Emerging Properties in Self-Supervised Vision Transformers

Also known as DINO

Javier Rando Ramirez

Mathilde Caron[1,2] Hugo Touvron[1,3] Ishan Misra[1] Hervé Jegou[1]
Julien Mairal[2] Piotr Bojanowski[1] Armand Joulin[1]

[1] Facebook AI Research [2] Inria* [3] Sorbonne University

Recap from previous presentations

- Transformer architecture: “Attention is all you need”

  *Attention-based **encoder + decoder** for Seq2Seq problems in NLP.*

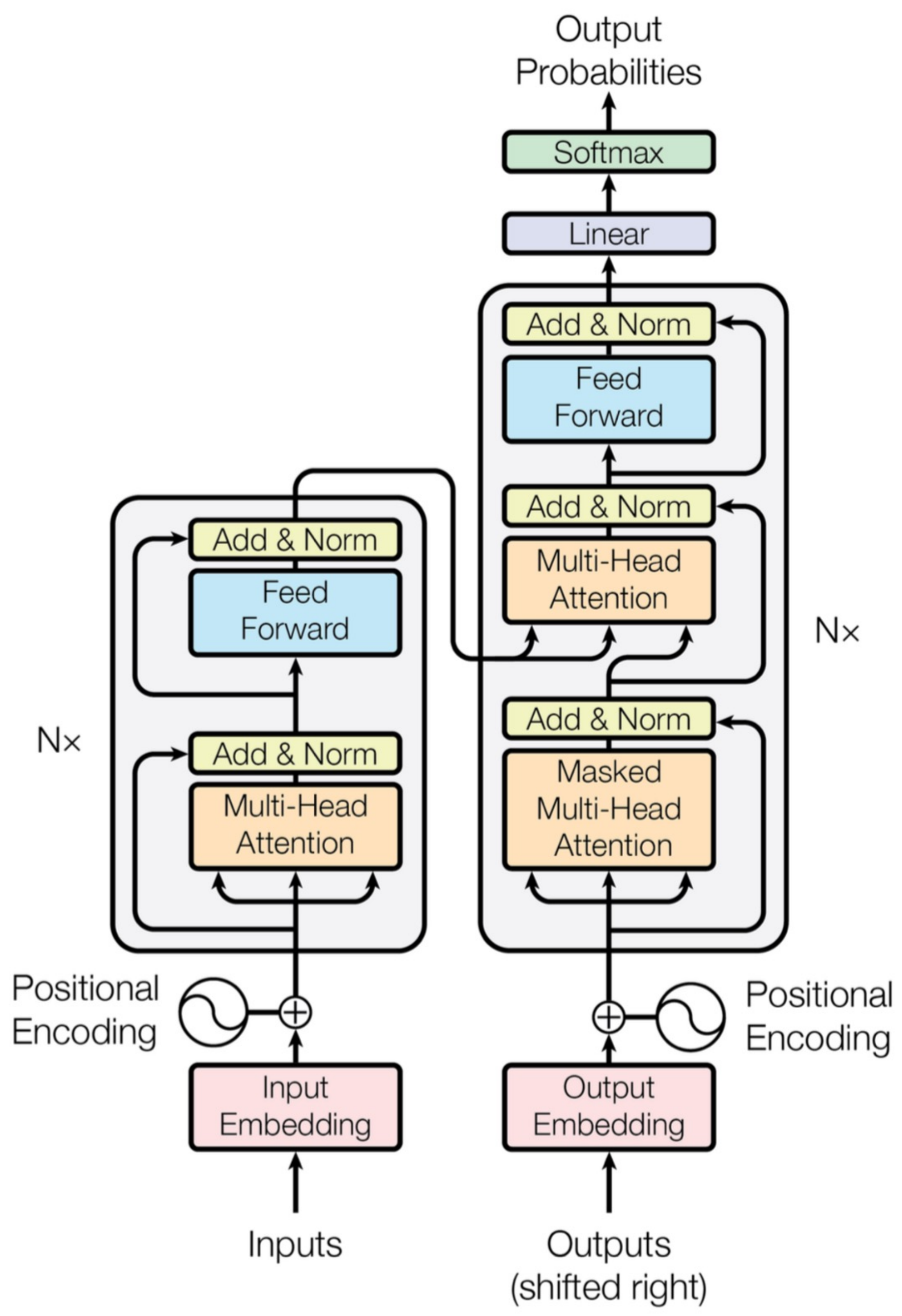
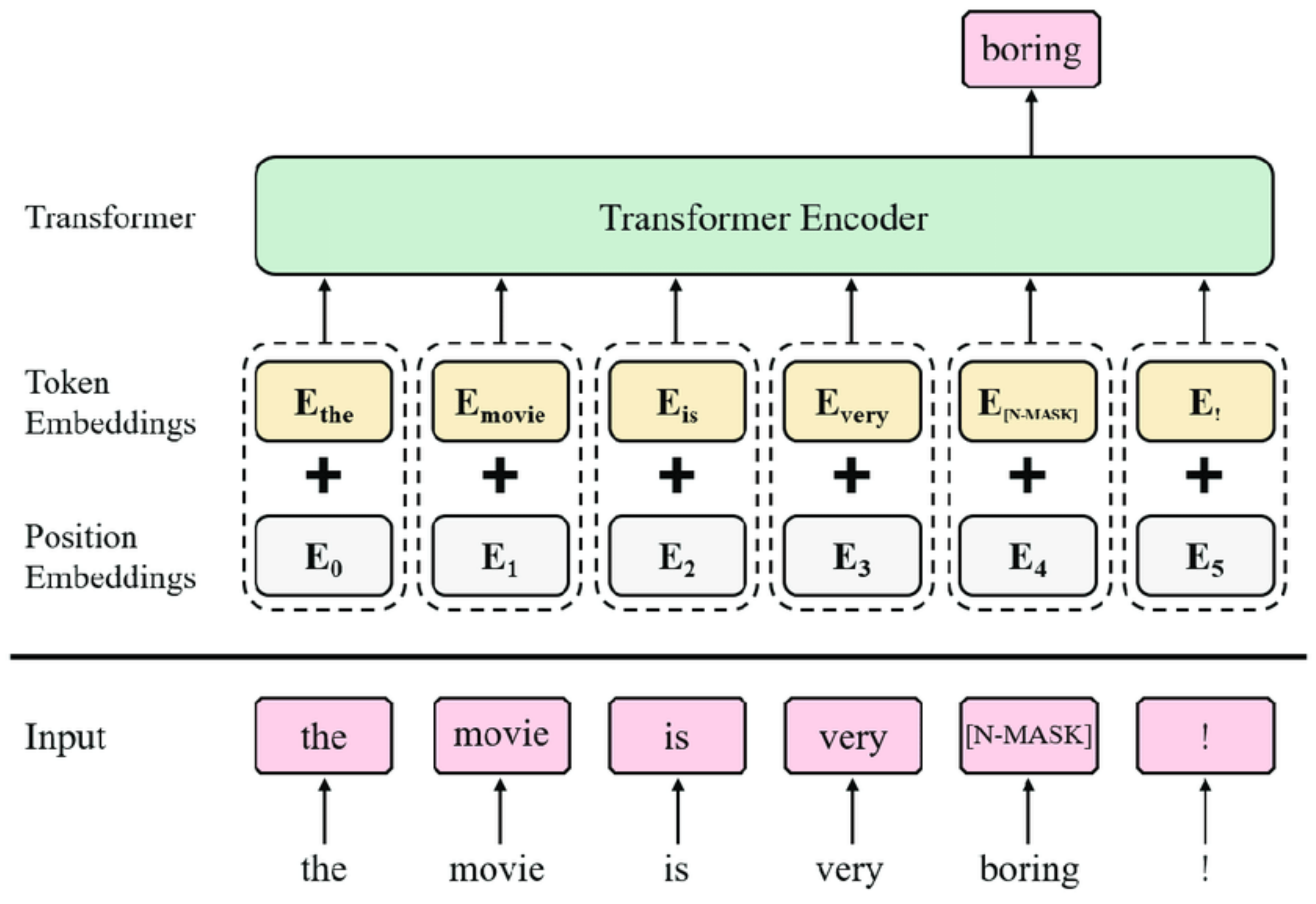


Figure 1: The Transformer - model architecture.

Recap from previous presentations

- Big language models can learn without supervision and be fine-tuned for different downstream tasks.

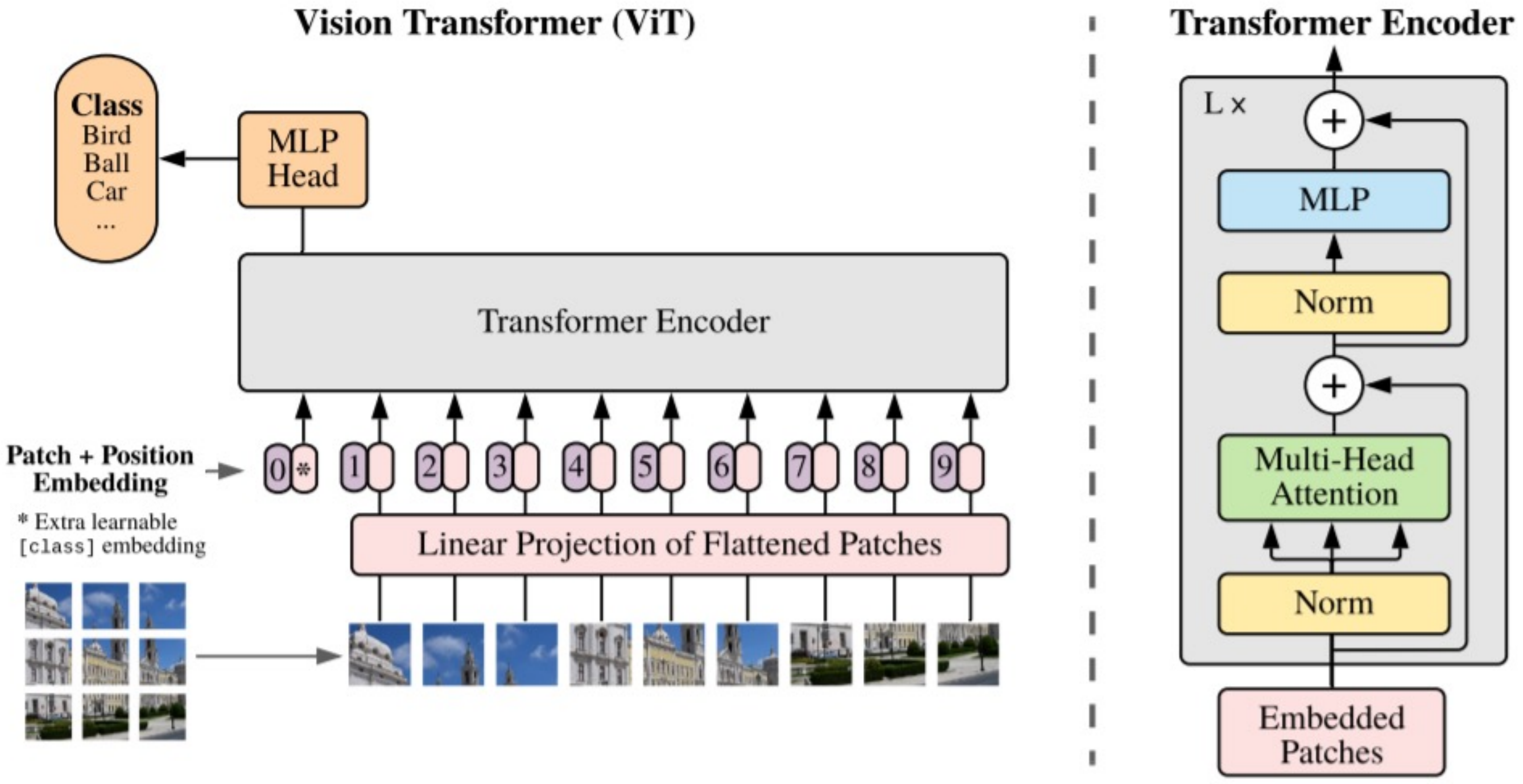
  State-of-the-art results given knowledge embedded in the Transformer.



Recap from previous presentations

- These same principles can be transferred to the Computer Vision domain: “An image is Worth 16x16 words”.

  *We can solve CV tasks relying only on attention (no need for CNNs) -> **Vision Transformers***



What we will see today

and how it is linked with previous presentations

- Success of Transformers in the NLP domain was mainly thanks to self-supervised pretraining
- So far, supervised ViTs had not yet delivered clear benefits over CNNs: *"computationally more demanding, require more training data, and their features do not exhibit unique properties"*

Can we train self-supervised ViTs? Will this bring interesting properties?

