

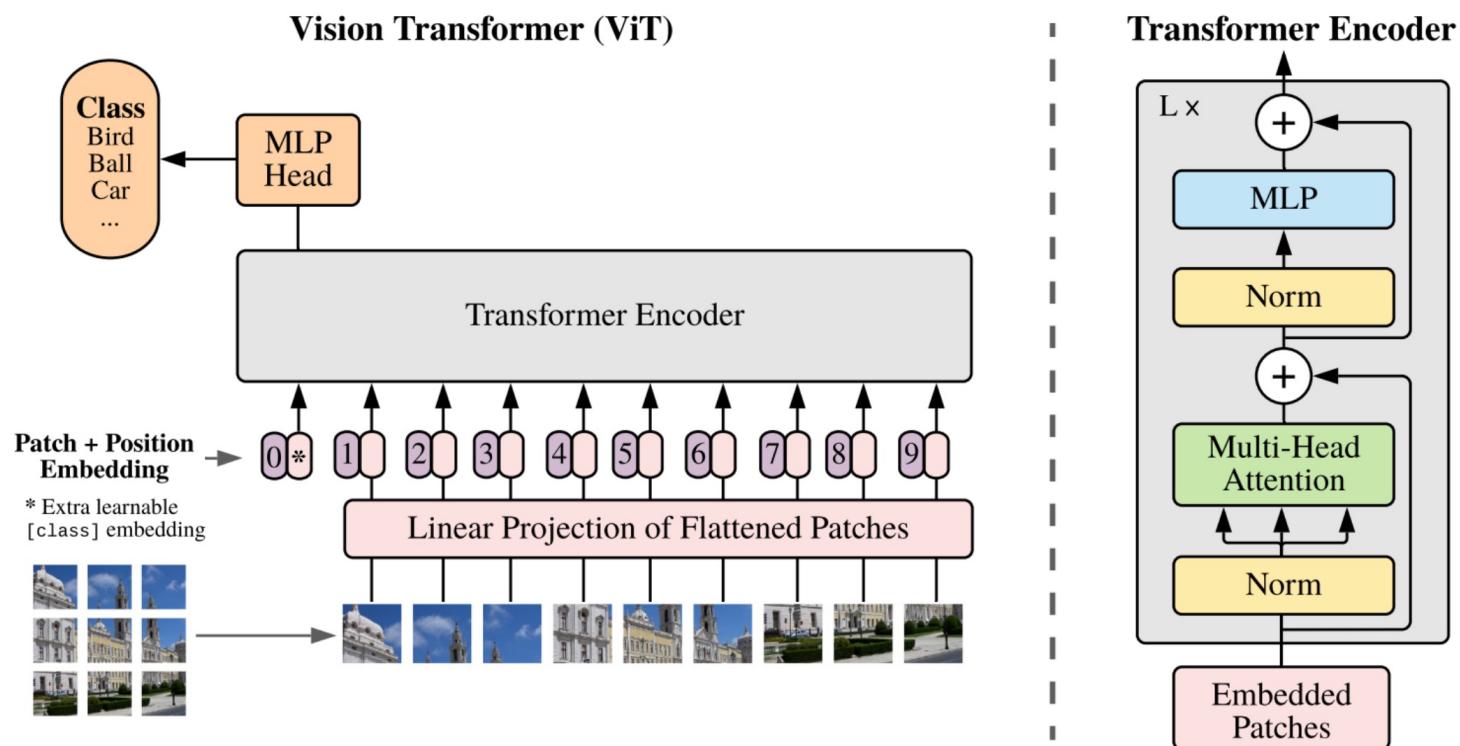
DINO: Emerging Properties in Self-Supervised Vision Transformers

Докладчик: Михненко Наталья, БПМИ182

Мотивация

- **Проблема:** трансформеры в распознавании образов хоть и конкурируют со сверточными сетями, но явных преимуществ не дают.
- **Идея:** в NLP задачах большую роль в успехе трансформера сыграло self-supervised предобучение (BERT, GPT)

