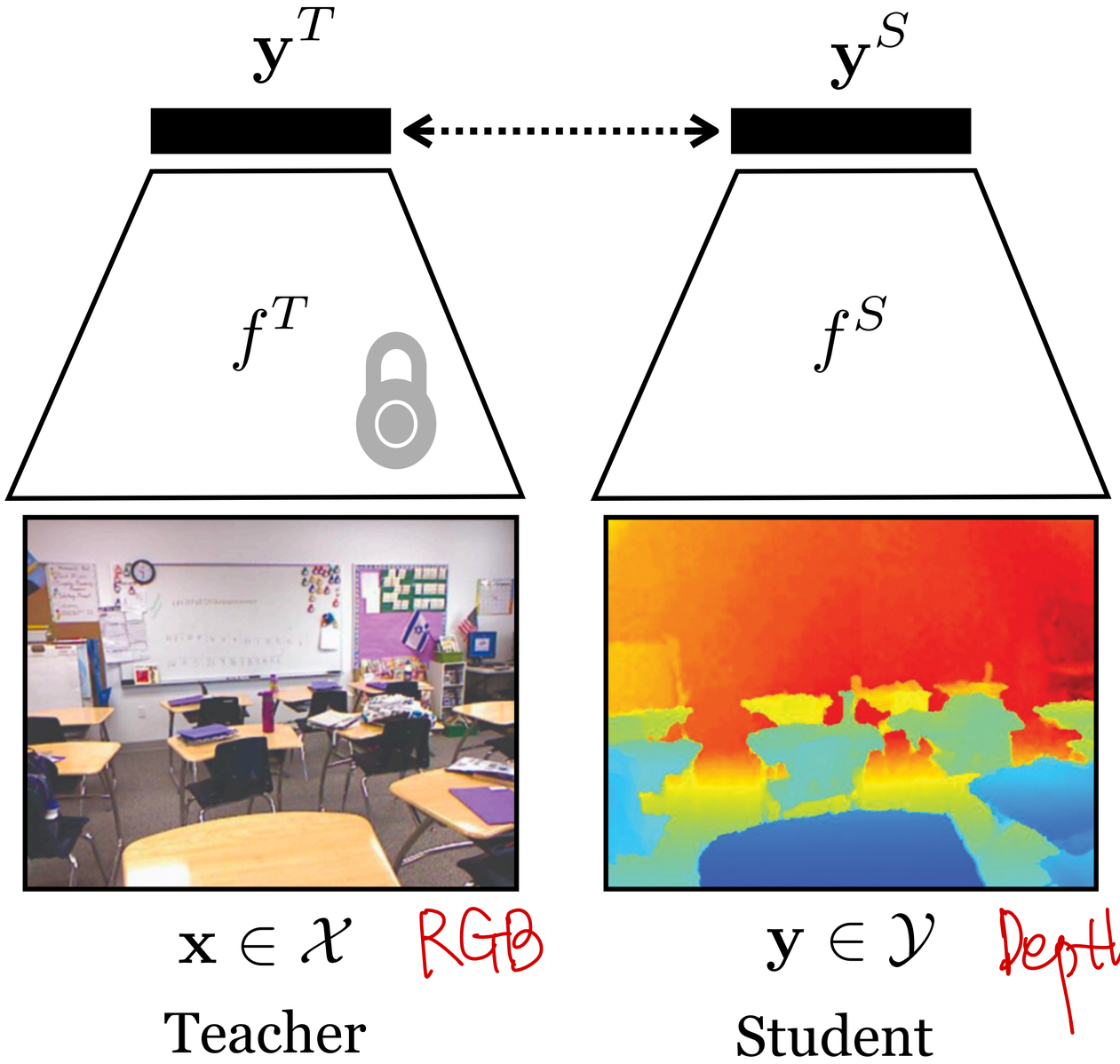
Hinton paper shows lower train accuracy but higher test accuracy on speech recognition with soft targets

System & training set	Train Frame Accuracy	Test Frame Accuracy
Baseline (100% of training set)	63.4%	58.9%
Baseline (3% of training set)	67.3%	44.5%
Soft Targets (3% of training set)	65.4%	57.0%

Table 5: Soft targets allow a new model to generalize well from only 3% of the training set. The soft targets are obtained by training on the full training set.

결론: 더 많은 정보를 가지고 있다.

Cross-Modal Distillation [Gupta, Hoffman, Malik, 2015]

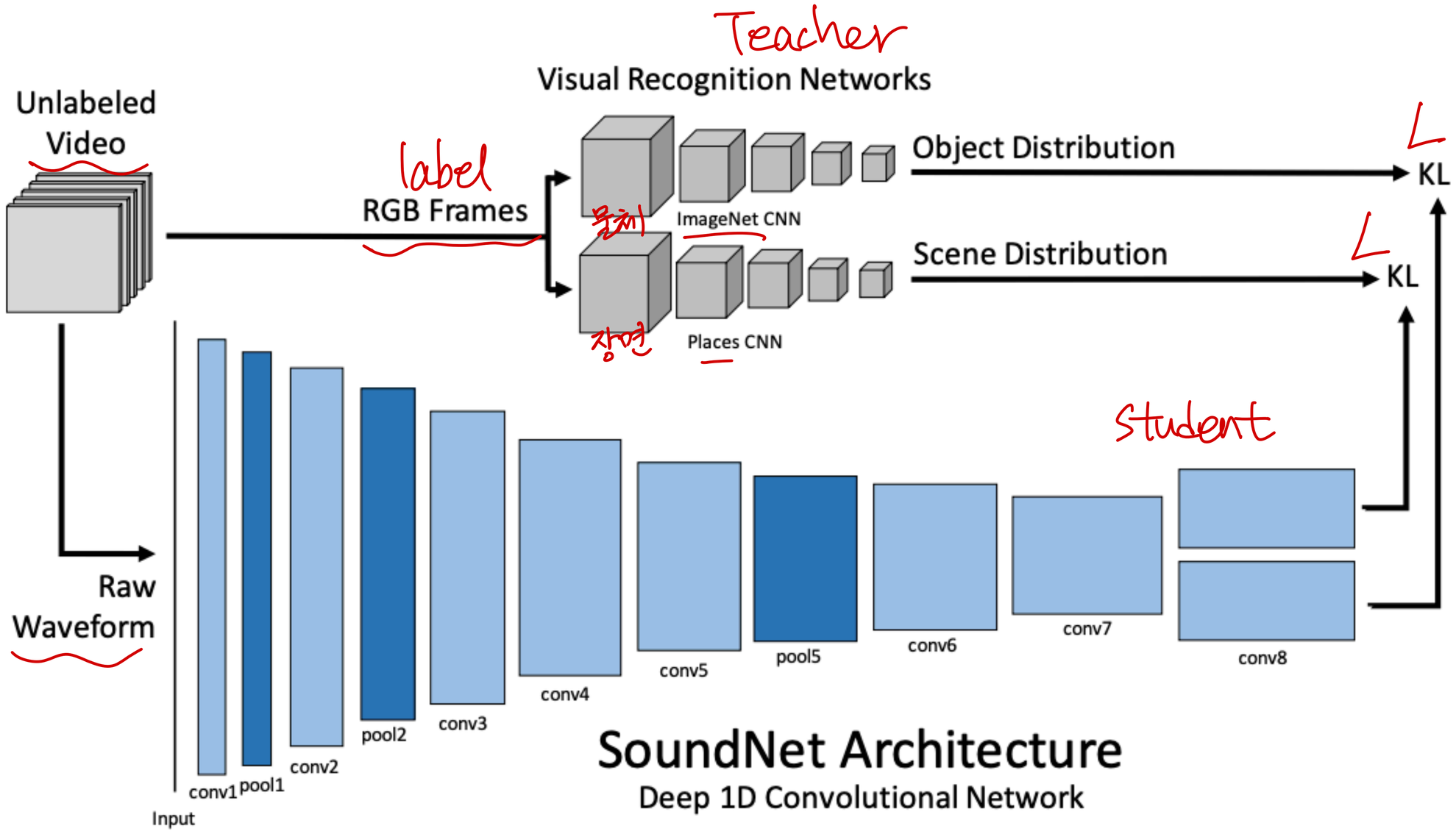


서로 다른 데이터 형식 간 distill

Useful if you have labeled data for one domain but not the other.

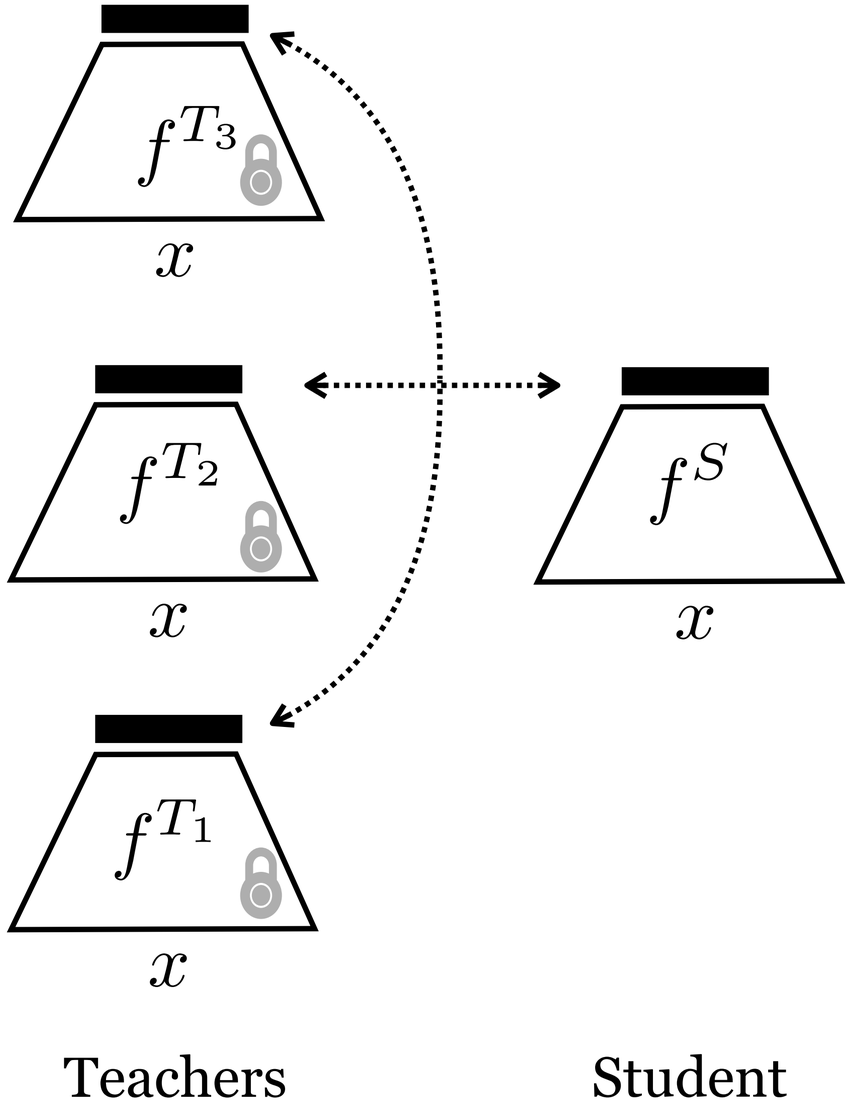
Cross-Modal Distillation: SoundNet

[Aytar*, Vondrick*, Torralba, 2016]

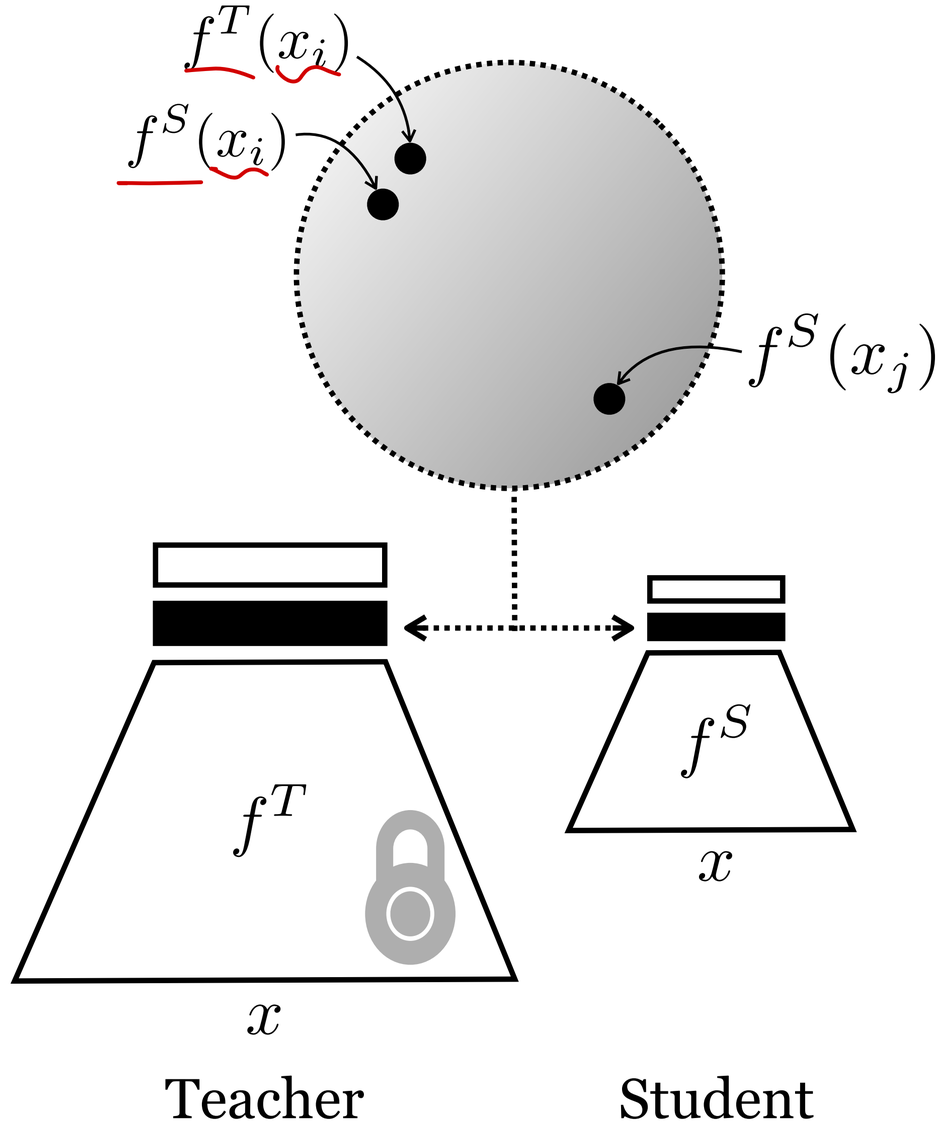


Distilling an ensemble

- Ensembling model outputs creates more accurate estimates.
- Why? Each model is imperfect in different random ways. Errors cancel out, truth is shared.
- Distill ensemble into a single model for fast inference.



