Self-Supervised Learning

Quick dive in

Andrey Popov 29.03.2024

План

- Введение: мотивация, возможности SSL методов
- Методы
- Валидация SSL подходов
- Техники обучения
- Ускорение обучения
- Другие подходы к SSL

Введение: мотивация и возможности

Что такое Self-Supervised Learning

- Self-Supervised Learning (SSL) - это тип обучения представлений, который позволяет получить хорошее представление данных из немаркированных наборов данных.

Введение: мотивация и возможности

Что такое Self-Supervised Learning

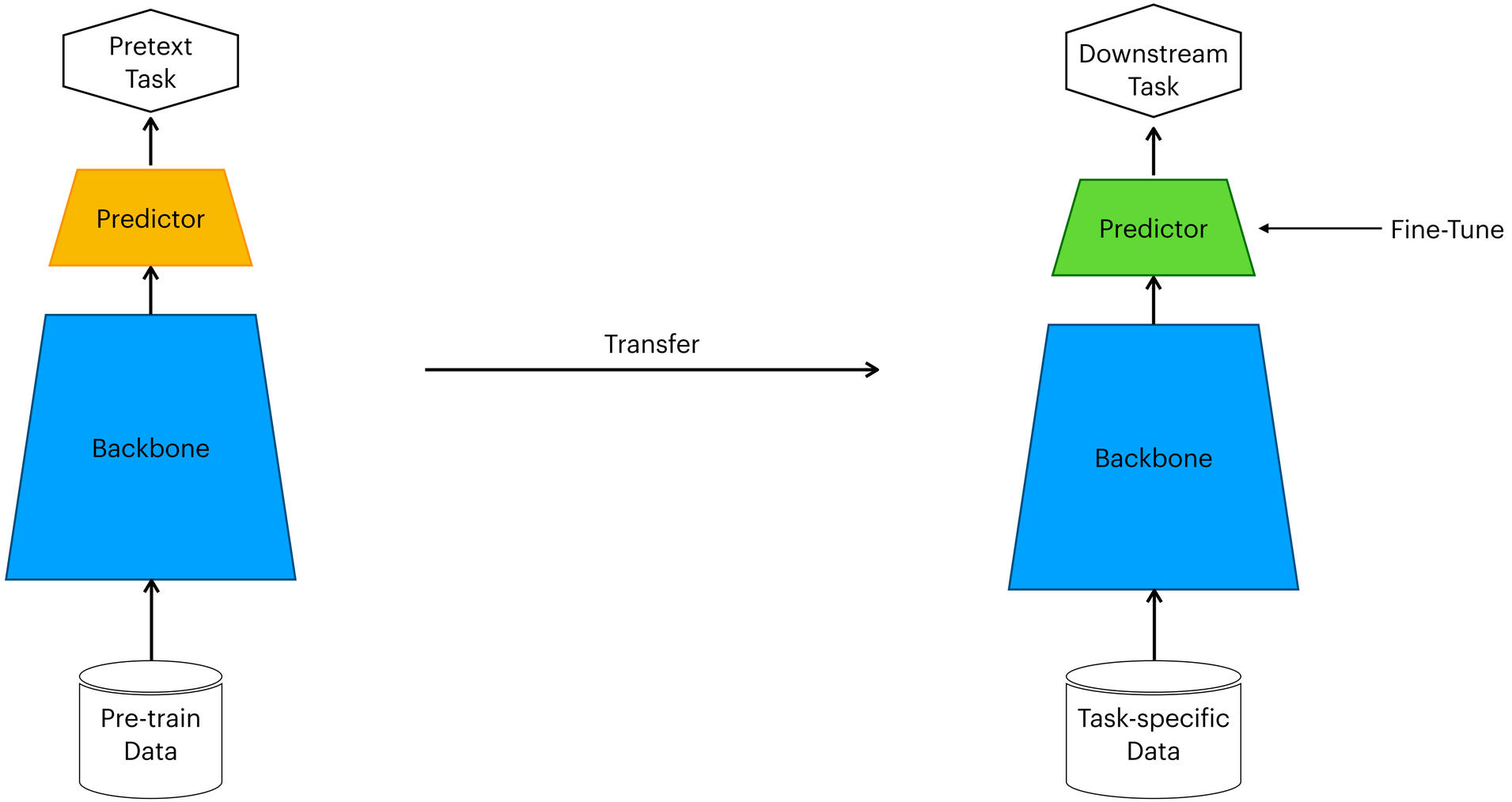
- Self-Supervised Learning (SSL) - это тип обучения представлений, который позволяет получить хорошее представление данных из немаркированных наборов данных. **Зачем?**

1. Разметка данных требует больших затрат, поэтому кол-во датасетов с качественными метками ограничено.
2. Обучение хорошему представлению (representation) облегчает перенос полезной информации в различные последующие задачи (downstream tasks).

Пайплайн обучения

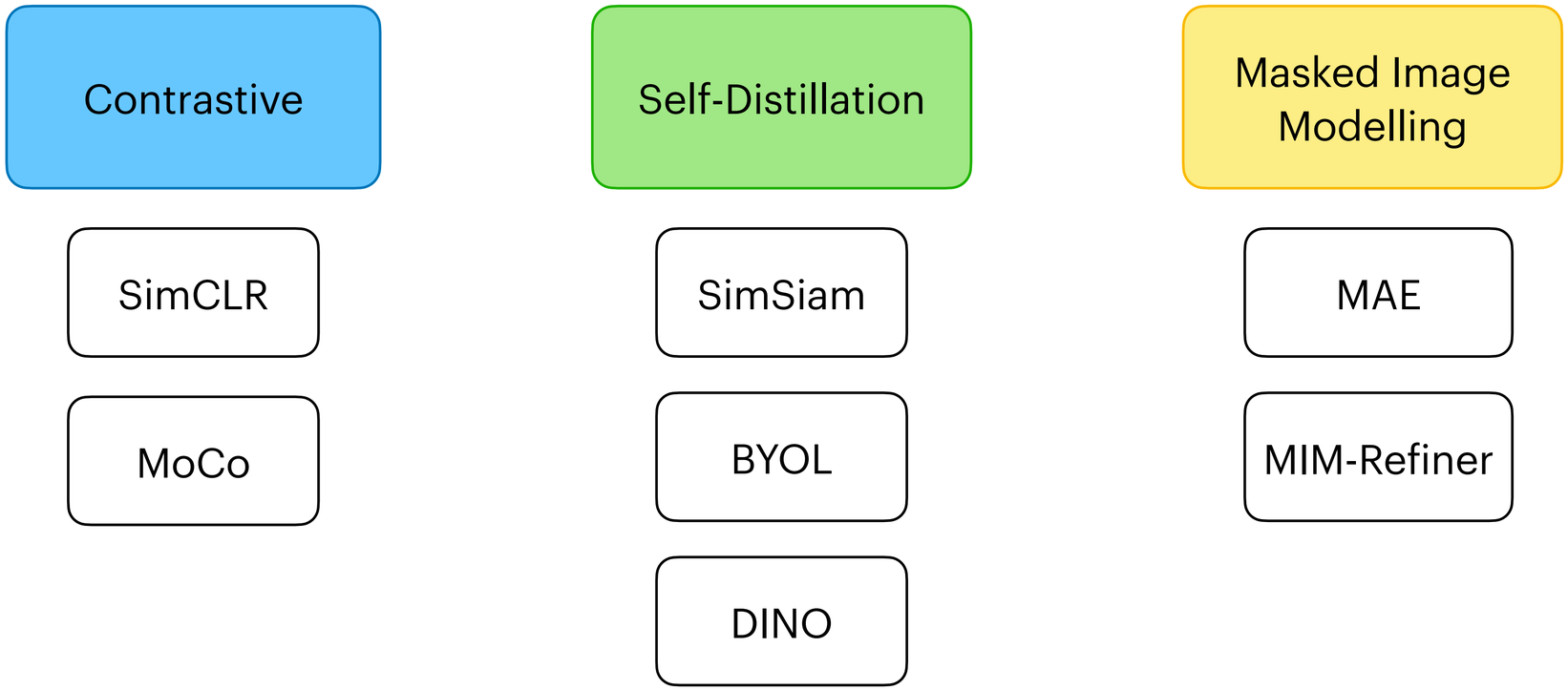
Step 1: Pre-train model for a pretext task

