Stacking for improving neural optimal transport based style-transfer models

Matvey Skripkin [1] **Mikhail Vulf** [1] **Nikita Ligostaev** [1] **Mikhail Koksharov** [1] **Antonina Kurdyukova** [1]

Abstract

Solving the continuous optimal transport problem by neural networks is a new promising approach for unsupervised style-transfer problem. The method uses a one-to-one mapping between source and target data distributions, producing good image quality. However, this image quality can be further improved by utilizing a stacking-like approach. This approach involves learning a series of OT maps between a new source distribution and the target distribution, which is created by applying the learned OT map to all input data. This method can result in higher image quality and can lead to more realistic and accurate style-transfer results.

Github repository: link to the project Github repository.

1. Introduction

Solving optimal transport (OT) problems with neural networks is a widespread topic in machine learning. It is shown that corresponding algorithms can be applied to large-scale computer vision tasks (Korotin et al., 2022). Also, such models as Wasserstain GANs uses optimal transport as a loss function to update the generator models in a generative simulation problem (Arjovsky et al., 2017).

In this paper the unsupervised style-transfer problem is being solved.

There are two approaches to solve the OT problem via neural networks. First approach is used in such models as GANs. OT cost is used as a loss function to compare two probability distributions (Sanjabi et al., 2018; Liu et al., 2019). We consider the second approach which is more novel. In contrast to GANs, optimal transport uses the input distribution and output distribution both at the same time, trains the optimal transport mapping between them and uses it as a generative model.

A novel neural-networks-based algorithm to compute optimal transport maps for strong transport costs is used (Korotin et al., 2022).

The main contributions of this report are as follows:

- Repeat the experiments presented in (Korotin et al., 2022)
- Apply a stacking-like approach to the algorithm, presented in (Korotin et al., 2022)
- Perform a series of experiments to improve the performance of the model. Use the learned OT map to construct new source distribution by applying this map to all input data and then learn a second OT map between the new source distribution and target distribution, and so on.

2. Preliminaries

Generally, the optimal transport transforms one probability distribution into another.

2.1. Strong OT formulation.

Given two probability distributions $\mathbb{P} \in \mathcal{P}(\mathcal{X})$ and $\mathbb{Q} \in \mathcal{P}(\mathcal{Y})$ from the respective Polish spaces $\mathcal{X}$ and $\mathcal{Y}$. The goal is to find a mapping T which would transform the distribution $\mathbb{P}$ to $\mathbb{Q}$.

While there are infinite amount of possible mappings, the aim is to identify the most optimal mapping based on a specific criterion. This criterion involves the cost of moving a point $x \in \mathbb{P}$ to another point $y \in \mathbb{Q}$ or, in other words, **OT cost** function.

Monge's primal formulation of the OT cost is

$$\text{Cost}(\mathbb{P}, \mathbb{Q}) = \inf_{T_{\#}\mathbb{P}=\mathbb{Q}} \int_{\mathcal{X}} c(x, T(x)) d\mathbb{P}(x), \qquad (1)$$

where the minimum is taken over measurable functions (transport maps) $T : X \to Y$ that map $\mathbb{P}$ to $\mathbb{Q}$. Let's denote

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Matvey Skripkin <Matvey.Skripkin@skoltech.ru>, Mikhail Vulf <Mikhail.Vulf@skoltech.ru>, Nikita Ligotsaev <Nikita.Ligostaev@skoltech.ru>, Mikhail Koksharov <Mikhail.Koksharov@skoltech.ru>, Antonina Kurdyukova <Antonina.Kurdyukova@skoltech.ru>.

Final Projects of the Machine Learning 2023 Course, Skoltech, Moscow, Russian Federation, 2023.

such functions as $T_{\#}P = \mathbb{Q}$ (Figure 3). The optimal T^* is called the **OT map**.

The objective is to find the mapping that is most cost-effective. In other words, gives the minimal value of the integral (1).

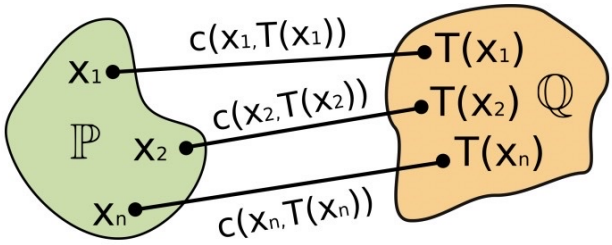

Figure 1. Monge's OT formulation (Korotin et al., 2022).

In the Monge's formulation an OT map may not exist. In other words, for some $\mathbb{P}, \mathbb{Q} \in \mathcal{P}(\mathcal{X}), \mathcal{P}(\mathcal{Y})$ there may be no T which map them one-to-one.