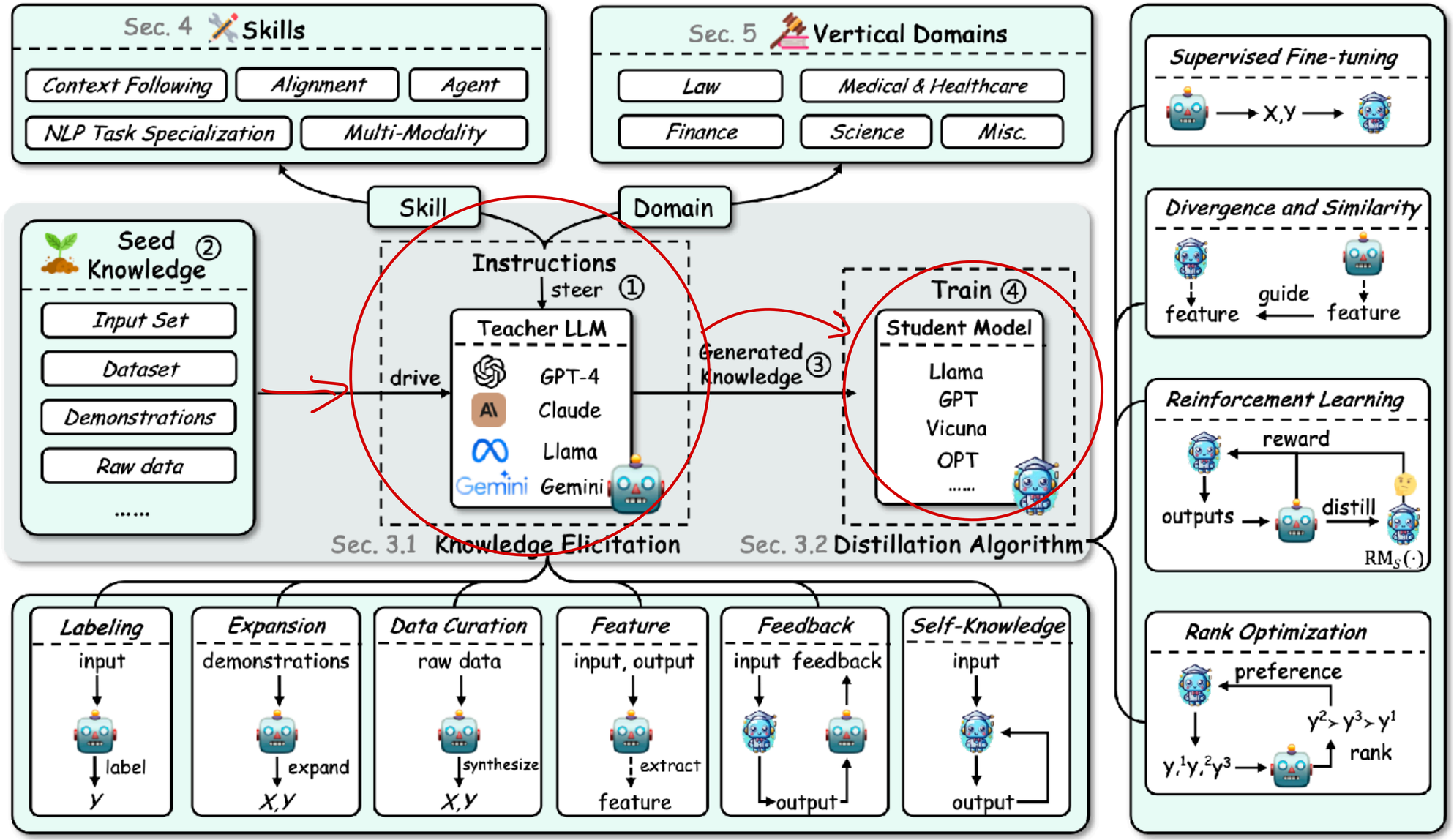
Contrastive Representation Distillation [Tian, Krishnan, Isola, 2020]



(T's view → S's view)
어떠한 관점변화 → 임베딩 변화 X

- Contrastive learning supervises embeddings to be invariant to some viewing transformation.
- The "viewing transformation" can be "teacher's view" (big net) vs "students view" (small net).
- Then the student will output embeddings that are the same as the teacher's.

“Distills” - Distillation in practice in the era of foundation models



https://arxiv.org/abs/2402.13116

Compression vs. Distillation

- Both give us smaller models with similar accuracy to large ones
- Distillation uses SGD to train a smaller model to match characteristics of a larger one
- Compression uses ***pruning*** or ***quantization*** to reduce the computational cost of a large model

Algorithm 1 Prune $k \leq d_{\text{col}}$ weights from row $\mathbf{w}$ with inverse Hessian $\mathbf{H}^{-1} = (2\mathbf{X}\mathbf{X}^\top)^{-1}$ according to OBS in $O(k \cdot d_{\text{col}}^2)$ time.

$M = \{1, \ldots, d_{\text{col}}\}$
for $i = 1, \ldots, k$ **do**
  $p \leftarrow \text{argmin}_{p \in M} \frac{1}{[\mathbf{H}^{-1}]_{pp}} \cdot w_p^2$
  $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{H}^{-1}_{:,p} \frac{1}{[\mathbf{H}^{-1}]_{pp}} \cdot w_p$
  $\mathbf{H}^{-1} \leftarrow \mathbf{H}^{-1} - \frac{1}{[\mathbf{H}^{-1}]_{pp}} \mathbf{H}^{-1}_{:,p} \mathbf{H}^{-1}_{p,:}$
  $M \leftarrow M - \{p\}$
end for

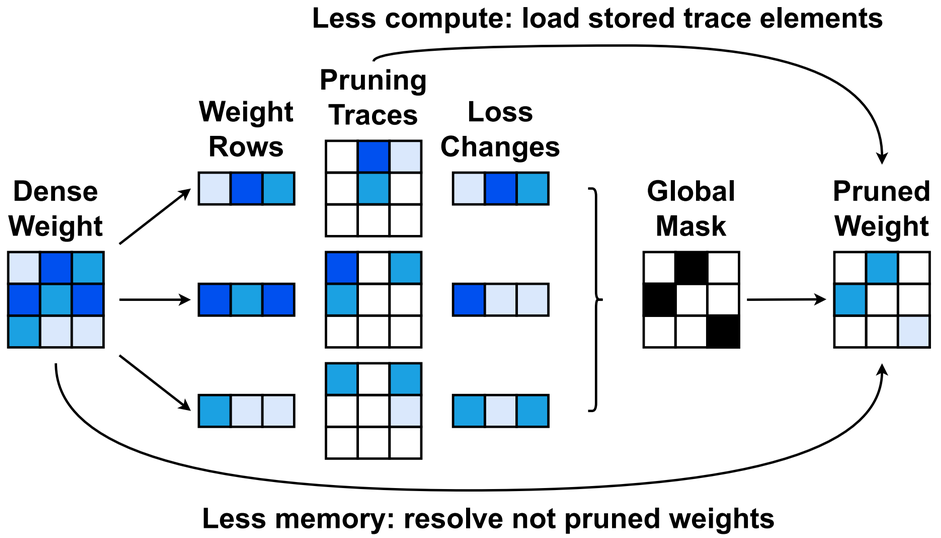
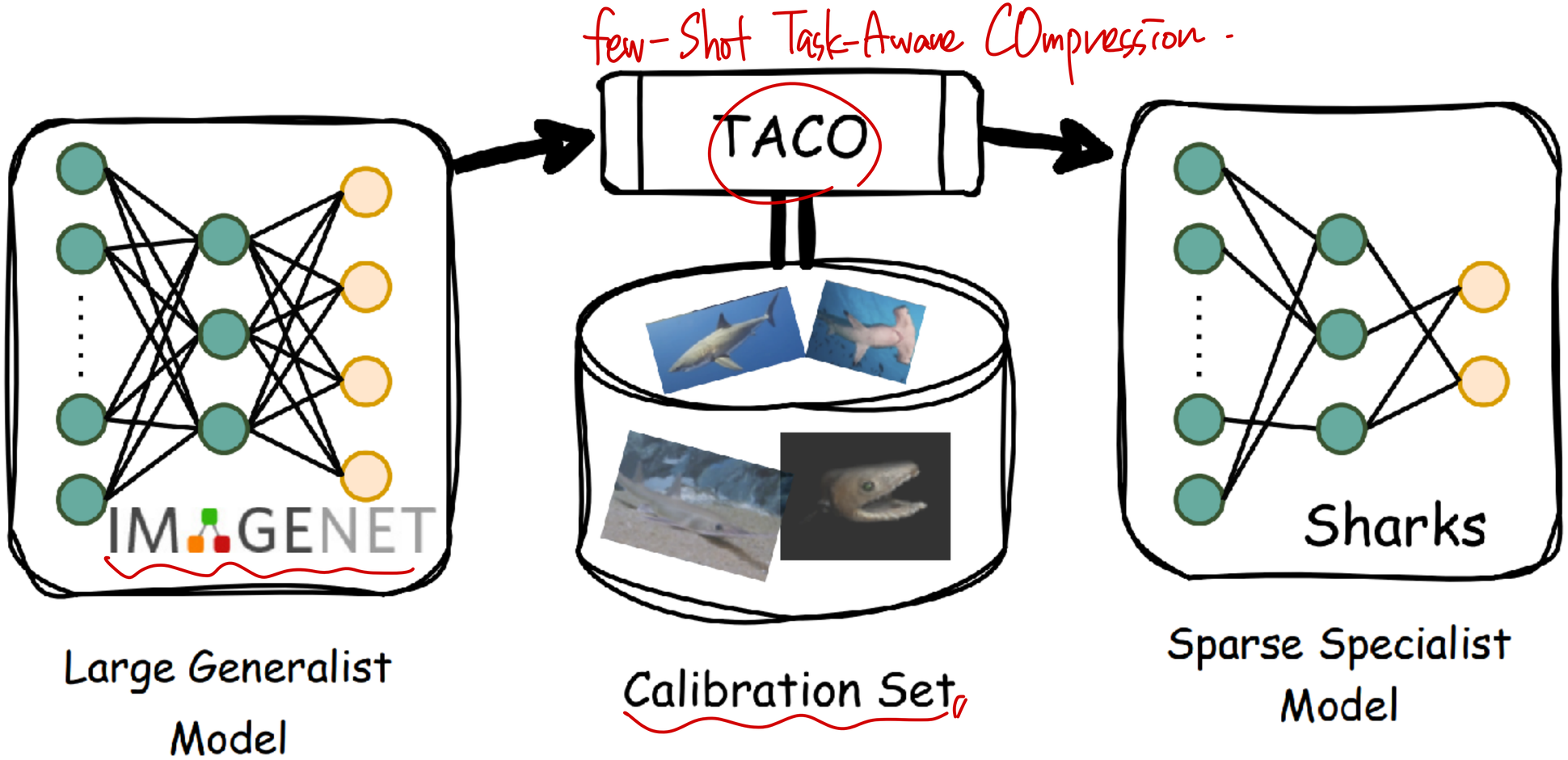


Figure 1: Efficient global OBS using the row-wise results.

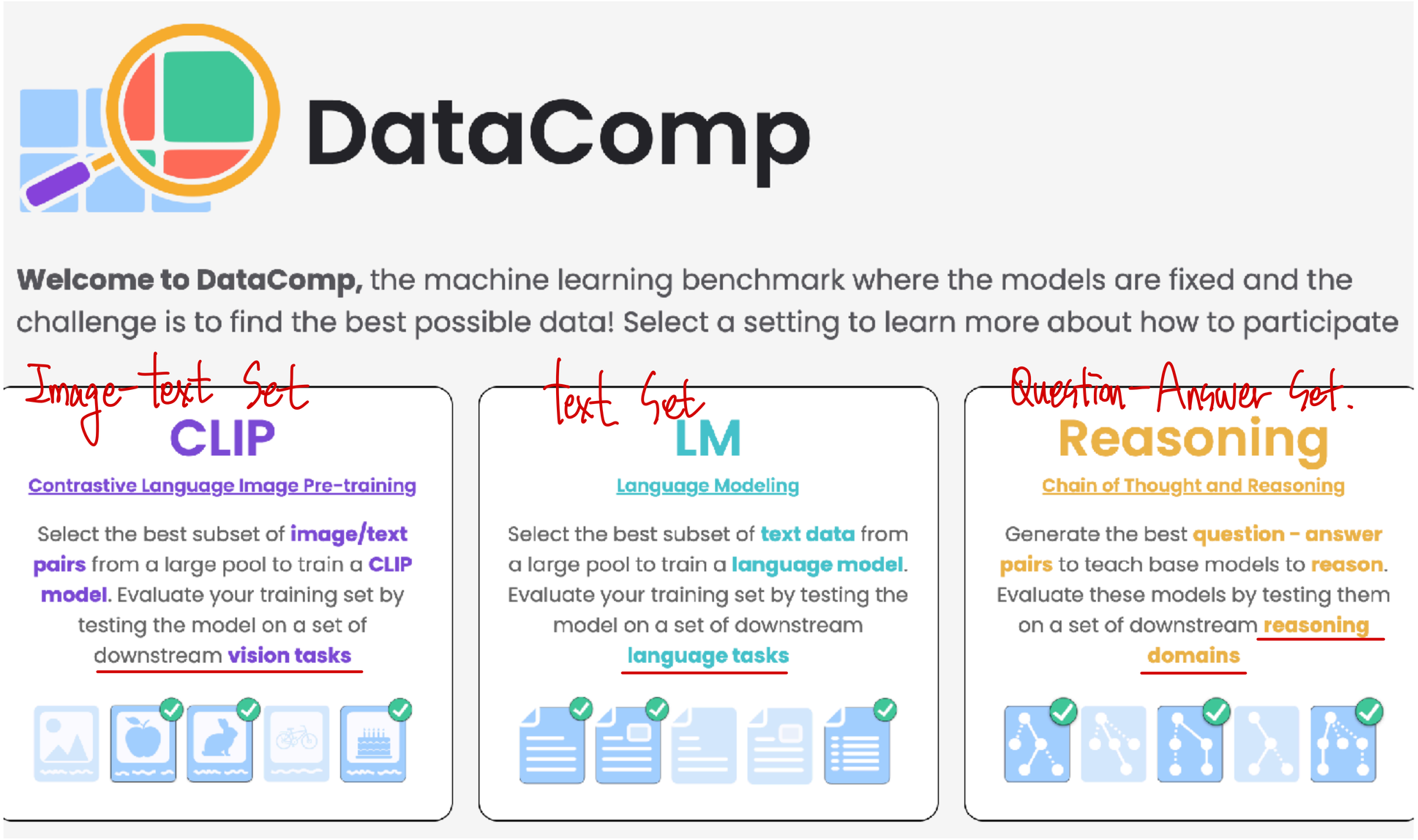
Optimal Brain Compression: A Framework for Accurate Post-Training Quantization and Pruning, Frantar et al., 2023

We almost never need a truly generalist model in practice. Specialization can enable much higher rates of compression with the same accuracy.



Vision Models Can Be Efficiently Specialized via Few-Shot Task-Aware Compression, Kuznedelev et al., 2023

What data is most useful when fine-tuning?

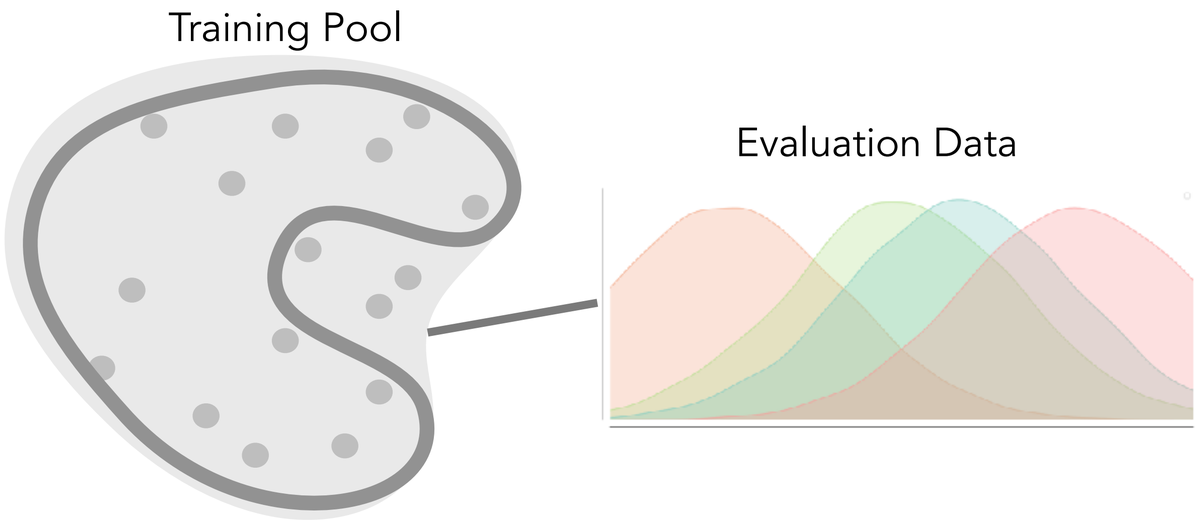


https://www.datacomp.ai/

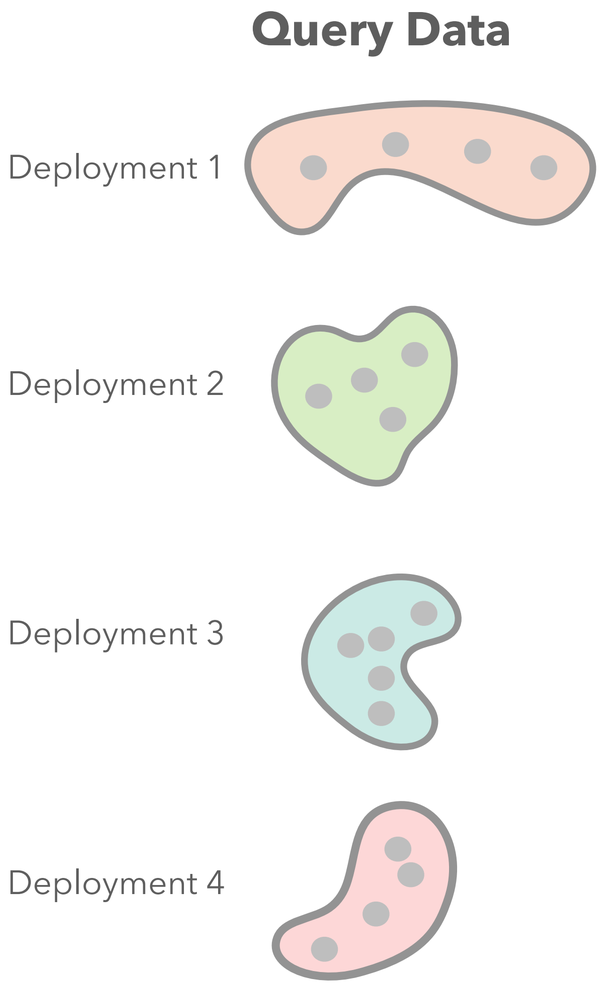
Curation for generalization vs. specialization

data subset selection으로부터 형성된다.