Step 2: Transfer to downstream task



Методы

Таксономия методов



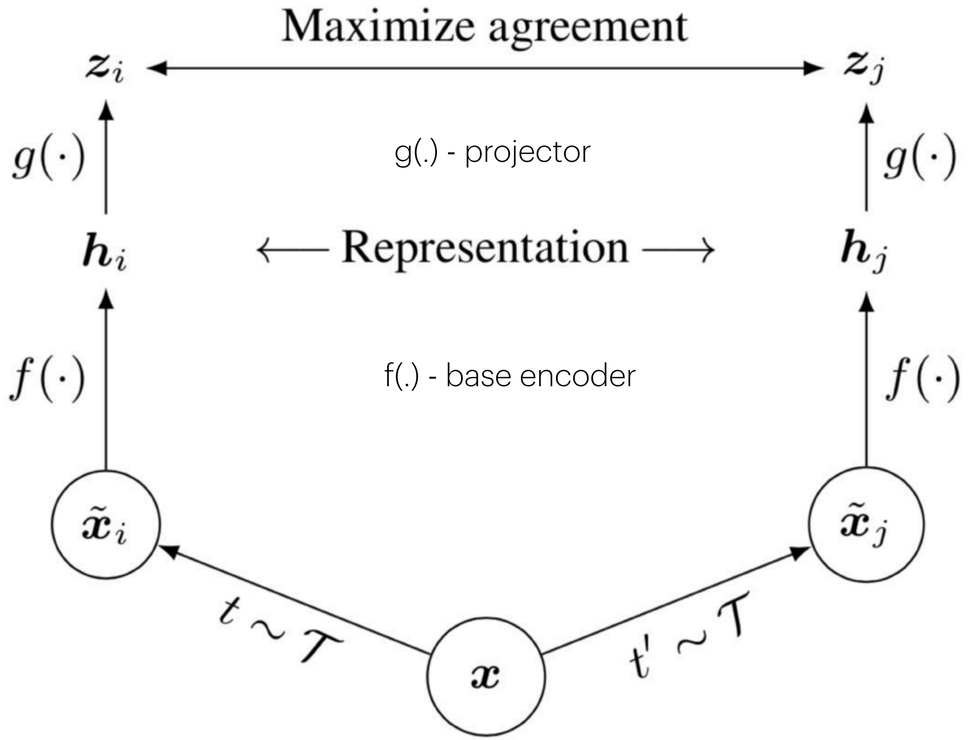
Contrastive SSL

Contrastive SSL

SimCLR (Simple framework for Contrastive Learning of visual Representations)

- SimCLR изучает визуальные признаки, оценивая сходство между двумя аугментированными представлениями изображения.
- Loss: InfoNCE
- Positive: аугментированное изображение
- Negative: другое изображение из mini-batch

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$



Source: Chen et al., 2020

Contrastive SSL

SimCLR - positive samples from augmentations

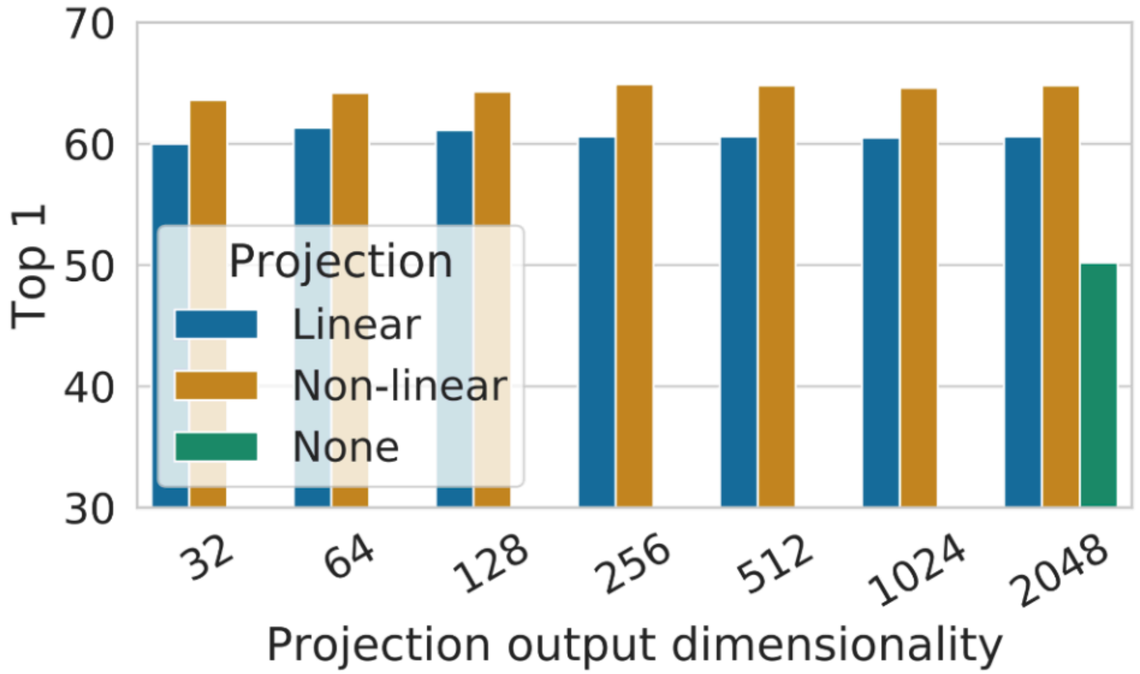
(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

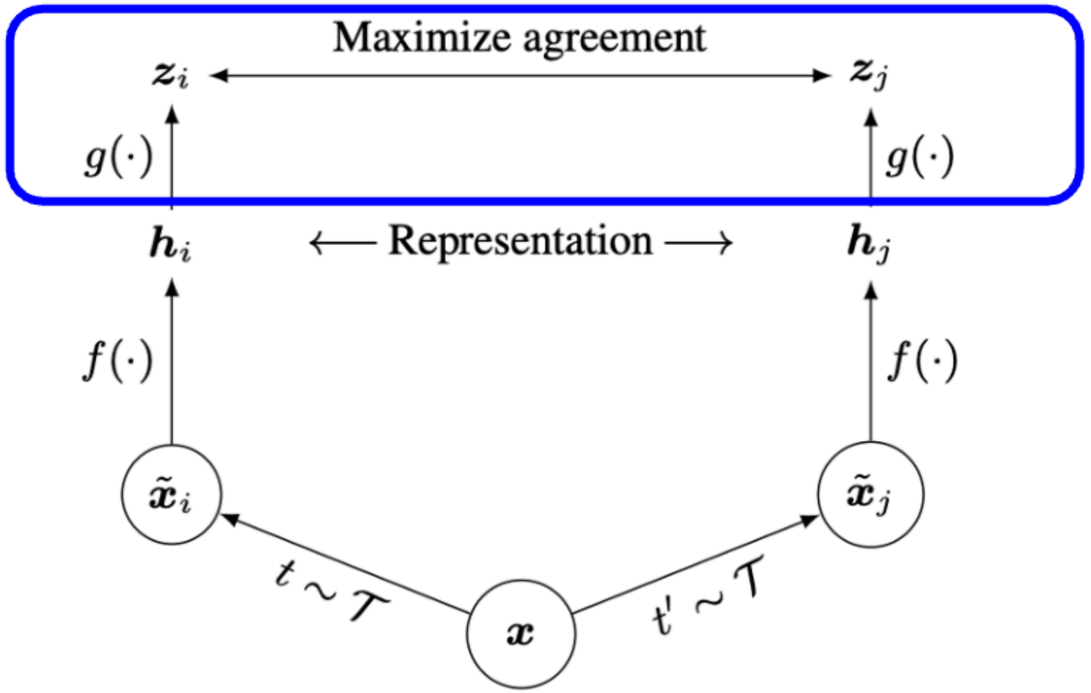
(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Source: Chen et al., 2020

Contrastive SSL

SimCLR - projection head





Source: Chen et al., 2020

Contrastive SSL
SimCLR - projection head

- SimCLR нужен **большой** batch size!

