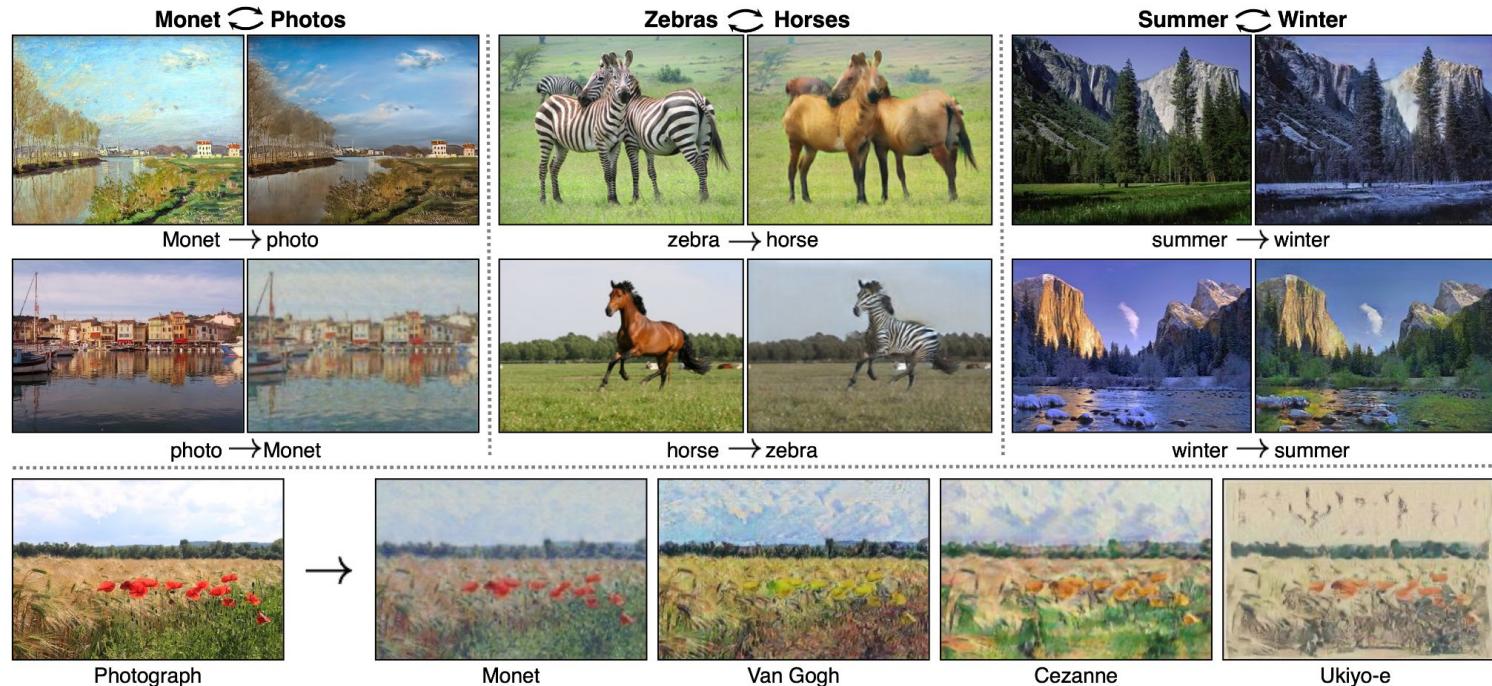


# **Contrastive Learning for Unpaired Image-to-Image Translation**

Павлов Вадим, Молодык Петр, Першин Максим, Сапожникова Дарья

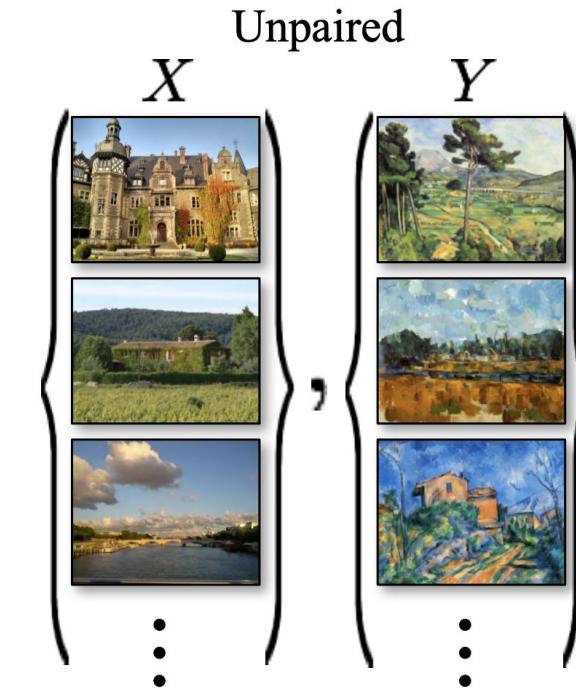
# Постановка задачи

Хотим научиться переводить картинку из одного домена в другой



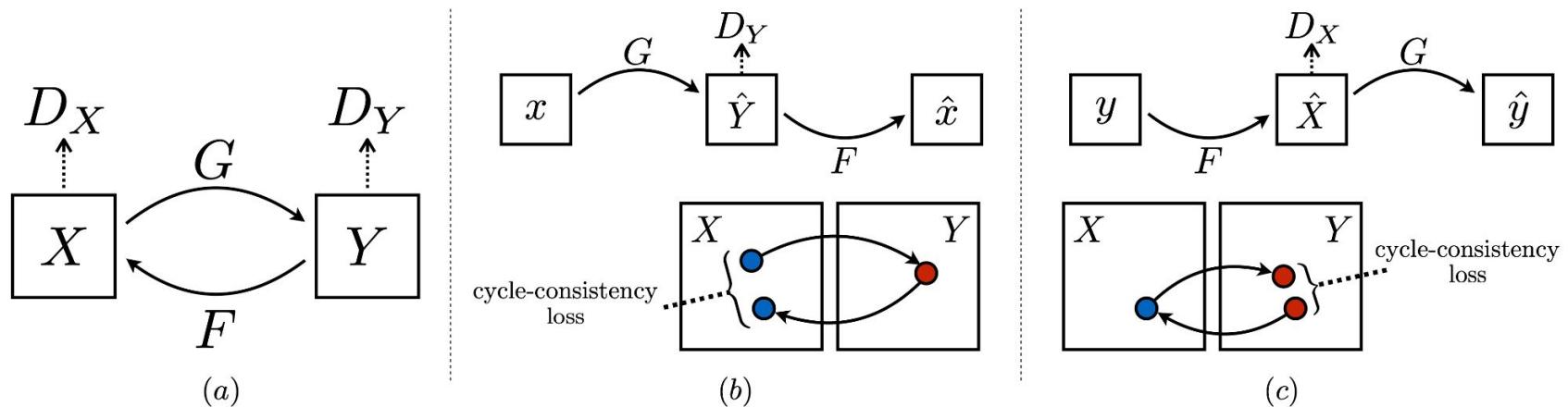
# Постановка задачи

Есть два типа подобных задач



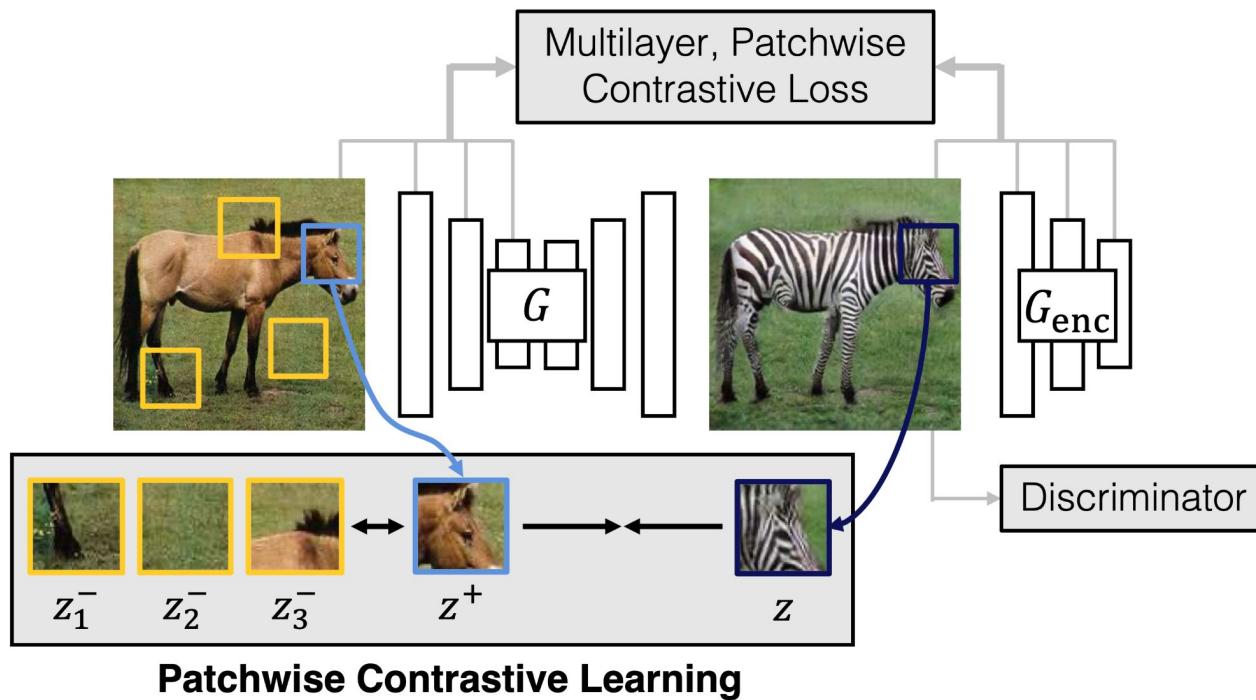
# Методы решения

Cycle GAN: учим дополнительный генератор и используем cycle-consistency loss



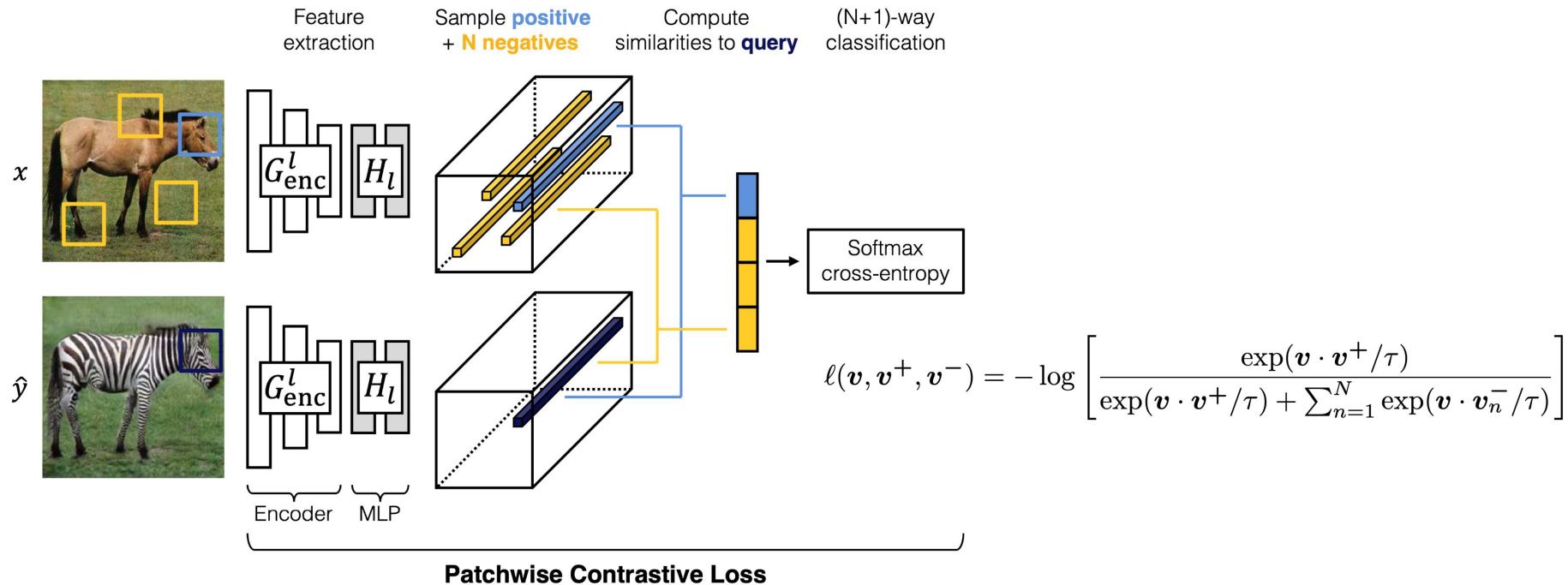
# Предлагаемый подход

Идея: patchwise contrastive loss