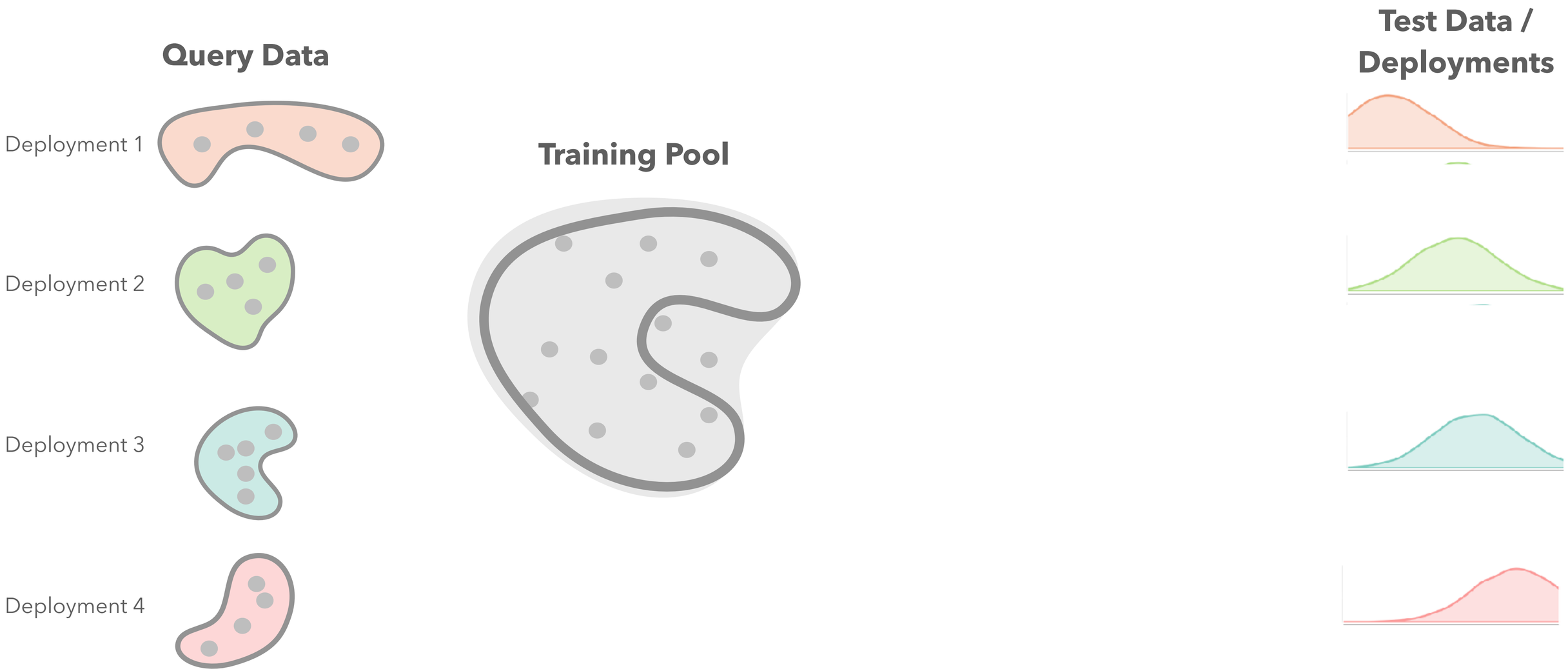
Foundation Model Setup:
Subset selection for Model *Generalization*



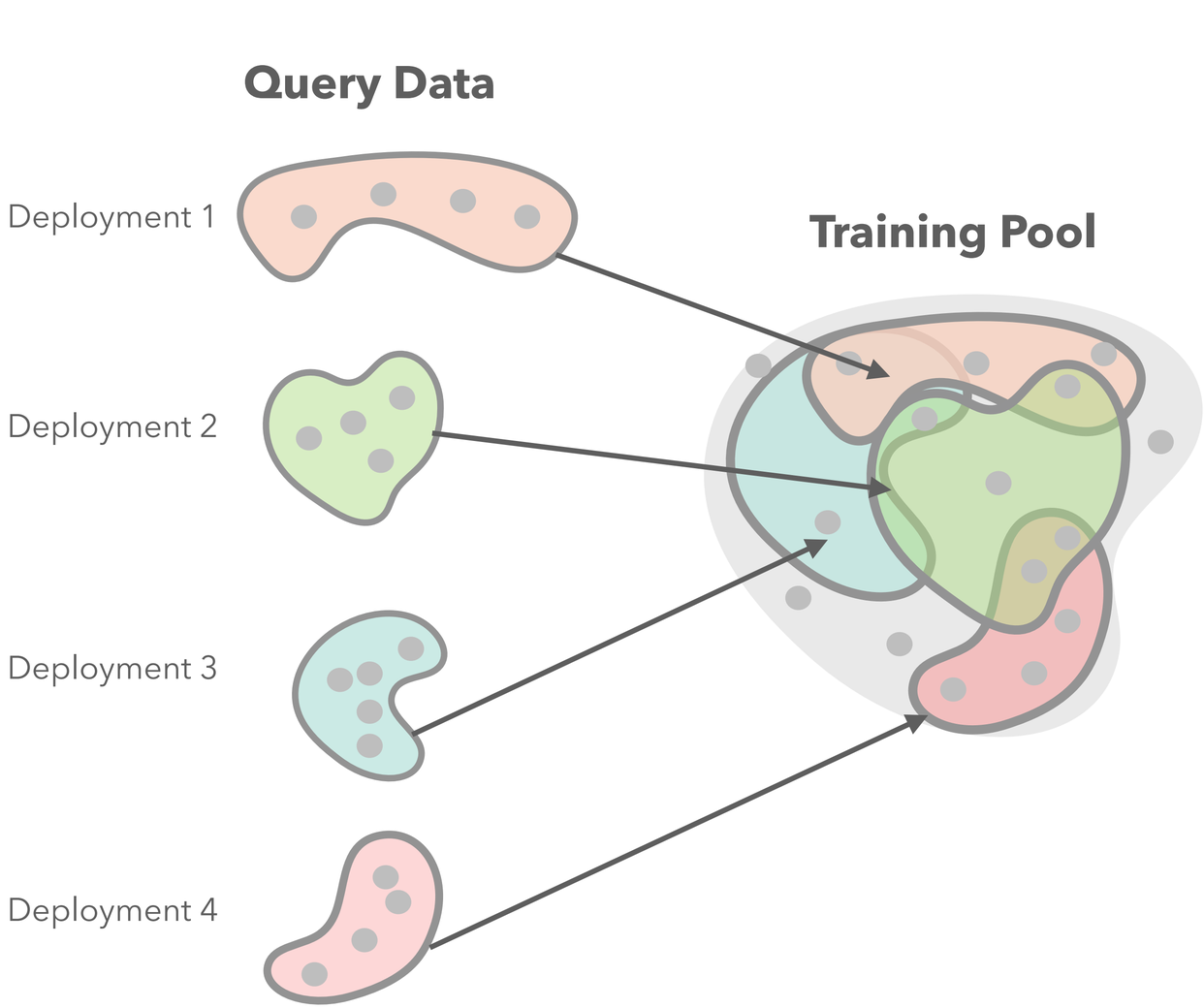
Dataset subset selection for specialization

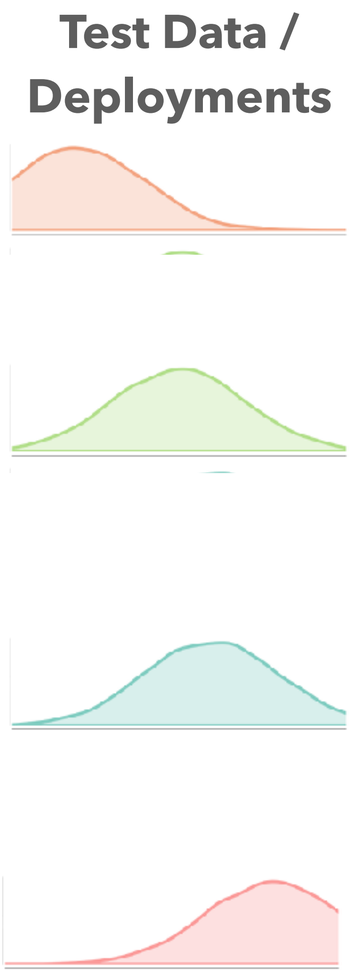


Dataset subset selection for specialization

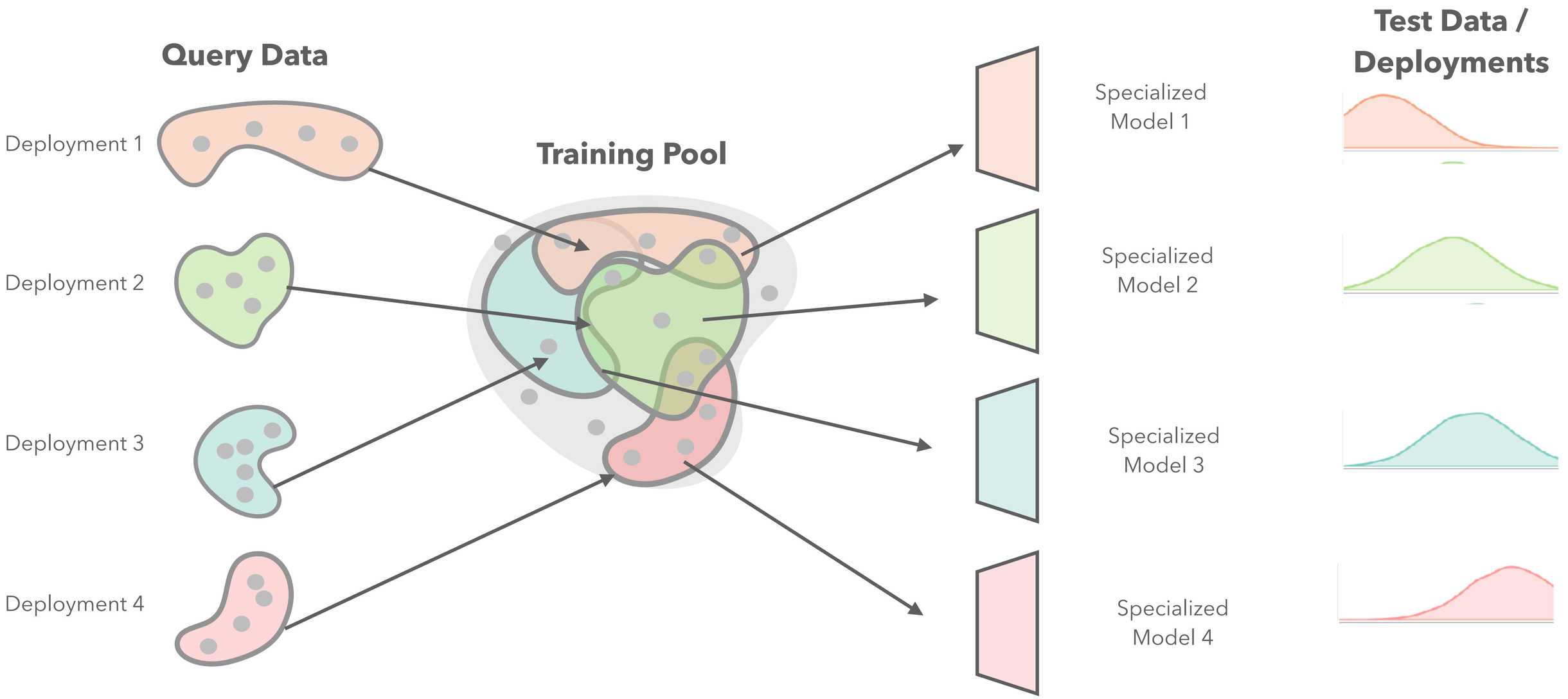


Dataset subset selection for specialization

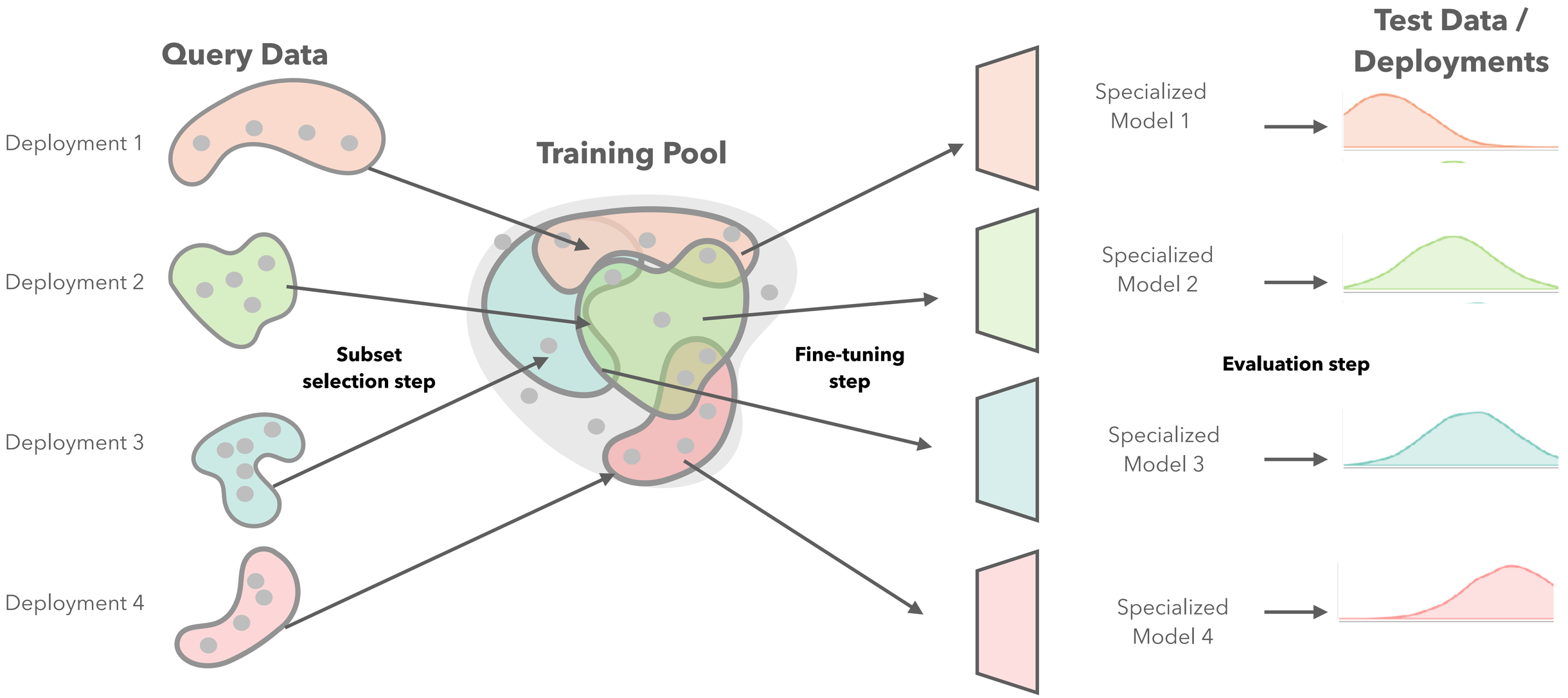




Dataset subset selection for specialization



Dataset subset selection for specialization



New benchmark: Data S^3

Subset Selection for Specialization.