Thus, (Kantorovitch, 1958) proposed the following symmetric relaxation for (1):

$$\text{Cost}(\mathbb{P}, \mathbb{Q}) = \inf_{\pi \in \prod(\mathbb{P}, \mathbb{Q})} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) d\pi(x, y), \quad (2)$$

where the optimal $\pi^* \in \prod(\mathbb{P}, \mathbb{Q})$ is called the **OT plan** (Figure 2).

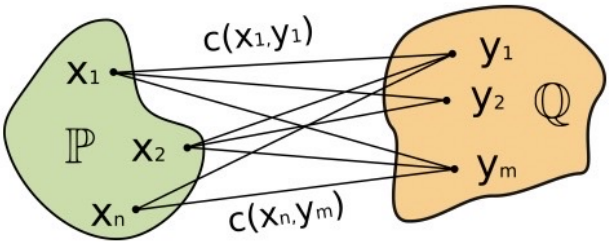

Figure 2. Strong OT formulation (Korotin et al., 2022).

2.2. Dual problem description

In this paper we solve strong OT problem, which is difficult to solve in forms (1) or (2). Therefore, we present the dual form of our problem to tackle this issue.

Dual problem for strong costs is well-known as the Kantorovich duality. A thorough explanation of conversion from primal to dual form is provided in (Korotin et al., 2022). Let us use the results from this paper.

The maxmin reformulation of the dual problem:

$$\text{Cost}(\mathbb{P}, \mathbb{Q}) = \sup_{f} \inf_{T} \mathcal{L}(f, T), \quad (3)$$

where the functional $\mathcal{L}$ is defined as:

$$\mathcal{L}(f, T) \stackrel{\text{def}}{=} \int \frac{\|x - T(x)\|_2^2}{2} p(x) dx + \\ + \int f(y) q(y) dy - \int f(T(x)) p(x) dx,$$

where $c(x, T(x)) = \frac{\|x - T(x)\|_2^2}{2}$ was taken in (1).

3. Related work

In this work we solve the practical task of neural style transfer (NST). NST is the problem of taking a content image and a style image as input, and outputting an image that has the content of the content image and the style of the style image.

Gatys et al (Gatys et al., 2015) proposed the first algorithm. In this algorithm, a VGG-16 architecture (Simonyan & Zisserman, 2014) pretrained on ImageNet (Deng et al., 2009) is used to extract the features that represent semantic content and style. Although this algorithm outputs images of very high quality, this algorithm will take a lot of time.

Figure 3. Style-transfer representation (Johnson et al., 2016).

Another approach to neural style transfer achieves real-time style transfer (Johnson et al., 2016). A feedforward convolutional neural network is trained in a supervised manner. The dataset used to train the feedforward convolutional neural network is the MS COCO dataset (Lin et al., 2014). The input image is passed through the Image Transform Net. The

output of the the Image Transform Net is passed throught the VGG-16. The main drawback of this algorithm is that each network is tied to a single style. For multiple styles large amount of memory required.

Mix fast style transfer is the task of performing real-time style transfer for multiple styles using one network. (Dumoulin et al., 2016) proposes the conditional instance normalization technique. The conditional instance normalization approach is efficient and produces very good outputs. However, conditional instance normalization layer needs to be manually implemented in most of today's popular deep-learning frameworks. This paper (Yanai & Tanno, 2017) proposes an approach for Mix Fast Style Transfer that is simpler to implement that the conditional normalization approach. There is an Image Transform Network followed by a VGG-16 network. If the network is trained on 4 different styles, to combine the 4 styles, we simply present the network with a content image and a 4-dimensional vector where each element is nonzero. The specific value of each element of the vector depends on how much you want to emphasize each style. However, in order to incoporate additional elements into this style transfer network, we have to train the network from scratch instead of finetuning like it was done in the previous approach.

4. Algorithm and Model

4.1. Motivation

In the context of neural networks, optimal transport has been used to design loss functions, to learn generative models, and to solve domain adaptation problems. However, traditional optimal transport algorithms are computationally expensive and do not scale well with the size of the data.

To overcome these limitations, researchers have proposed several neural network-based approaches that approximate the optimal transport distance efficiently. These methods leverage the expressive power of deep neural networks to learn the transportation plan between two distributions directly from the data.

Neural optimal transport has several advantages over traditional optimal transport algorithms. Firstly, it allows us to work with high-dimensional data without requiring any explicit feature engineering. Secondly, it provides a flexible framework for designing loss functions that can capture complex relationships between the data. Finally, it can be trained end-to-end using stochastic gradient descent, making it scalable to large datasets.