Архитектура ViT

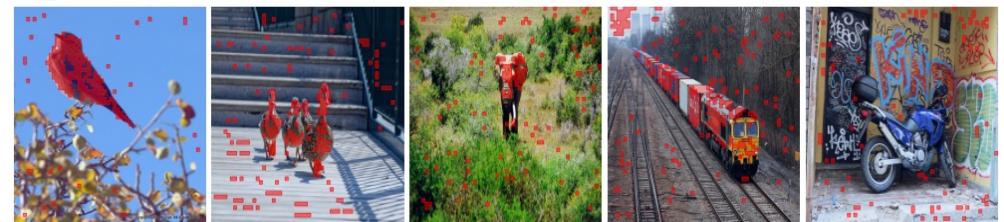


Результаты ViT

ViT хорош не только в supervised режиме:

- карты признаков содержат почти семантическую сегментацию изображения
- эти карты признаков сами по себе дают 78.3% accuracy на ImageNet на kNN

Supervised



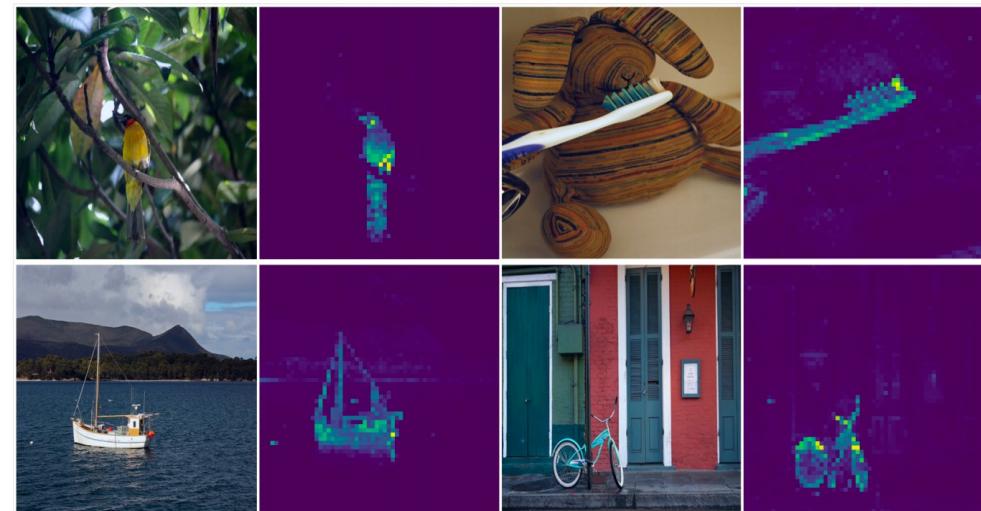
DINO



DINO: self-[di]stillation with [no] labels

Используем:

- momentum encoder
- multi-crop аугментация
- маленький размер патча



Итог: уже 80.1% на ImageNet

DINO: обучение

Пусть:

g_{θ_s} - ученик, g_{θ_t} - учитель

θ_s, θ_t - соответственные параметры

x – изображение на входе

P_s, P_t - вероятности на выходе

Применяем softmax с температурным шкалированием:

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^K \exp(g_{\theta_s}(x)^{(k)}/\tau_s)}$$