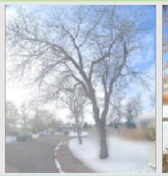
Dataset	AutoArborist	iWildCam	GeoDE	NuScenes	FishDetection
Input	Urban street images	Camera trap	Objects images	Driving footage	Underwater Videos
Prediction	Tree species	Animal species	Object recognition	Steering Angle	Fish Bounding Boxes
Train Pool	781k imgs, 304 classes	130k imgs, 182 classes	53k imgs, 40 classes	100k imgs	781k imgs
Deploy 1	Surrey	Southeast Asia	Indonesia	Seaport, Boston	Freshwater, PNW Lakes
Deploy 2	Calgary	Central America	Nigeria	Queenstown, Singapore	Pacific Ocean
Deploy 3	Los Angeles	Southern Africa	Indoor Objects	One North, Singapore	East Atlantic Ocean
Deploy 4	Washington DC	Central Africa	Outdoor Objects	Holland Village, Singapore	Gulf of Mexico

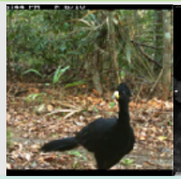


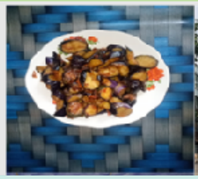


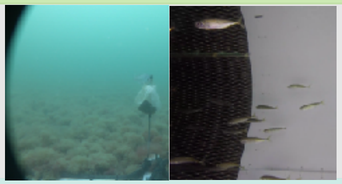








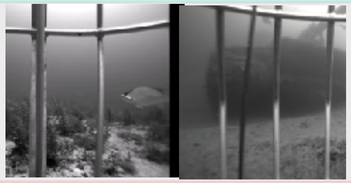


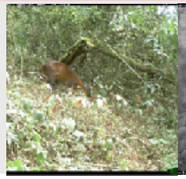





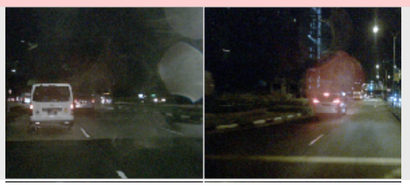
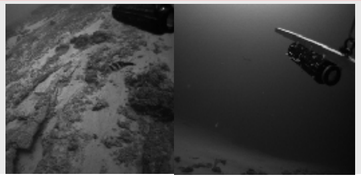




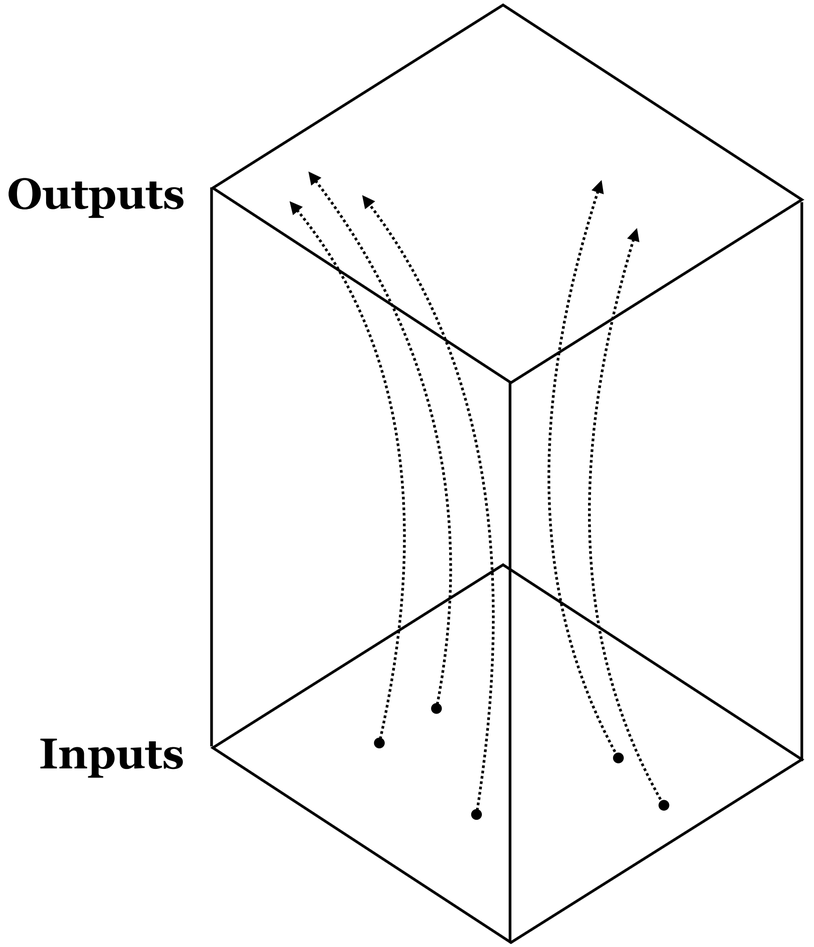


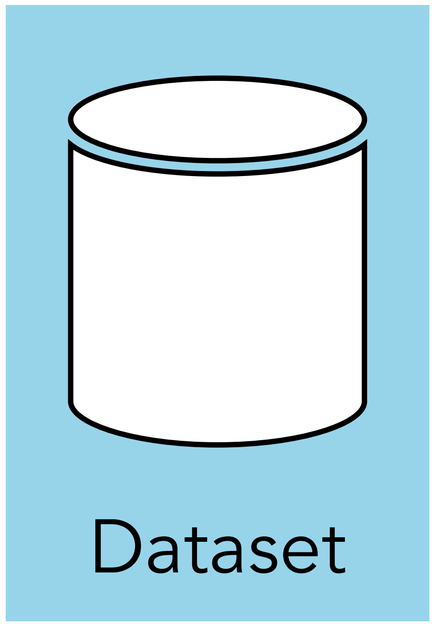


1. Knowledge about the mapping
2. Knowledge about the outputs
3. **Knowledge about the inputs**
4. Learning to learn fast

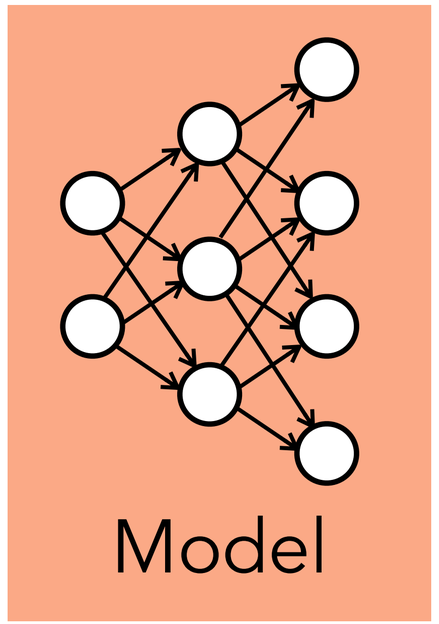


Transfer Learning



Dataset

Generative modeling



Model

data

data (++)

보안 생성 변형 ...

"This release is the culmination of many hours of collective effort to create a single file that compresses the visual information of humanity into a few gigabytes."

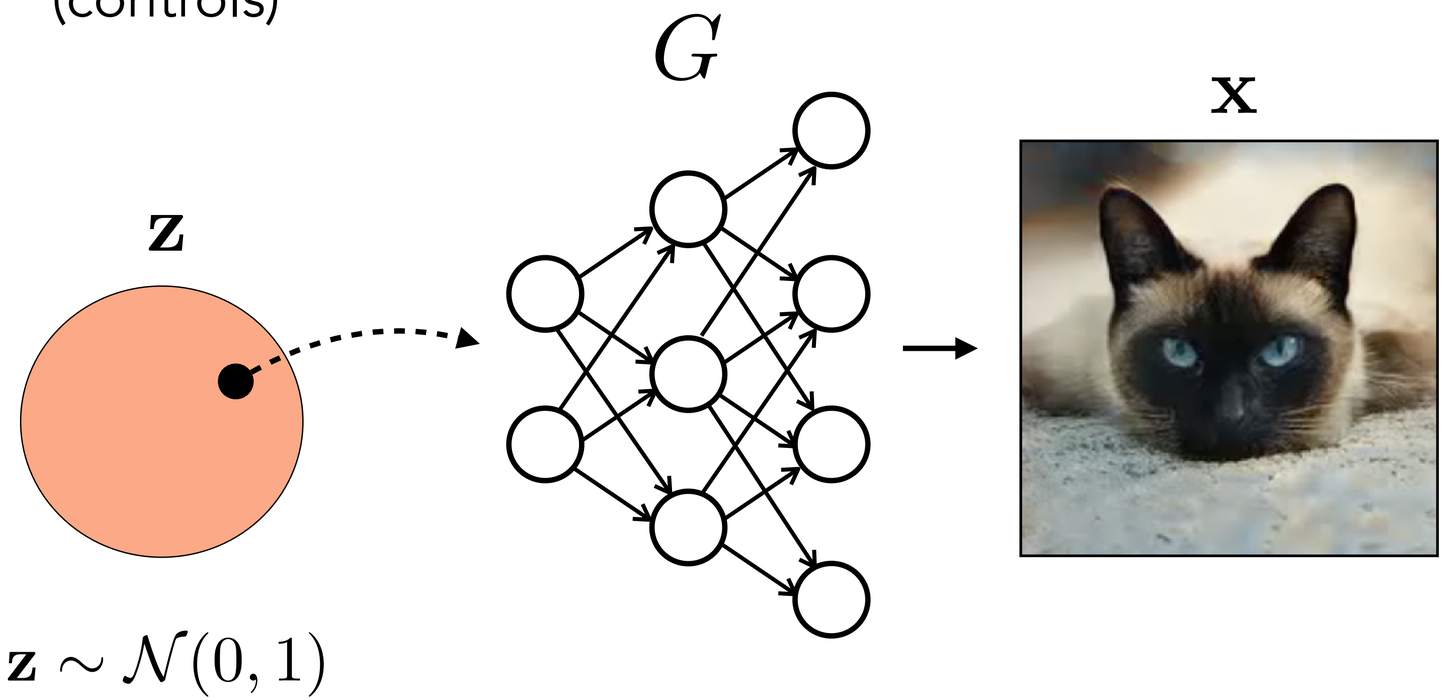
Emad Mostaque [https://stability.ai/blog/stable-diffusion-public-release]

Generative Models

Latent variables

(controls)

Synthesized Image

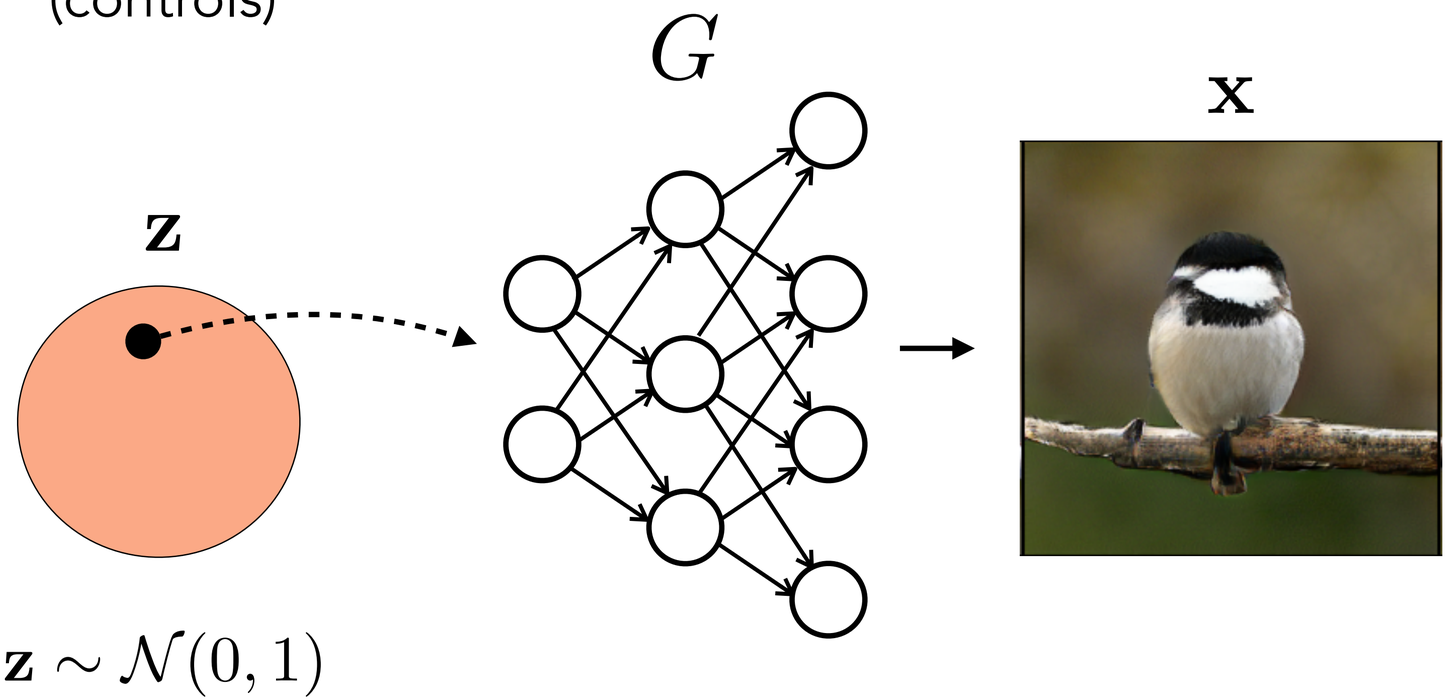


Generative Models

Latent variables

(controls)

Synthesized Image

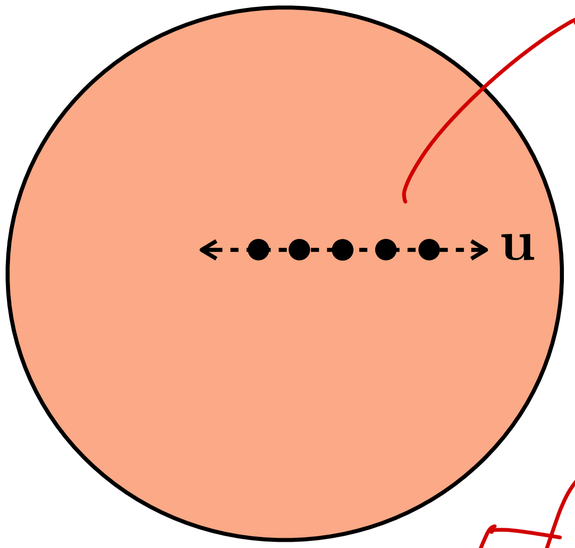


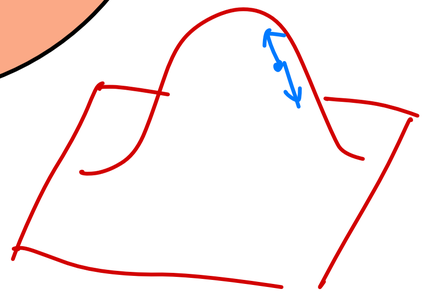
Synthesized Images

Latent variables

(controls)

latent distribution의 특정 값과 새의 방향과 Mapping.







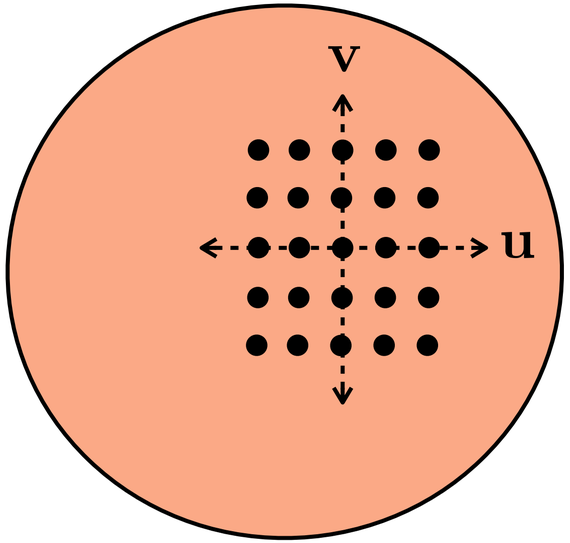
v

u

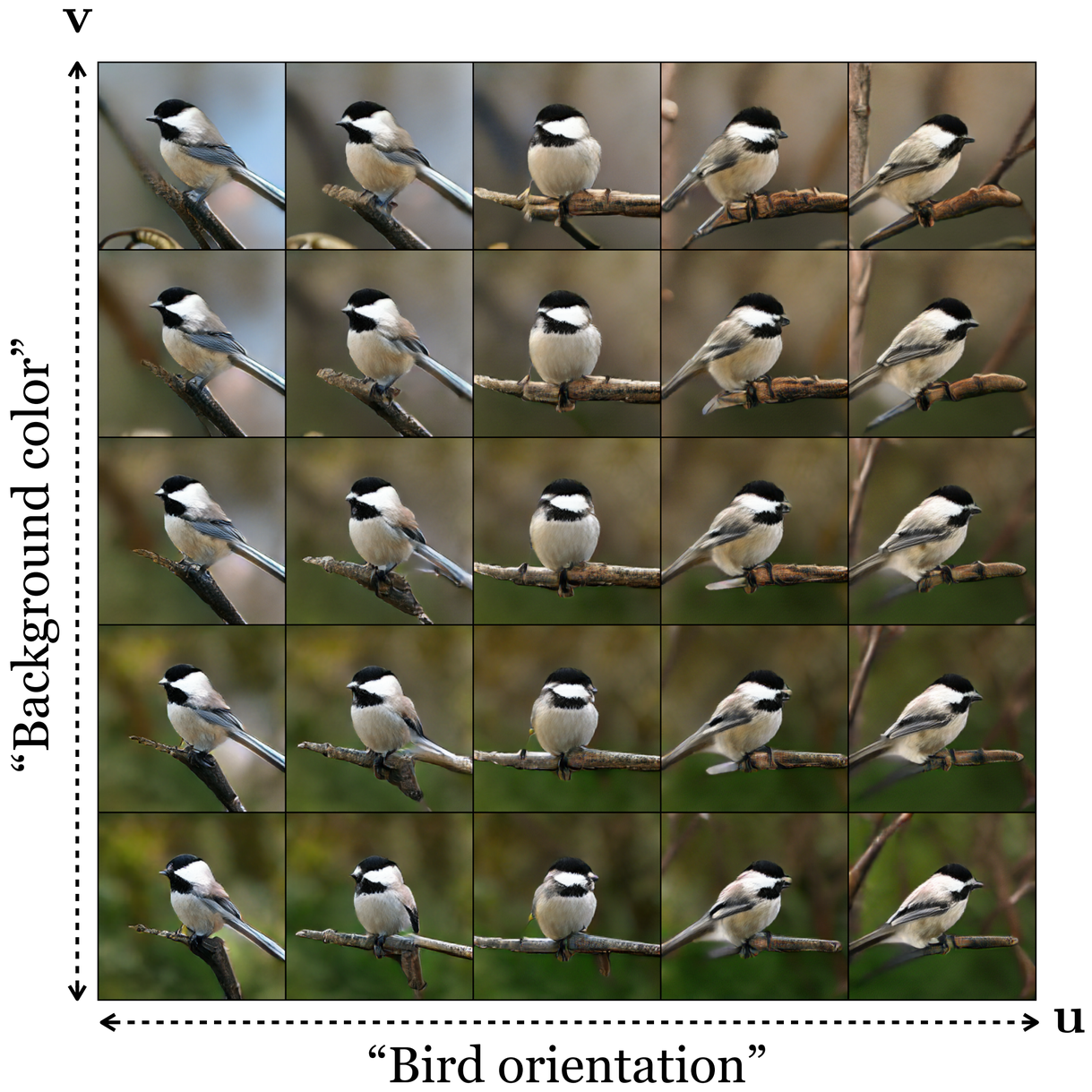
"Bird orientation"