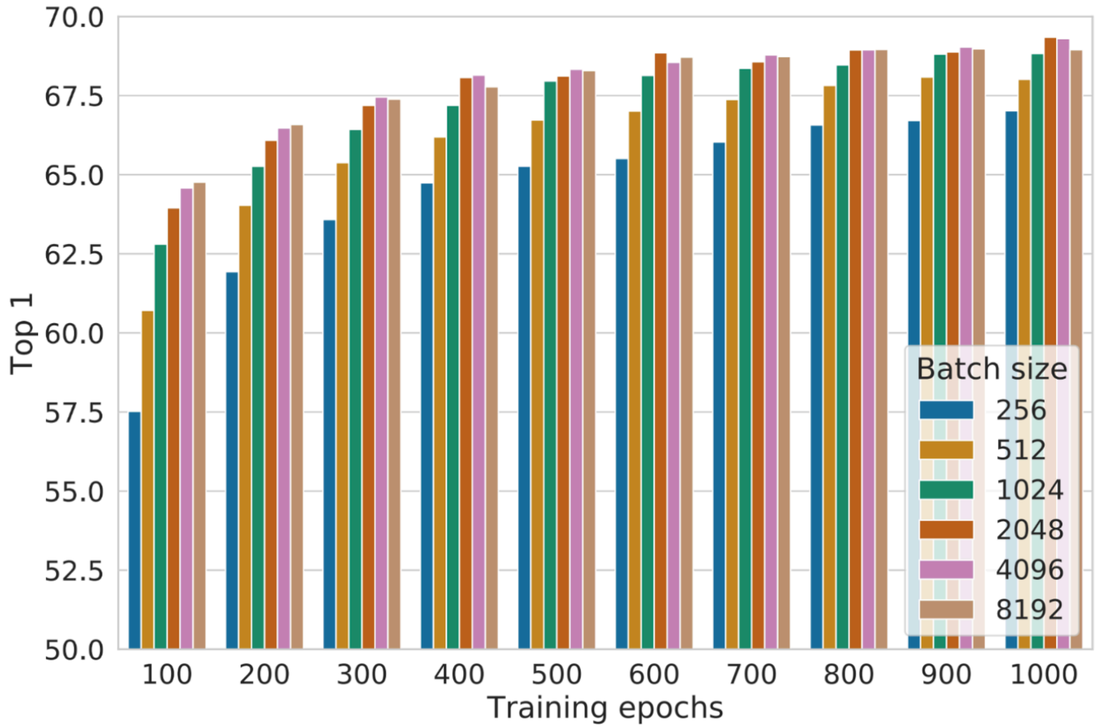


Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

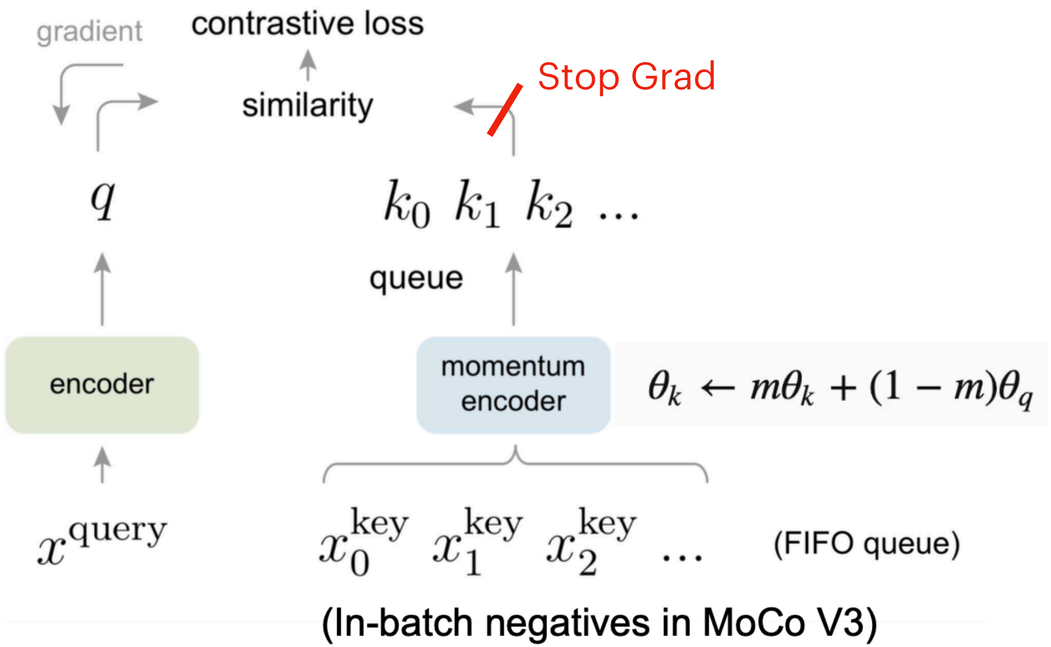
Source: Chen et al., 2020

Contrastive SSL

MoCo (Momentum Contrastive Learning)

- Очередь ключей (отрицательных образцов)
- Независимость размера батча от количества ключей: можно поддерживать большое количество отрицательных образцов.
- Кодировщик ключей медленно улучшается благодаря momentum update:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q$$



Source: He et al., 2020

Contrastive SSL

MoCo (Momentum Contrastive Learning)

Algorithm 1 Pseudocode of MoCo in a PyTorch-like style.

```
# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature

f_k.params = f_q.params # initialize
for x in loader: # load a minibatch x with N samples
   x_q = aug(x) # a randomly augmented version
   x_k = aug(x) # another randomly augmented version

   q = f_q.forward(x_q) # queries: NxC
   k = f_k.forward(x_k) # keys: NxC
   k = k.detach() # no gradient to keys

   # positive logits: Nx1
   l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

   # negative logits: NxK
   l_neg = mm(q.view(N,C), queue.view(C,K))

   # logits: Nx(1+K)
   logits = cat([l_pos, l_neg], dim=1)

   # contrastive loss, Eqn.(1)
   labels = zeros(N) # positives are the 0-th
   loss = CrossEntropyLoss(logits/t, labels)

   # SGD update: query network
   loss.backward()
   update(f_q.params)

   # momentum update: key network
   f_k.params = m*f_k.params+(1-m)*f_q.params

   # update dictionary
   enqueue(queue, k) # enqueue the current minibatch
   dequeue(queue) # dequeue the earliest minibatch
```

bmm: batch matrix multiplication; mm: matrix multiplication; cat: concatenation.

