

# Self-supervised learning

Виданов Андрей Николаевич  
МФТИ

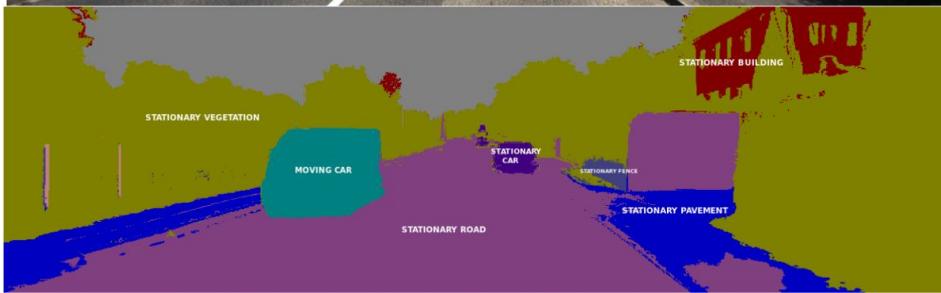
Курс:ИИН практика/Группа Б05-021  
Эксперт: Полина Барабанщикова

$x_1$

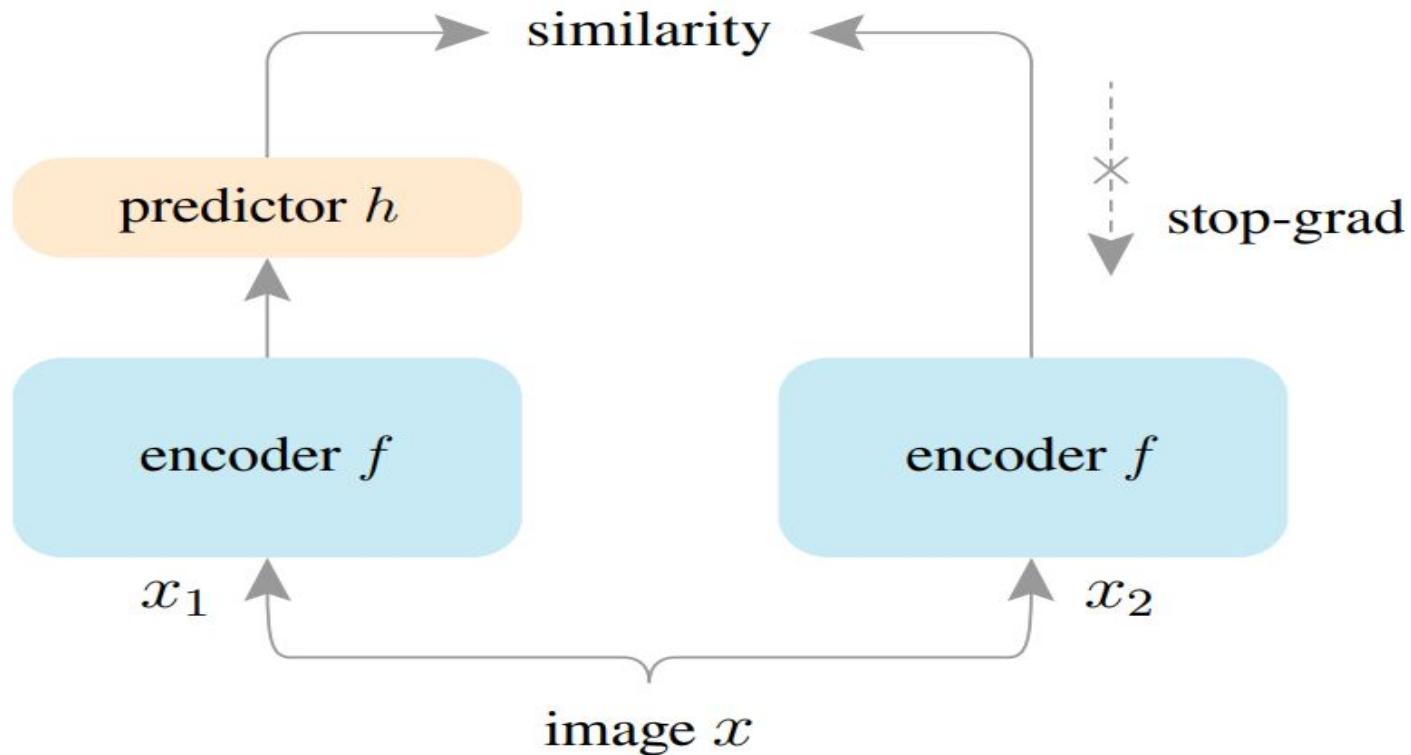
$x_2$

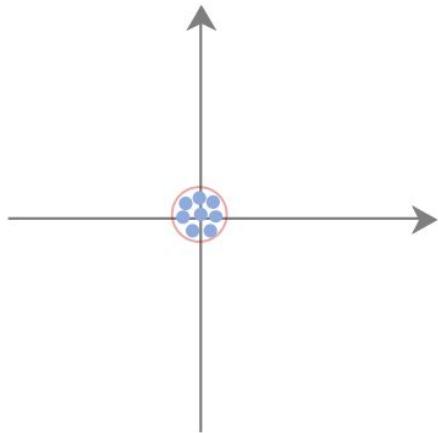
image  $x$

Целью обучения является генерация векторных представлений, применимых в задачах классификации, детекции и сегментации