Latent variables

(controls)

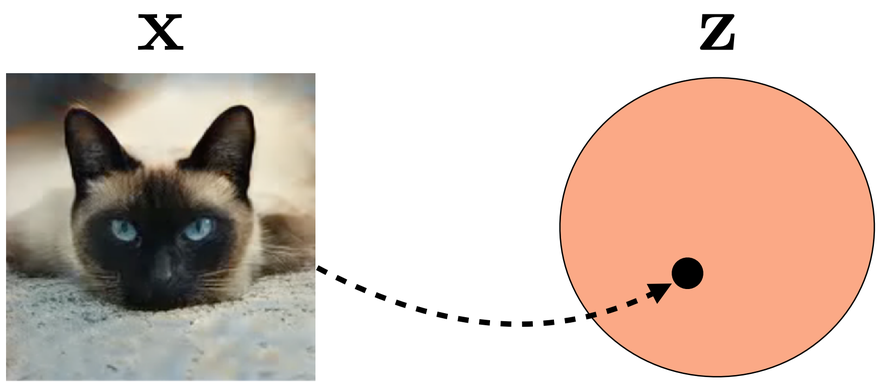


Synthesized Images



Data++ supports counterfactual reasoning

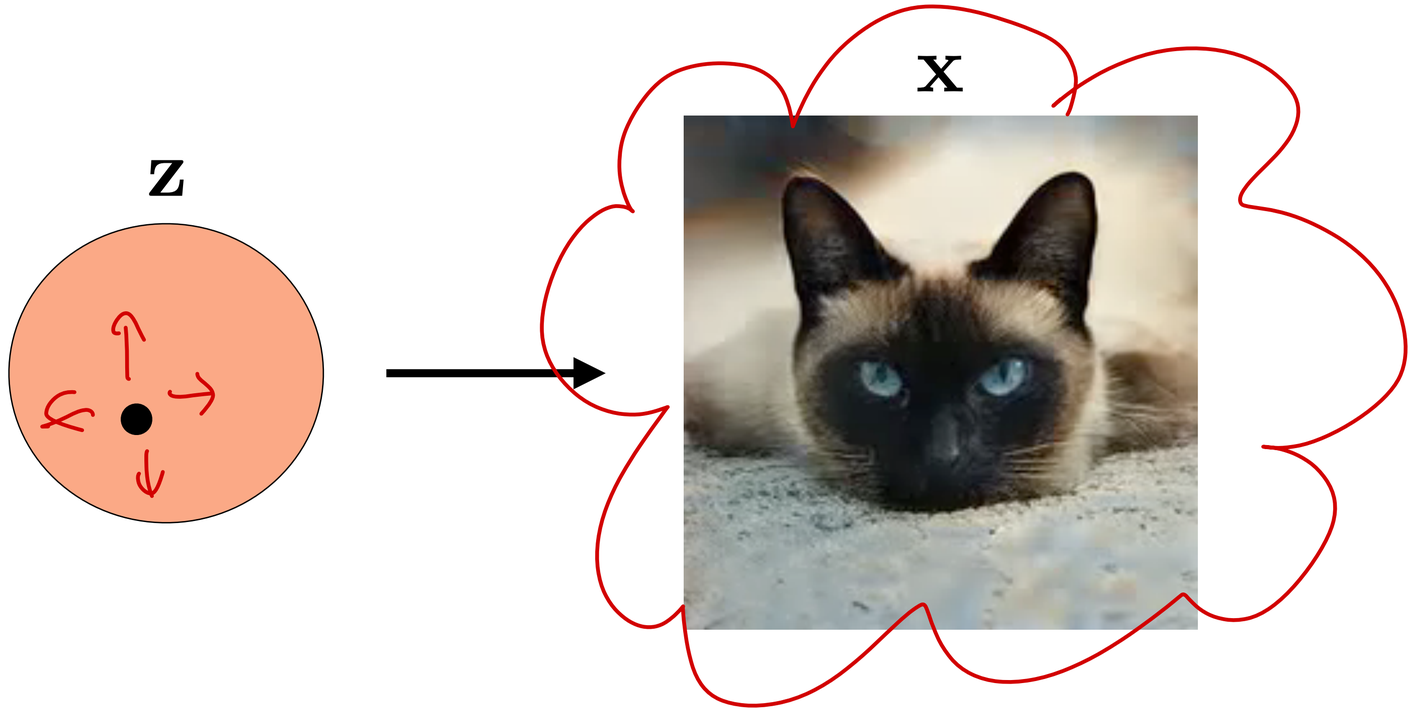
"What would it look like if…"



반사실적

Data++ supports counterfactual reasoning

"What would it look like if…"



latent variable은 어떻게 생겼냐 → 비현실 image도 있을수있다.

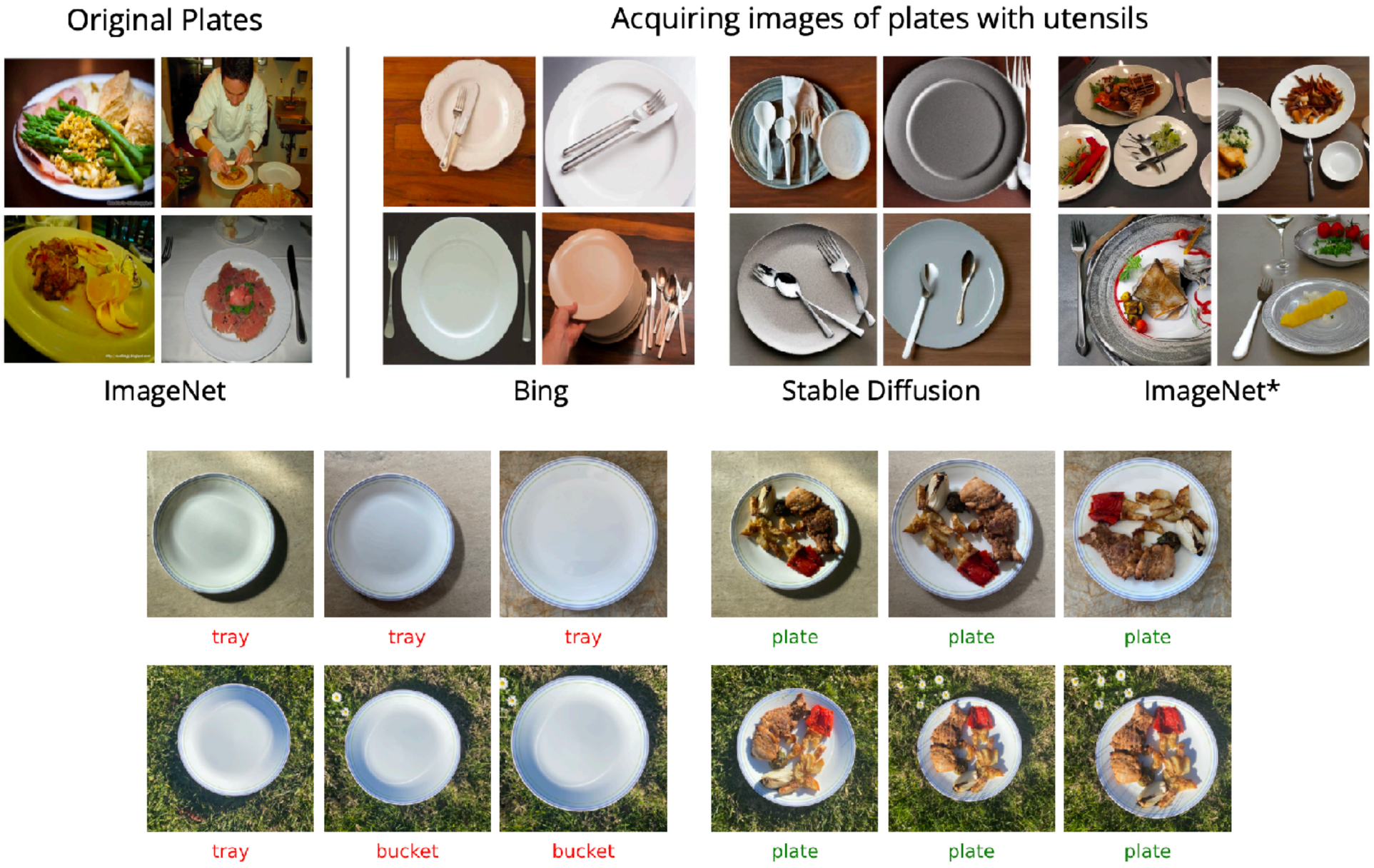
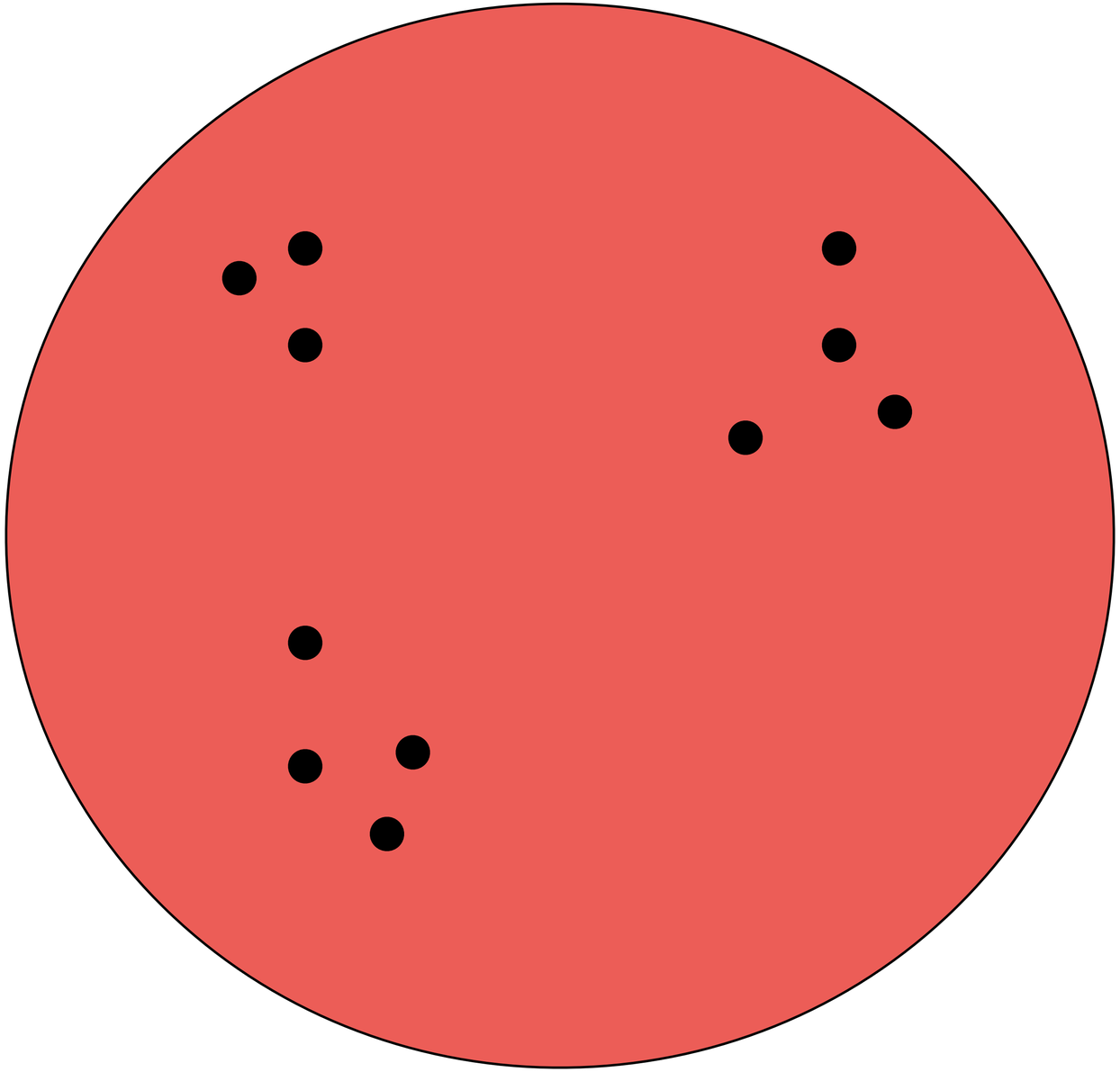


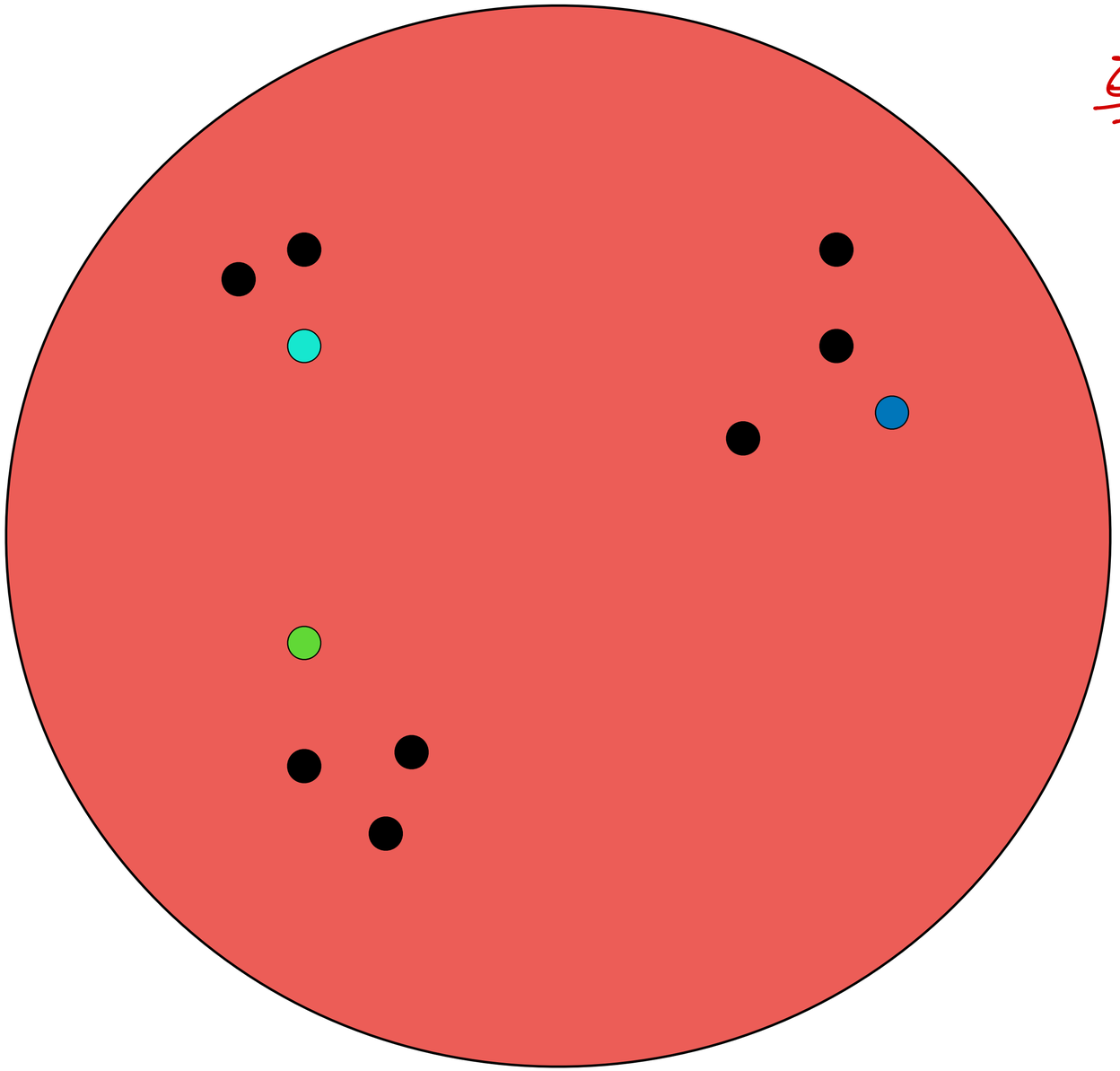
Figure 8: Real images of plates, with and without food and either on a table or in the grass. Below each image is the predicted class by an ImageNet-trained ResNet50.