# Предлагаемый подход

Идея: patchwise contrastive loss на разных уровнях энкодера



# Предлагаемый подход

Итоговая функция потерь: важно сначала прогнать через  $H_l$

$$\{\mathbf{z}_l\}_L = \{H_l(G_{\text{enc}}^l(\mathbf{x}))\}_L \quad l \text{ — номер слоя}$$

$$\{\hat{\mathbf{z}}_l\}_L = \{H_l(G_{\text{enc}}^l(G(\mathbf{x})))\}_L$$

$$\mathcal{L}_{\text{PatchNCE}}(G, H, X) = \mathbb{E}_{\mathbf{x} \sim X} \sum_{l=1}^L \sum_{s=1}^{S_l} \ell(\hat{\mathbf{z}}_l^s, \mathbf{z}_l^s, \mathbf{z}_l^{S \setminus s}) \quad \begin{matrix} \text{берем каждый четвертый} \\ \text{слой энкодера} \end{matrix}$$

$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) = \mathbb{E}_{\mathbf{y} \sim Y} \log D(\mathbf{y}) + \mathbb{E}_{\mathbf{x} \sim X} \log(1 - D(G(\mathbf{x})))$$

Итоговый функционал:

$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) + \lambda_X \mathcal{L}_{\text{PatchNCE}}(G, H, X) + \lambda_Y \mathcal{L}_{\text{PatchNCE}}(G, H, Y)$$

# Предлагаемый подход

Другая возможная опция: негативные примеры из других картинок

$$\{\mathbf{z}_l\}_L = \{H_l(G_{\text{enc}}^l(\mathbf{x}))\}_L \quad l \text{ — номер слоя}$$

$$\{\hat{\mathbf{z}}_l\}_L = \{H_l(G_{\text{enc}}^l(G(\mathbf{x})))\}_L$$

$$\mathcal{L}_{\text{external}}(G, H, X) = \mathbb{E}_{\mathbf{x} \sim X, \tilde{\mathbf{z}} \sim Z^-} \sum_{l=1}^L \sum_{s=1}^{S_l} \ell(\hat{\mathbf{z}}_l^s, \mathbf{z}_l^s, \tilde{\mathbf{z}}_l) \text{ , слои - 5 штук равномерно}$$

$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) = \mathbb{E}_{\mathbf{y} \sim Y} \log D(\mathbf{y}) + \mathbb{E}_{\mathbf{x} \sim X} \log(1 - D(G(\mathbf{x})))$$

Итоговый функционал:

$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) + \lambda_X \mathcal{L}_{\text{external}}(G, H, X) + \lambda_Y \mathcal{L}_{\text{external}}(G, H, Y)$$

# Предлагаемый подход

Итоговая модель:

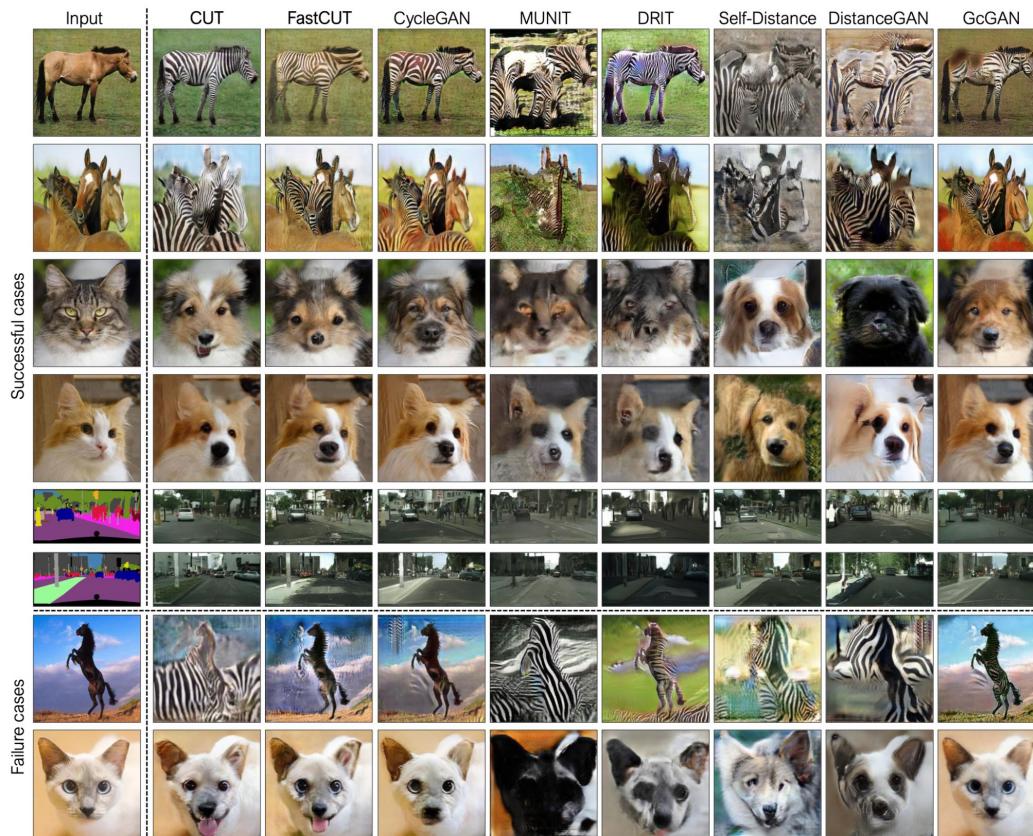
$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) + \lambda_X \mathcal{L}_{\text{PatchNCE}}(G, H, X) + \lambda_Y \mathcal{L}_{\text{PatchNCE}}(G, H, Y)$$

$\lambda_X = 1, \lambda_Y = 1$  — Contrastive Unpaired Translation (CUT)

$\lambda_X = 10, \lambda_Y = 0$  — FastCUT

FastCUT работает быстрее и требует меньше памяти, но в среднем хуже качество

# Эксперименты



# Эксперименты

Для Cityscapes дополнительно использовали DRN для семантической сегментации результатов и смотрели качество на созданных картинках

Method	Cityscapes				Cat→Dog		Horse→Zebra		
	mAP↑	pixAcc↑	classAcc↑	FID↓	FID↓	FID↓	sec/iter↓	Mem(GB)↓	
CycleGAN [89]	20.4	55.9	25.4	76.3	85.9	77.2	0.40	4.81	
MUNIT [44]	16.9	56.5	22.5	91.4	104.4	133.8	0.39	3.84	
DRIT [41]	17.0	58.7	22.2	155.3	123.4	140.0	0.70	4.85	
Distance [4]	8.4	42.2	12.6	81.8	155.3	72.0	<b>0.15</b>	2.72	
SelfDistance [4]	15.3	56.9	20.6	78.8	144.4	80.8	0.16	2.72	
GCGAN [18]	21.2	63.2	26.6	105.2	96.6	86.7	0.26	2.67	
CUT	<b>24.7</b>	<b>68.8</b>	<b>30.7</b>	<b>56.4</b>	<b>76.2</b>	<b>45.5</b>	0.24	3.33	
FastCUT	19.1	59.9	24.3	68.8	94.0	73.4	<b>0.15</b>	<b>2.25</b>	