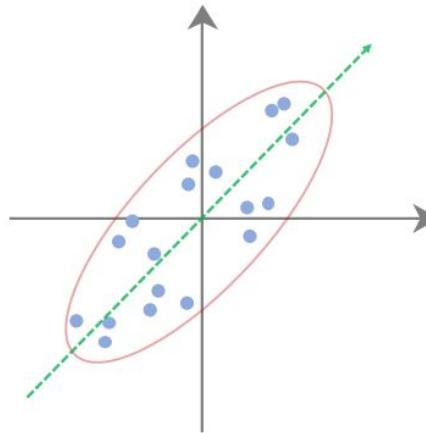


## Сиамские сети

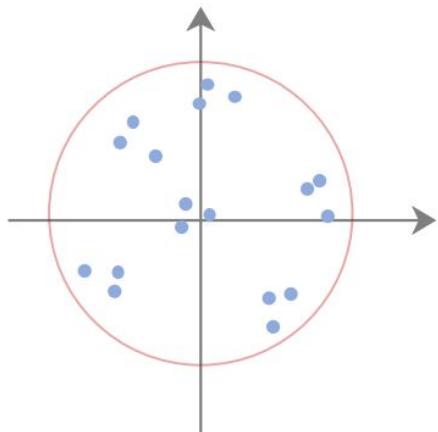




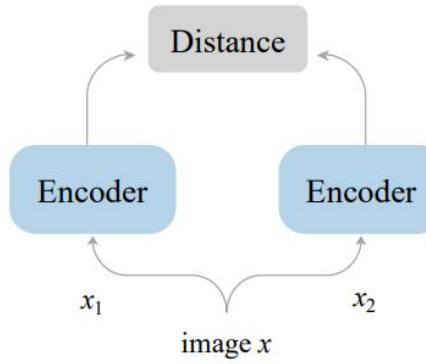
(a) complete collapse



(b) dimensional collapse



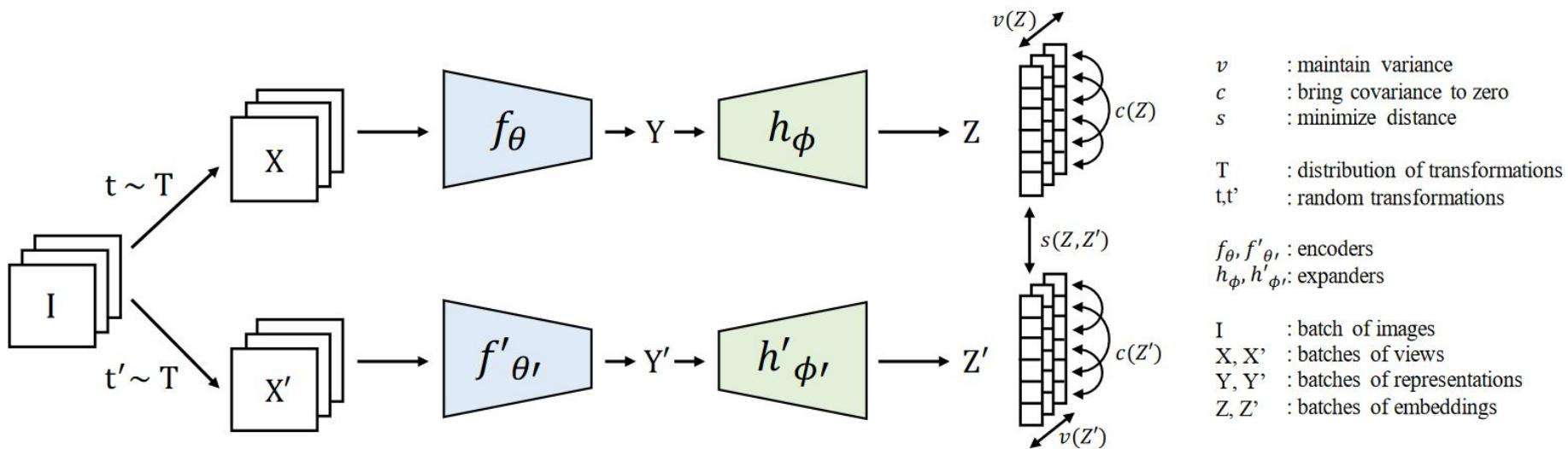
(c) decorrelated



(d) the concise framework

# VICReg

$$L(Z, Z') = \lambda s(Z, Z') + \mu[v(Z, Z')] + \nu[c(Z, Z')],$$