https://arxiv.org/abs/2302.07865

Labeling Data++



Labeling Data++

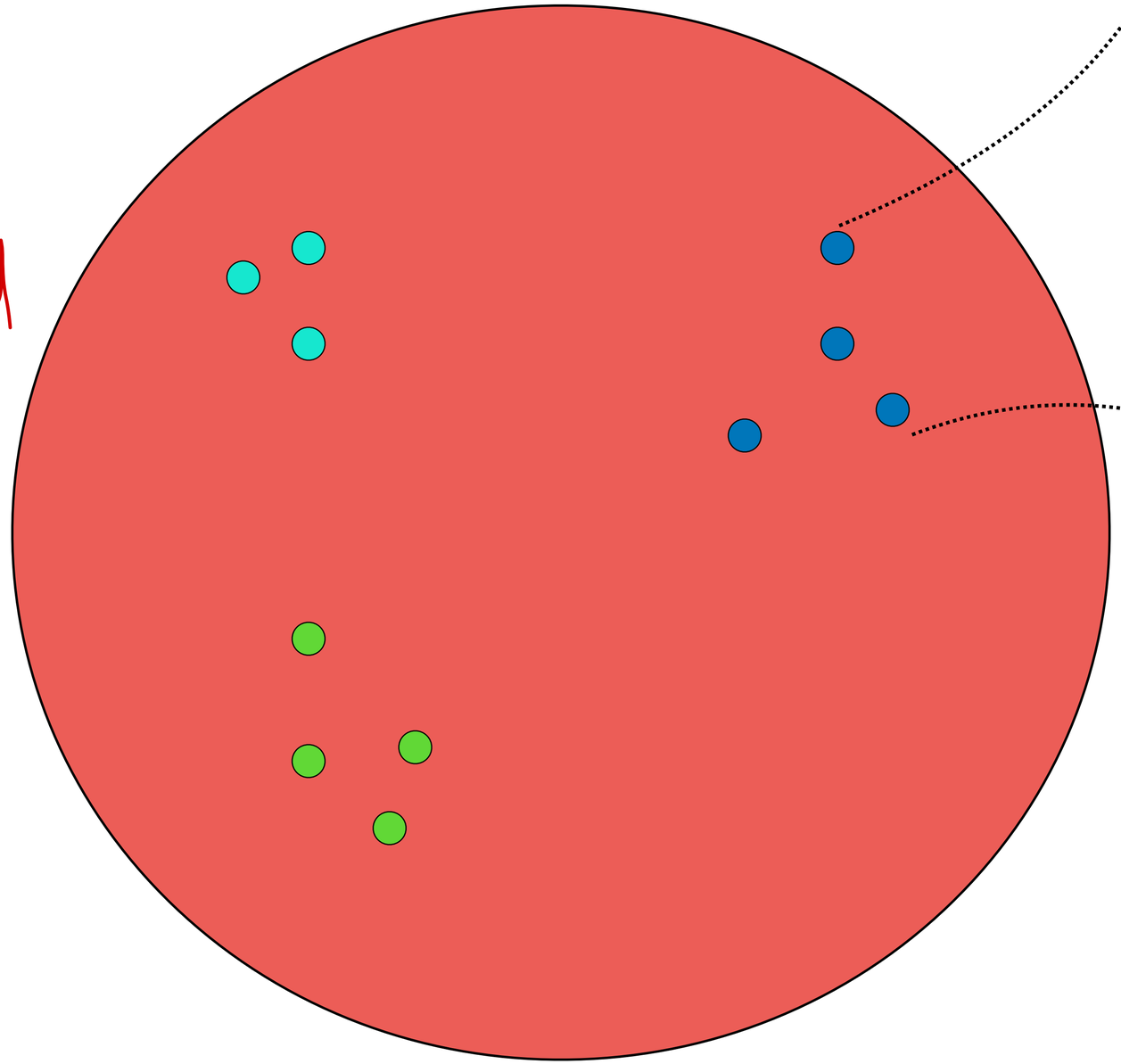


특정 variable의 labeling.

Labeling Data++

latent space 에서

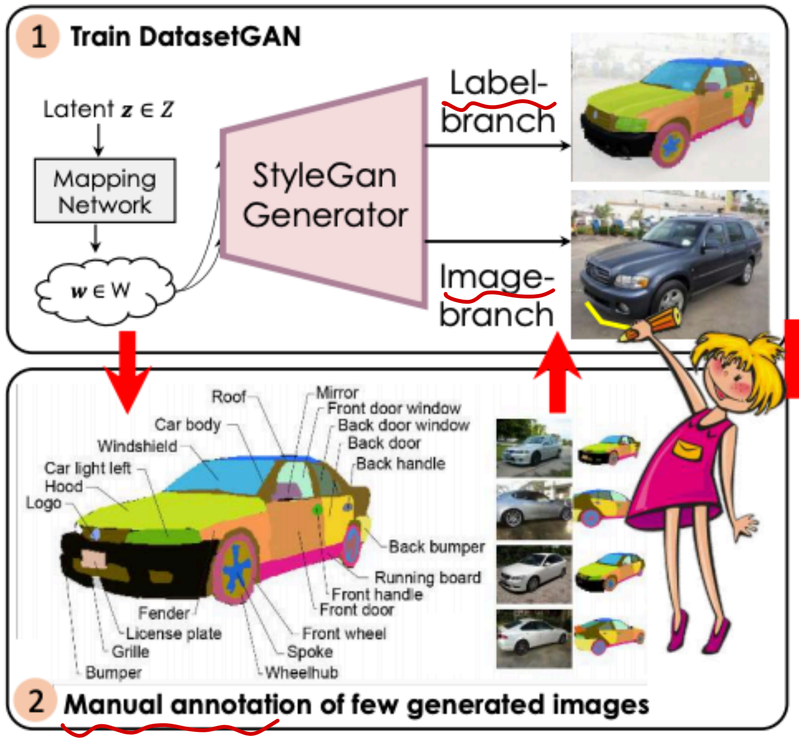
인접할 = 같은 label





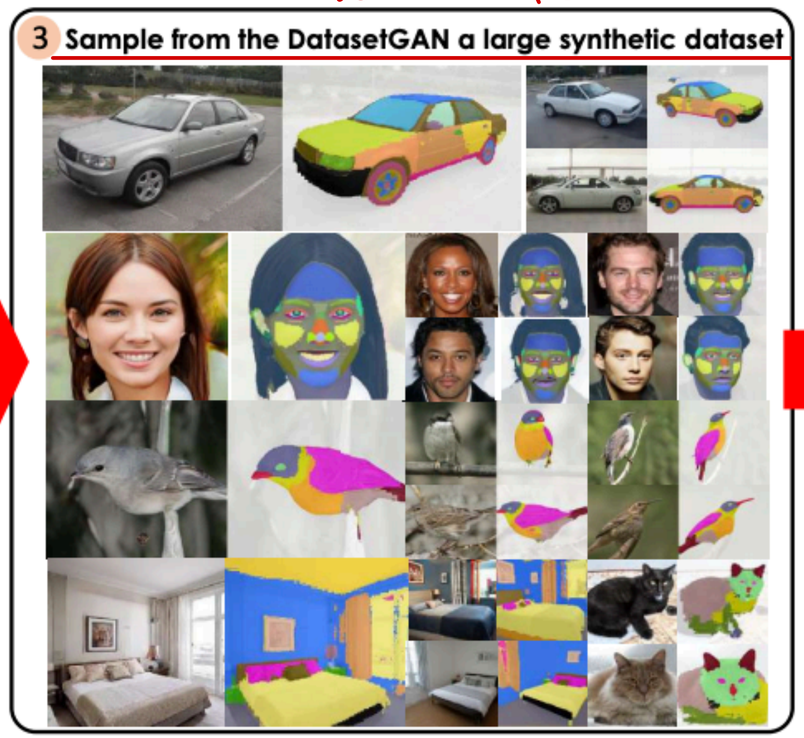
DatasetGAN

[Zhang*, Ling*, Gao, Yin, Lafleche, Barriuso, Torralba, Fidler 2021]

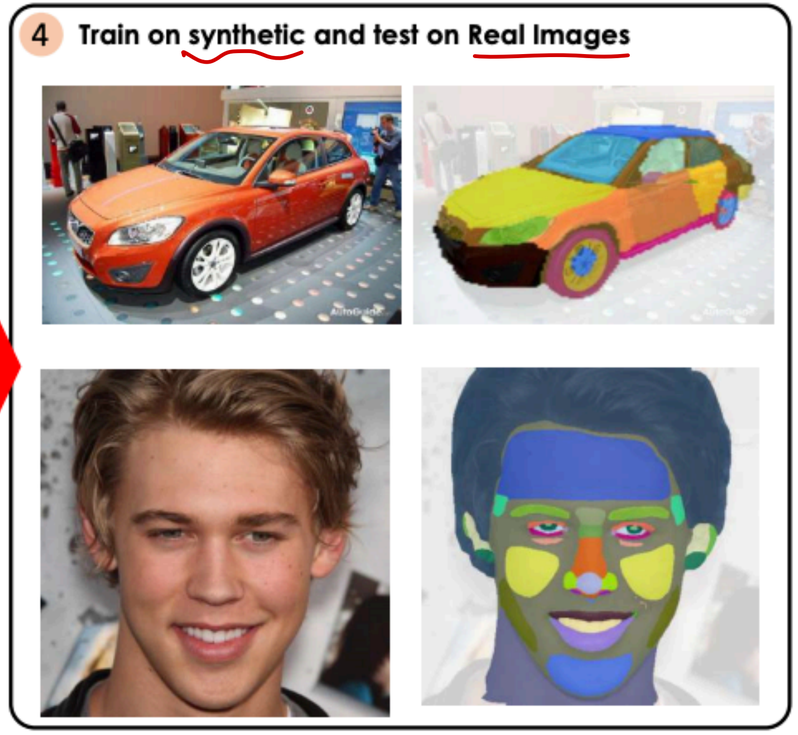


소수 생성된 이미지에 직접 주석달기

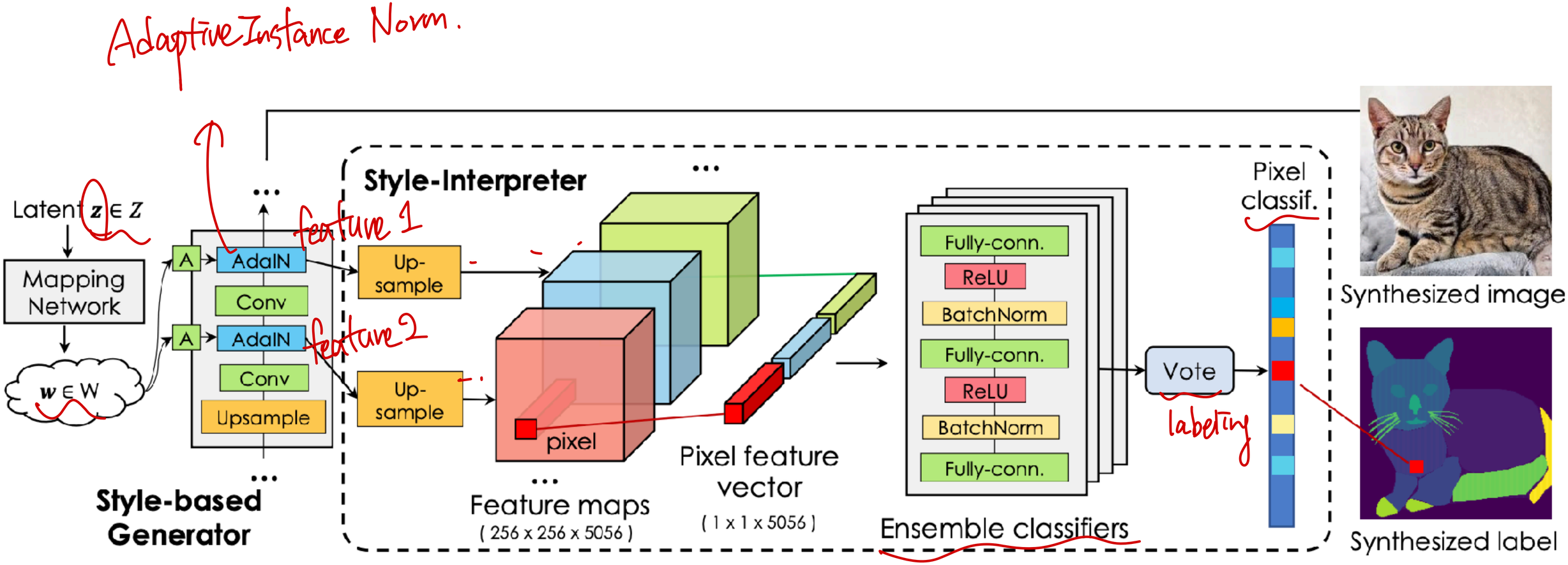
생성 데이터셋 추출



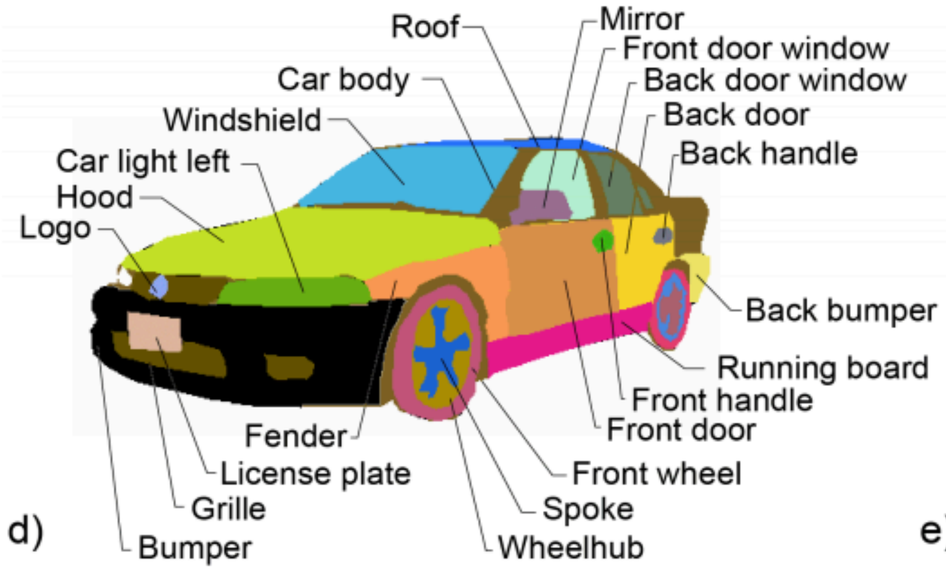
학습 & test



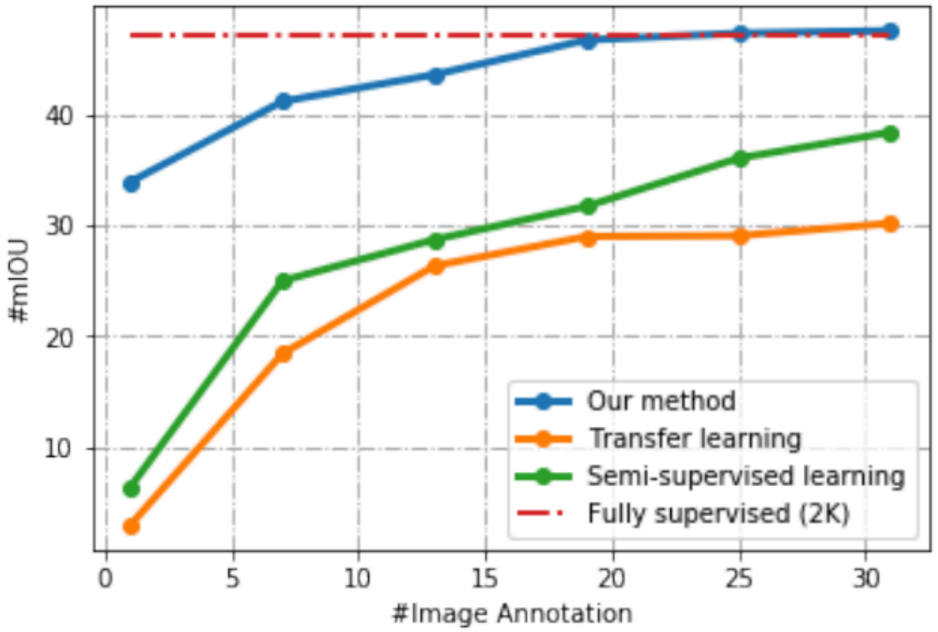
see also [Tritrong*, Rewatbowornwong*, Suwajanakorn, CVPR 2021]
[Li, Yang, Kries, Torralba, Fidler, CVPR 2021]