Related work

Self-supervised learning in CV

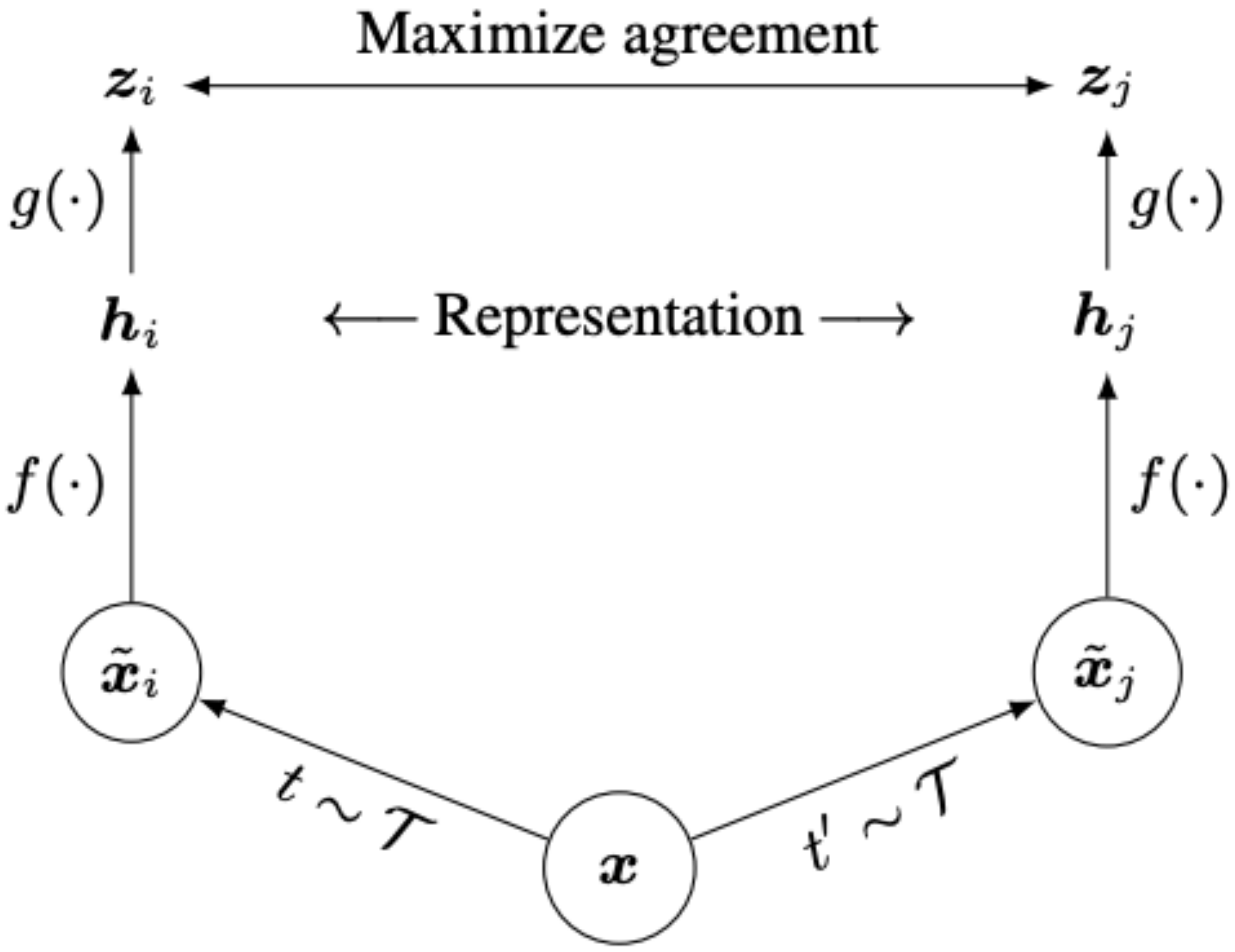
Contrastive Learning [1]

1. Apply two different transformations to a sample *x*

2. Obtain their representation using an encoder f(·), e.g. ResNet

3. Map them to a space using an MLP g(·)

4. Apply contrastive loss (match predictions)



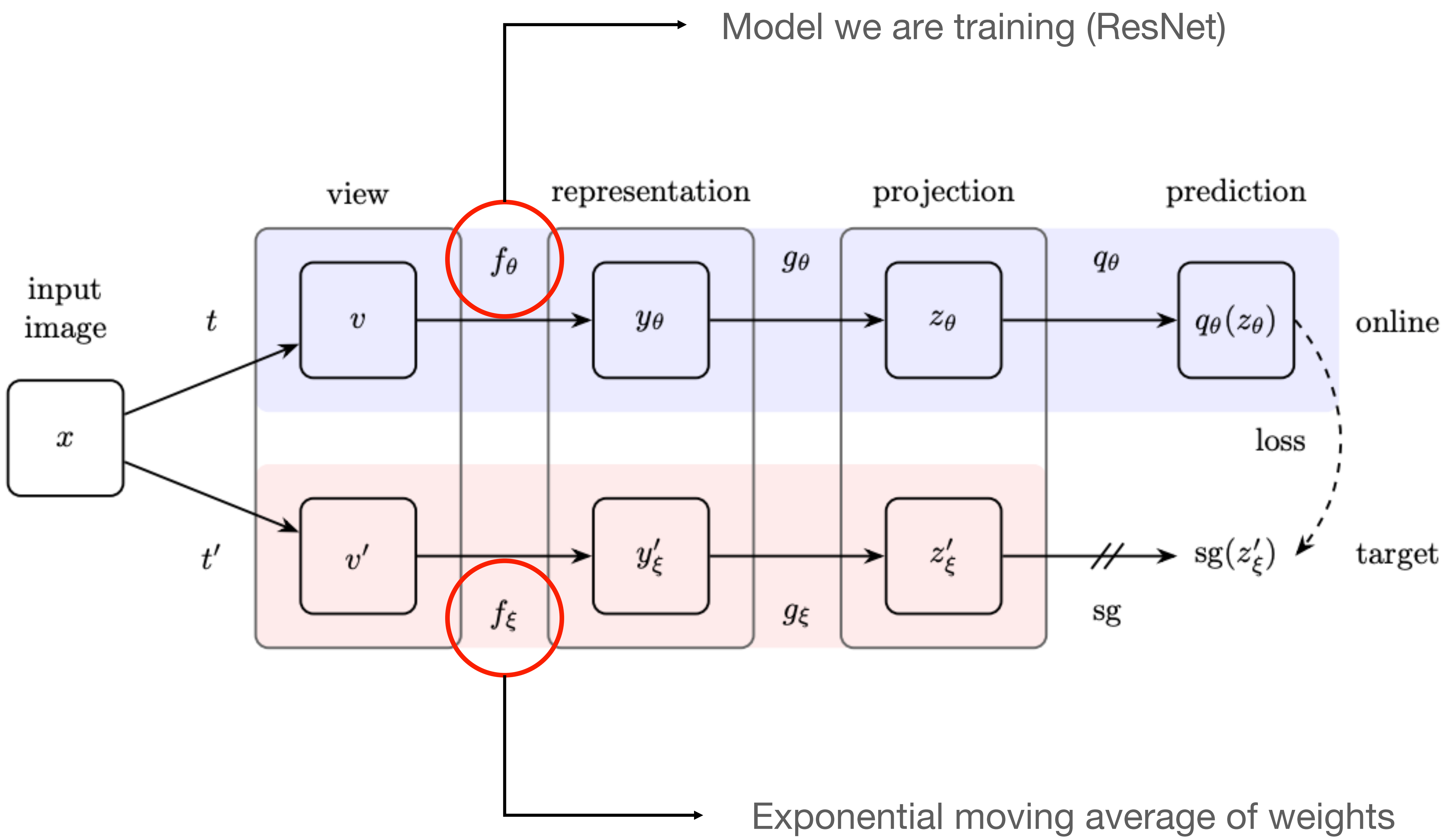
[1] Chen et al. *A simple framework for contrastive learning of visual representations*.

Related work

Self-supervised learning in CV

BYOL [2]

1. Apply two transformations to a sample *t.*
2. Define your network f_θ and let f_ξ be a (moving average) copy of it
3. Make our network predict the representation that a similar network will produce for the other transformation.



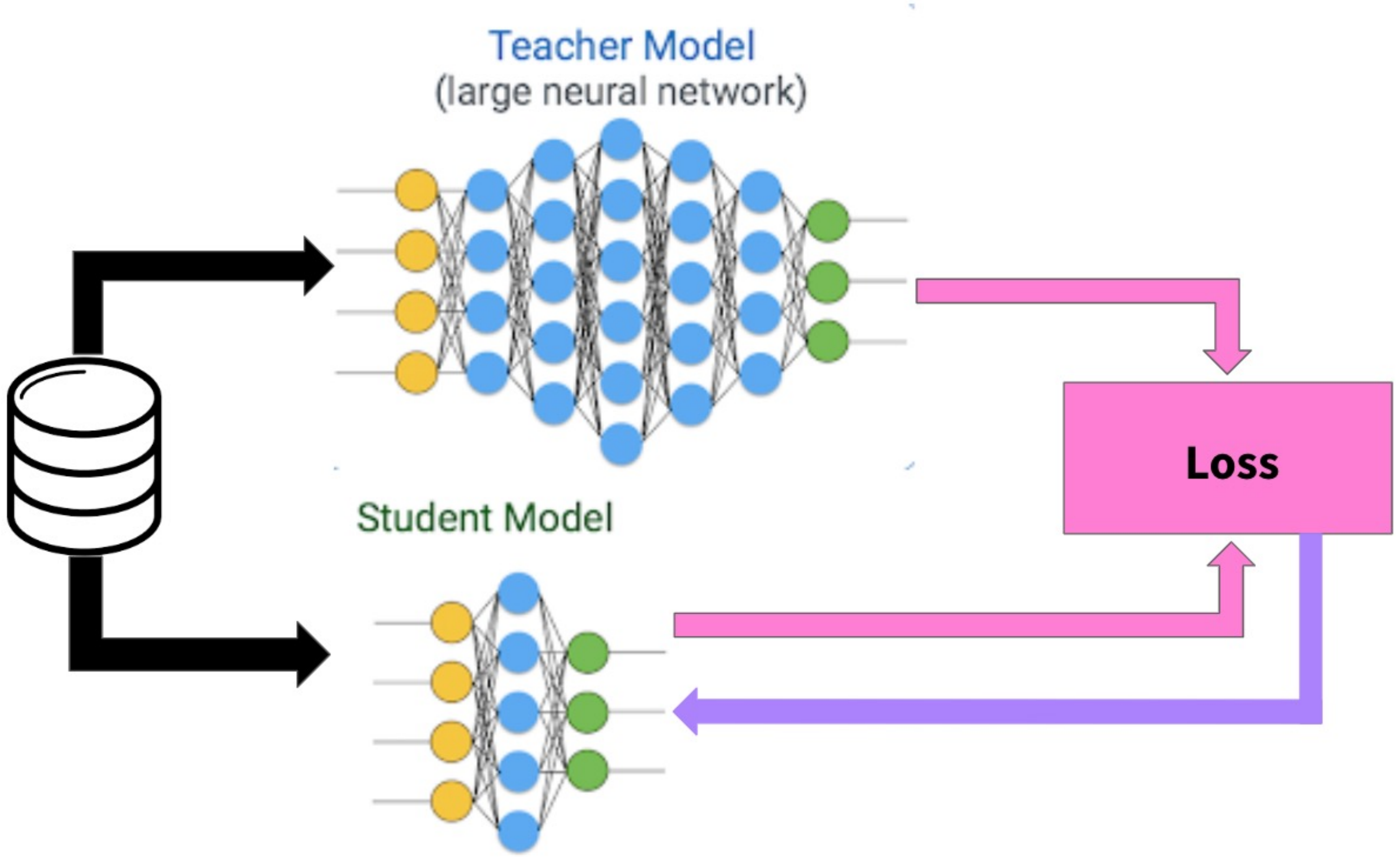
[2] Grill et al. *Bootstrap your own latent: A new approach to self-supervised Learning.*

Related work

Knowledge distillation (will see later in the course)

Distillation [3]

1. Train a large model (Teacher)
2. Generate soft labels for input data using the Teacher
3. Train a smaller model (Student) to predict the labels generated by the teacher.



[3] Hinton et al. *Distilling the Knowledge in a Neural Network*

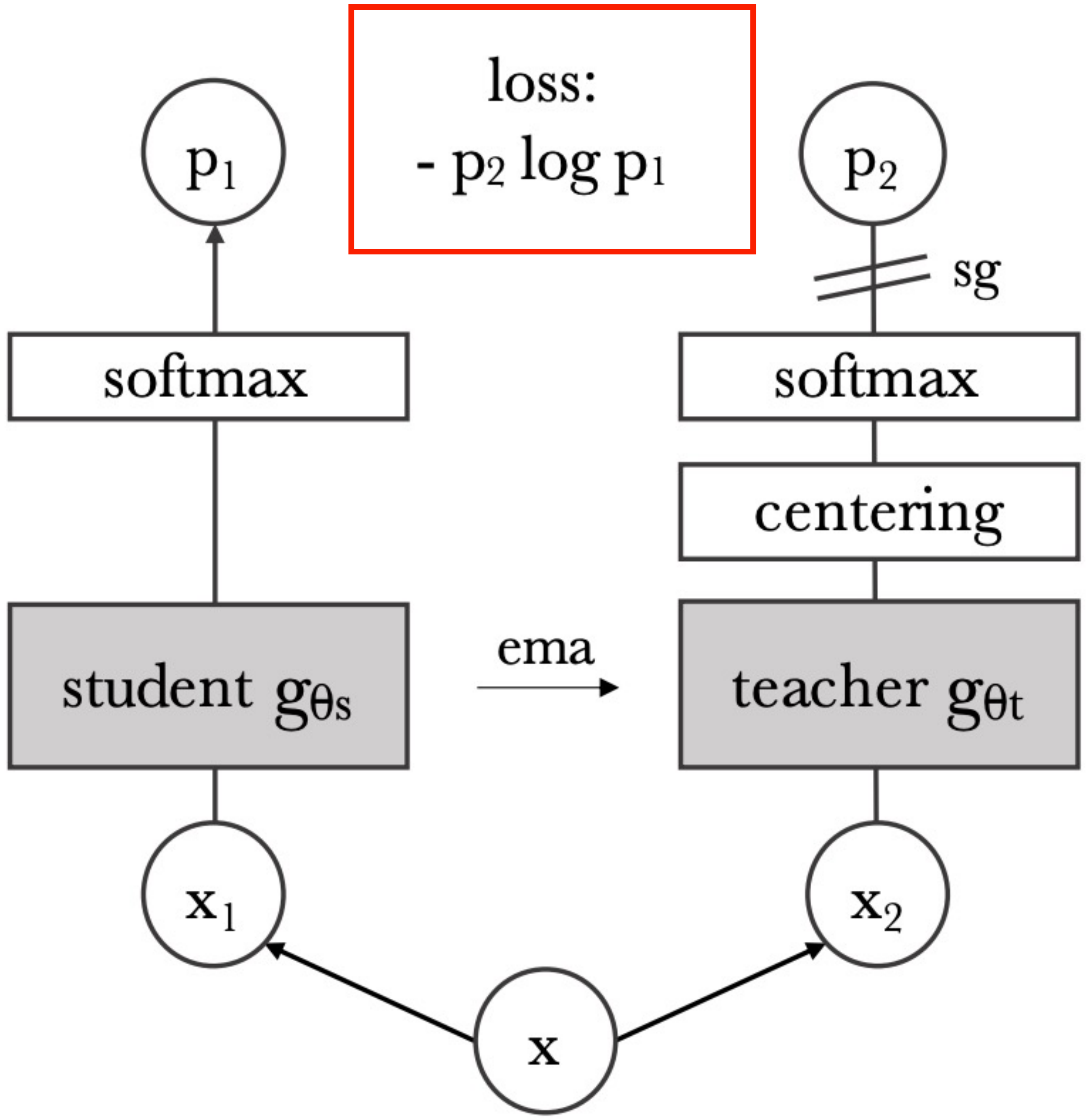
Self-distillation with no labels (DINO)

Combine the previous to create a self-supervised method

Overall idea

- Augment data
- Train a student network to predict the representation generated by a teacher network on a different variation of the same image.