Generate a positive pair by sampling data augmentation functions

No gradient through the key

Use the running queue of keys as the negative samples

InfoNCE loss

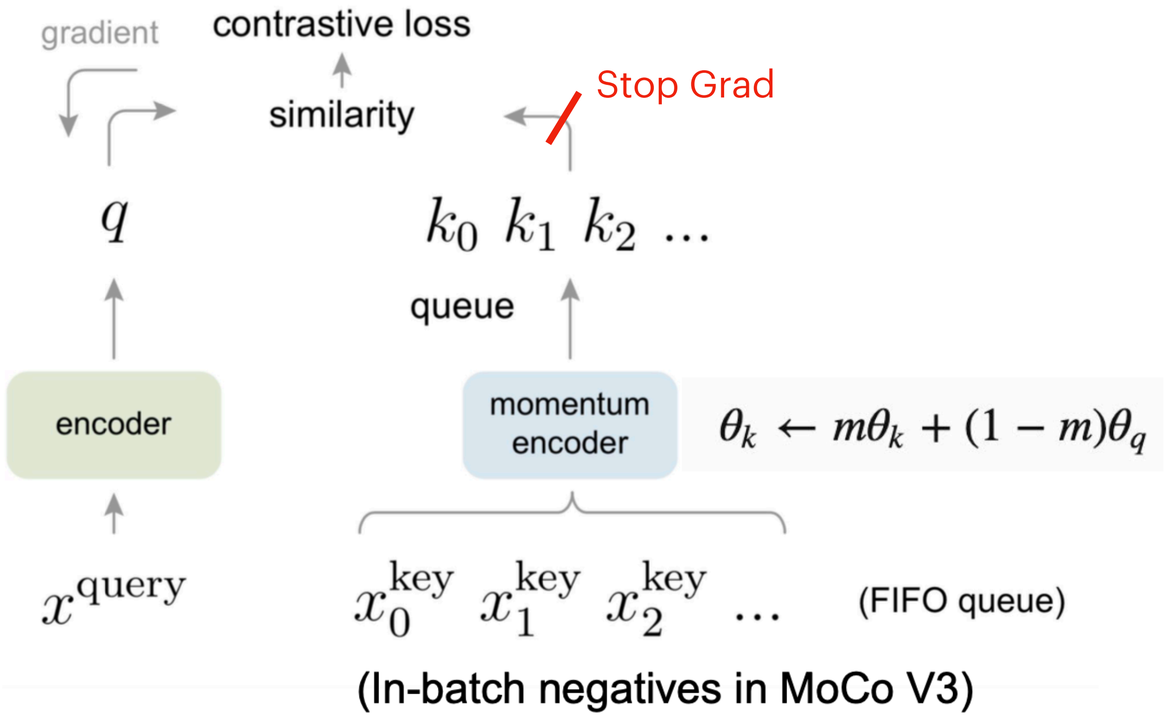
Update f_k through momentum

Update the FIFO negative sample queue

Source: cs231n 2023

Contrastive SSL
MoCo V2, V3

- Из SimCLR
  - Сложные аугментации
  - Нелинейный projector
- Из MoCo
  - Очередь негативных примеров
  - Momentum update для энкодера негативных примеров
- Vision Transformers
- In-batch negatives



Source: Chen et al., 2020

Contrastive SSL
Отличия от Metric Learning

Deep Metric Learning		Contrastive SSL
pos/neg пары получаются из лэйблов или из фиксированных трансформаций (напр. левая и правая часть изображения)	⇨	Позитивные пары получаются из рандомных аугментаций данных, негативные пары - это все не позитивные пары.
Hard negative mining	⇨	Random sampling
Encoder NN	⇨	Encoder NN + projector MLP
Small dataset (<200k)	⇨	Large dataset
Zero-shot kNN validation	⇨	Zero-shot kNN, zero-shot/few-shot/fine-tuning linear probing

Self-Distillation

Self-Distillation
SimSiam & BYOL

- Поиск сходства двух аугментированных изображений без учета негативных примеров
- Минимизация L2 расстояния между online и target сэмплами

- BYOL
  - Momentum encoder (EMA)

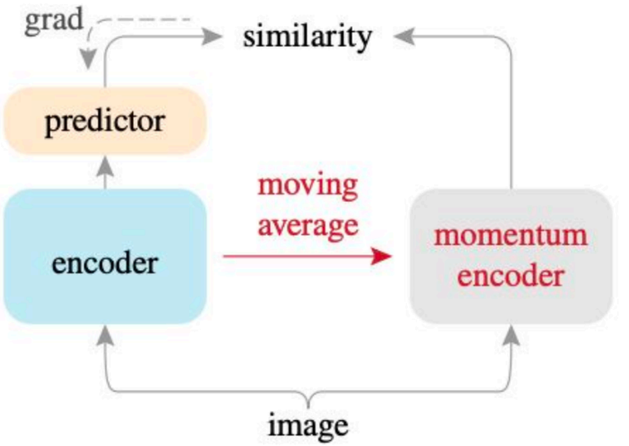
$$\mathcal{L}_{\mathrm{BYOL}}\left(\theta_{\mathrm{s}}, \gamma\right)=\mathbb{E}_{(\boldsymbol{x}, t_1, t_2) \sim\left(X, T_1, T_2\right)}\left[\left\|\operatorname{renorm}\left(p_\gamma\left(f_{\theta_{\mathrm{s}}}\left(t_1(\boldsymbol{x})\right)\right)\right)-\operatorname{renorm}\left(f_{\theta_{\mathrm{t}}}\left(t_2(\boldsymbol{x})\right)\right)\right\|_2^2\right]$$

Predictor · augumented samples · student NN · teacher NN

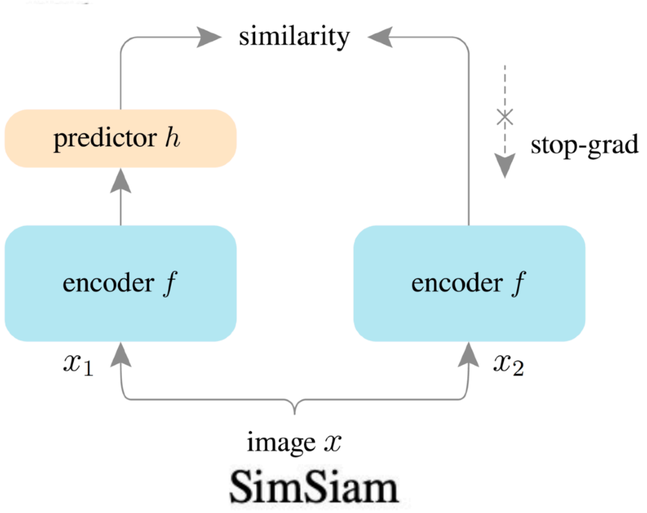
- SimSiam
  - No EMA
  - online encoder == target encoder

$$\mathcal{L}_{\mathrm{SimSIAM}}\left(\theta_{\mathrm{s}}, \gamma\right)=\mathbb{E}_{(\boldsymbol{x}, t_1, t_2)}\left[\left\|\operatorname{renorm}\left(p_\gamma\left(f_{\theta_{\mathrm{s}}}\left(t_1(\boldsymbol{x})\right)\right)\right)-\operatorname{sg}\left(\operatorname{renorm}\left(f_{\theta_{\mathrm{s}}}\left(t_2(\boldsymbol{x})\right)\right)\right)\right\|_2^2\right]$$

- *“BN is helpful for optimization when used appropriately”*



BYOL

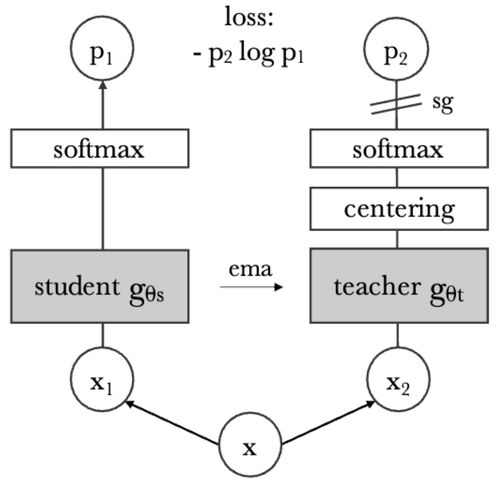


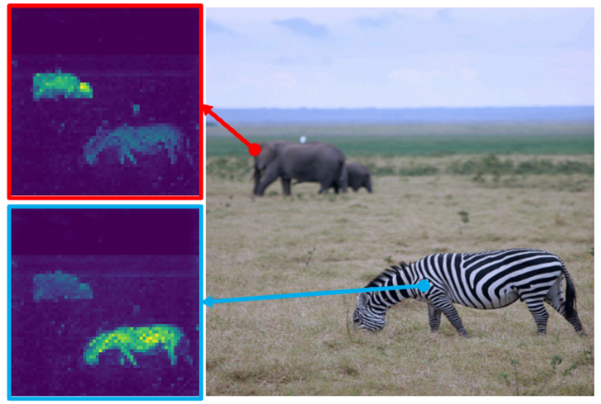
SimSiam

Sources: Grill et al., 2020, Chen and He, 2021

Self-Distillation
DINO

- Cross entropy loss
- Centring + softmax for teacher(target) NN
- Вернули EMA :)
- Vision Transformer





Source: Caron et al. [2021, DINO]

Masked Image Modeling

Masked Image Modeling

BEiT

Segmentation

Models	ADE20K
Supervised Pre-Training on ImageNet	45.3
DINO [CTM+21]	44.1
BEiT (ours)	45.6
BEiT + Intermediate Fine-Tuning (ours)	**47.7**

Table 3: Results of semantic segmentation on ADE20K. We use SETR-PUP [ZLZ+20] as the task layer and report results of single-scale inference.