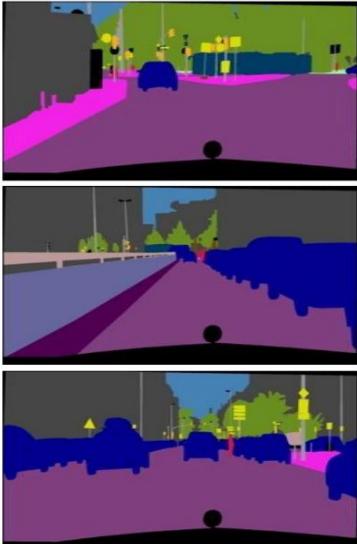
# Ablation study

- Слагаемое  $\lambda_Y \mathcal{L}_{\text{PatchNCE}}(G, H, Y)$  в функции потерь стабилизирует обучение (Id)
- Лучше использовать негативные примеры из других патчей той же самой картинки, чем из других (Int > Ext)
- Важно использовать несколько слоев энкодера для функции потерь, в отличие от обычного contrastive learning (All > Last)

Method	Training settings					Testing datasets		
	Id Negs		Layers Int Ext			Horse → Zebra		Cityscapes
	FID↓	FID↓	mAP↑					
CUT (default)	✓	255	All	✓	✗	45.5	<b>56.4</b>	<b>24.7</b>
no id	✗	255	All	✓	✗	39.3	68.5	22.0
no id, 15 neg	✗	15	All	✓	✗	44.1	59.7	23.1
no id, 15 neg, last	✗	15	Last	✓	✗	<b>38.1</b>	114.1	16.0
last	✓	255	Last	✓	✗	441.7	141.1	14.9
int and ext	✓	255	All	✓	✓	56.4	64.4	20.0
ext only	✓	255	All	✗	✓	53.0	110.3	16.5
ext only, last	✓	255	Last	✗	✓	60.1	389.1	5.6