4.2. Stacking-like approach

The main idea of our work is to apply a stacking-like approach to the algorithm, considered in (Korotin et al., 2022). The demonstrative scheme of the stacking process is presented on the Figure 4.

We start from two distributions $\mathbb{P}, \mathbb{Q}$. Use the learned OT map to construct new source distribution by applying this map to all input data and then learn a second OT map between the new source distribution and target distribution, and so on. There are k steps of stacking on the Figure 4. The hypothesis is that the OT map should converge to one-to-one map between distributions $\mathbb{P}, \mathbb{Q}$.

4.3. Algorithm description

To approach the problem (3, 4) in practice, we use neural networks $T_\theta : \mathbb{R}^P \to \mathbb{R}^Q$ and $f_\omega : \mathbb{R}^P \to \mathbb{R}$.

Firstly, for fixed f_θ solve the inner problem:

$$T_\varphi^* = \text{arginf}_T \int \frac{\|T_\varphi(x) - x\|^2}{2} p(x)dx + \\ + \int f_\theta(y)p(y)dy - \int f_\theta(T_\varphi(x))p(x)dx \tag{4}$$

$$T_\varphi^* = \text{arginf}_T \int \frac{\|T_\varphi(x) - x\|_2^2}{2} p(x)dx - \\ - \int f_\theta\left(T_\varphi(x)\right) p(x)dx \tag{5}$$

Calculate the gradient:

$$\nabla_\varphi \left(\int \frac{\|T_\varphi(x) - x\|_2^2}{2} p(x)dx - \int f_\theta\left(T_\varphi(x)\right) p(x)dx \right) = \\ = \int \nabla_\varphi \left(\frac{\|T_\varphi(x) - x\|_2^2}{2} \right) p(x)dx - \\ - \int \nabla_\varphi \left(f_\theta\left(T_\varphi(x)\right) \right) p(x)dx \tag{6}$$

Note, that last two terms in (6) are expectations. Thus, we can estimate them by mean:

$$\int \nabla_\varphi \left(\frac{\|T_\varphi(x) - x\|_2^2}{2} \right) p(x)dx + \\ - \int \nabla_\varphi \left(f_\theta\left(T_\varphi(x)\right) \right) p(x)dx \approx \\ \approx \frac{1}{N} \sum_{i=1}^N \nabla_\varphi \left(\frac{\|T_\varphi\left(x_i\right) - x_i\|_2^2}{2} \right) + \\ - \frac{1}{N} \sum_{i=1}^N \nabla_\varphi \left(f_\theta\left(T_\varphi\left(\widetilde{x}_i\right)\right) \right), \tag{7}$$

where the mean is calculated for batches x and $\tilde{x}$ of samples from the $p(x)$.

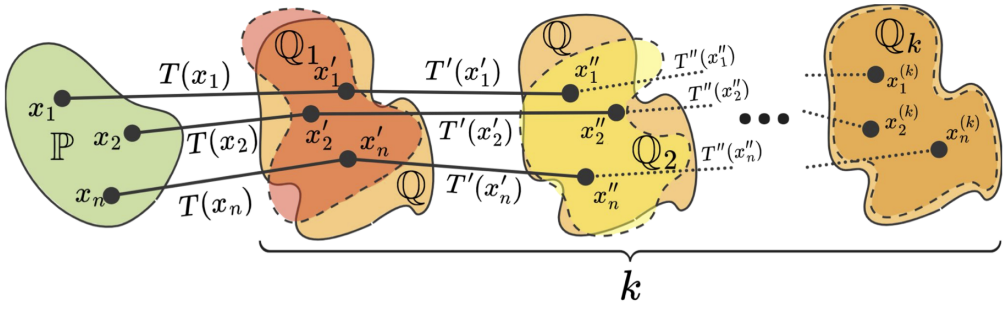

Figure 4. Stacking-like Neural Optimal Transport.

The mean values (7) is calculated by backpropagation for the loss:

$$\ell = \frac{1}{N}\sum_{i=1}^{N}\nabla_\varphi\left(\frac{\|T_\varphi(x_i)-x_i\|_2^2}{2}\right) + \\ -\frac{1}{N}\sum_{i=1}^{N}\nabla_\varphi\left(f_\theta\left(T_\varphi\left(\widetilde{x}_i\right)\right)\right) \tag{8}$$

Finally, fix T_φ^* and solve the outer problem in the same way.

For the pseudocode of the full algorithm with stacking see Algorithm 1. The parameters of the neural networks T_θ, f_ω are trained with the Adam optimizer by using random batches from $\mathbb{P}, \mathbb{Q}$.