DINO: обучение

Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

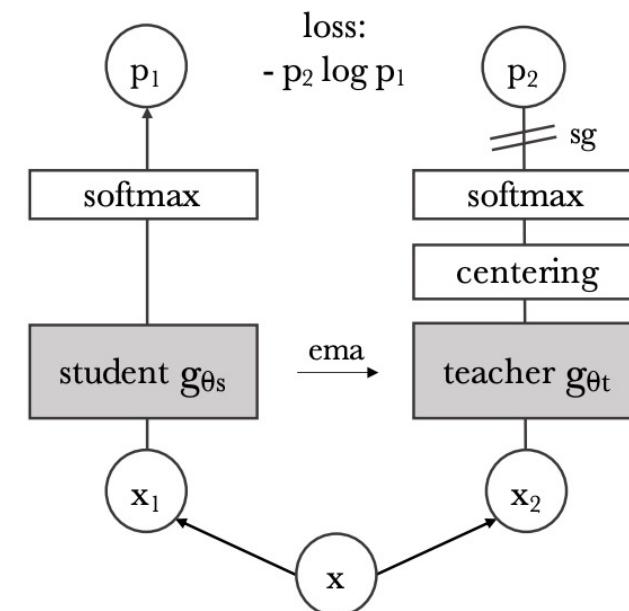
```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```



DINO: функция потерь

$$\min_{\theta_s} H(P_t(x), P_s(x)) \quad H(a, b) = -a \log b$$

Адаптируем под self-supervised:

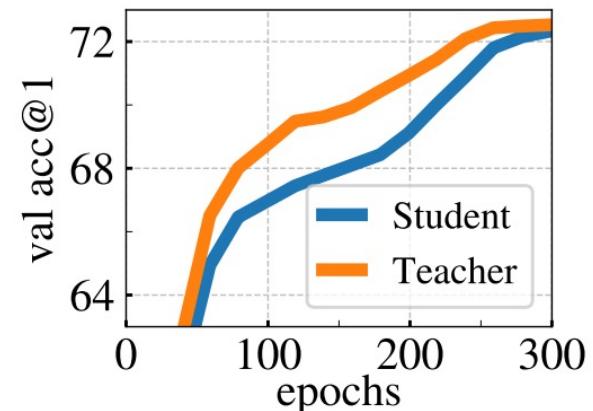
$$\min_{\theta_s} \sum_{x \in \{x_1^g, x_2^g\}} \sum_{\substack{x' \in V \\ x' \neq x}} H(P_t(x), P_s(x'))$$

v – маленькие части изображения,
g - большие

DINO: сеть-учитель

- Изначально обученного учителя нет, поэтому строим его из весов студента
- Хорошо работает фиксировать веса на всю эпоху
- Хорошо работает momentum encoder:
λ обновляется по косинусоиде от 0.996 до 1

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



Teacher	Top-1
Student copy	0.1
Previous iter	0.1
Previous epoch	66.6
Momentum	72.8

DINO: как избежать коллапса

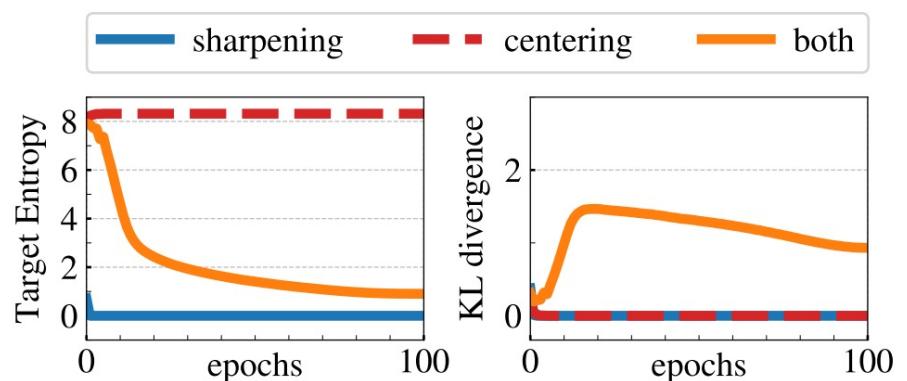
- Centering и sharpening

$$g_t(x) \leftarrow g_t(\hat{x}) + c$$

$$c \leftarrow mc + (1 - m) \frac{1}{B} \sum_{i=1}^B g_{\theta_t}(x_i)$$

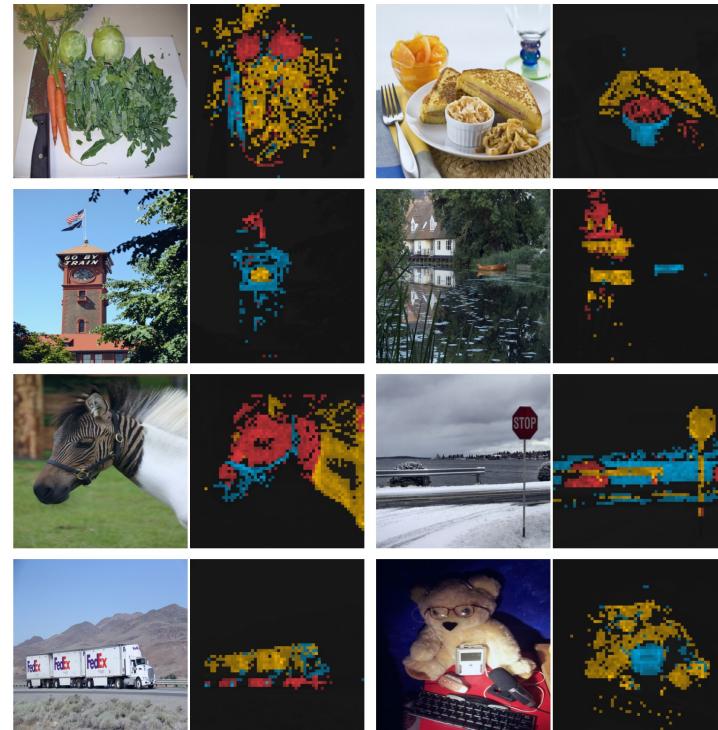
m – гиперпараметр,

B – размер батча



DINO: гиперпараметры

- ViT-S/16