Cross-entropy loss

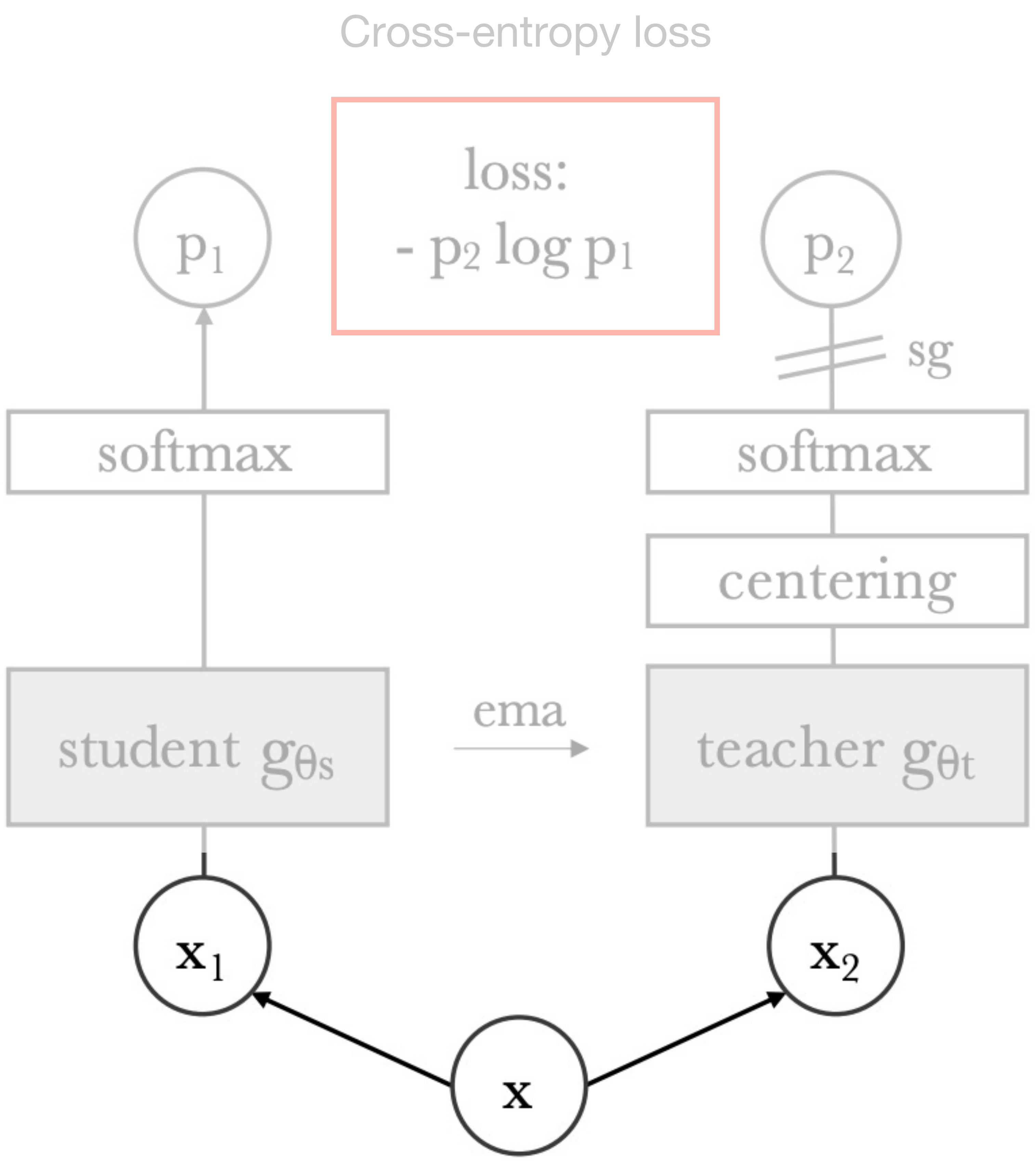


Self-distillation with no labels (DINO)

Data Augmentation

For each image, they generate a set of **global** and **local** views.

These views are further augmented using color jittering, Gaussian blur and solarization.

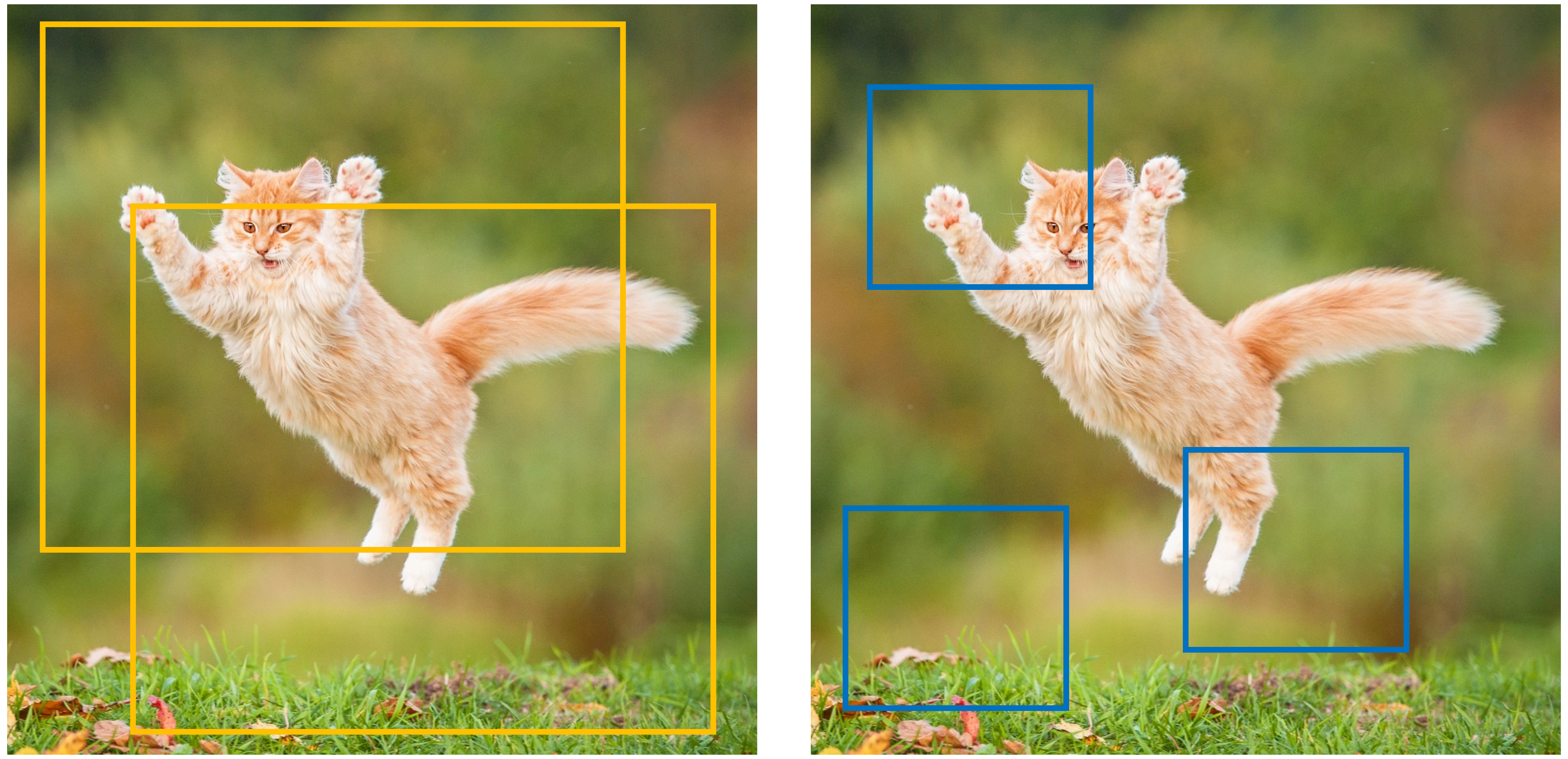


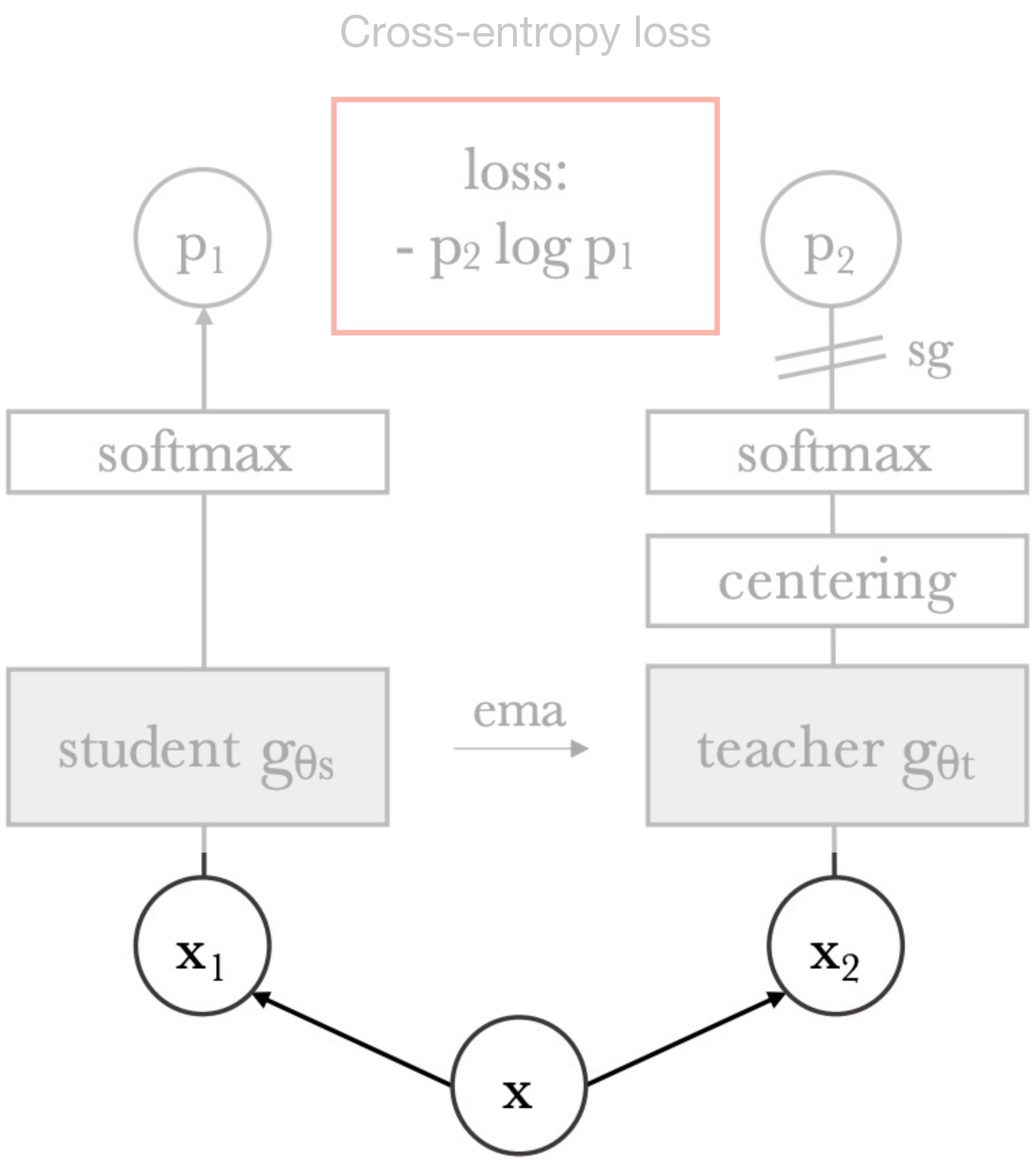
Self-distillation with no labels (DINO)