Classification

Models	Model Size	Resolution	ImageNet
Training from scratch (i.e., random initialization)			
ViT$_{384}$-B [DBK+20]	86M	384^2	77.9
ViT$_{384}$-L [DBK+20]	307M	384^2	76.5
DeiT-B [TCD+20]	86M	224^2	81.8
DeiT$_{384}$-B [TCD+20]	86M	384^2	83.1
Supervised Pre-Training on ImageNet-22K (using labeled data)			
ViT$_{384}$-B [DBK+20]	86M	384^2	84.0
ViT$_{384}$-L [DBK+20]	307M	384^2	85.2
Self-Supervised Pre-Training on ImageNet-1K (without labeled data)			
iGPT-1.36B† [CRC+20]	1.36B	224^2	66.5
ViT$_{384}$-B-JFT300M‡ [DBK+20]	86M	384^2	79.9
MoCo v3-B [CXH21]	86M	224^2	83.2
MoCo v3-L [CXH21]	307M	224^2	84.1
DINO-B [CTM+21]	86M	224^2	82.8
BEiT-B (ours)	86M	224^2	83.2
BEiT$_{384}$-B (ours)	86M	384^2	84.6
BEiT-L (ours)	307M	224^2	85.2
BEiT$_{384}$-L (ours)	307M	384^2	**86.3**

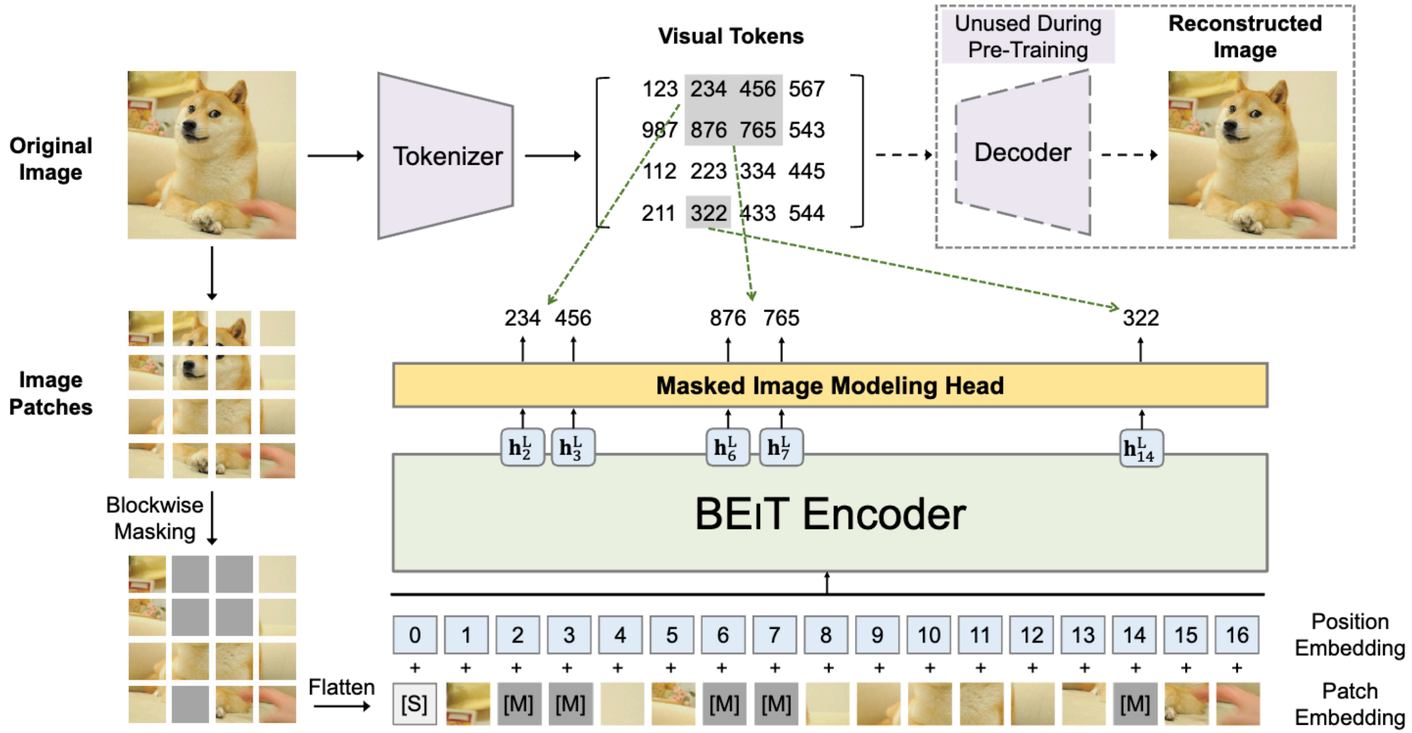


Figure 1: Overview of BEiT pre-training. Before pre-training, we learn an "image tokenizer" via autoencoding-style reconstruction, where an image is tokenized into discrete visual tokens according to the learned vocabulary. During pre-training, each image has two views, i.e., image patches, and visual tokens. We randomly mask some proportion of image patches (gray patches in the figure) and replace them with a special mask embedding [M]. Then the patches are fed to a backbone vision Transformer. The pre-training task aims at predicting the visual tokens of the *original* image based on the encoding vectors of the *corrupted* image.

Source: Hangbo Bao et. al.

Masked Image Modeling

MAE

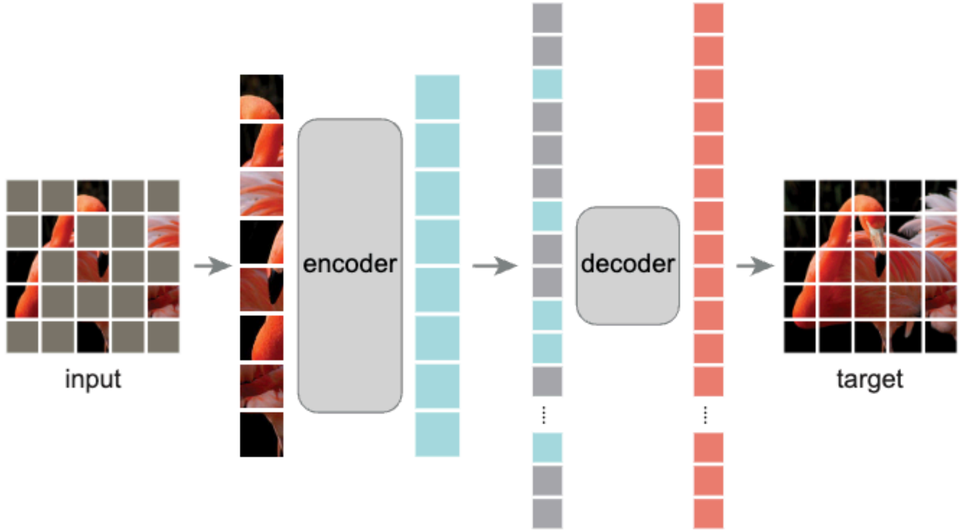


Figure 1. **Our MAE architecture**. During pre-training, a large random subset of image patches (*e.g.*, 75%) is masked out. The encoder is applied to the small subset of *visible patches*. Mask tokens are introduced *after* the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images (full sets of patches) for recognition tasks.