- AdamW
- Размер батча 1024
- 16 gpus, 3 дня
- learning rate scheduling
- аугментации



Эксперименты

Method	Arch.	Param.	im/s	Linear	<i>k</i> -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
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

Эксперименты

DINO + ResNet50 < supervised ViT < DINO + ViT

Method	Arch.	Param.	im/s	Linear	<i>k</i> -NN
<i>Comparison across architectures</i>					
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

Image Retrieval

Pretrain	Arch.	Pretrain	$\mathcal{R}\text{Ox}$		$\mathcal{R}\text{Par}$	
			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

Copy detection

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

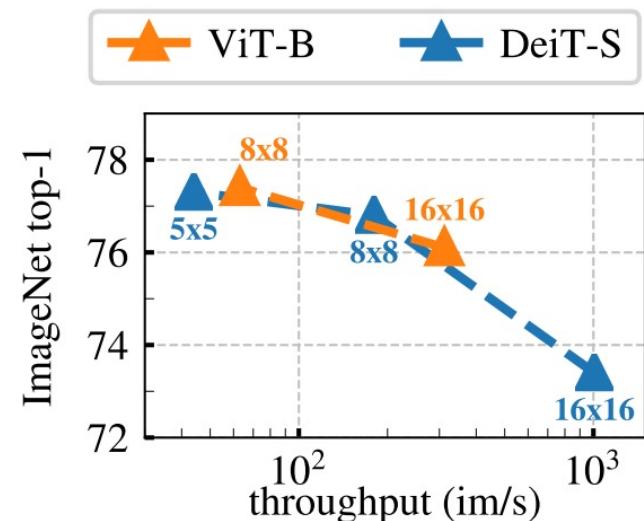
Трансферное обучение

	Cifar ₁₀	Cifar ₁₀₀	INat ₁₈	INat ₁₉	Flwrs	Cars	INet
<i>ViT-S/16</i>							
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
<i>ViT-B/16</i>							
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

Влияние компонентов и параметров

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



Итоги

- Self-supervised методы позволяют добиться лучших результатов
- Карты признаков содержат почти семантическую сегментацию изображения