Data Augmentation

Global: more than 50% of the image

Local: less than 50% of the image



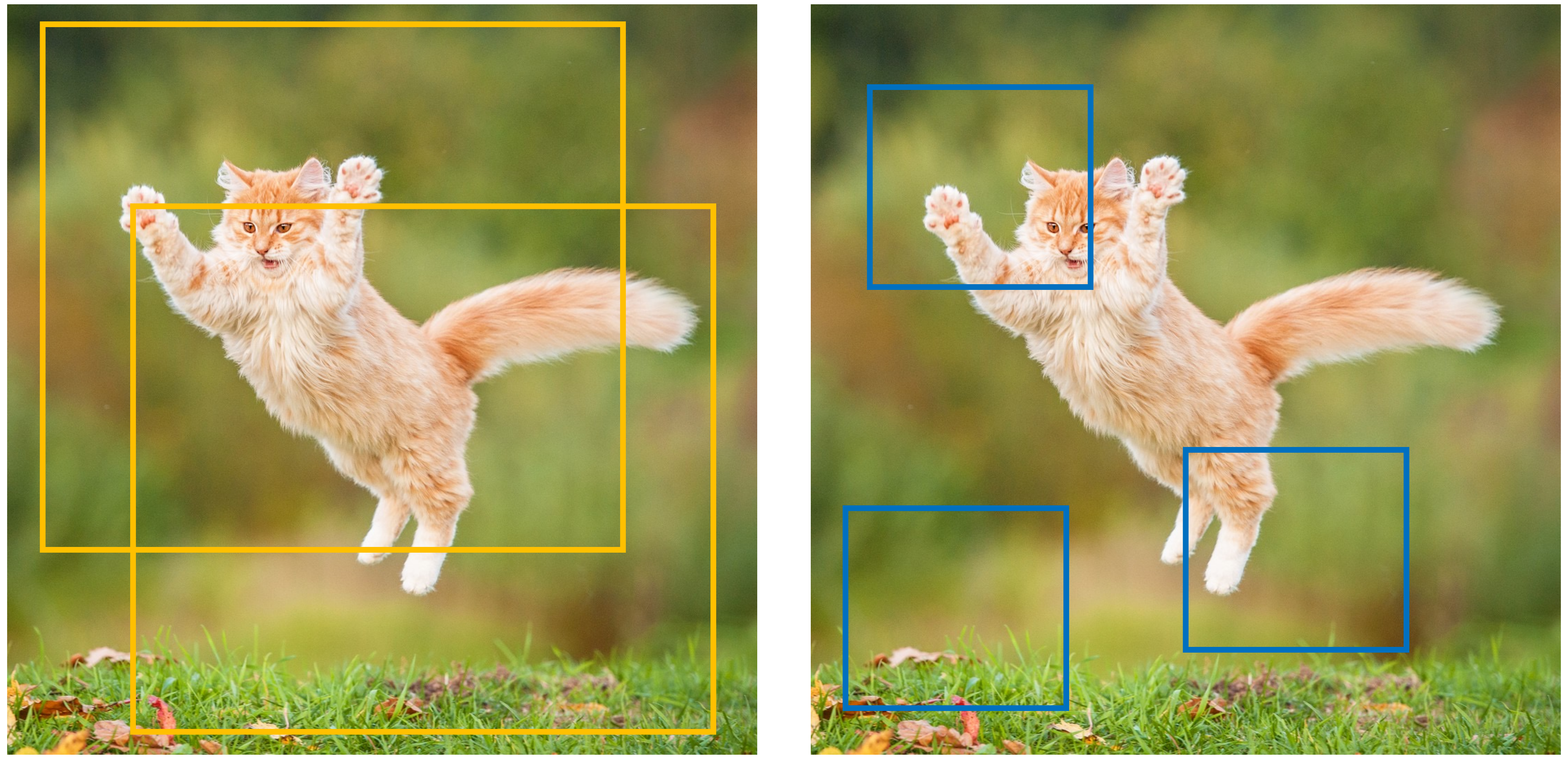


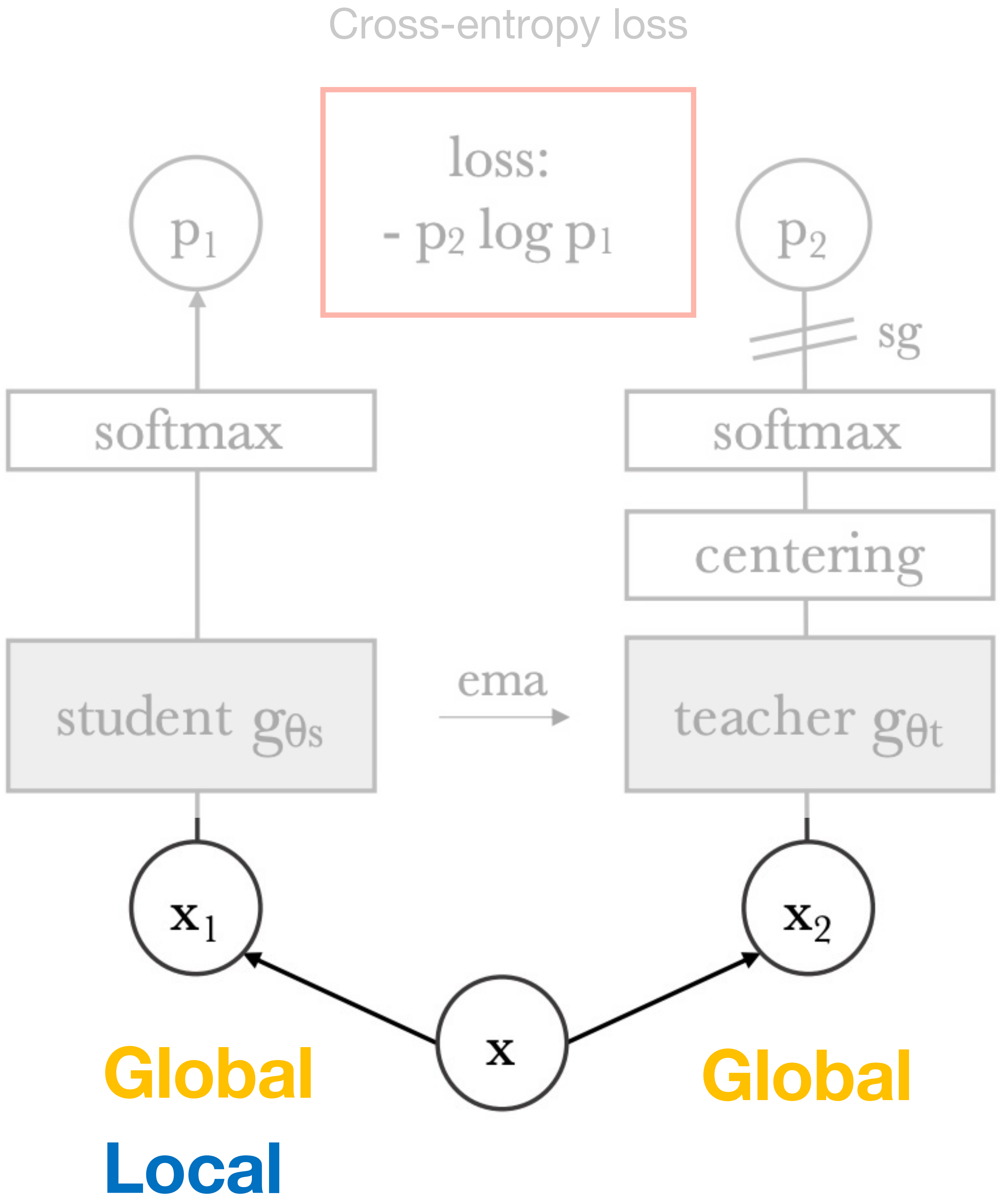
Self-distillation with no labels (DINO)

Data Augmentation

Global: student and teacher

Local: only for the student



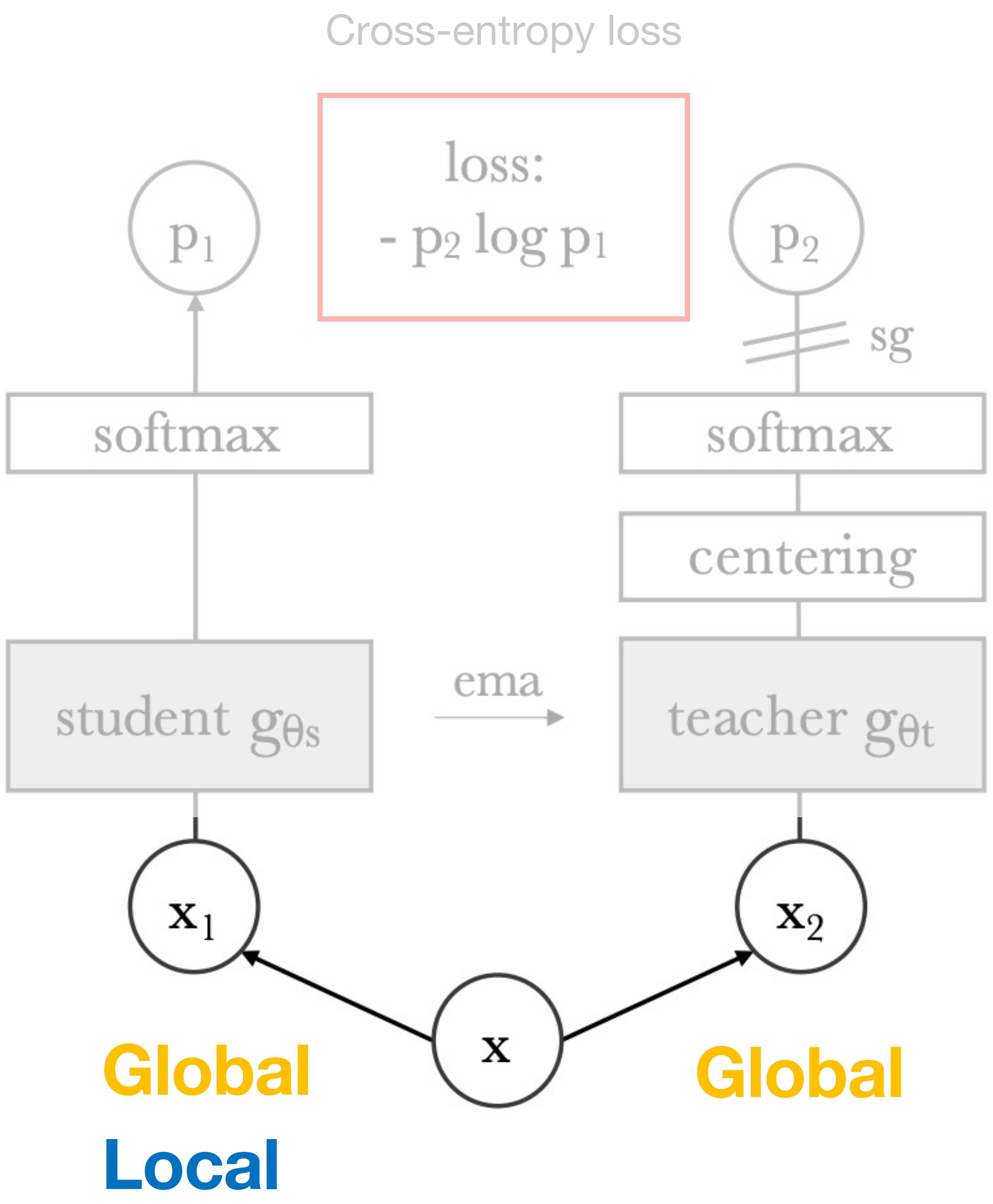


Self-distillation with no labels (DINO)

Data Augmentation

Motivation: from local to global

=

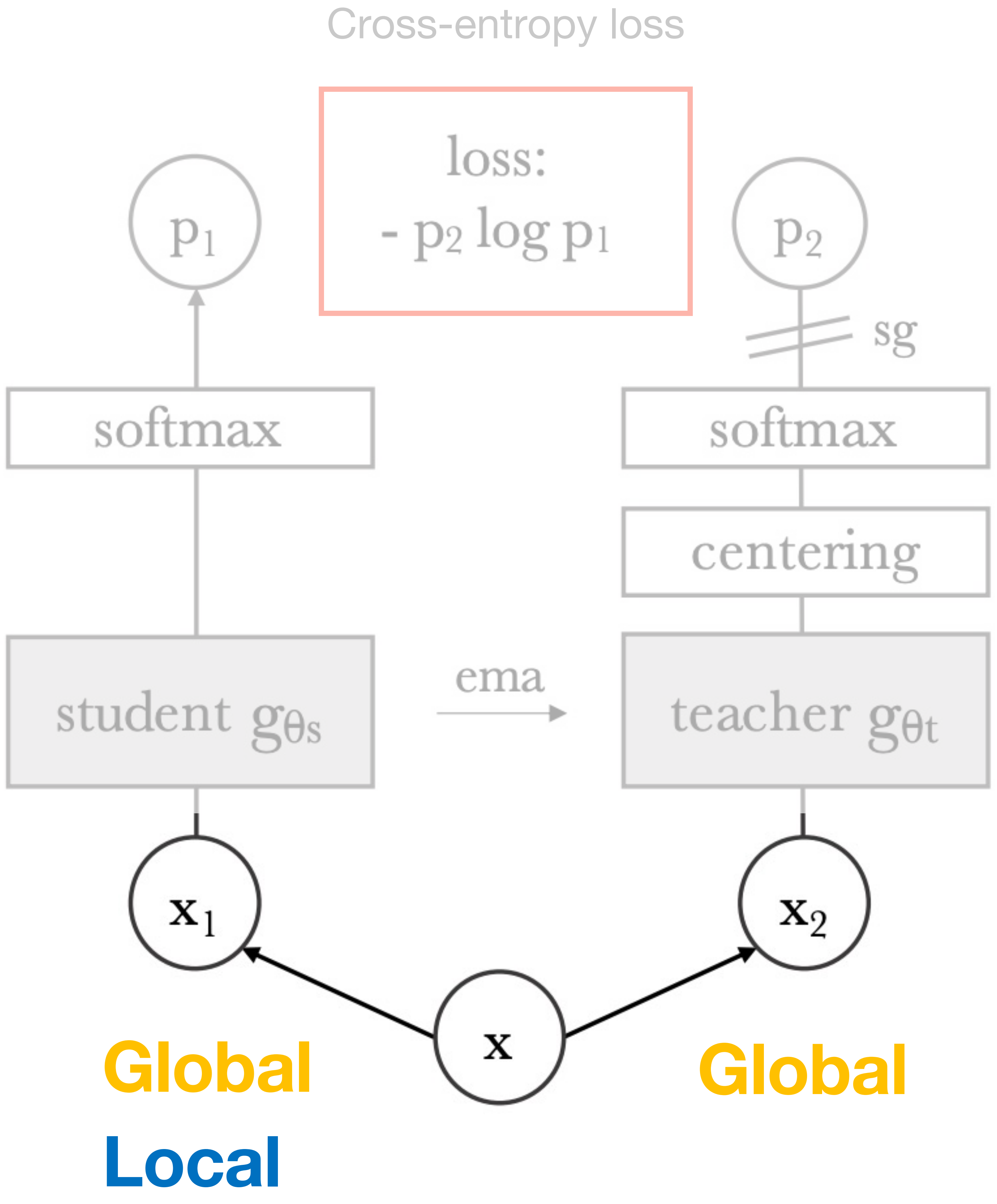


Self-distillation with no labels (DINO)