[Zhang*, Ling*, Gao, Yin, Lafleche, Barriuso, Torralba, Fidler, CVPR 2021]



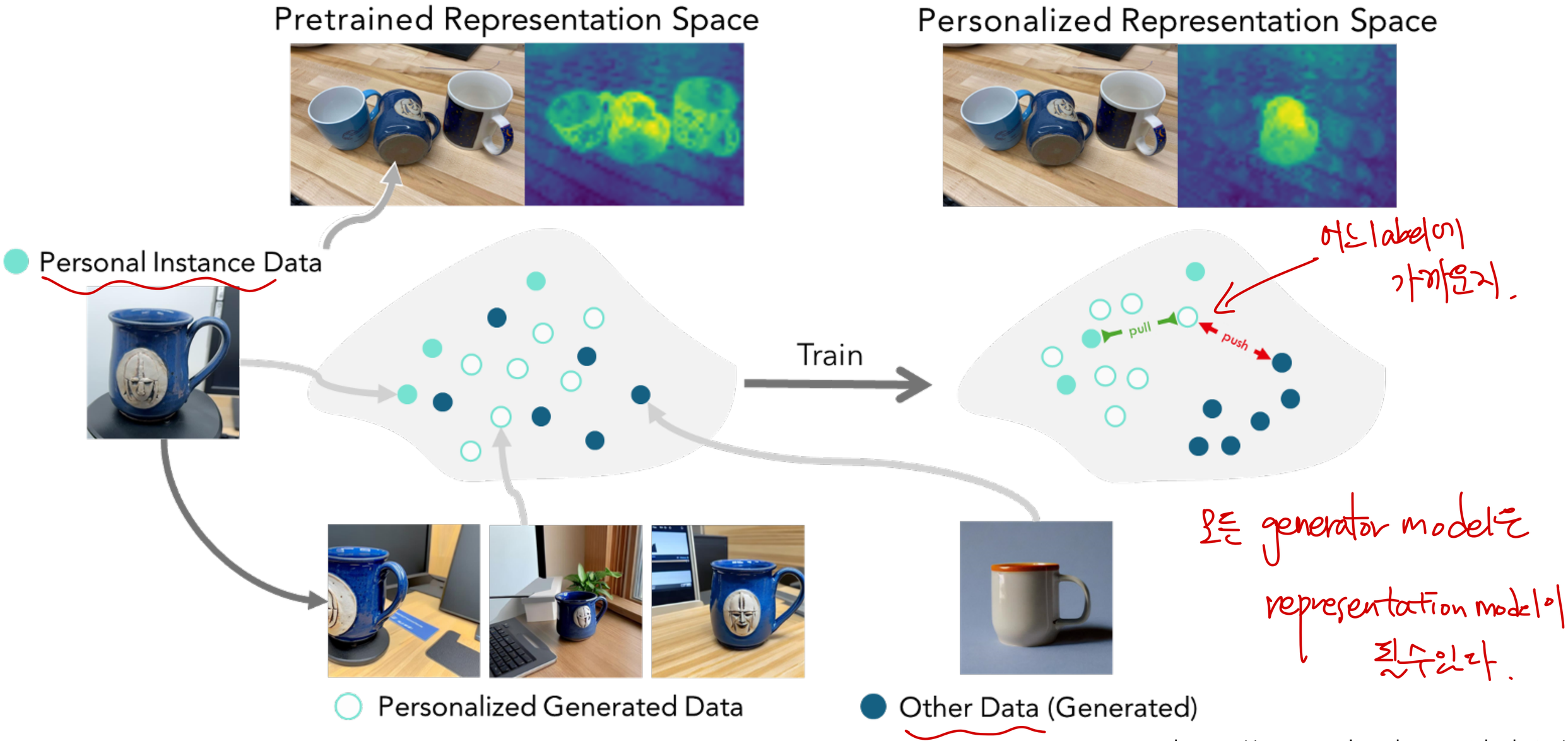




1 labeled GAN image is worth

~100 labeled regular images!

Personalized representation from personalized generation



어느 label에 가까운지.

또는 generator model은 representation model이 필수입니다.

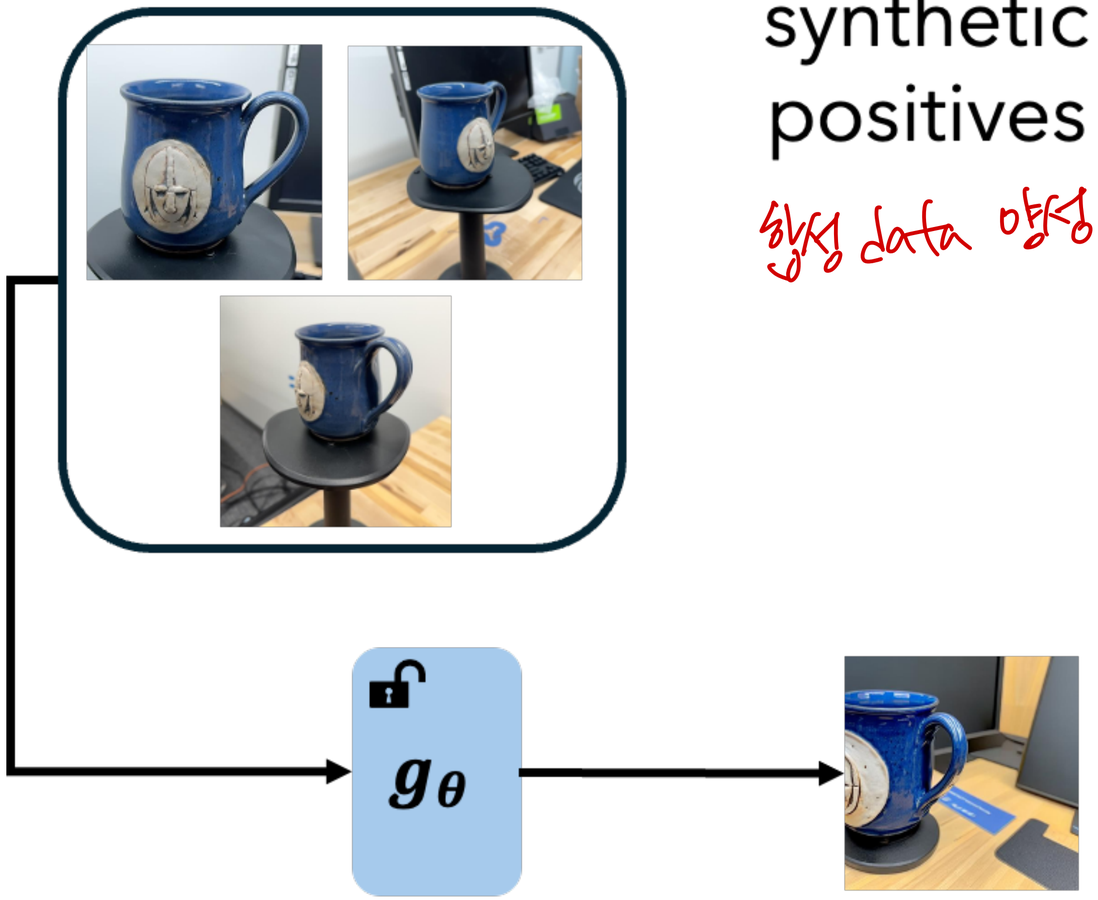
https://personalized-rep.github.io/

Step 1: Finetune Stable Diffusion on D_{real} (DreamBooth)

실제 data



https://personalized-rep.github.io/



Step 2:
Generate synthetic positives

양성 data 양성

CFG = 4



CFG = 7.5



A <new1> mug on a windowsill



A <new1> mug on a park bench



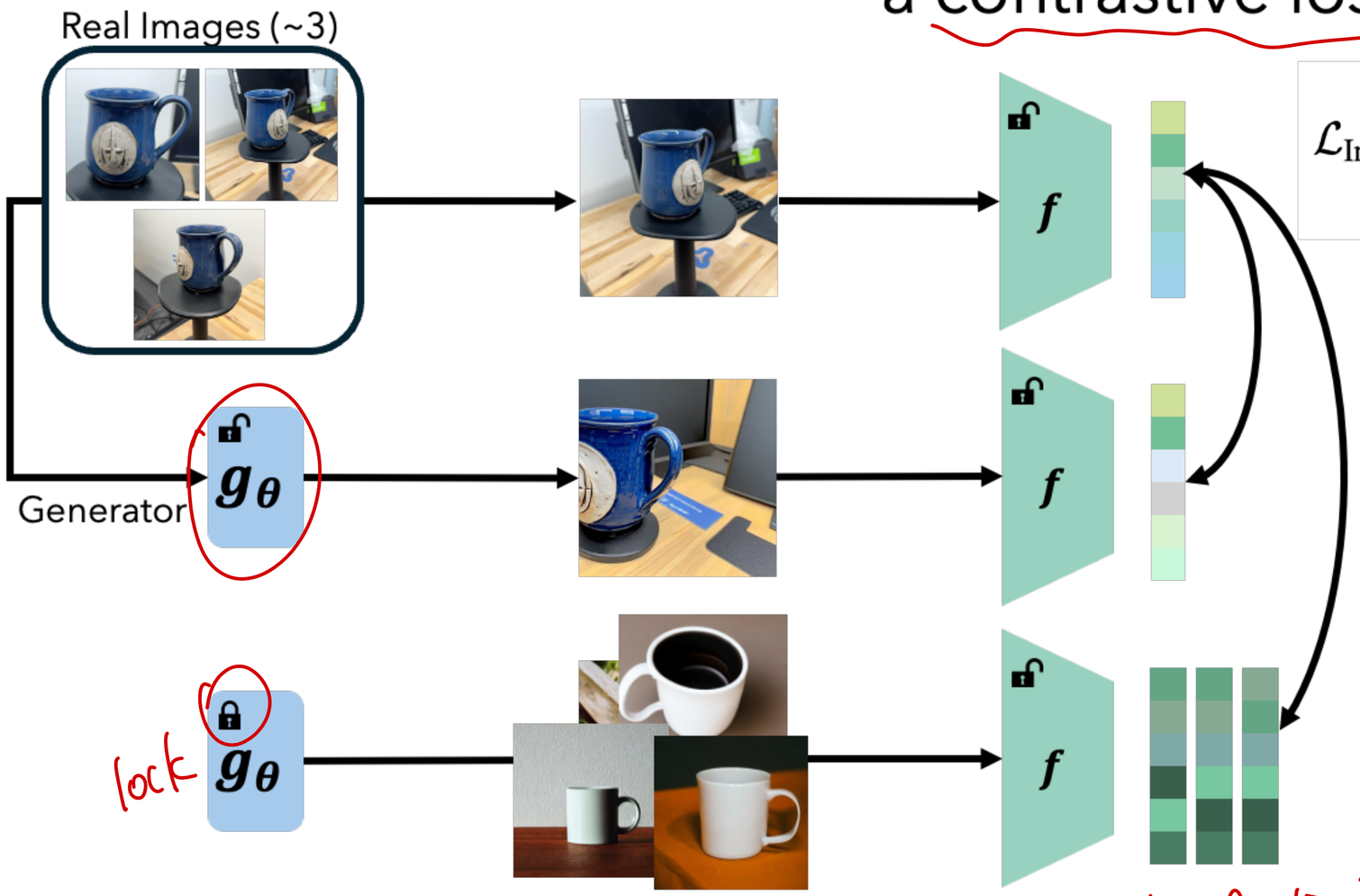
A <new1> mug on a kitchen island



https://personalized-rep.github.io/

Low rank Adaptation

Step 3: LoRA Finetune f_ϕ with a contrastive loss



$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\text{sim}(\mathbf{x}_0, \mathbf{x}_+)/\tau)}{\sum_{i=1}^{N} \exp(\text{sim}(\mathbf{x}_0, \mathbf{x}_i)/\tau)}$$

*Supervise on *both* CLS token and avg-pooled patch features

https://personalized-rep.github.io/

The Data++ pareto frontier - generation vs representation

If the goal is to use generative data to improve training, where does this go wrong?

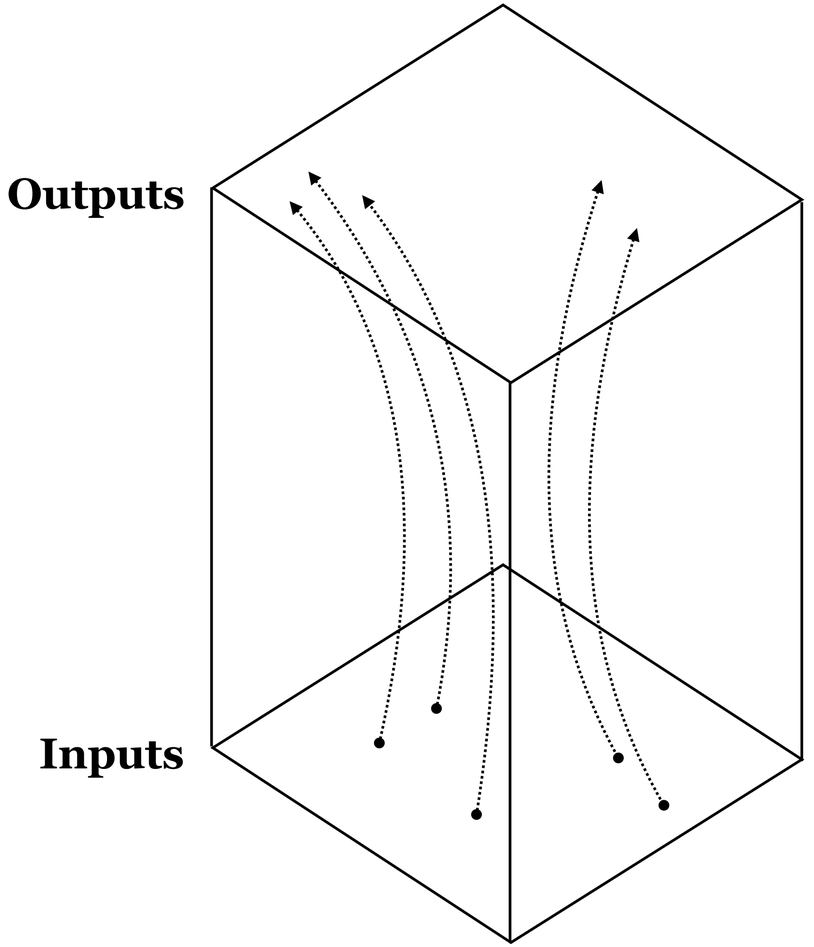
생성형 Data는 통한 학습이 잘못될 수 있는 지점.

- You need enough data to train a good generator, and you also need enough data to learn a good representation, if both datasets are the same, how much can you gain?
  - If the goal is generating data for something rare — generator is bad, so learned representation from generated data is also bad
  - If the goal is generating data for something common — generator is good, but the baseline is also good! Hard to see gains

생성기 성능 ↑, 데이터생성 ↑.

