Final project: “Stacking for improving neural optimal transport based style-transfer models”

Daniil Panov [1] **Anastasia Gavrish** [1] **Nikita Bogdanov** [1] **Nikita Vasilev** [1] **Nikolay Kashin** [1]

Abstract

In this project, a stacking approach was applied to a neural optimal transport(NOT) problem of image style transfer. The main objective is to evaluate the effect of the stacking on improvement of the NOT performance. The stacking was done over the strong neural optimal optimal transport realization from (Korotin et al., 2022). The stacking was applied for shoes to bags style transfer problem. The unpaired datasets were taken from open sources. Evaluation of the results was done based on Frechet Inception Distance(FID). Up to three iterations of the stacking were conducted to see which of the iterations would have the lowest FID score.

Github repo: our project github repo link here

1. Introduction

The solution of problem of finding the optimal way of moving one distribution of mass were proposed as large-scale OT (Seguy et al., 2017) and Wasserstein GANs (Arjovsky et al., 2017). In most of the methods for solving OT, the loss is being calculated and used in the update of generator in generative models (Gulrajani et al., 2017; Liu et al., 2017; Sanjabi et al., 2018; Petzka et al., 2017). One of the most recent works present the neural-networks-based algorithms to compute optimal transport maps and plans for strong and weak transport costs (Korotin et al., 2022).

Existing methods are designed only for strong OT formulation. Most of them search for a deterministic solution, i.e., for a map $T^\star$ rather than a stochastic plan $\pi^\star$, although $T^\star$ might not always exist.

To compute the OT plan (map), (Lu et al., 2020; Xie et al., 2019) approach the primal formulation. Their methods imply using generative models and yield complex optimization objectives with several adversarial regularizers, e.g., they are used to enforce the boundary condition ($T_\# P = Q$). As a result, the methods are hard to setup since they require careful selection of hyperparameters.

In contrast, methods based on the dual formulation have simpler optimization procedures. Most of such methods are designed for OT with the quadratic cost, i.e., the Wasserstein-2 distance (W_2^2). An evaluation of these methods is provided in (Korotin et al., 2021a).

Beside the high potential of the application of the NOT, it is still can produce artifacts or defects in the image. In the report, we propose stacking of a few NOT models to increase the performance of the NOT.

1.1. The main contributions of this report

- In Section ”Algorithms and Models”, we provide a theoretical description of what loss function is used and how the stacking is applied for model improvement.
- In Section ”Experiments and Results.”, we show how stacking influence our predictions. Also data and its preprocessing is described.

2. Algorithms and Models

In our project we use next experimental setup for numerical experiments: one premium google Colab and one home personal computer. The source code and requirments could be find in the project Github repository (GitHub)

In this section, we provide some concepts of the strong Optimal Transport (OT) theory.

Notations. We use X, Y, Z to denote Polish spaces and $P(X), P(Y), P(Z)$ to denote the respective sets of probability distributions on them. We denote the set of probability distributions on X × Y with marginals P and Q by $\Pi(P, Q)$. For a measurable map $T : X \times Z \to Y$ (or $T : X \to Y$),

Strong OT formulation. For $\mathbb{P} \in P(X)$, $\mathbb{Q} \in P(Y)$ and a cost function $c : X \times Y \to R$, Monge’s primal formulation

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Nikita Gushchin <Nikita.Gushchin@skoltech.ru>.

Final Projects of the Machine Learning 2020 Course, Skoltech, Moscow, Russian Federation, 2020.

of the OT cost is:

$$Cost(\mathbb{P}, \mathbb{Q}) = \inf_{T} \int_{\chi} c(x, T(x))\, d\mathbb{P}(x) \quad (1)$$

where the minimum is taken over measurable functions (transport maps) $T : X \to Y$ that map P to Q (Figure 1).

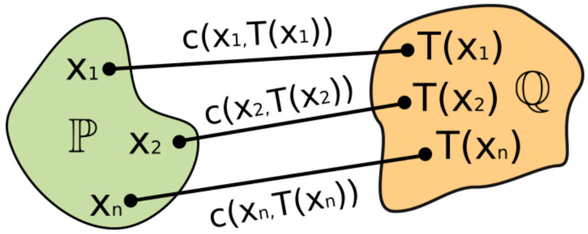


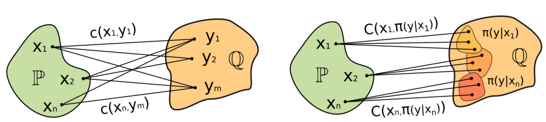
Figure 1. Monge's OT formulation (Korotin et al., 2022)

The optimal T^* is called the OT map.