5. Experiments and Results

5.1. Image datasets

We use the following publicly available datasets as $\mathbb{P}, \mathbb{Q}$: shoes (Yu & Grauman, 2014) and Amazon handbags. The size of datasets varies from 45K to 125K images. In our experiments we learned an OT mapping from bags images to shoes images.

The datasets are publicly available here (Zhu, 2019).

5.1.1. Shoes dataset

The shoes dataset (Yu & Grauman, 2014) is a collection of images of shoes that contains a total of 124,891 images of shoes, which are divided into 4 different categories: athletic, boots, sandals, and shoes.

Each image in the dataset is a color image with a resolution of 64×64 pixels. The shoes in the images are photographed in such way that all images are displayed in the same angular perspective.

5.1.2. Handbags dataset

A handbag images dataset is a collection of images of different types of handbags, purses, clutches, and other similar

Figure 5. Shoes dataset.

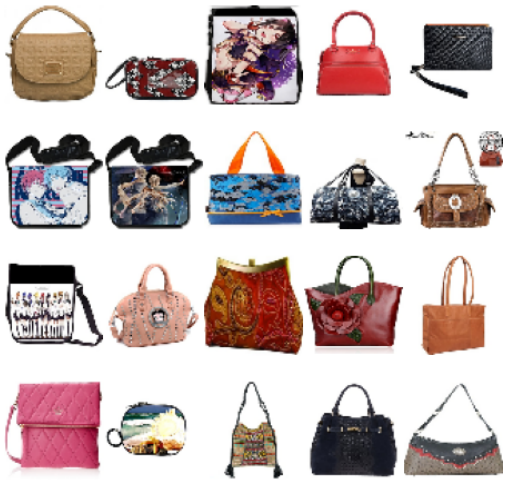

Figure 6. Handbags dataset.

Algorithm 1 Stacking-like strong neural optimal transport (SNOT)

Input: distributions $\mathbb{P}, \mathbb{Q}$ accesible by samples;
potential network $f_\omega : \mathbb{R}^P \to \mathbb{R}$;
identity map $T_1^\theta : \mathbb{R}^P \to \mathbb{R}^P$;
mapping network $T_n^\theta : \mathbb{R}^P \to \mathbb{R}^Q$ at the n-th iteration of stacking;
number of inner iterations K_T; number of stacking iterations K_{st}
Output: learned deterministic OT map T_θ between distributions $\mathbb{P}, \mathbb{Q}$;
for $n = 1, 2, ..., K_{st}$ **do**
  repeat
    Sample batches $Y \sim \mathbb{Q},\ X \sim \mathbb{P}$;
    $X \leftarrow (T_n^\theta \circ T_{n-1}^\theta \circ \cdots \circ T_1^\theta)(X)$;
    $\mathcal{L}_f \leftarrow \frac{1}{|X|} \sum_{x \in X} f_\omega(T_{n+1}^\theta(x)) - \frac{1}{|Y|} \sum_{y \in Y} f_\omega(y)$;
    Update ω by using $\frac{\partial \mathcal{L}_f}{\partial \omega}$;
    for $k_T = 1, 2, ..., K_T$ **do**
      Sample batch $X \sim \mathbb{P}$;
      $X \leftarrow (T_n^\theta \circ T_{n-1}^\theta \circ \cdots \circ T_1^\theta)(X)$;
      $\mathcal{L}_T \leftarrow \frac{1}{|X|} \sum_{x \in X} \left[\frac{\|T_{n+1}^\theta(x) - x\|^2}{2} - f_\omega(T_{n+1}^\theta(x)) \right]$;
      Update θ by using $\frac{\partial \mathcal{L}_T}{\partial \theta}$;
    end for
  until not converged;
end for

accessories. The dataset is a collection of images of handbags that contains a total of 45,023 images of handbags.

As a previous dataset, resolution of each 3-channel image in the handbags dataset is 64×64 pixels.

5.2. Training description and evaluation

To evaluate our model the Frechet Inception Distance (FID) is calculated. The FID score was proposed and used by Martin Heusel, et al (Heusel et al., 2017). This metric calculates the distance between two distributions of images.

A common standardization of images and mapping to $[-1, 1]$ was performed while preprocessing.

Firstly, the baseline algorithm (Korotin et al., 2022) was launched. Then, the first step of stacking was performed, which gave some decrease in FID-score. The second step of stacking increased the FID-score compared with the first step and with the baseline (Figure 8). The scores and hyperparameters are collected in Table 1

Then we tried to change the hyperparameters during the first step of stacking. But it led to the increase in FID-score (Figure 9). The detailed scheme of this experiments with hyperparameters changes is presented on Figure 7. All FID-scores and hyperparameters are collected in the Table 2

5.3. Computing infrastructure

A personal computer with CPU Intel Core i7-9700K, and GPU Nvidia GeForce GTX 1080 Ti was used.