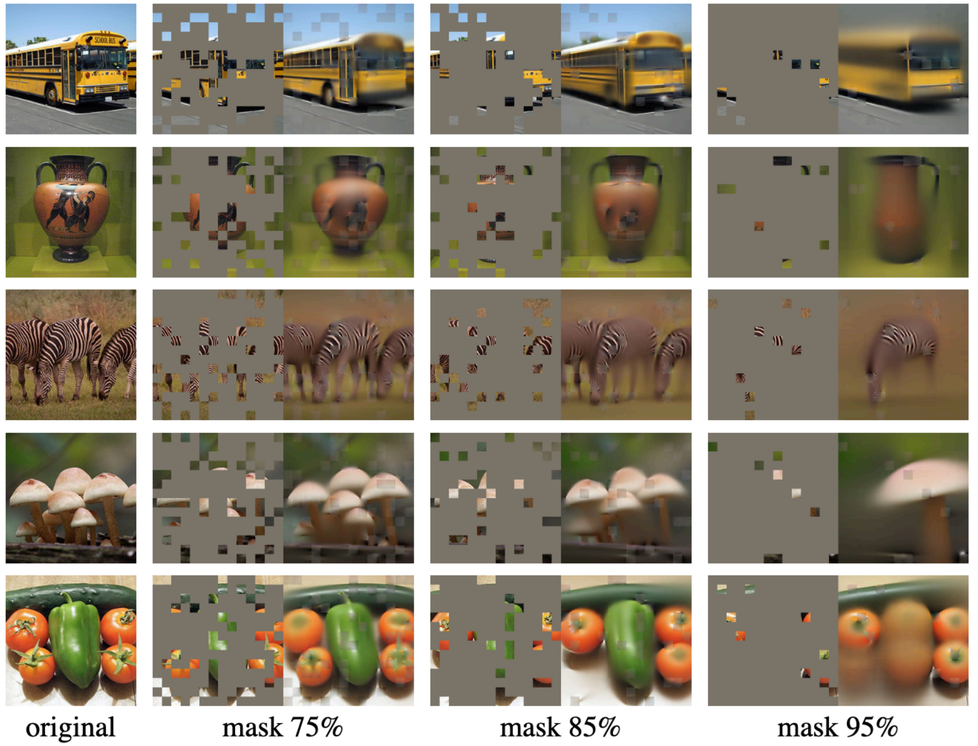


Figure 4. Reconstructions of ImageNet *validation* images using an MAE pre-trained with a masking ratio of 75% but applied on inputs with higher masking ratios. The predictions differ plausibly from the original images, showing that the method can generalize.

Source: Kaiming He et al.

Гибридные SSL модели

Гибридные SSL модели

iBOT, DINOv2 - merge self-distillation with MIM

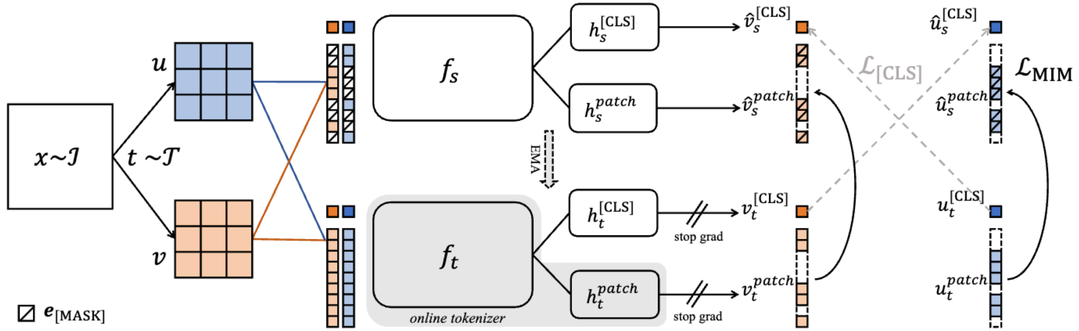


Figure 3: **Overview of iBOT framework, performing masked image modeling with an *online tokenizer*.** Given two views $\boldsymbol{u}$ and $\boldsymbol{v}$ of an image $\boldsymbol{x}$, each view is passed through a teacher network $h_t \circ f_t$ and a student network $h_s \circ f_s$. iBOT minimizes two losses. The first loss $\mathcal{L}_{\texttt{[CLS]}}$ is self-distillation between cross-view `[CLS]` tokens. The second loss $\mathcal{L}_{\text{MIM}}$ is self-distillation between in-view patch tokens, with some tokens masked and replaced by $\boldsymbol{e}_{\texttt{[MASK]}}$ for the student network. The objective is to reconstruct the masked tokens with the teacher networks' outputs as supervision.

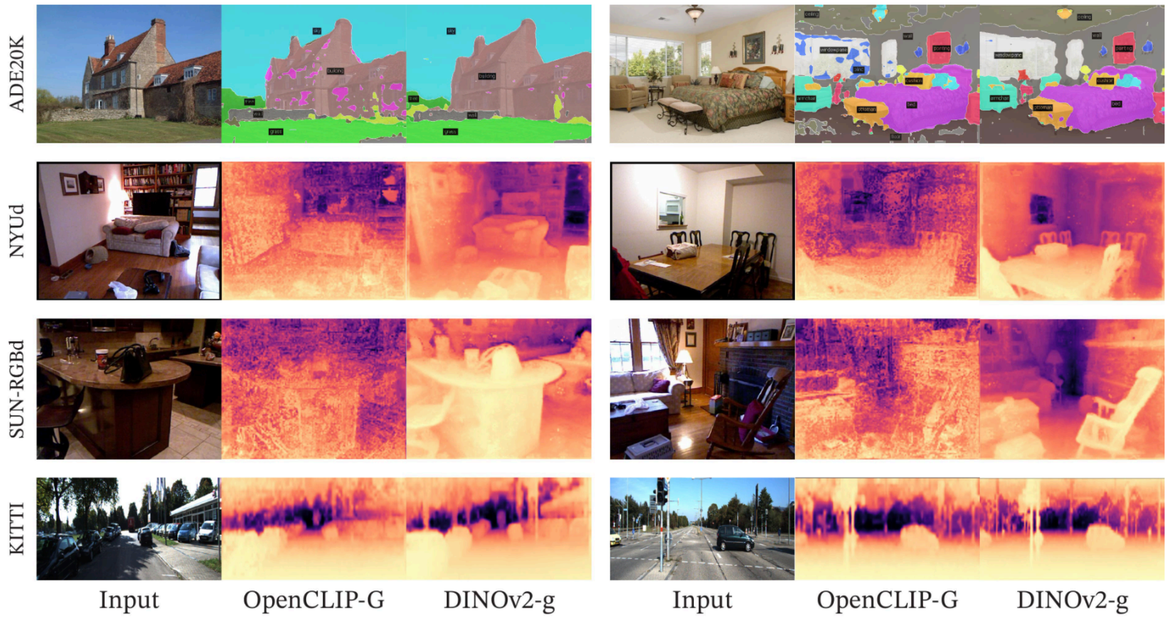


Figure 7: **Segmentation and depth estimation with linear classifiers.** Examples from ADE20K, NYUd, SUN RGB-D and KITTI with a linear probe on frozen OpenCLIP-G and DINOv2-g features.