Data Augmentation

Motivation: from local to global

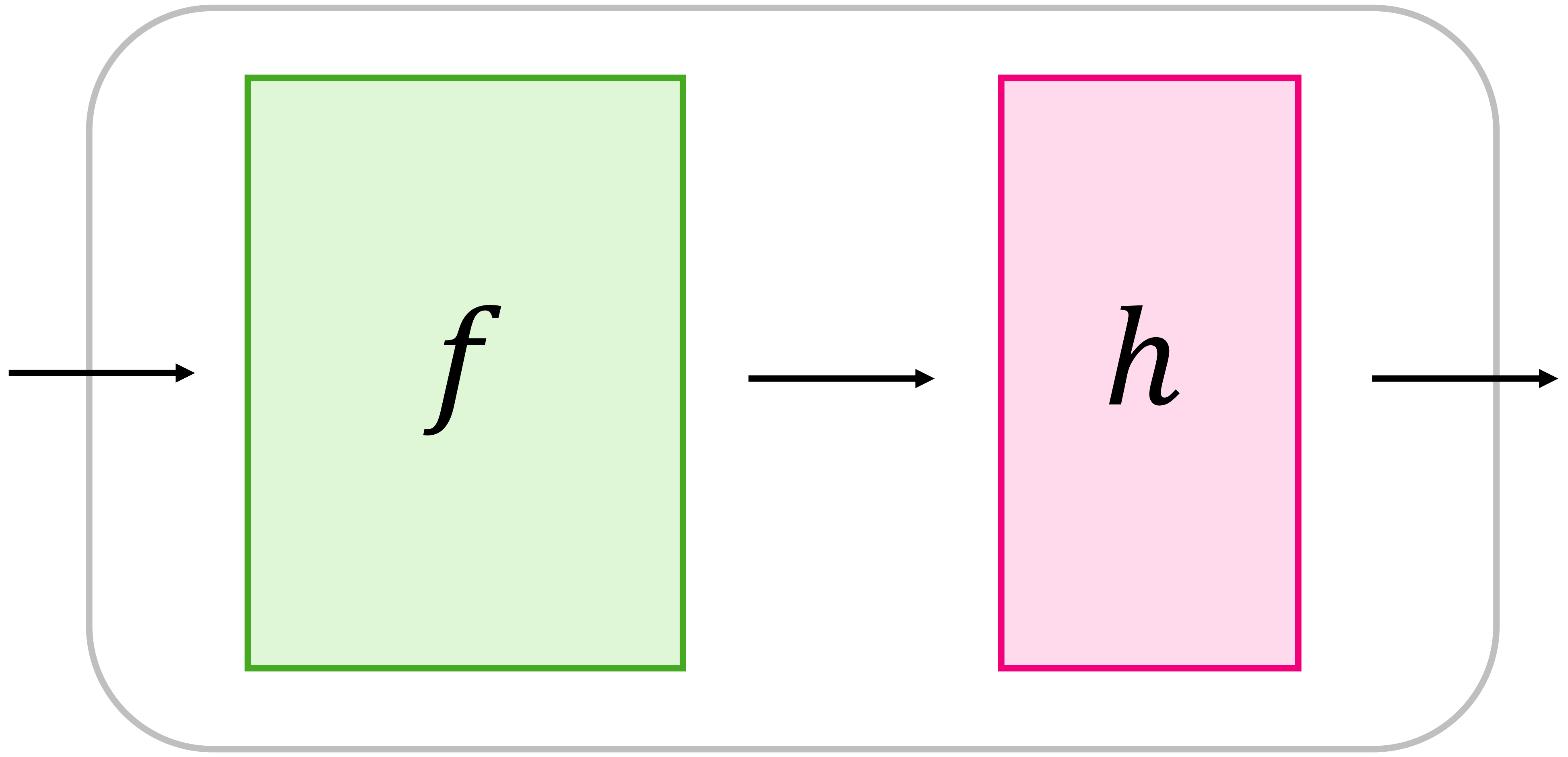
=

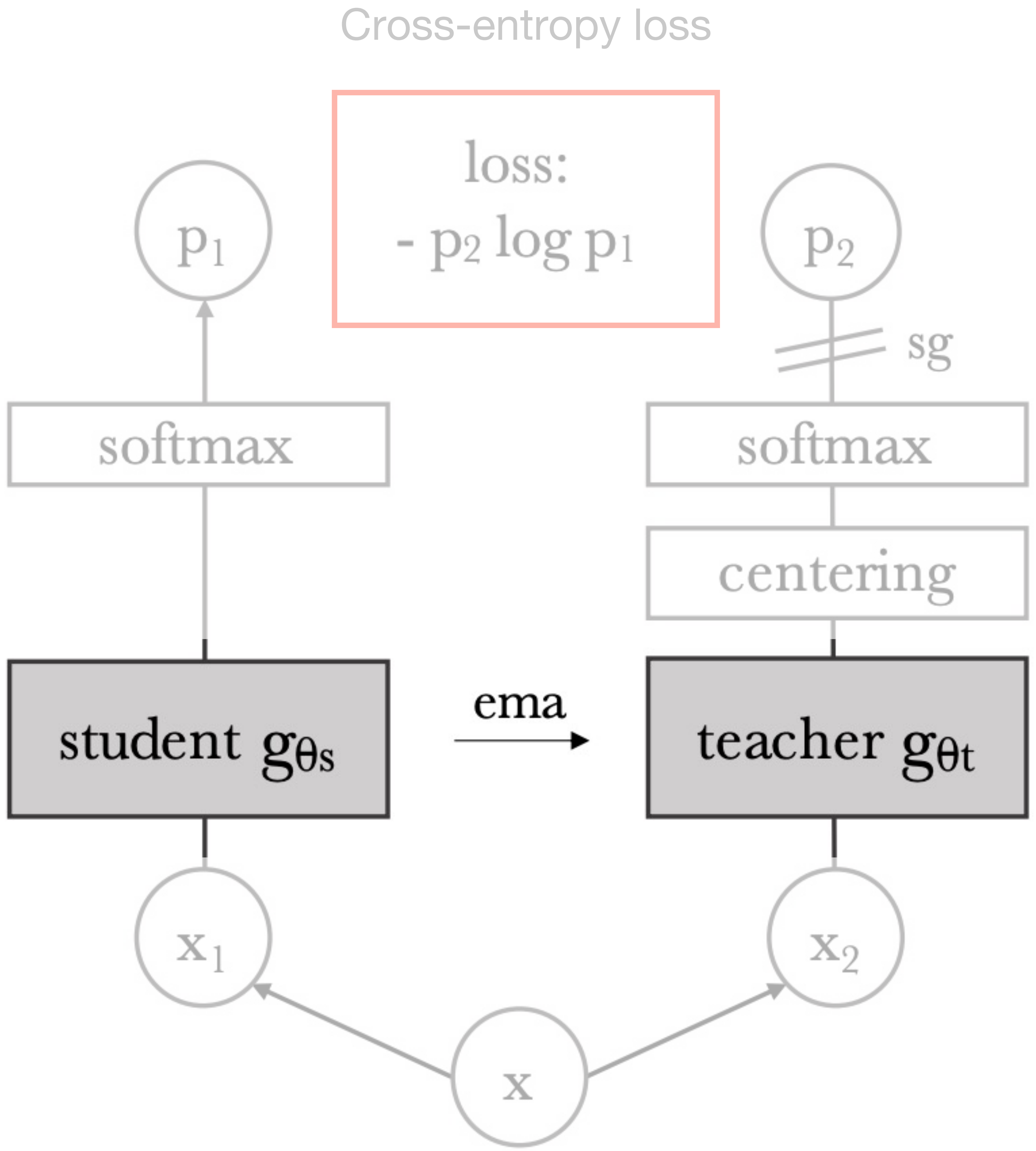


Self-distillation with no labels (DINO)

Network architectures

$$g = h \circ f$$





Self-distillation with no labels (DINO)

Network architectures

f

"Backbone"

(ViT or ResNet)

h

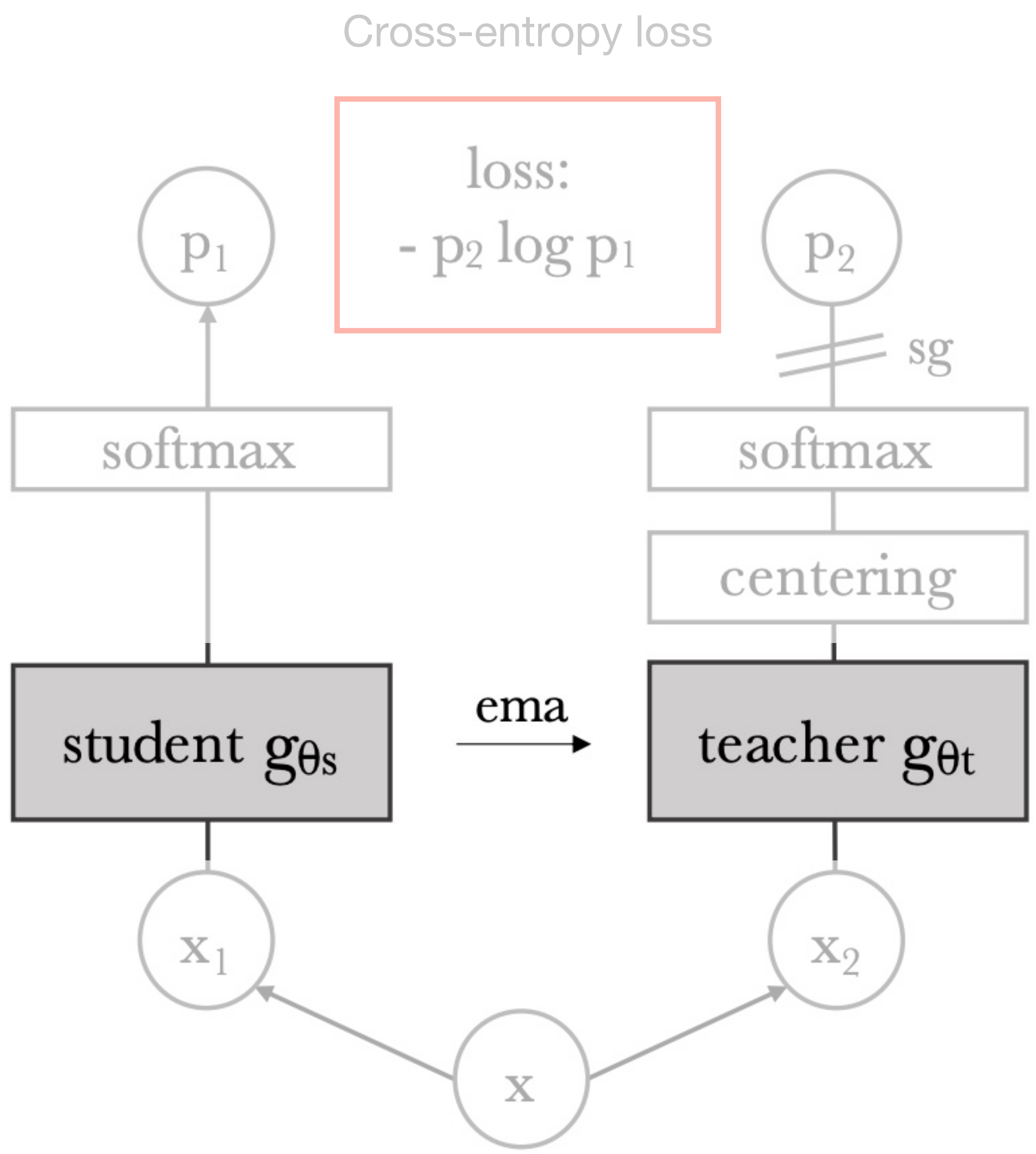
"Projection head"

3-layer MLP

Hidden dimension 2048 and l_2 norm.

Output to K dimensions.

K	1024	4096	16384	65536	262144
k-NN top-1	67.8	69.3	69.2	69.7	69.1



Self-distillation with no labels (DINO)

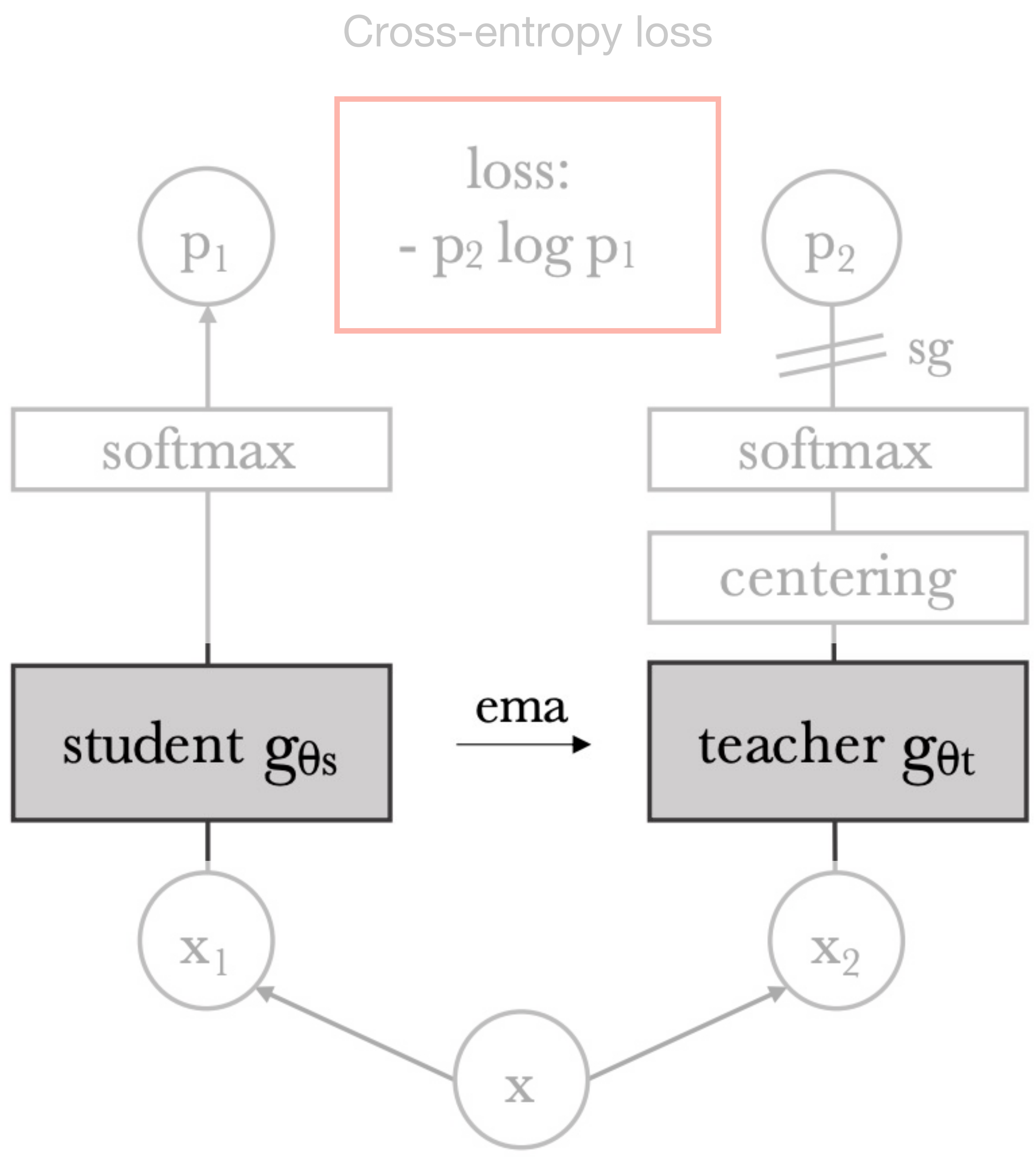
Network architectures

Teacher network

Exponential moving average of the student weights: **momentum encoder.**

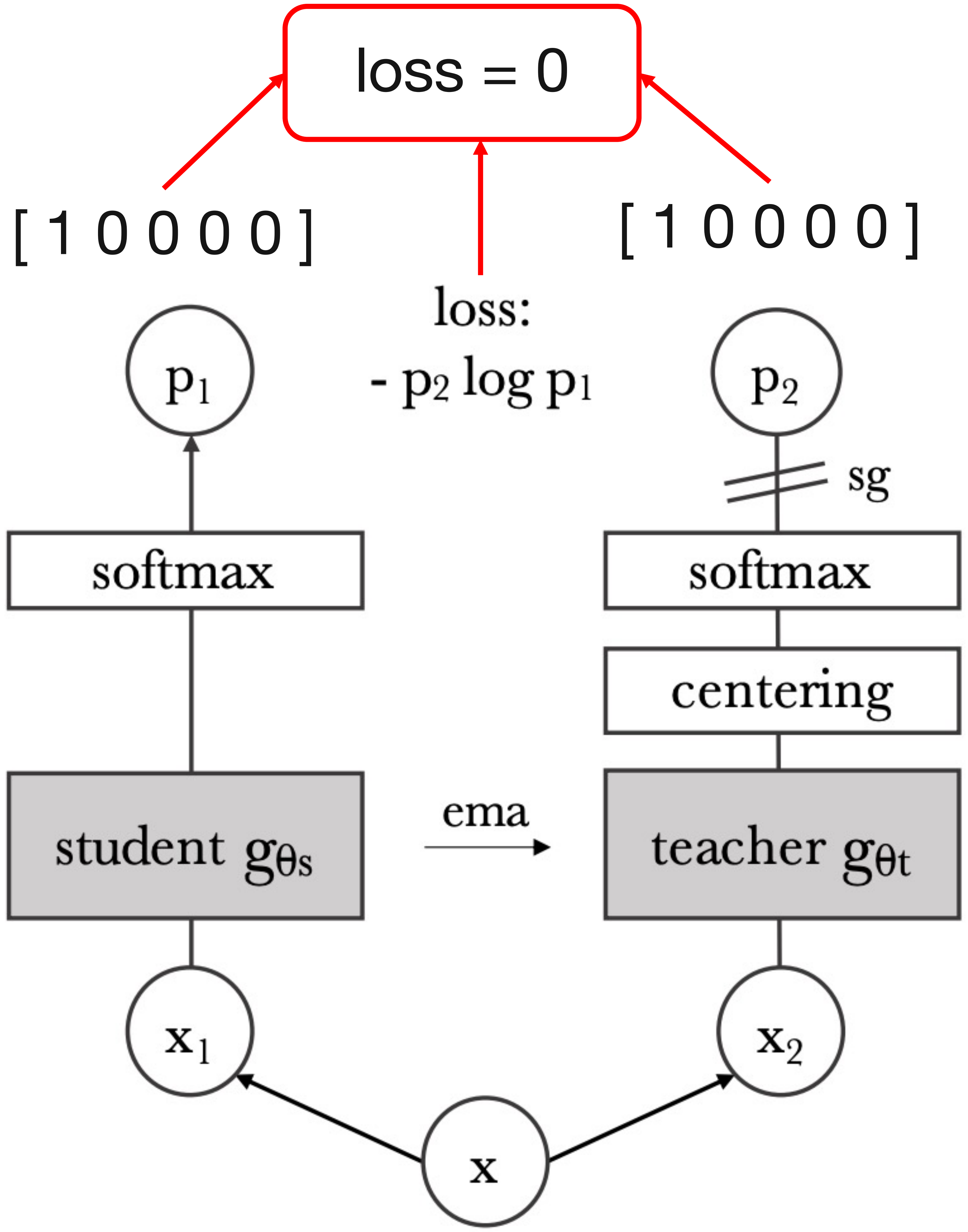
$$\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$$

λ follows a cosine schedule from 0.996 to 1 during training



Self-distillation with no labels (DINO)

What can go wrong?