- SSL может быть ключом для разработки модели, основанной на ViT, подобной BERT в NLP

Рецензия

Сильные стороны:

- Актуальность
- Новые дальнейшие изучения на основе данной статьи
- Высокая обобщающая способность

Слабые стороны:

- Отсутствуют примеры плохих результатов модели
- Нет описания инициализации весов

Практик-исследователь

Кто: Facebook AI Research, Inria, Sorbonne University

Когда: весна 2021

Цитаты и продолжения: Под 90 цитирований, но ничего
интересного

(кроме “torch.manual seed(3407) is all you need”)

Связь с другими работами

Method	Mom.	SK	MC	Loss	Pred.	<i>k</i> -NN	Lin.
DINO	✓	✗	✓	CE	✗	72.8	76.1
BYOL	✓	✗	✗	MSE	✓	66.6	71.4
MoCov2	✓	✗	✗	INCE	✗	62.0	71.6
SwAV	✗	✓	✓	CE	✗	64.7	71.8

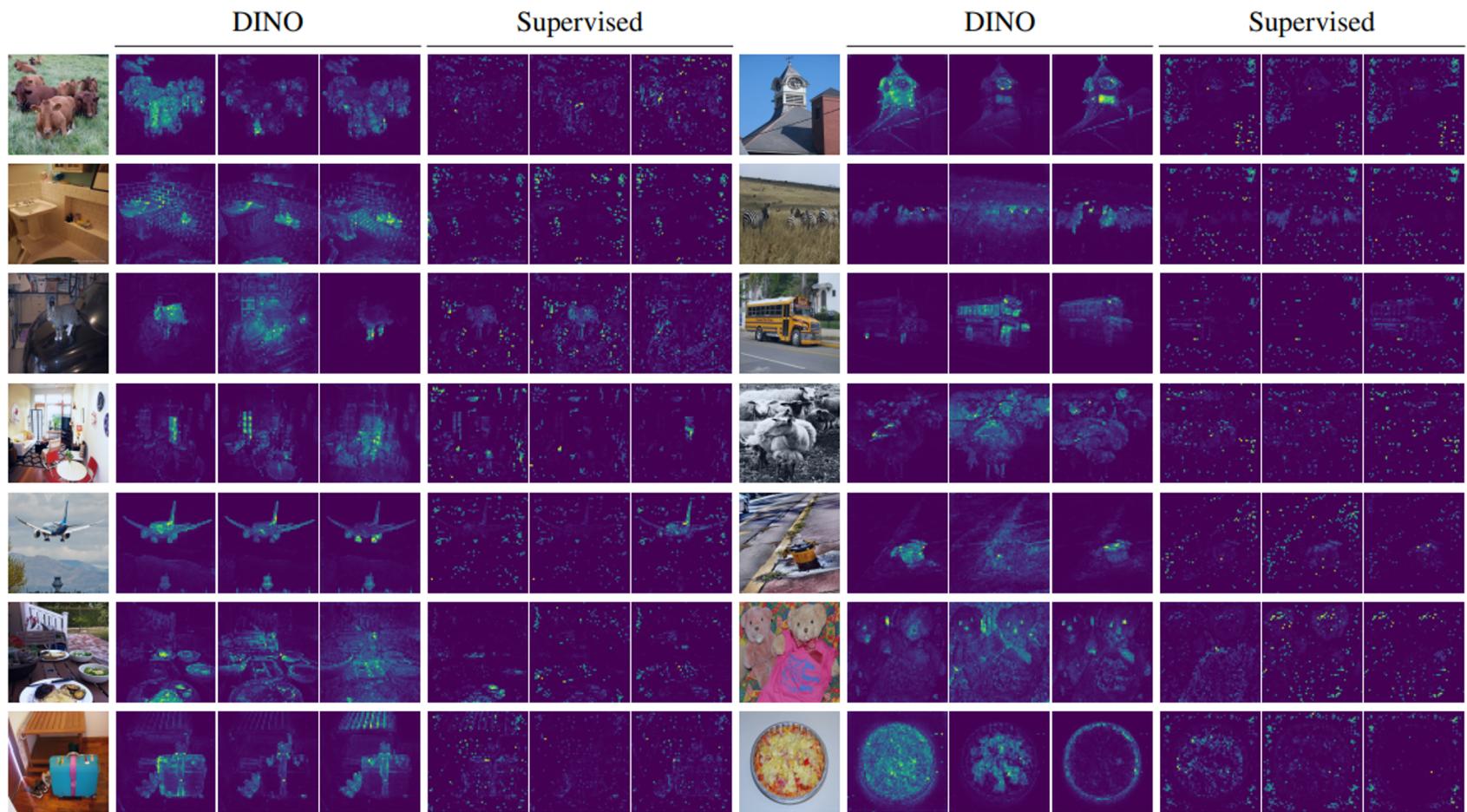
SK: Sinkhorn-Knopp, MC: Multi-Crop, Pred.: Predictor

CE: Cross-Entropy, MSE: Mean Square Error, INCE: InfoNCE

CsMI - Вышла в то же время. Другой loss и конволюции вместо трансформера.

UIC - По-другому борются с коллапсами и не используют скользящее среднее.

Применение



Применение

Query



DINO

96.4%

AVERAGE PRECISION



Multigrain architecture

90.7%

AVERAGE PRECISION



Supervised ViT

89%

AVERAGE PRECISION