Sources: DINOv2, iBOT

Гибридные SSL модели

DINOv2 - подготовка данных

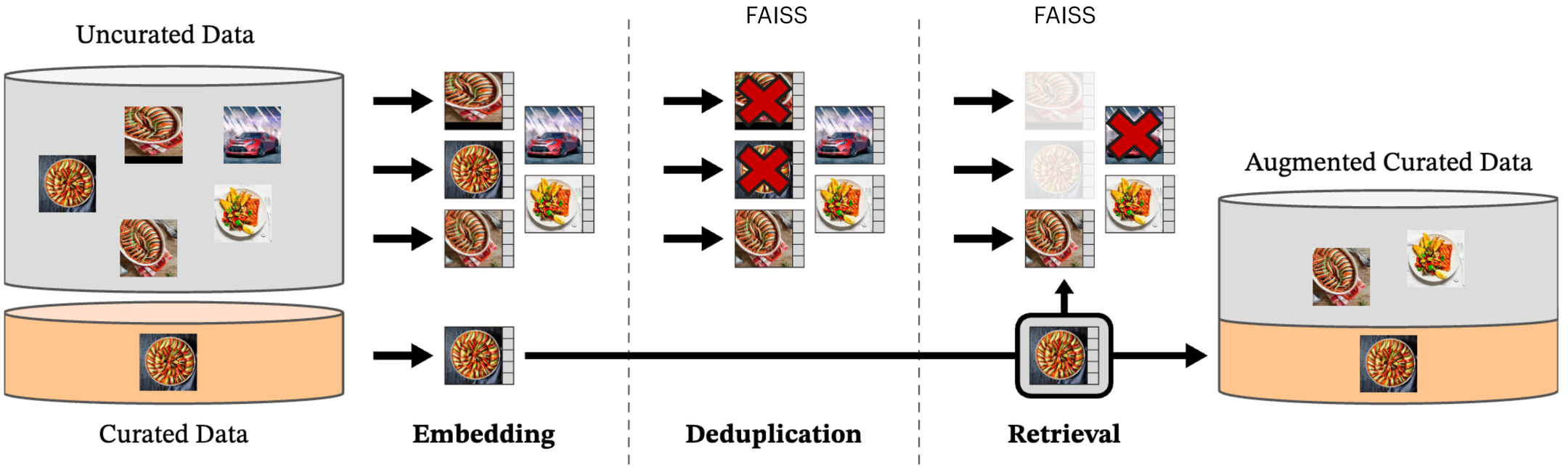


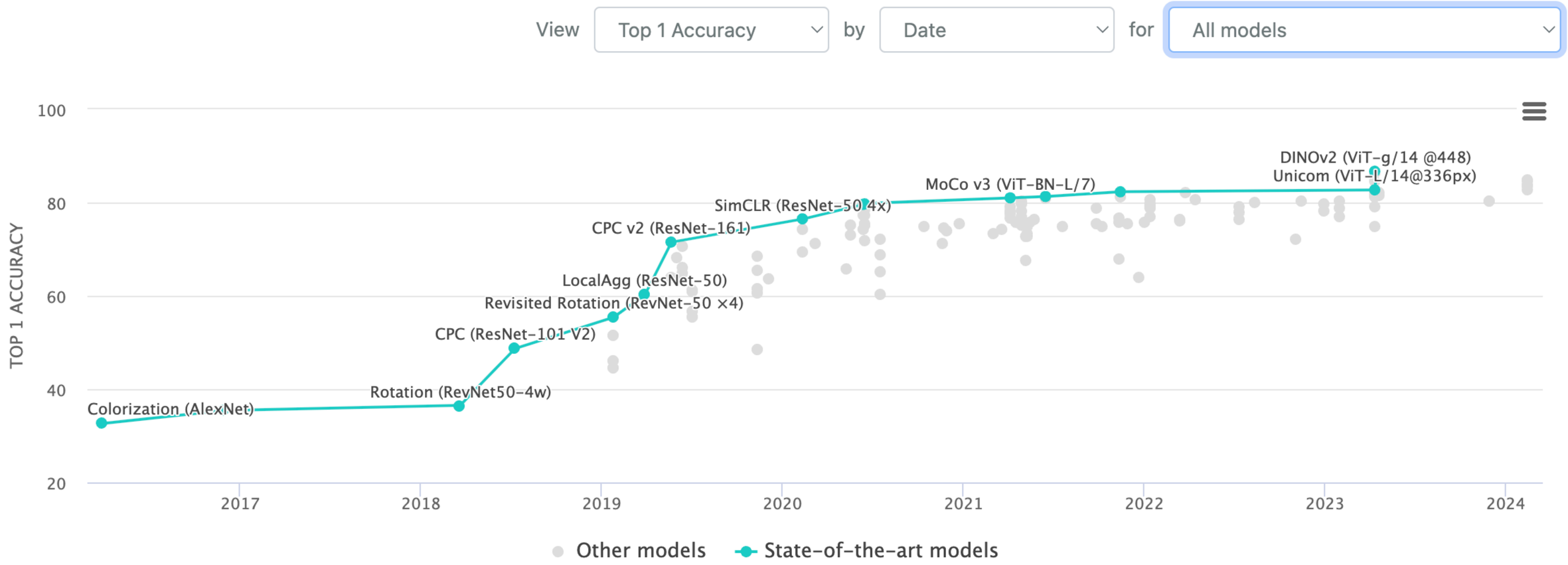
Figure 3: **Overview of our data processing pipeline.** Images from curated and uncurated data sources are first mapped to embeddings. Uncurated images are then deduplicated before being matched to curated images. The resulting combination augments the initial dataset through a self-supervised retrieval system.

Dataset: LVD-142M

Sources: DINOv2

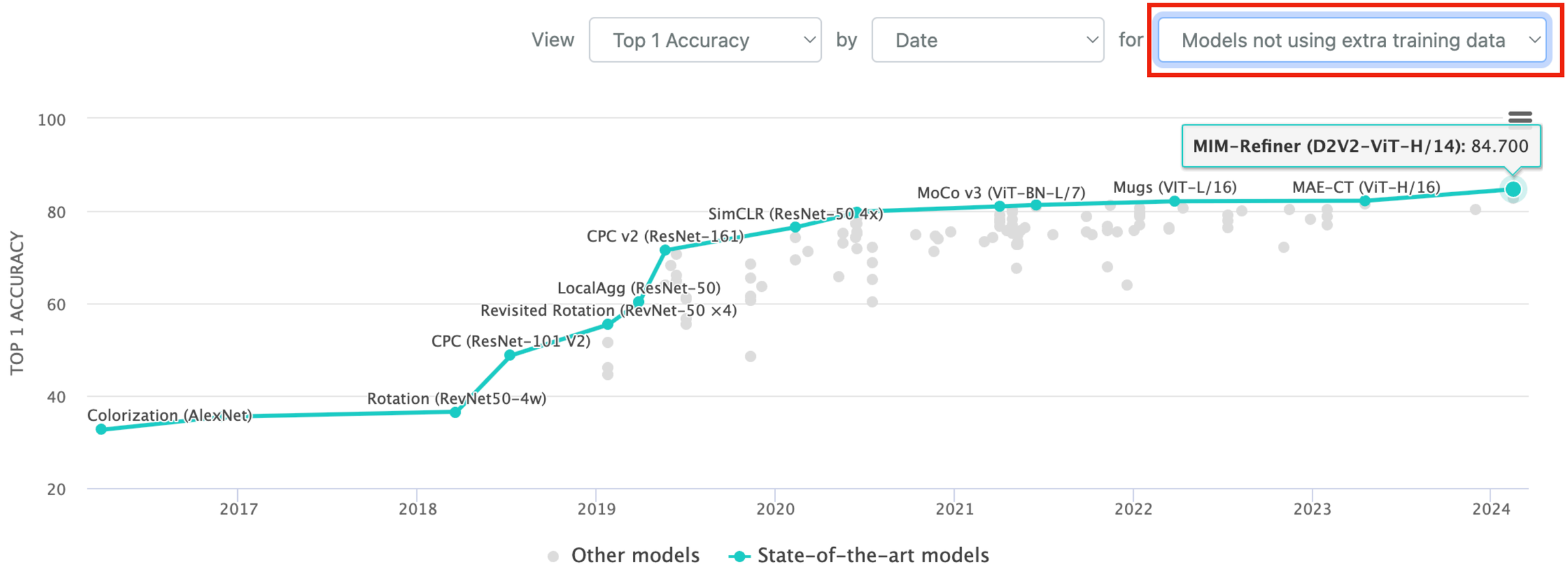
Валидация SSL моделей

Валидация
Бэнчмарки



https://paperswithcode.com/sota/self-supervised-image-classification-on

Валидация
Бэнчмарки



https://paperswithcode.com/sota/self-supervised-image-classification-on

Source: MIM-Refiner

Валидация

Как считать?