Collapse!

The loss goes to 0 if both networks always output the same constant value (**no need to learn anything**).

Self-distillation with no labels (DINO)

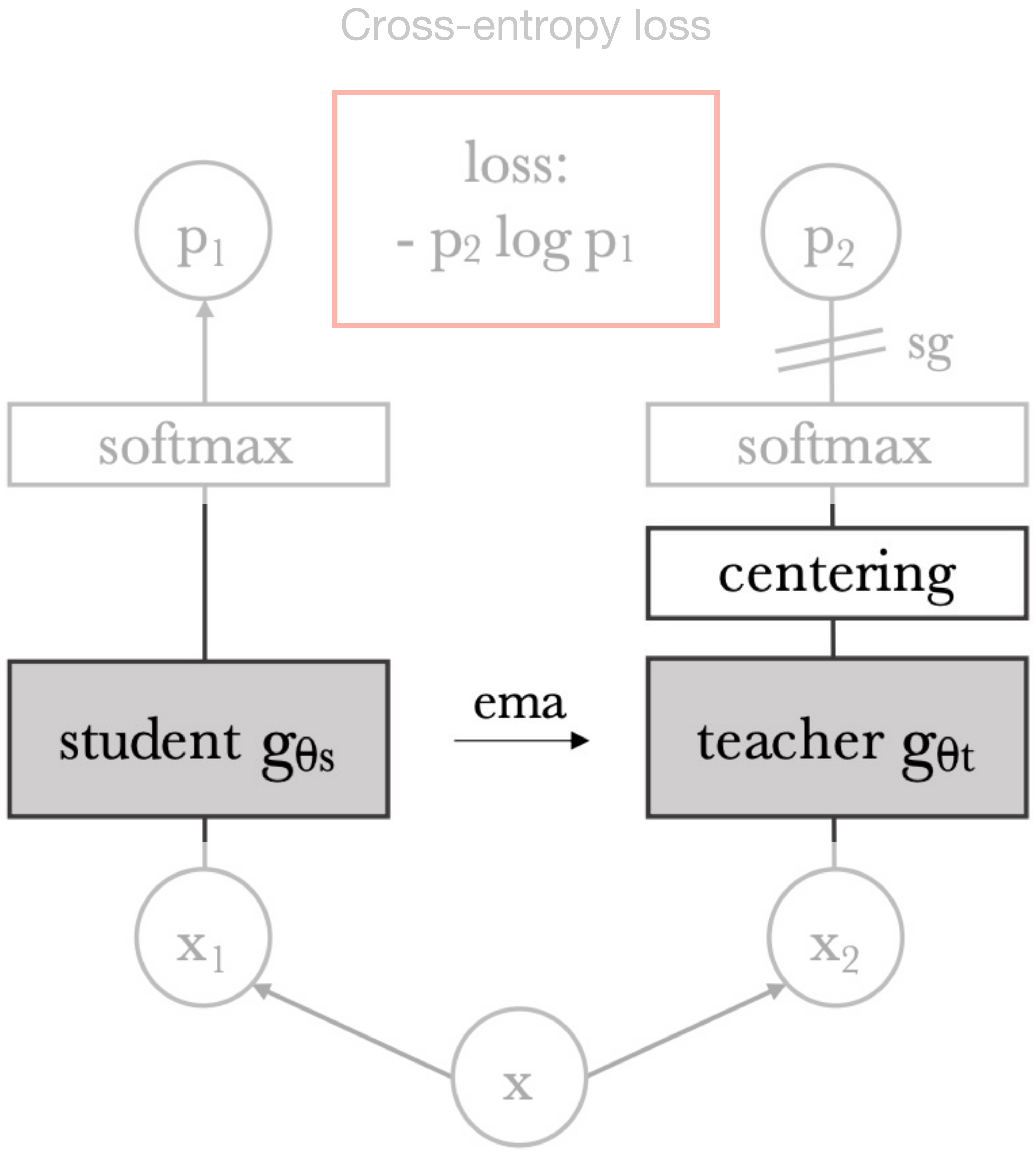
Avoiding collapse

1. Centering

Keep **running average of all representations** seen by the teacher and add it as bias.

"*Avoids the collapse induced by a dominant dimension, but encourages an uniform output*".

$$g_t \leftarrow g_t(x) - c$$



Self-distillation with no labels (DINO)

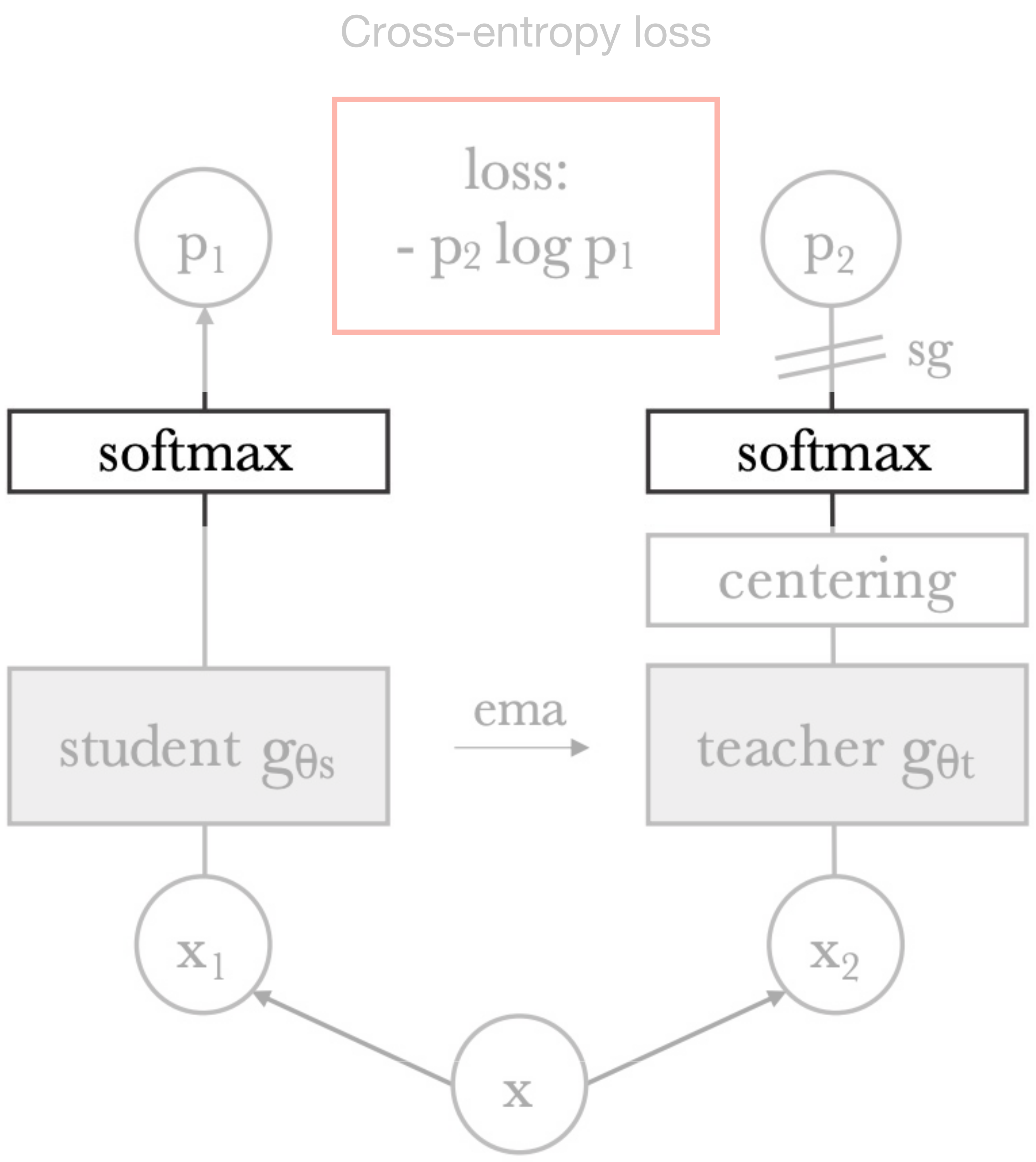
Avoiding collapse

2. Sharpening

Apply different temperature (τ) in the *softmax* for teacher and student.

$$P\left(x^{(i)}\right) = \frac{\exp\left(g_\theta\left(x^{(i)}\right)/\tau\right)}{\Sigma_{k=1}^{K}\exp\left(g_\theta\left(x^{(k)}\right)/\tau\right)}$$

$$\tau_{teacher} \ll \tau_{stud}$$

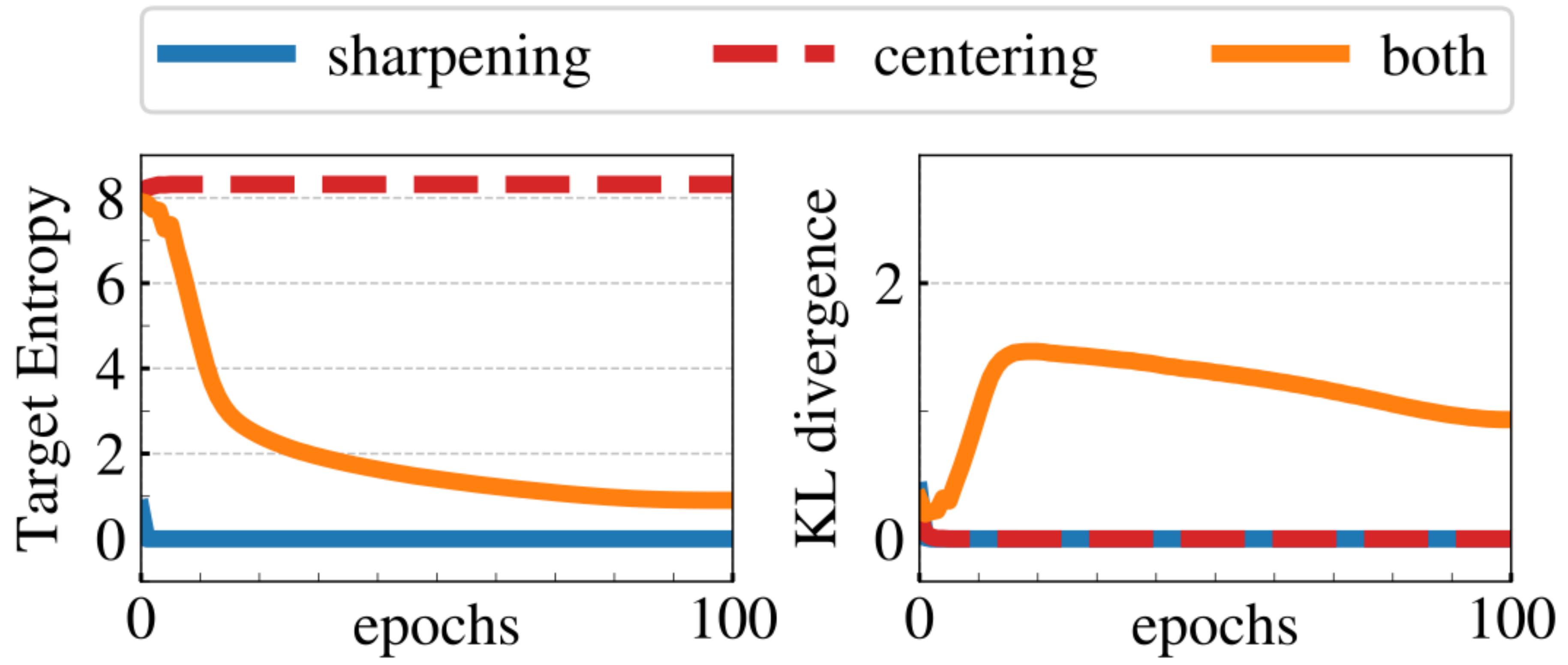


Self-distillation with no labels (DINO)