## Инвариантность

$$s(Z, Z') = \frac{1}{n} \sum_i \|z_i - z'_i\|_2^2.$$

Уменьшает расстояние  
между  
аугментированными  
копиями изображений

## Дисперсия

$$v(Z) = \frac{1}{d} \sum_{j=1}^d \max(0, \gamma - S(z^j, \epsilon)),$$

$$S(x, \epsilon) = \sqrt{\text{Var}(x) + \epsilon},$$

Предотвращает  
коллапс

## Ковариация

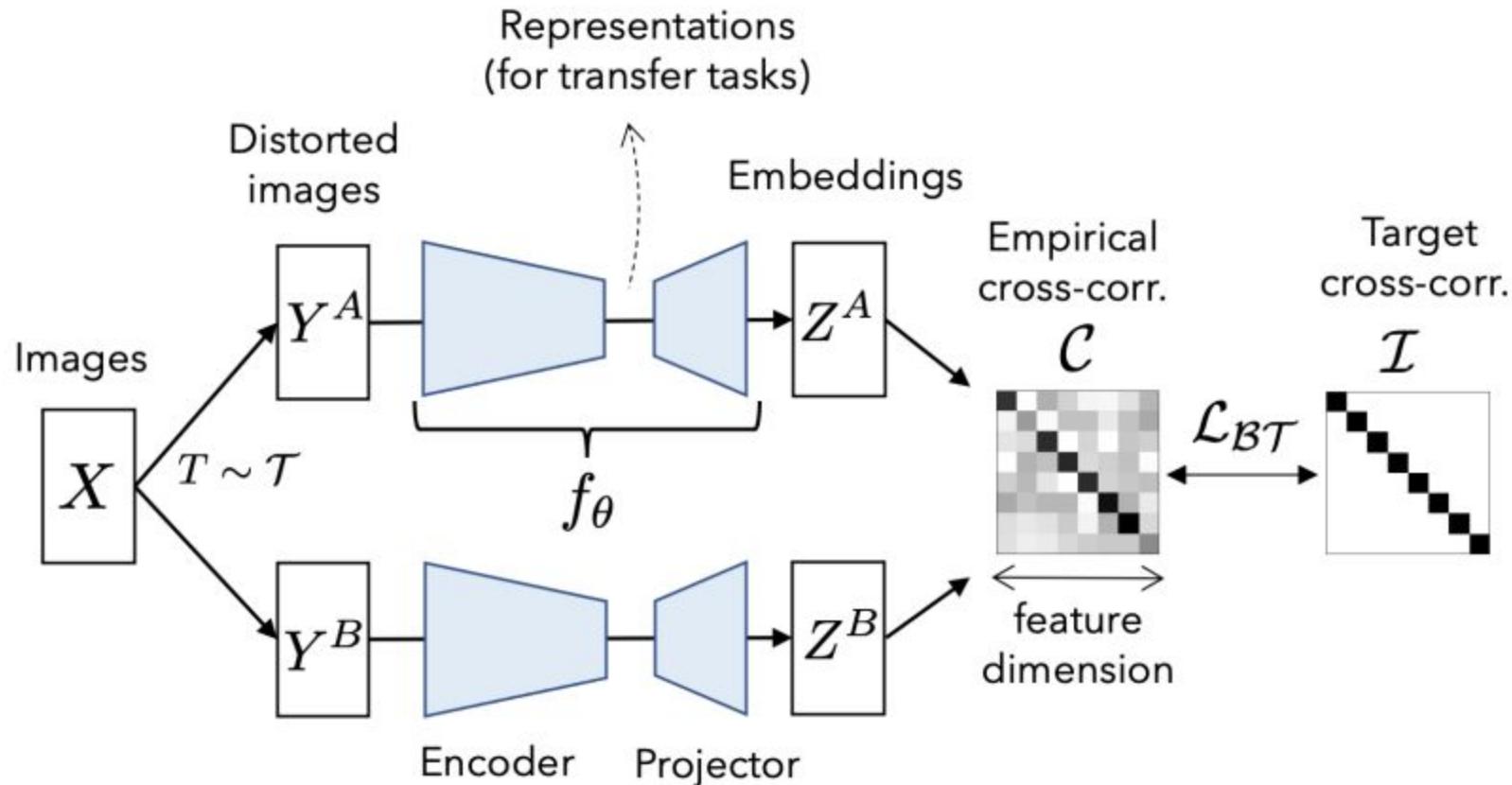
$$c(Z) = \frac{1}{d} \sum_{i \neq j} [C(Z)]_{i,j}^2.$$