5.4. Code availability

The code is written in PyTorch framework and is publicly available at the project Github repository.

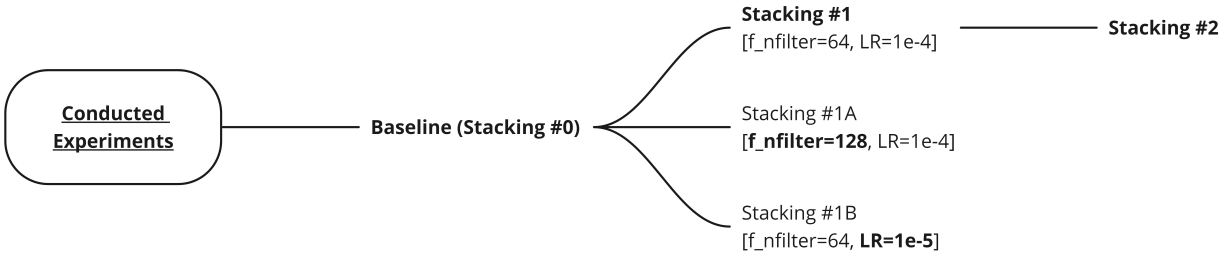

Figure 7. Block-scheme of the conducted experiments.

6. Conclusion and further objectives

The field of optimal transport has seen a tremendous growth in recent years, with numerous applications in image processing, computer vision, machine learning, and statistics. The optimal transport theory provides a powerful mathematical framework for solving problems related to the transportation of mass between two probability distributions.

The stacking-like strategy utilized in this paper is a method for combining the results of earlier models to achieve performance that is superior to that of any one model working alone. The purpose of this study was to test the hypothesis that using a stacking-like approach will increase neural optimal transport performance.

We applied the neural optimal transport algorithm in a stacking-like manner to the initial distribution $\mathbb{P}$ and obtained a modified distribution $\mathbb{Q}_1$ that was different from the final desired distribution $\mathbb{Q}$. We examined the hypothesis that the distribution $\mathbb{Q}_k$ increasingly approaches the real distribution $\mathbb{Q}$. Convergence was assessed using the FID-score.

We concluded from our experiments that the FID-metric did not demonstrate a statistically significant improvement over the strong NOT. The subsequent step toward generating a statistically meaningful improvement in the FID-metric might be to test a stacking-like approach on weak neural optimal transport.

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dumoulin, V., Shlens, J., and Kudlur, M. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

Gatys, L. A., Ecker, A. S., and Bethge, M. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pp. 694–711. Springer, 2016.

Kantorovitch, L. On the translocation of masses. *Management science*, 5(1):1–4, 1958.

Korotin, A., Selikhanovych, D., and Burnaev, E. Neural optimal transport. *arXiv preprint arXiv:2201.12220*, 2022.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Liu, H., Gu, X., and Samaras, D. Wasserstein gan with quadratic transport cost. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4832–4841, 2019.

Sanjabi, M., Ba, J., Razaviyayn, M., and Lee, J. D. On the convergence and robustness of training gans with regularized optimal transport. *Advances in Neural Information Processing Systems*, 31, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Yanai, K. and Tanno, R. Conditional fast style transfer network. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pp. 434–437, 2017.

Yu, A. and Grauman, K. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 192–199, 2014.

Zhu, J.-Y. hdf5 datasets. https://github.com/junyanz/iGAN/blob/master/train_dcgan/README.md, 2019. [Online; accessed 20-Feb-2023].

Table 1. FID-scores and hyperparameters of first series of experiments.

STACKING NAME	FID-SCORE	T_ITERS	BATCH_SIZE	F_NFILTER	LR, 10^{-4}
BASELINE	15.204	10	64	64	1
STACKING 1	15.163	10	64	64	1
STACKING 2	15.495	10	64	64	1