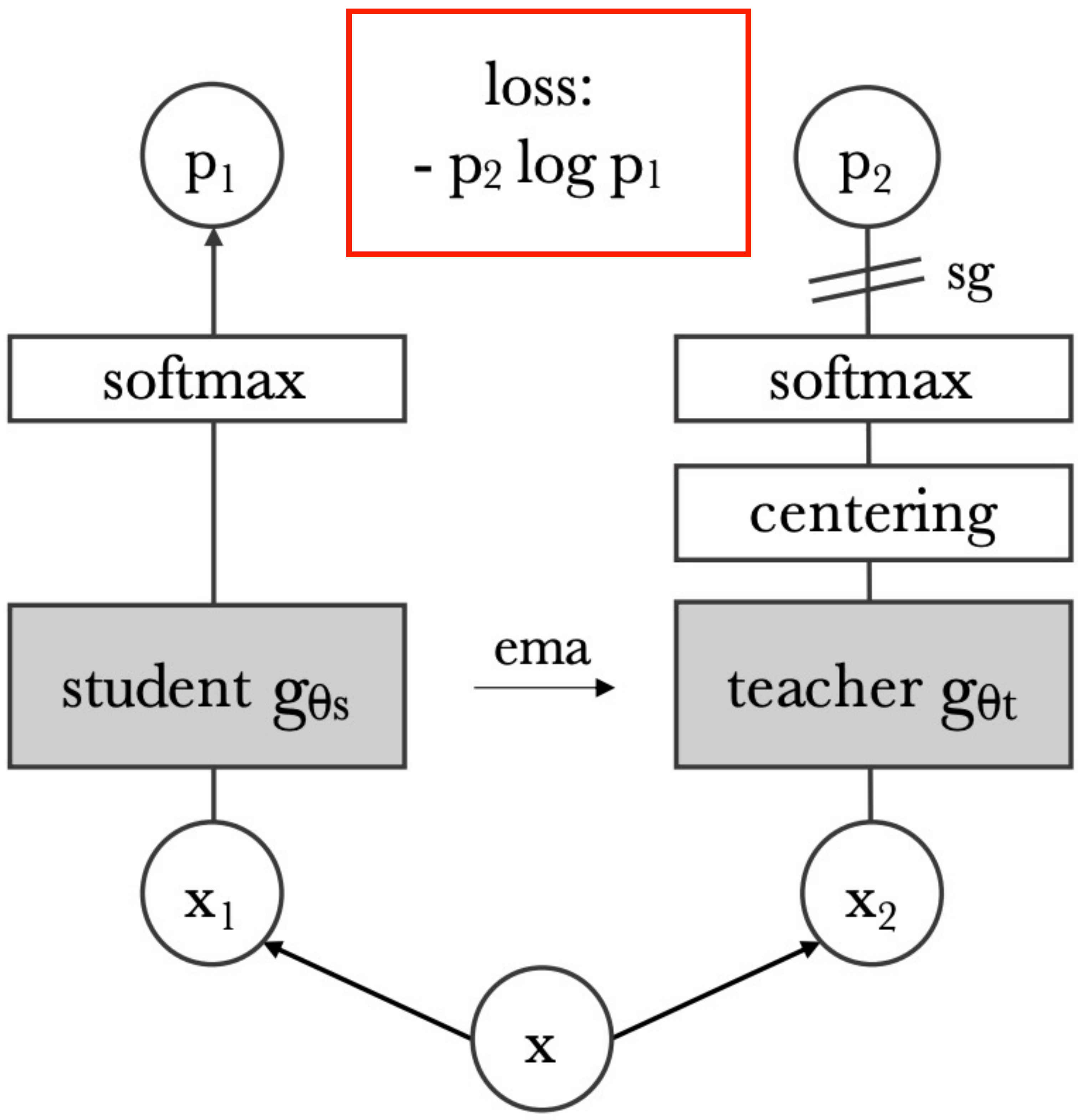
Avoiding collapse

3. Momentum teacher

Having such an architecture combined with the previous methods is enough to avoid collapse under this setup.



Cross-entropy loss



Self-<u>di</u>stillation with <u>no</u> labels (DINO)

Putting it all together

Animation from the DINO official repository: https://github.com/facebookresearch/dino

Implementation and evaluation

Experimental setup

- Train on ImageNet dataset **without labels**
- Architectures:
    - ViTs with different depths and patch sizes (8, 16)
    - ResNet with different hyperparameters and sizes

"Training DINO with ViT takes ***just*** two 8-GPU servers over 3 days to achieve 76.1% on ImageNet benchmark"

Implementation and evaluation

Evaluation results across training strategies

Results on ResNet-50

Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [12]	RN50	23	1237	69.1	60.7
MoCov2 [15]	RN50	23	1237	71.1	61.9
InfoMin [67]	RN50	23	1237	73.0	65.3
BarlowT [81]	RN50	23	1237	73.2	66.0
OBoW [27]	RN50	23	1237	73.8	61.9
BYOL [30]	RN50	23	1237	74.4	64.8
DCv2 [10]	RN50	23	1237	75.2	67.1
SwAV [10]	RN50	23	1237	**75.3**	65.7
DINO	RN50	23	1237	**75.3**	**67.5**

Results on ViT-S

Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	ViT-S	21	1007	79.8	79.8
BYOL* [30]	ViT-S	21	1007	71.4	66.6
MoCov2* [15]	ViT-S	21	1007	72.7	64.4
SwAV* [10]	ViT-S	21	1007	73.5	66.3
DINO	ViT-S	21	1007	**77.0**	**74.5**

Best results obtained when combined with ViT

Implementation and evaluation

Evaluation results across architectures

Method	Arch.	Param.	im/s	Linear	k-NN
SCLR [12]	RN50w4	375	117	76.8	69.3
SwAV [10]	RN50w2	93	384	77.3	67.3
BYOL [30]	RN50w2	93	384	77.4	–
DINO	ViT-B/16	85	312	78.2	76.1
SwAV [10]	RN50w5	586	76	78.5	67.1
BYOL [30]	RN50w4	375	117	78.6	–
BYOL [30]	RN200w2	250	123	79.6	73.9
DINO	ViT-S/8	21	180	79.7	**78.3**
SCLRv2 [13]	RN152w3+SK	794	46	79.8	73.1
DINO	ViT-B/8	85	63	**80.1**	77.4

“A base ViT with 8x8 patches achieves 80.1% top-1 accuracy with 10x less parameters and 1.4x faster run time than previous state of the art”

Implementation and evaluation

Performance on k-NN shows latent space preserves properties

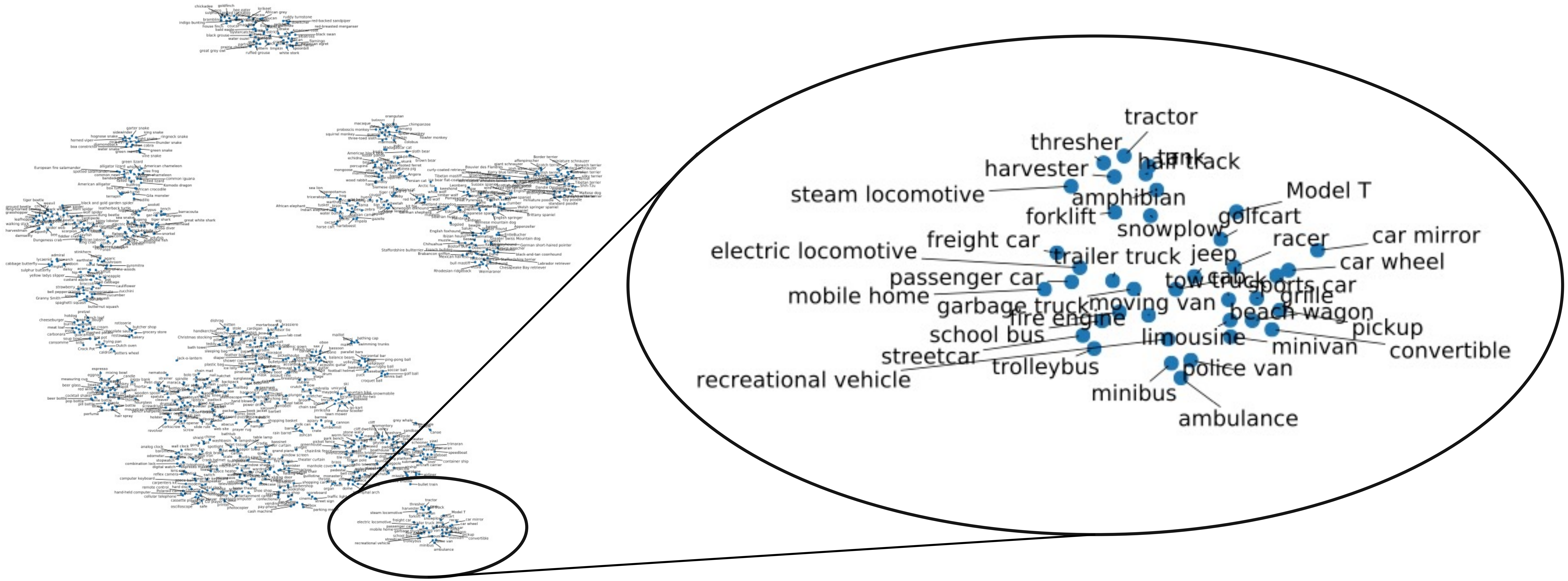


Figure 11: t-SNE visualization of ImageNet classes as represented using DINO. For each class, we obtain the embedding by taking the average feature for all images of that class in the validation set.

Similar classes are found close in space

Properties of ViT trained with DINO

They outperform supervised training in different tasks

Table 3: **Image retrieval.** We compare the performance in retrieval of off-the-shelf features pretrained with supervision or with DINO on ImageNet and Google Landmarks v2 (GLDv2) dataset. We report mAP on revisited Oxford and Paris. Pretraining with DINO on a landmark dataset performs particularly well. For reference, we also report the best retrieval method with off-the-shelf features [57].

			$\mathcal{R}$Ox		$\mathcal{R}$Par	
Pretrain	Arch.	Pretrain	M	H	M	H
Sup. [57]	RN101+R-MAC	ImNet	49.8	18.5	74.0	**52.1**
Sup.	ViT-S/16	ImNet	33.5	8.9	63.0	37.2
DINO	ResNet-50	ImNet	35.4	11.1	55.9	27.5
DINO	ViT-S/16	ImNet	41.8	13.7	63.1	34.4
DINO	ViT-S/16	GLDv2	**51.5**	**24.3**	**75.3**	51.6

Table 4: **Copy detection.** We report the mAP performance in copy detection on Copydays "strong" subset [21]. For reference, we also report the performance of the multigrain model [5], trained specifically for particular object retrieval.

Method	Arch.	Dim.	Resolution	mAP
Multigrain [5]	ResNet-50	2048	224^2	75.1
Multigrain [5]	ResNet-50	2048	largest side 800	82.5
Supervised [69]	ViT-B/16	1536	224^2	76.4
DINO	ViT-B/16	1536	224^2	81.7
DINO	ViT-B/8	1536	320^2	**85.5**

Properties of ViT trained with DINO

They generalize better to downstream tasks

Table 6: **Transfer learning by finetuning pretrained models on different datasets.** We report top-1 accuracy. Self-supervised pretraining with DINO transfers better than supervised pretraining.

	Cifar_{10}	Cifar_{100}	INat_{18}	INat_{19}	Flwrs	Cars	INet
ViT-S/16							
Sup. [69]	**99.0**	89.5	70.7	76.6	98.2	92.1	79.9
DINO	**99.0**	**90.5**	**72.0**	**78.2**	**98.5**	**93.0**	**81.5**
ViT-B/16							
Sup. [69]	99.0	90.8	**73.2**	77.7	98.4	92.1	81.8
DINO	**99.1**	**91.7**	72.6	**78.6**	**98.8**	**93.0**	**82.8**

Properties of ViT trained with DINO

Model learns understandable features -> object segmentation

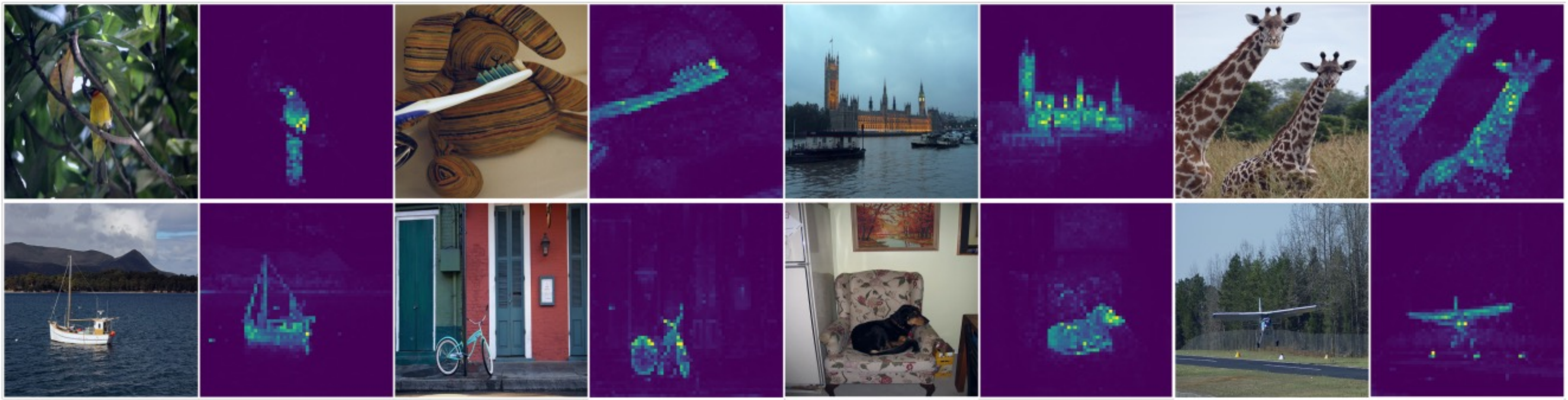


Figure 1: **Self-attention from a Vision Transformer with 8×8 patches trained with no supervision.** We look at the self-attention of the [CLS] token on the heads of the last layer. This token is not attached to any label nor supervision. These maps show that the model automatically learns class-specific features leading to unsupervised object segmentations.

Properties of ViT trained with DINO

Model learns understandable features -> object segmentation

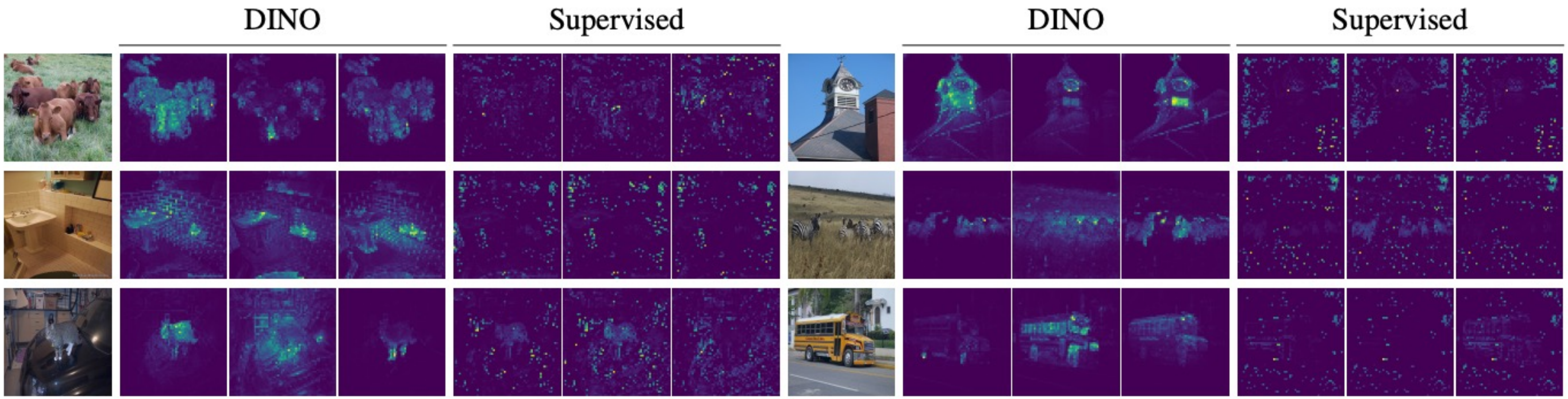


Figure 10: **Self-attention heads from the last layer.** We look at the attention map when using the [CLS] token as a query for the different heads in the last layer. Note that the [CLS] token is not attached to any label or supervision.