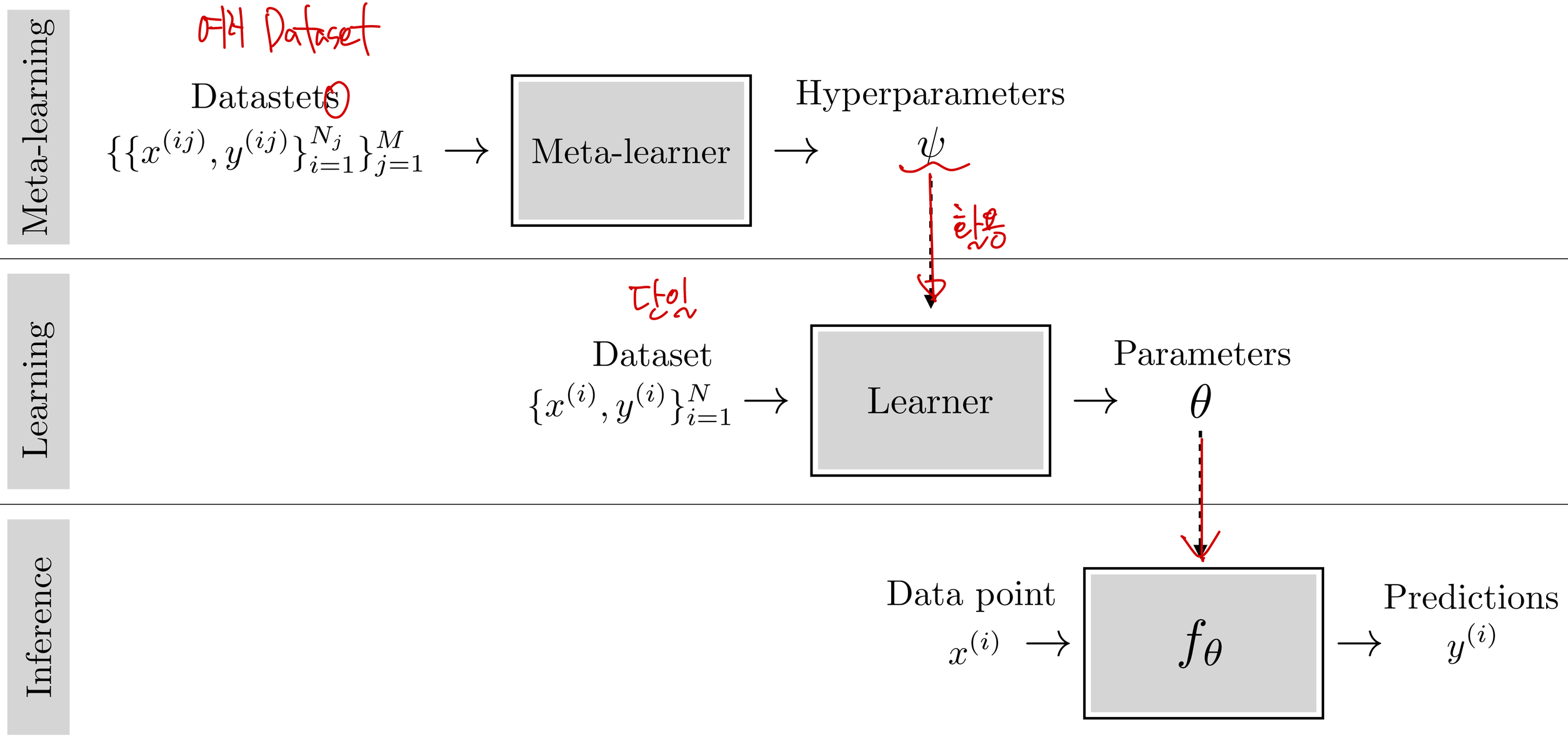
생성기 성능 ↑. but baseline 또한 좋다 → improvement 찾기 어려움.

1. Knowledge about the mapping
2. Knowledge about the outputs
3. Knowledge about the inputs
4. **Learning to learn fast**



Transfer Learning

Meta-learning: learning to learn fast



MAML: Model-Agnostic Meta-Learning

[Finn, Abbeel, Levine 2017]

Model에 구애받지 않는 Meta learning

- So far we saw transfer learning as: given some prior model/representation, how can we quickly adapt it?
- Could we *learn* to do transfer learning?
- Consider finetuning: given an init, update it with SGD — Stocastic gradient descent
- MAML: *learn* an init such just a few steps of SGD will update it very effectively

MAML: Model-Agnostic Meta-Learning

[Finn, Abbeel, Levine 2017]

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks

Require: α, β: step size hyperparameters

1: randomly initialize θ

2: **while** not done **do**

3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$

4: **for all** $\mathcal{T}_i$ **do**

5:

6: 1. Starting with init, run a few steps of SGD on task i
2. Compute loss after these steps

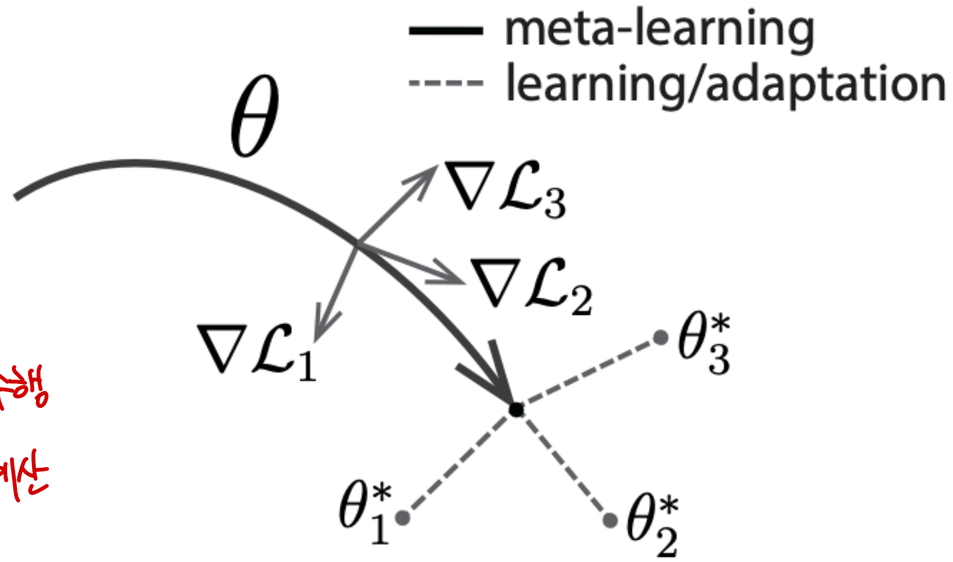
SGD 수행

Loss 계산

7: **end for**

8: init <— backprop through

9: **end while**



MAML: Model-Agnostic Meta-Learning

[Finn, Abbeel, Levine 2017]



How to backprop through  ?

1 step SGD:

forward → backward → forward

 = `forward`(x, `backward`(`forward`(x, θ)) + θ)

2 steps SGD:

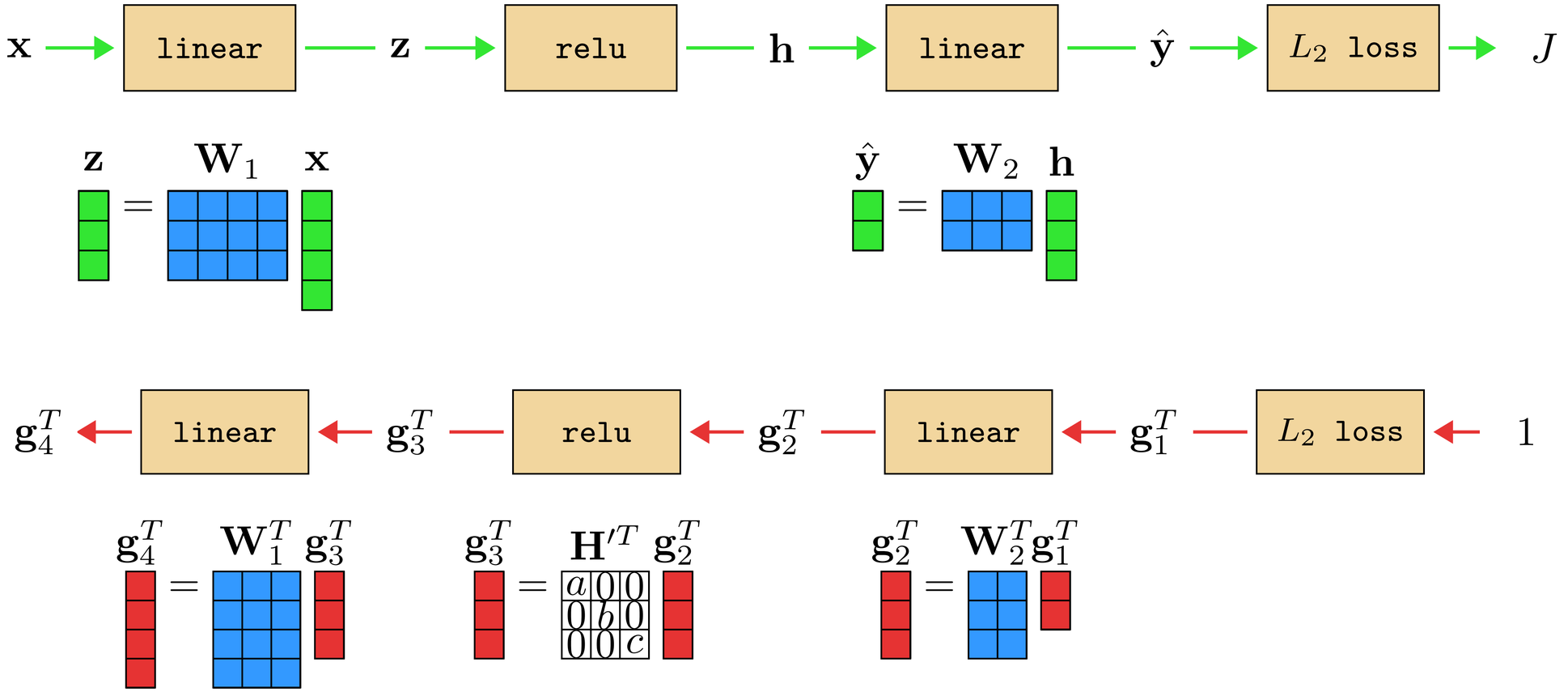
 = `forward`(x, `backward`(`forward`(x, `backward`(`forward`(x, θ)) + θ), θ) + θ)

You can backprop through backward!

backward를 통과해서 backprop이 가능하다
=backward는 미분가능한 연산으로만

```
forward(x, backward(forward(x, backward(forward(x, θ)) + θ), θ) + θ)
```

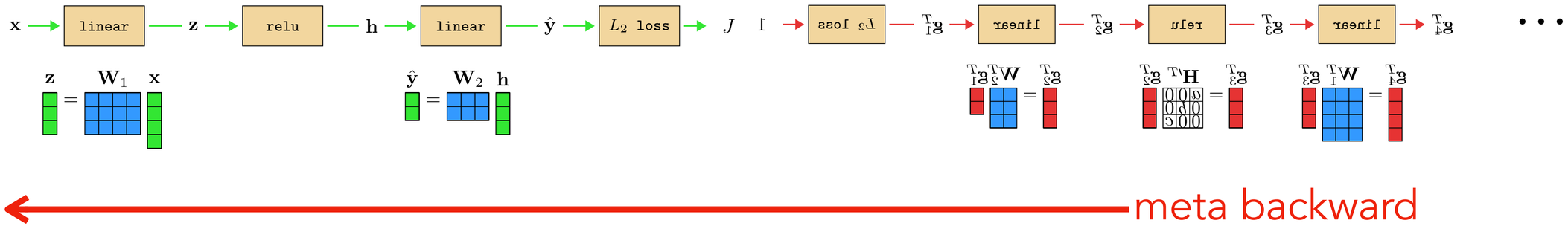
Example forward and backward through a simple net:



You can backprop through backward!

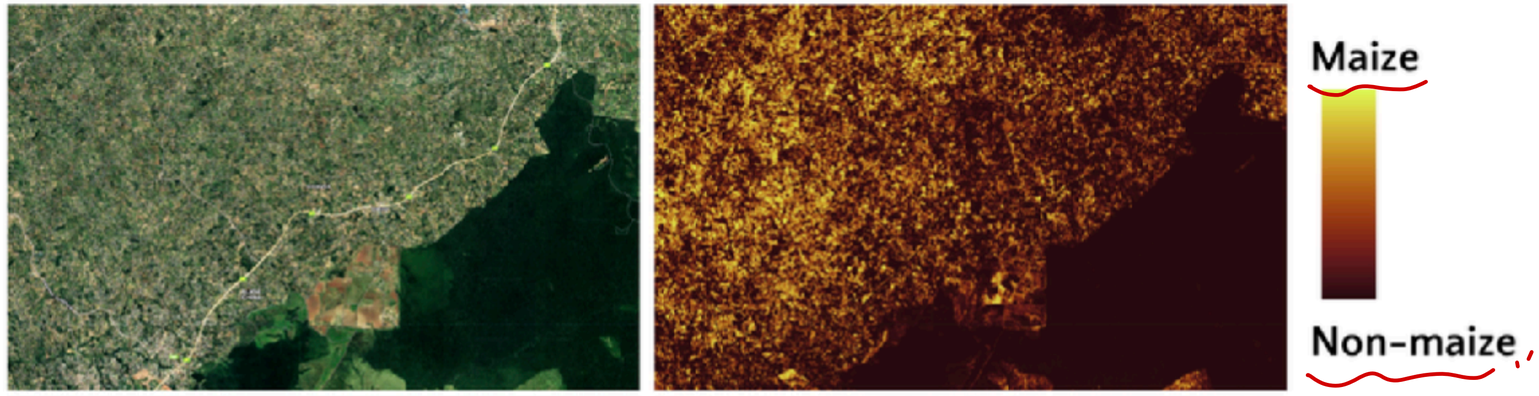
`forward`(x, `backward`(`forward`(x, `backward`(`forward`(x, θ)) + θ), θ) + θ)

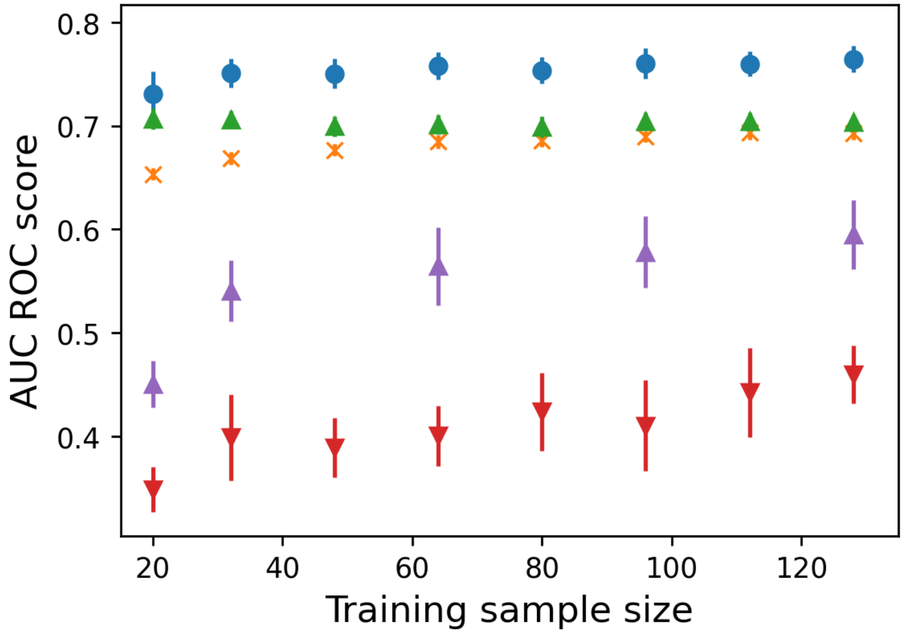
Example forward and backward through a simple net:



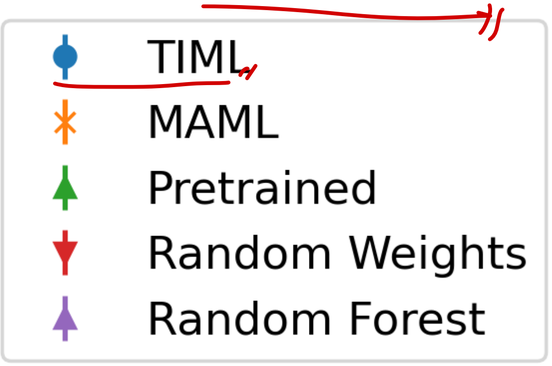
Task-Informed MAML for Agriculture

- Tseng et al. introduce a geospatial embedding that encodes task-specific context into meta-learning, and helps performance





(a) Kenya: Maize vs. Rest



TIML: Task-Informed Meta-Learning for Agriculture, Tseng et al., 2022

Meta-learning by sequence modeling

- Definition of learning:
  - input: a sequence x0, y0, x1, y1, x2, y2, ...
  - output: a function f: x—> y
- x0, y0, x1, y1, x2, y2, x3, ?
- RNN state + params is f, apply that f to x3

Model이 시퀀스에서 규칙을 추출

반영

입력 시퀀스의 변화.

y_3

improvement → in-context learning.

So foundation models are meta-learners?

Transfer Learning

- Learning with little data
- Transferring knowledge about the mapping (finetuning and domain adaptation)
- Transferring knowledge about the outputs (distillation)
- Transferring knowledge about the inputs (generative models as data)
- Learning to learn quickly (meta-learning)