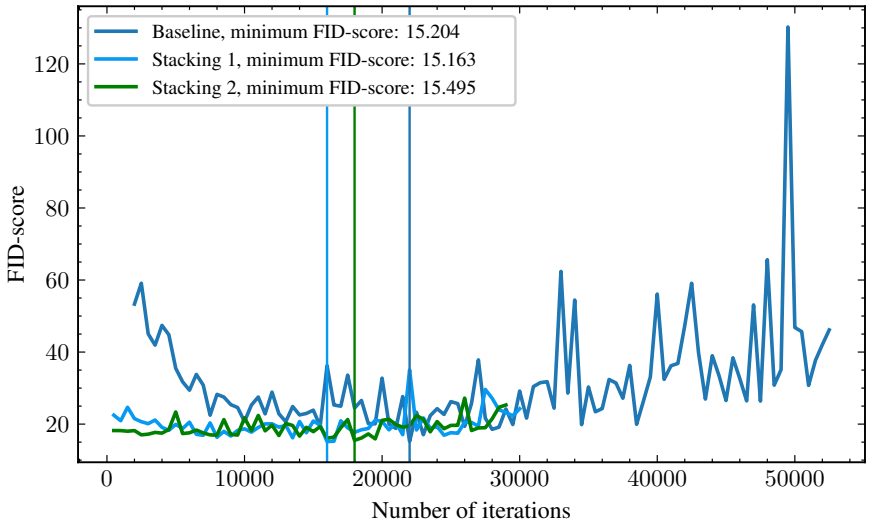

Figure 8. First series of experiments: running a baseline and two runs of stacking-like approach.

Table 2. FID-scores and hyperparameters of second series of experiments.

STACKING NAME	FID-SCORE	F_NFILTER	LR, 10^{-4}
STACKING 1	15.163	64	1
STACKING 1A	16.158	128	1
STACKING 1B	18.07	64	0.1

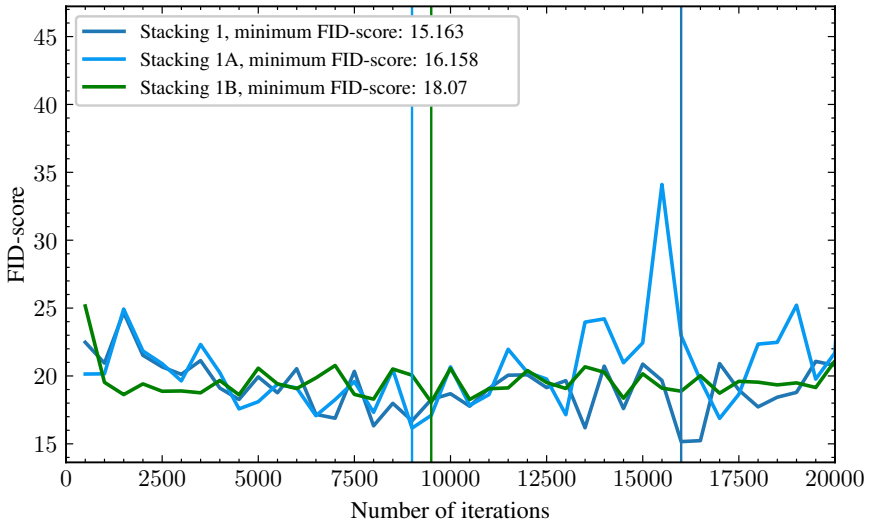

Figure 9. Second series of experiments: changing hyperparameters in stacking-like approach.

A. Experimental results

Figure 10. Baseline style-transfer with strong OT maps: handbags $\rightarrow$ shoes translation, 64×64, number of iterations $\in [0; 52500]$