$$C(Z) = \frac{1}{n-1} \sum_{i=1}^n (z_i - \bar{z})(z_i - \bar{z})^T$$

where  $\bar{z} = \frac{1}{n} \sum_{i=1}^n z_i.$

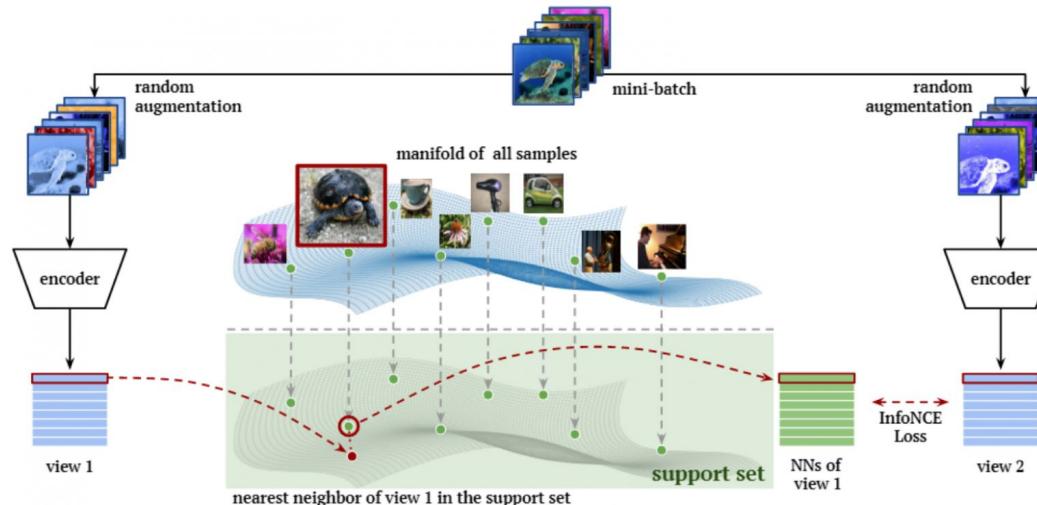
Предотвращает  
переобучение на шум

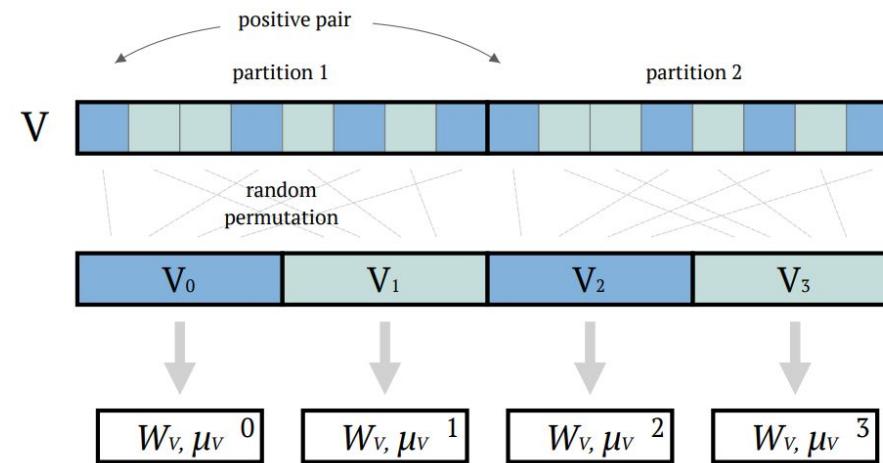
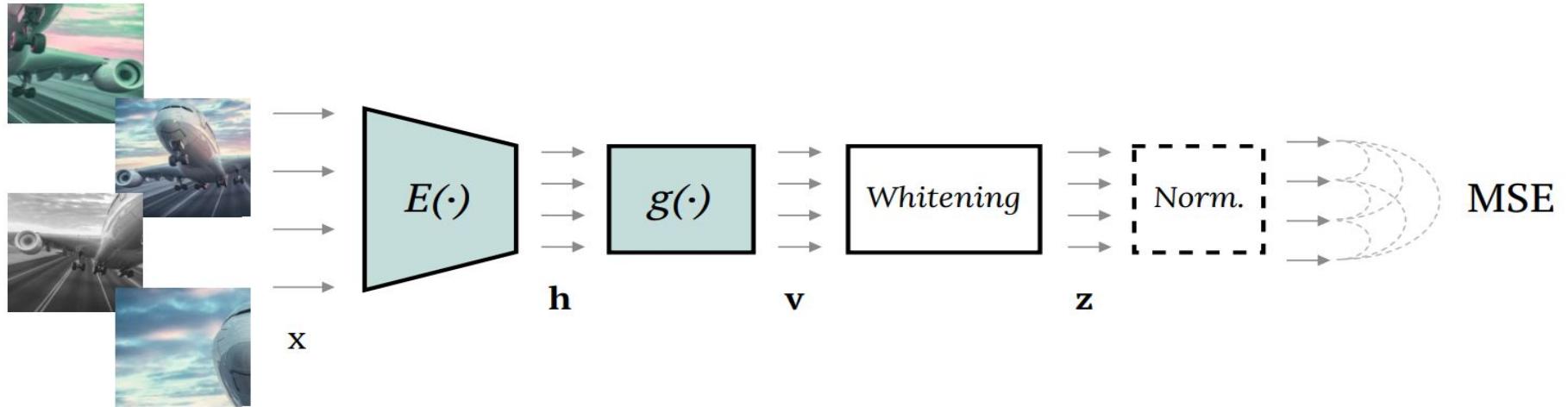