# Ablation study

Вход



CUT



$\lambda_Y = 0$



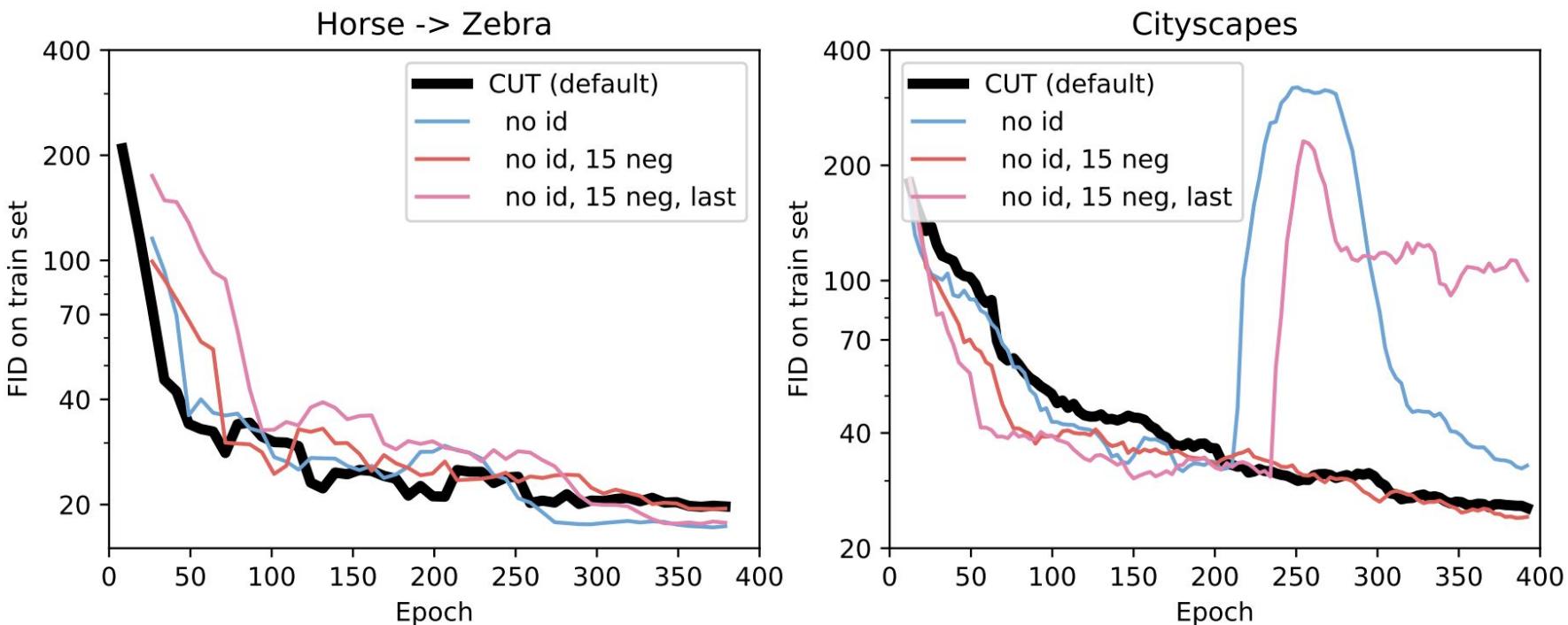
Один слой энкодера  
в функции потерь



$\mathcal{L}_{\text{external}}(G, H, X)$

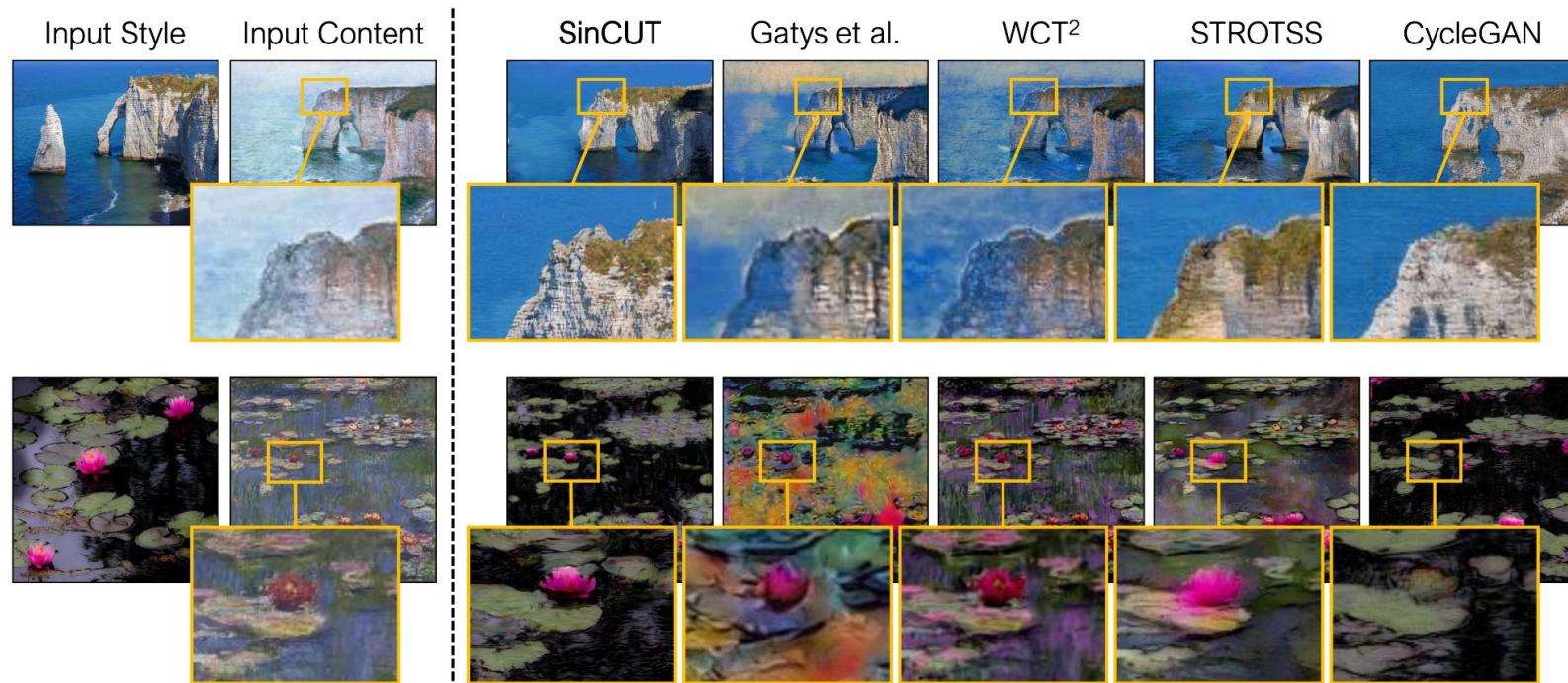


# Ablation study



# SinCUT

- Задача рисунок -> фотография, при этом в обоих доменах по одной картинке в HD разрешении
- Не влезает на GPU, учат патчами 128×128
- Ограничивают область видимости дискриминатора + gradient penalty



# Итог

- Получили SOTA подход к решению задачи Image-to-Image Translation
- Есть быстрая модификация, которая дает чуть хуже качество

# Рецензия

# Содержание и вклад

## Результаты:

- Авторы создали модель для перевода изображений, которая не использует циклическую структуру и не требует негативных семплов из других изображений
- Модель может обучаться даже по 1 изображению
- CUT бьет предыдущие модели по качеству в экспериментах
- Есть версия FastCut, которая при адекватном качестве показывает самое низкое потребление памяти и времени

## Значимость:

- Перевод изображений - очень актуальная тема, используется, например, для переноса стиля
- Ускорение работы существенно, так как прошлые лучшие модели - тяжеловесные ганы

# Сильные стороны

- Получилось избавиться от использования циклических архитектур и от использования дополнительных генераторов и дискриминаторов
- Не нужны негативные примеры с разных картинок, из-за этого модель можно обучать, используя всего 2 картинки - 1 из исходного домена и 1 из целевого
- CUT бьет все прошлые модели с запасом по всем параметрам