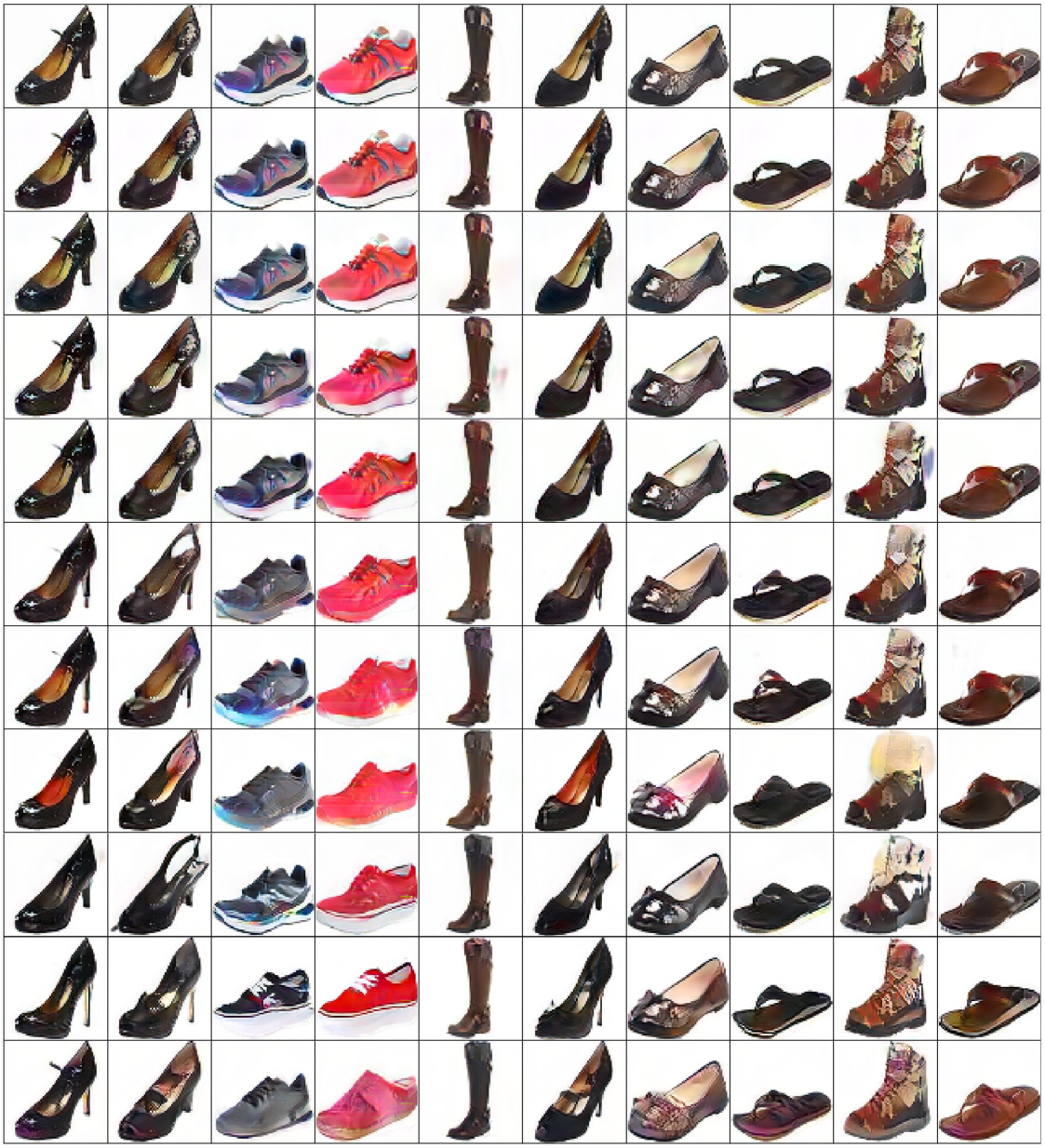

Figure 11. First stacking style-transfer with strong OT maps: handbags $\rightarrow$ shoes translation, 64×64, number of iterations $\in [0; 30000]$