# Barlow Twins



# Методы повышения качества представлений

1. Изменение функции потерь
2. Whitening(отбеливание)





## Проводимые эксперименты

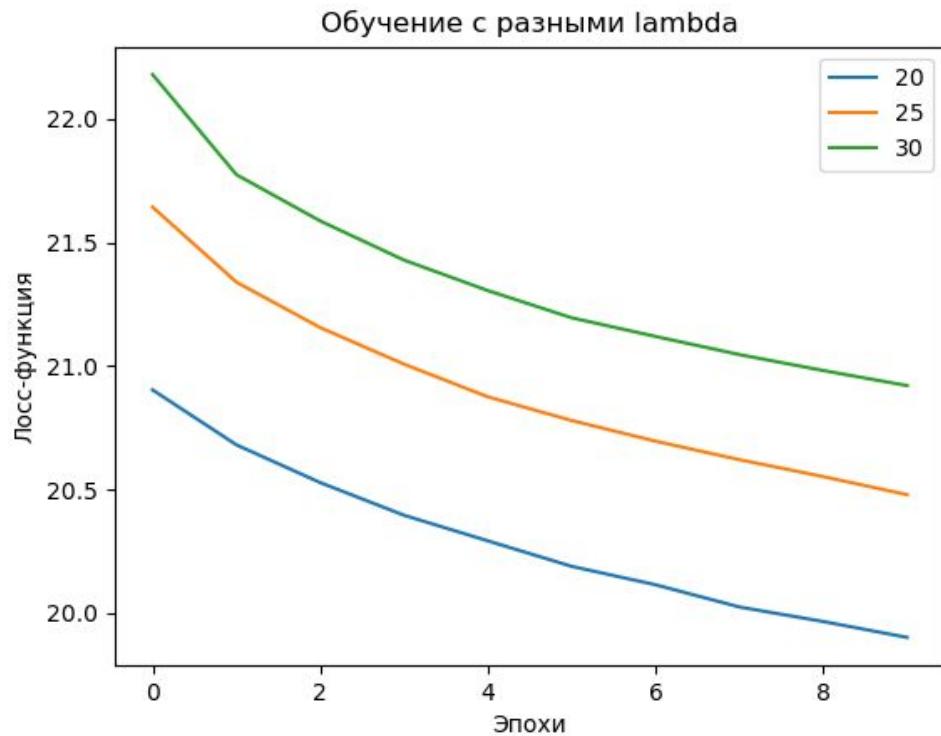
Пока лучших результатов удалось добиться с параметрами:

$$\lambda = 20$$

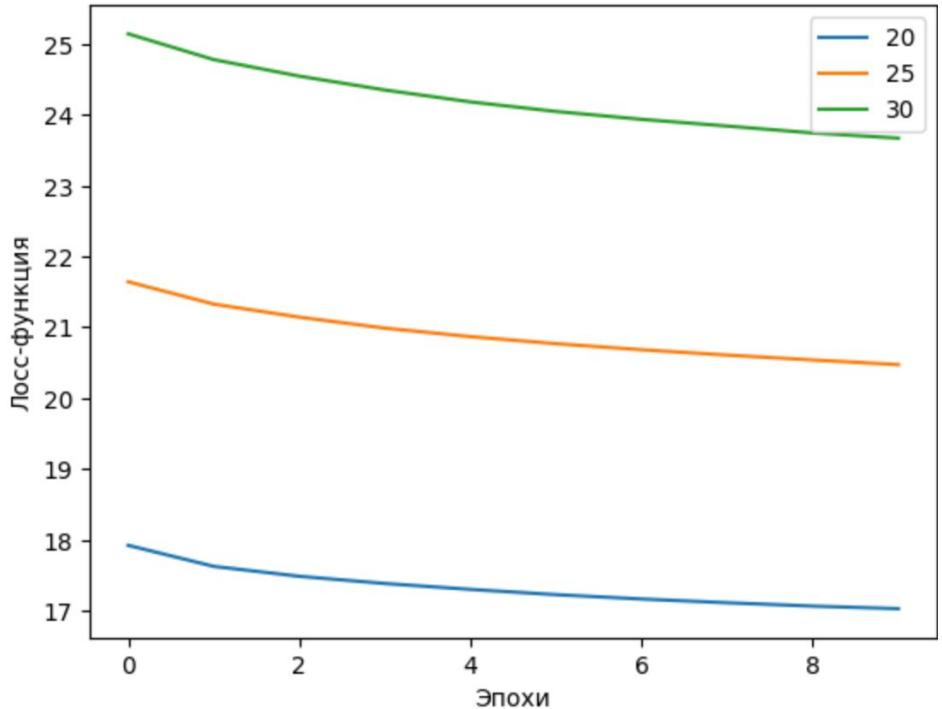
$$\mu = 25$$

$$\nu = 1$$

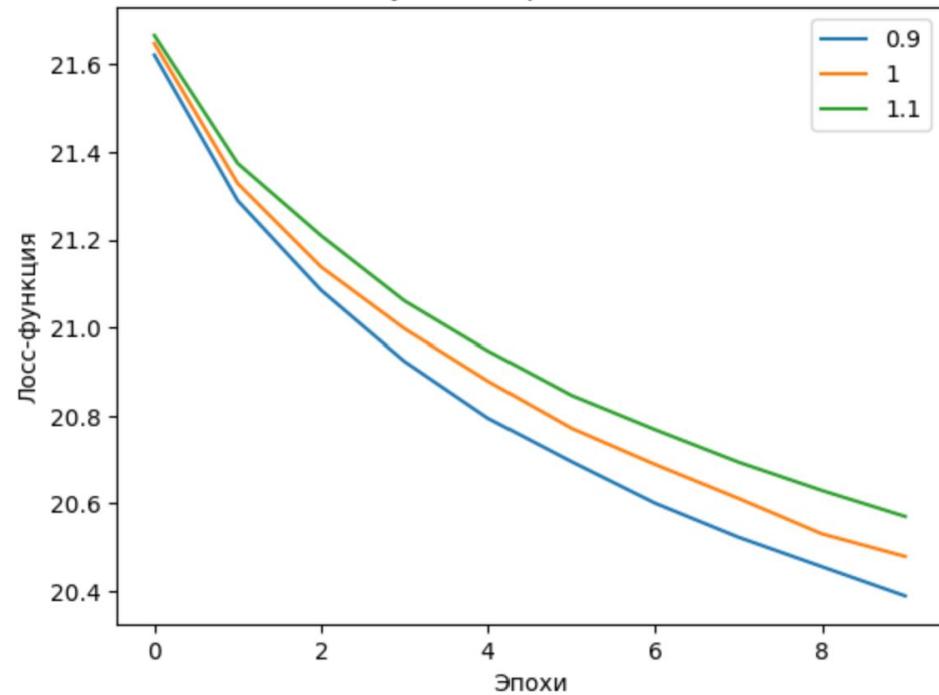
$$L(Z, Z') = \lambda s(Z, Z') + \mu[v(Z, Z')] + \nu[c(Z, Z')],$$