- На downstream задачах

- Zero-shot kNN
- Linear
- MLP
- Full fine-tune

Техники обучения

Техники обучения
Data Augmentation

- **Basic Image Augmentation**
  - *Random crop*
  - *Color distortion*
  - *Gaussian blur*
  - *Color jittering*
  - *Random flip/rotation*
  - *etc.*
- Augmentation Strategies
- Image Mixture

https://neurips.cc/virtual/2021/tutorial/21895

Техники обучения
Data Augmentation

- Basic Image Augmentation
- **Augmentation Strategies**
  - *AutoAugment (Cubuk, et al. 2018): inspired by NAS (Neural Architecture Search)*
  - *RandAugment (Cubuk et al. 2019): reduces NAS search space in AutoAugment.*
  - *PBA (Population based augmentation; Ho et al. 2019): evolutionary algorithm*
  - *UDA (Unsupervised Data Augmentation; Xie et al. 2019): minimize the KL*
  - *MultiCrop (SwAV, Mathilde Caron et al. 2020): small (96x96) & big (224x224) crops*
- Image Mixture

https://neurips.cc/virtual/2021/tutorial/21895

Техники обучения
Data Augmentation

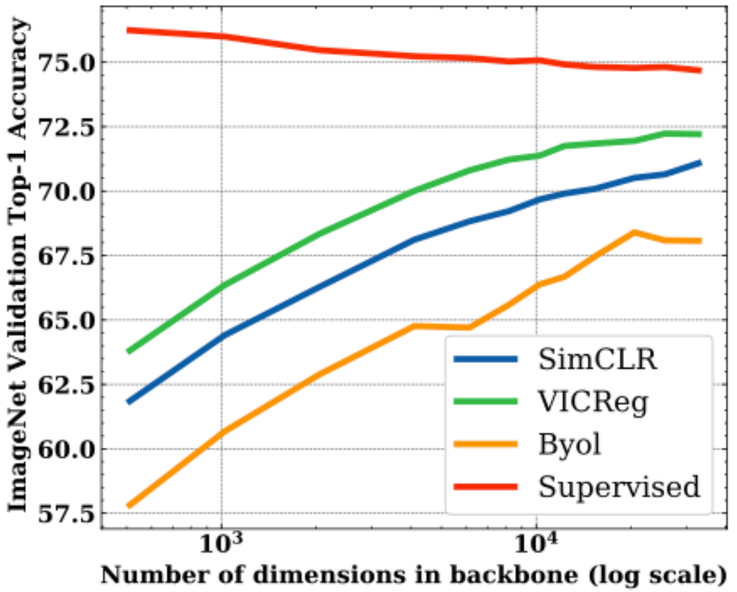
- Basic Image Augmentation
- Augmentation Strategies
- **Image Mixture**
  - *Mixup (Zhang et al 2018): weighted pixel-wise combination of two images.*
  - *Cutmix (Yun et al 2019): mix in a local region of one image into the other.*
  - *MoCHi (“Mixing of Contrastive Hard Negatives”; Kalantidis et al 2020): mixture of*

https://neurips.cc/virtual/2021/tutorial/21895

Техники обучения

Какая роль у Projector и Backbone?

- **Иногда** добавляет качества на downstream задачах, если использовать projector во время SSL
- Помогает модели справляться с **шумными аугментациями**, которые могли бы навредить финальному качеству модели
- Увеличение размера backbone позволяет **повышать финальное качество** (у supervised подходов ровно наоборот)



Техники обучения
Гиперпараметры

- Batch Size
- Learning Rate
- Optimizators
- Weight Decay
- ViT hyperparameters
  - Patch size
  - Stochastic depth - randomly drops blocks of the ViT as a regularization
  - Layer decay
  - Layer scale
  - [CLS] token

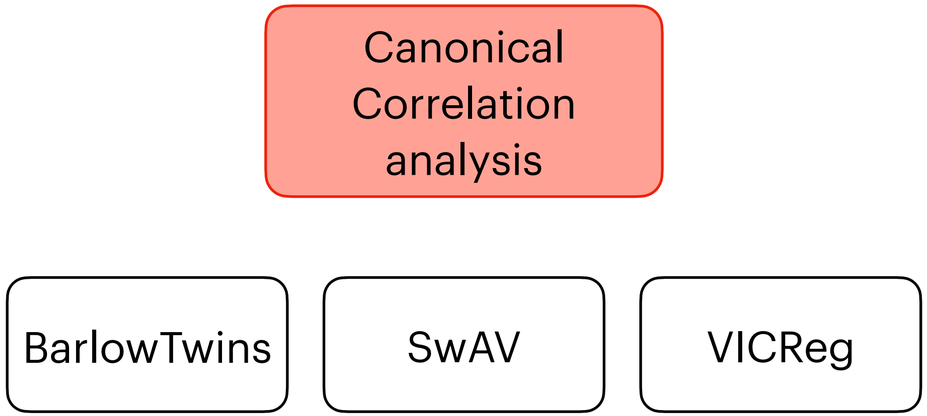
Техники обучения

Ускорение обучения

- Если есть несколько GPU
  - DDP: Distributed Data Parallel
  - FSDP: Fully Sharded Data Parallel
  - SyncBatchNorm - синхронизация батчнормов со всех GPU
  - Loss Aggregation
- Fast Augmentation
  - DALI
  - FFCV
  - FFCV-SSL
- ViT
  - Flash Attention
  - X Formers
  - Float16 —> bfloat16

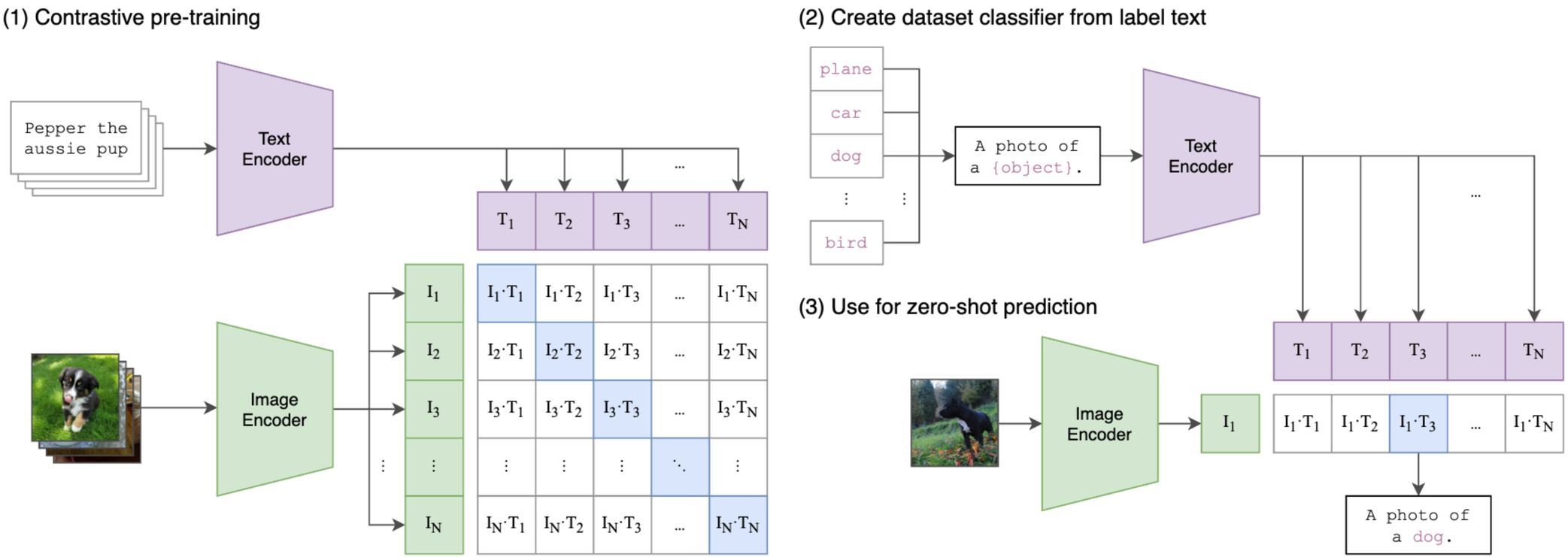
Другие подходы к SSL

CCA



Другие подходы к SSL
Multimodal

- CLIP
- BLIP
- CoCa



References

- SSL Cookbook - https://arxiv.org/abs/2304.12210
- NeurIPS OpenAI tutorial on SSL - https://neurips.cc/virtual/2021/tutorial/21895
- Stanford CS231n - http://cs231n.stanford.edu/slides/2023/lecture_13.pdf