- FastCut побивает по FID прошлые решения и при этом занимает меньше памяти и времени
- Представлено большое количество экспериментов, причем есть сравнения не только с другими актуальными моделями, но и с разными настройками и режимами CUT

# Слабые стороны

- В статье написано, что патчи для негативных семплов выбираются случайно. Мне это кажется немного странным, ведь могут получиться просто одинаковые куски земли
- В статье упоминается, что можно также обучать и тестировать модель на датасете, где есть пары и метки таргетов, но нет подробного объяснения, как именно они это делают и что это дает
- В статье упоминается возможность использования только части слоев при подсчете лосса по патчам, но экспериментов по сравнению подходов нет, есть только примеры с использованием либо всех слоев, либо только одного.

# Комментарии и выводы

## **Читаемость и понятность:**

Написано неплохо, все формулы поясняются, и поэтому читать несложно.

Немного запутанная структура статьи: перемешано описание архитектуры и сравнение с прошлыми моделями.

## **Воспроизводимость:**

В статье приведена ссылка на гитхаб, где есть исходный код и ReadMe с инструкцией.

## **Вывод:**

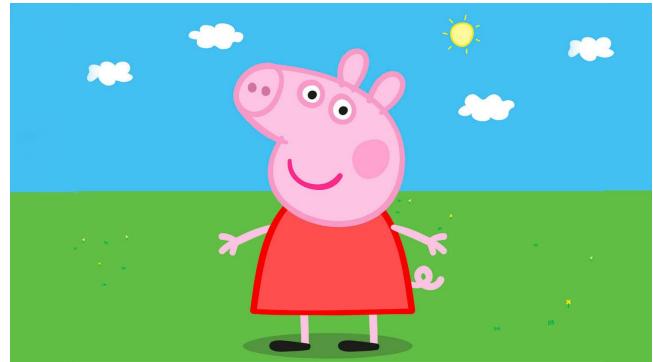
Очень сильная статья, предлагает новый подход к решению задачи, который дает лучше качество и при этом еще и показывает лучшую производительность.

Оценка по NIPS: 9 (уверенность 3)

# О статье



# Эксперимент 1



# Эксперимент 1



# Эксперимент 1



# Эксперимент 1



## Эксперимент 2



# Эксперимент 3