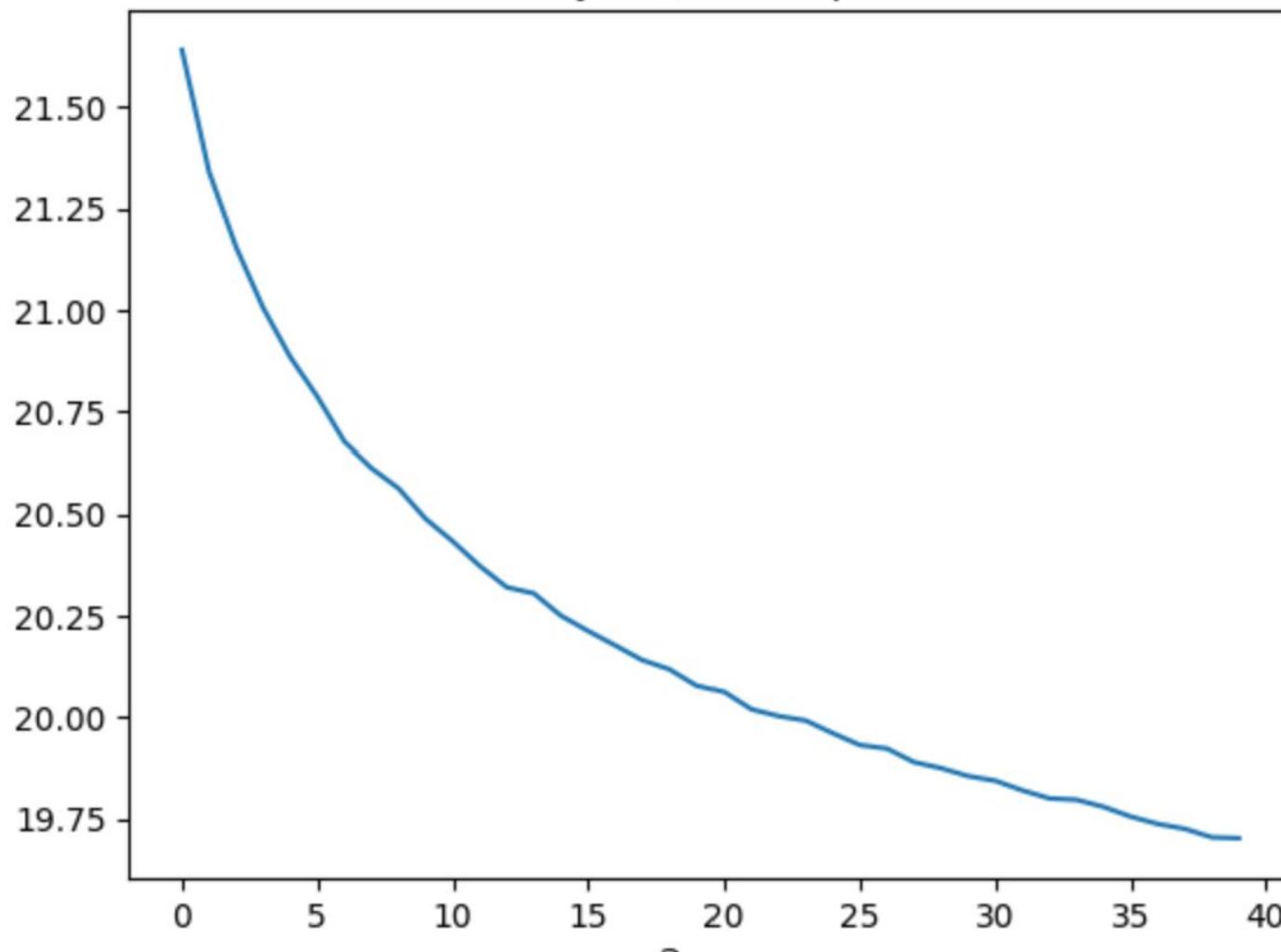
Обучение с разными  $\eta$



Обучение с разными  $n$



## ФУНКЦИЯ ПОТЕРЬ

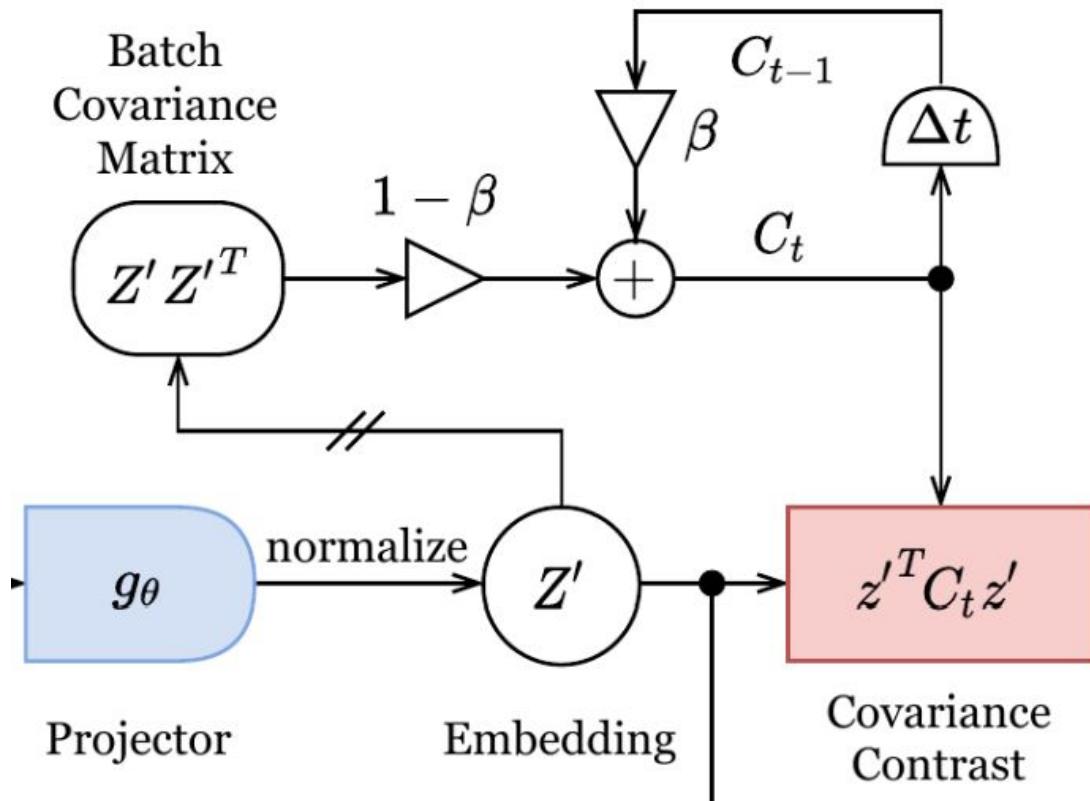


ViCReg ковариационная компонента

$$C(Z) = \frac{1}{n-1} \sum_1^n (z_i - \bar{z})(z_i - \bar{z})^T$$

$$c(Z) = \frac{1}{d} \sum_{i \neq j} [C(Z)]_{i,j}^2$$

TiCo ковариационная компонента



# Модификация функции потерь

```
N = z_a.size(0)
D = z_a.size(1)

# variance term of the loss
std_x = torch.sqrt(z_a.var(dim=0) + self.eps)
std_y = torch.sqrt(z_b.var(dim=0) + self.eps)
std_loss = 0.5 * (torch.mean(F.relu(1 - std_x)) + torch.mean(F.relu(1 - std_y)))

# covariance term of the loss
cov_x = (z_a.T @ z_a) / (N - 1)
cov_y = (z_b.T @ z_b) / (N - 1)

# compute off-diagonal elements
n, _ = cov_x.shape
off_diag_cov_x = cov_x.flatten()[:-1].view(n - 1, n + 1)[:, 1:].flatten()
off_diag_cov_y = cov_y.flatten()[:-1].view(n - 1, n + 1)[:, 1:].flatten()

cov_loss = off_diag_cov_x.pow_(2).sum().div(D) + off_diag_cov_y.pow_(
    2
).sum().div(D)

# loss
loss = (
    self.lambda_param * repr_loss
    + self.mu_param * std_loss
    + self.rho * (torch.mm(z_a, C) * z_a_tic).sum(dim=1).mean()
)
```

# Результаты

Итоговая точность предсказаний(Accuracy) на задаче классификации		
TiCo	Custom	VicReg
0.304	<b>0.377</b>	0.421

# Литература



Самообучение. Проблематика и постановка задачи



Задача классификации по представлениям

Модель VicReg



Модели сиамских сетей



Набор данных



CIFAR10