Properties of ViT trained with DINO

Model learns understandable features -> object segmentation

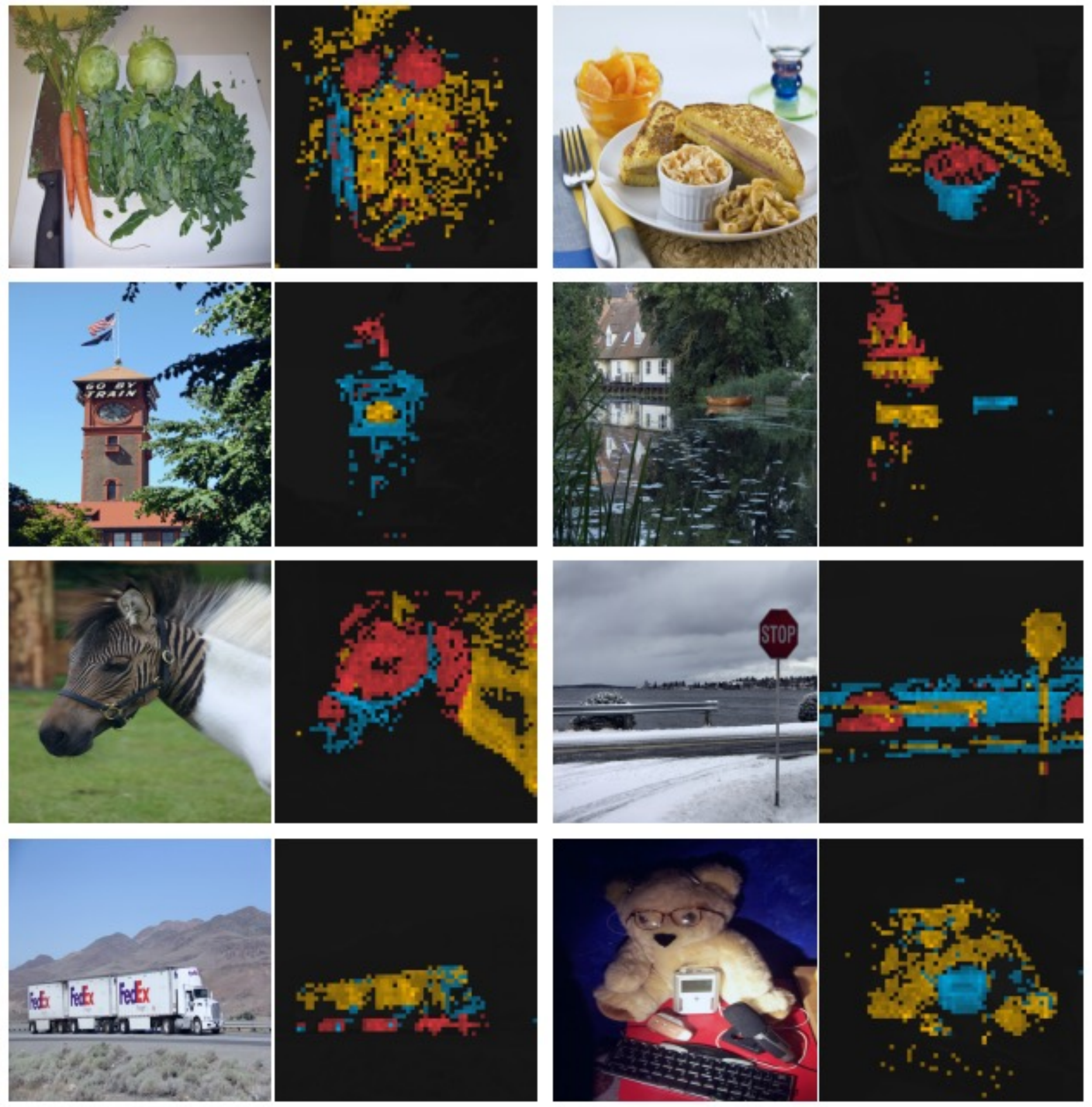


Figure 3: **Attention maps from multiple heads.** We consider the heads from the last layer of a ViT-S/8 trained with DINO and display the self-attention for `[CLS]` token query. Different heads, materialized by different colors, focus on different locations that represents different objects or parts (more examples in Appendix).

Ablation study

How changing the architecture impacts the performance

	Method	Mom.	SK	MC	Loss	Pred.	k-NN	Lin.
1	DINO	✓	✗	✓	CE	✗	72.8	76.1
2		✗	✗	✓	CE	✗	0.1	0.1
3		✓	✓	✓	CE	✗	72.2	76.0
4		✓	✗	✗	CE	✗	67.9	72.5
5		✓	✗	✓	MSE	✗	52.6	62.4
6		✓	✗	✓	CE	✓	71.8	75.6
7	BYOL	✓	✗	✗	MSE	✓	66.6	71.4
8	MoCov2	✓	✗	✗	INCE	✗	62.0	71.6
9	SwAV	✗	✓	✓	CE	✗	64.7	71.8

SK: Sinkhorn-Knopp, MC: Multi-Crop, Pred.: Predictor
CE: Cross-Entropy, MSE: Mean Square Error, INCE: InfoNCE

"The best combination is the momentum encoder with the multicrop augmentation and the cross-entropy loss"

Ablation study

How changing the architecture impacts the performance

Patch size

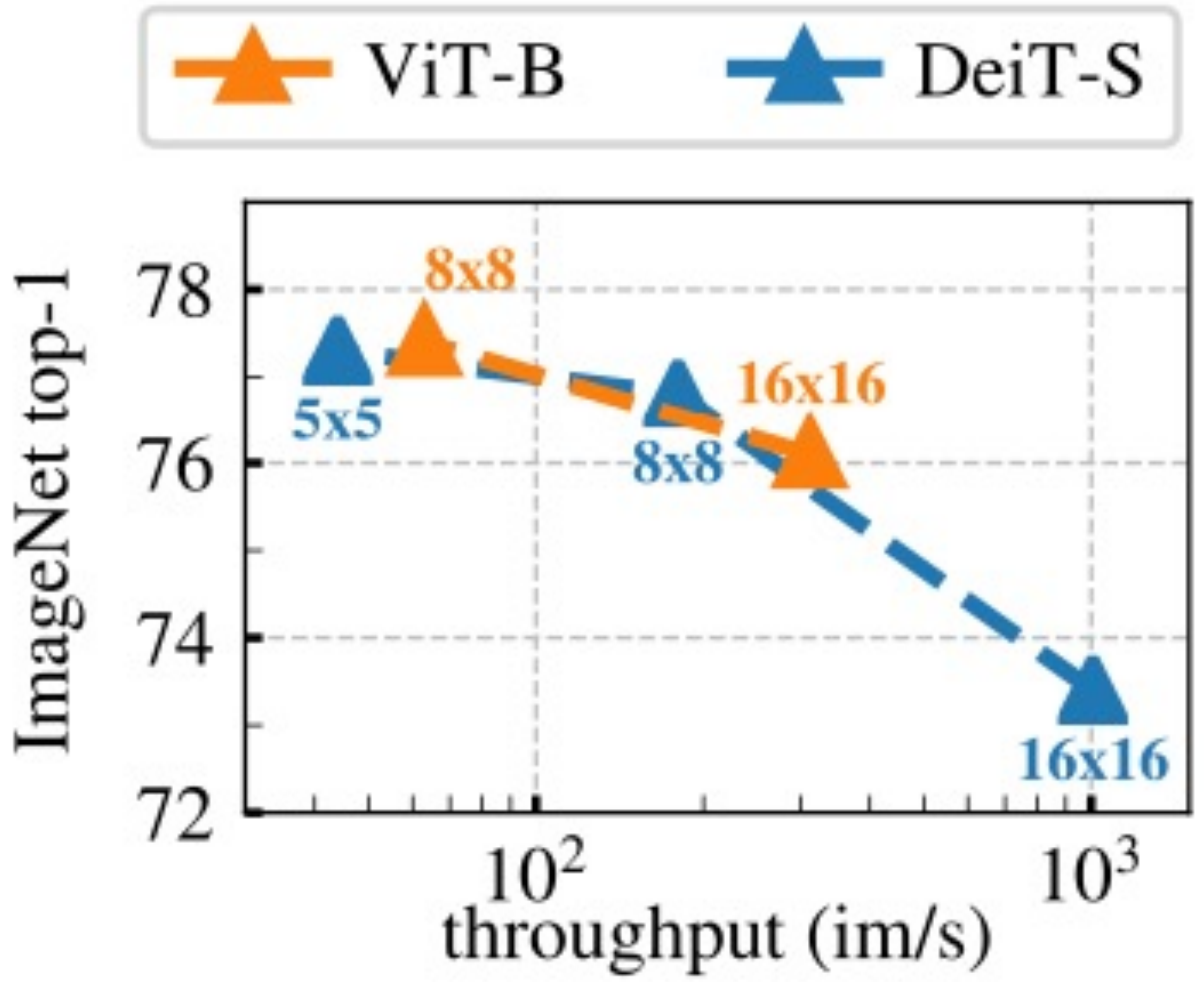


Figure 5: **Effect of Patch Size.** k-NN evaluation as a function of the throughputs for different input patch sizes with ViT-B and ViT-S. Models are trained for 300 epochs.

Batch size

bs	128	256	512	1024
top-1	57.9	59.1	59.6	59.9

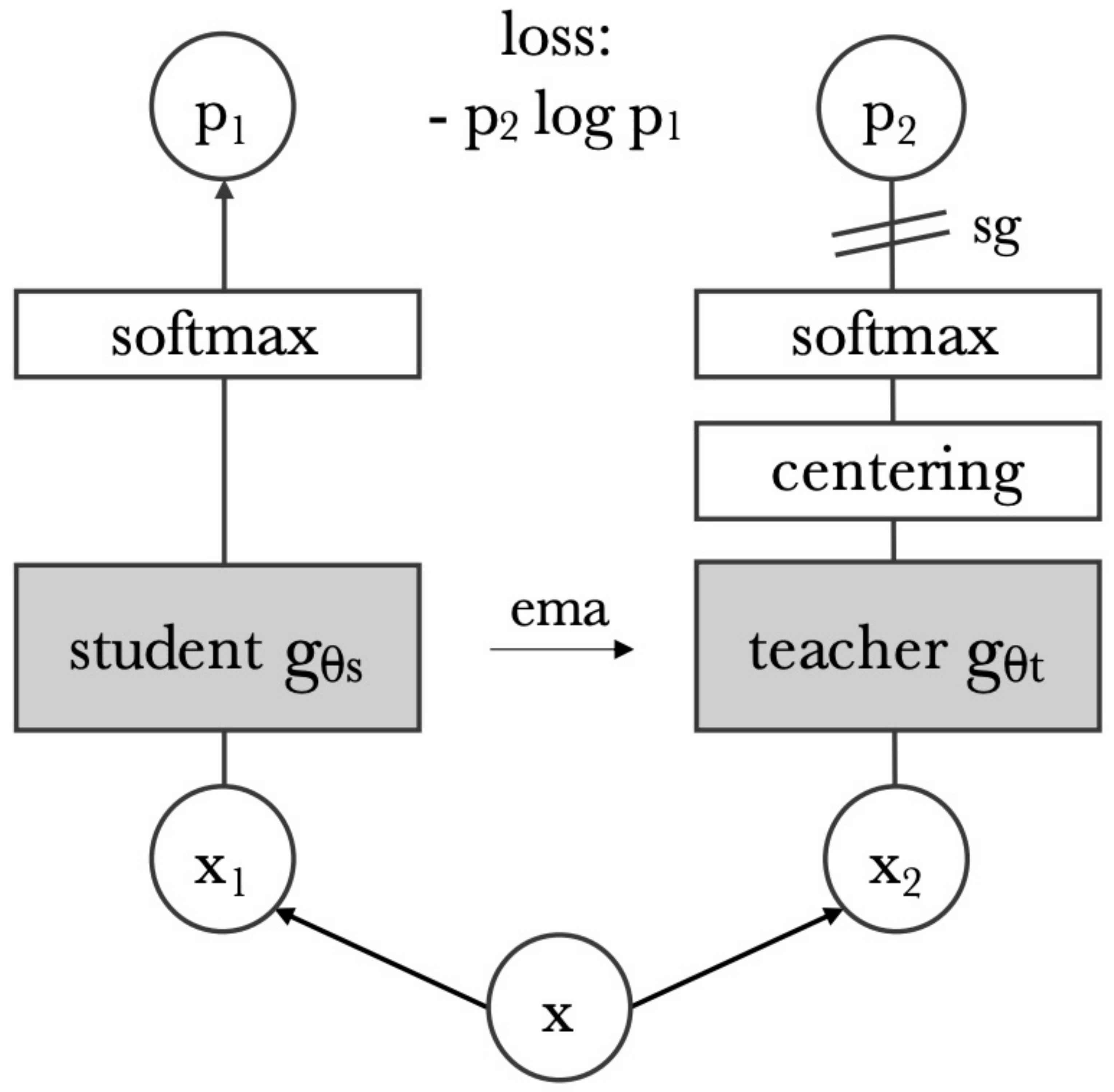
Table 9: **Effect of batch sizes.** Top-1 with k-NN for models trained for 100 epochs without multi-crop.

Conclusion

Self-Supervised Learning for Vision Transformers

Novel self-supervised training approach to unlock the potential of Vision Transformers:

- Can train on unlabeled data
- Learn representations have interesting properties
- Results are comparable with state-of-the-art supervised strategies



Emerging Properties in Self-Supervised Vision Transformers

Also known as DINO

Mathilde Caron[1,2] Hugo Touvron[1,3] Ishan Misra[1] Hervé Jegou[1]
Julien Mairal[2] Piotr Bojanowski[1] Armand Joulin[1]

Javier Rando Ramirez

[1] Facebook AI Research [2] Inria* [3] Sorbonne University