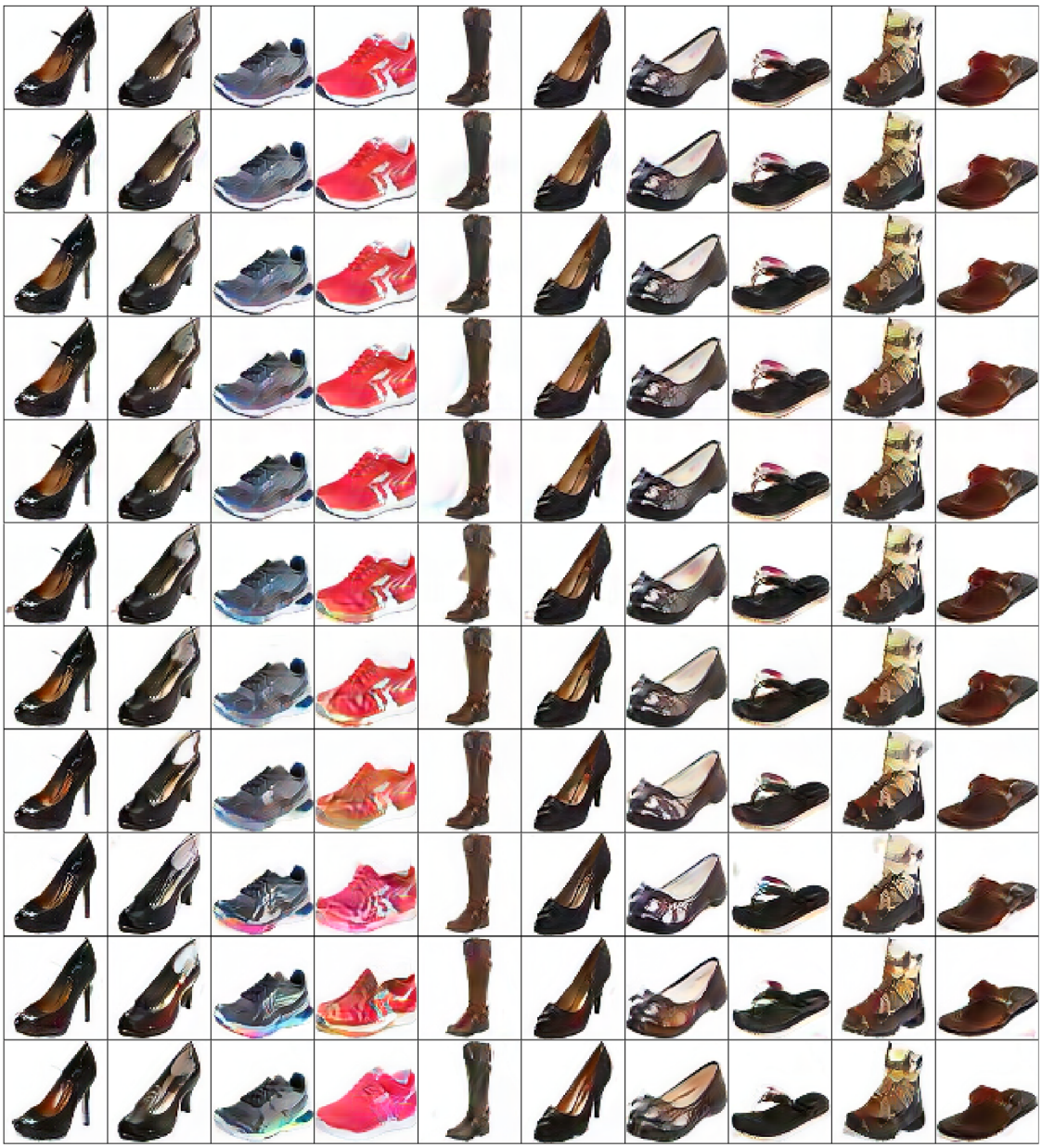

Figure 12. Second stacking style-transfer with strong OT maps: handbags $\rightarrow$ shoes translation, 64×64, number of iterations $\in [0; 29300]$

A. Team member's contributions

Explicitly stated contributions of each team member to the final project.

Matvey Skripkin (20% of work)

- Coding the main algorithm
- Preparing the GitHub Repo

Mikhail Vulf (20% of work)

- Team lead
- Coding the main algorithm
- Preparing the GitHub Repo

Nikita Logitsaev (20% of work)

- Coding the main algorithm
- Preparing the report

Mikhail Koksharov (20% of work)

- Preparing the presentation slides
- Perform the theoretical research

Antonina Kurdyukova (20% of work)

- Preparing the report
- Reviewing literature on the topic

B. Reproducibility checklist

Answer the questions of following reproducibility checklist. If necessary, you may leave a comment.

1. A ready code was used in this project, e.g. for replication project the code from the corresponding paper was used.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   General comment: If the answer is **yes**, students must explicitly clarify to which extent (e.g. which percentage of your code did you write on your own?) and which code was used.

   Students' comment: About 50% of code is written by our team. This project was based on this paper by Alexander Korotin, Daniil Selikhanovych and Evgeny Burnaev and this GitHub repository.

2. A clear description of the mathematical setting, algorithm, and/or model is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

3. A link to a downloadable source code, with specification of all dependencies, including external libraries is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

4. A complete description of the data collection process, including sample size, is included in the report.

   ☐ Yes.
   ☐ No.
   ☑ Not applicable.

   Students' comment: The data was taken from the public domain that is available at this link.

5. A link to a downloadable version of the dataset or simulation environment is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

6. An explanation of any data that were excluded, description of any pre-processing step are included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

7. An explanation of how samples were allocated for training, validation and testing is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

8. The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results are included in the report.

   ☐ Yes.
   ☐ No.
   ☑ Not applicable.

   Students' comment: Our task was to test the hypothesis that stacking would improve the metric of the model with optimized hyper-parameters. One calculation took about 40 hours, so the influence of only a few hyper-parameter values was estimated (see Figure 7).

9. The exact number of evaluation runs is included.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

10. A description of how experiments have been conducted is included.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None

11. A clear definition of the specific measure or statistics used to report results is included in the report.

    ☐ Yes.
    ☐ No.
    ☑ Not applicable.

Students' comment: A Brief description of the well-known Frechet Inception Distance (FID) is introduced in the report as well as a reference with detailed description.

12. Clearly defined error bars are included in the report.

    ☐ Yes.
    ☐ No.
    ☑ Not applicable.

    Students' comment: None

13. A description of the computing infrastructure used is included in the report.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None