Note that (1) is not symmetric and does not allow mass splitting, i.e., for some $\mathbb{P}, \mathbb{Q} \in P(X), P(Y)$, there may be no T satisfying $T_{\#}P = Q$. Thus, (Kantorovitch, 1958) proposed the following relaxation:

$$Cost(\mathbb{P}, \mathbb{Q}) = \inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \int_{\chi \times y} c(x, y)\, d\pi(x, y) \quad (2)$$

where the minimum is taken over all transport plans π(Figure 2a), i.e., distributions on $X \times Y$ whose marginals are $\mathbb{P}$ and $\mathbb{Q}$. The optimal $\pi\star \in \Pi(\mathbb{P}, \mathbb{Q})$ is called the optimal transport plan. If $\pi\star$ is of the form $[id, T\star]\,\mathbb{P} \in \Pi(\mathbb{P}, \mathbb{Q})$ for some $T\star$, then $T\star$ minimizes (1). In this case, the plan is called deterministic. Otherwise, it is called stochastic (nondeterministic)



(a) Strong OT formulation (2). (b) Weak OT formulation (3).

Figure 2. Strong (Kantorovitch, 1958) and weak (Gozlan et al., 2017) optimal transport fomulations

An example of OT cost for $X = Y = \mathbb{R}^D$ is the (p power of) Wasserstein-p distance W_p, i.e., formulation (2) with $c(x, y) = \|x - y\|^p$. Two its most popular cases are $p = 1, 2 (\mathbb{W}_1, \mathbb{W}_2^2)$.

Weak OT formulation (Gozlan et al., 2017). Let $C : X \times P(Y) -> \mathbb{R}$ be a weak cost, i.e., a function which takes a point $x \in \chi$ and a distribution of $y \in Y$ as input. The weak OT cost between $\mathbb{P}, \mathbb{Q}$ is

$$Cost(\mathbb{P}, \mathbb{Q}) = \inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \int_{\chi} C(x, \pi(\cdot|x))\, d\pi(x) \quad (3)$$

where $\pi(\cdot|x)$ denotes the conditional distribution (Figure 3b). Note that (3) is a generalization of (2). Indeed, for cost $C(x, \mu) = \int_Y c(x, y)\, d\mu(y)$, the weak formulation (3) becomes strong (2). example of a weak OT cost for $X = Y = \mathbb{R}^D$ is the γ-weak ($\gamma \geq 0$) Wasserstein-2 (W_2, γ):

$$C(x, \mu) = \int_Y \frac{1}{2}\|x - y\|^2 d\mu(y) - \frac{\gamma}{2} Var(\mu) \quad (4)$$

Existence and duality. Throughout the paper, we consider weak costs C(x, μ) which are lower bounded, convex in μ and jointly lower semicontinuous in an appropriate sense. Under these assumptions, (Backhoff-Veraguas et al., 2019) prove that the minimizer π^* of (3) always exists With mild assumptions on c, strong costs satisfy these assumptions. In particular, they are linear w.r.t. μ, and, consequently, convex. The γ-weak quadratic cost (4) is lower-bounded (for $\gamma \leq 1$) and is also convex since the functional Var(μ) is concave in μ. For the costs in view, the dual form of (3) is

$$Cost(\mathbb{P}, \mathbb{Q}) = \sup_{f} \int_X f^C(x) d\mathbb{P}(x) + \int_Y f(y) d\mathbb{P}(y) \quad (5)$$

where f are the upper-bounded continuous functions with not very rapid growth (Backhoff-Veraguas et al., 2019) and f^C is the weak C-transform of f, i.e

$$f^C(x) \overset{def}{=} \inf_{\mu \in P(Y)} \{C(x, \mu) - \int_Y f(y) d\mu(y)\} \quad (6)$$

Note that for strong costs C, the infimum is attained at any $\mu \in P(Y)$ supported on the $arg\inf_{y \in Y}\{c(x, y) - f(y)\}$ set. Therefore, it suffices to use the strong c-transform:

$$f^C(x) = f_c(x) \overset{def}{=} \inf_{y \in Y} \{c(x, y) - f(y)\}. \quad (7)$$

For strong costs (2), formula (5) with (7) is the well known Kantorovich duality (Villani, 2008) In this project we use two neural networks: ResNet and Unet.A residual neural network (ResNet) is an artificial neural network (ANN). It is a gateless or open-gated variant of the HighwayNet, the first working very deep feedforward neural network with hundreds of layers, much deeper than previous neural networks. Unet network is based on the fully convolutional network and its architecture was modified and extended to work with fewer training images and to yield more precise segmentations.

2.1. Dataset and preprocessing

In this section, we give explanation of datasets which we use. First dataset is shoes, which is provided by Yu and Grauman. It is a large shoe dataset consisting of 50K catalog images collected from Zappos.com (reference to dataset). The images are divided into 4 major categories — shoes, sandals, slippers, and boots — followed by functional types and individual brands. The shoes are centered on a white background and pictured in the same orientation for convenient analysis. Another dataset 137K handbag images is downloaded from Amazon (reference to dataset). In both datasets, image have 3 RGB channel with size 64x64 pixel. After loading datasets we apply normalization to them. Then, split data into to train (90 % of whole dataset) and test part (10 % of whole dataset).

2.2. Training parameters and metric

For training Unet and ResNet models we use learning rate which is 10^{-4}, weight decay 10^{-10} and batch size is 128. The training of the models was carried out as follows, each iteration of the model consisted of 10,000 epochs. In our project, we have implemented 3 epochs. To evaluate the accuracy of the result obtained, the metrica FID is used. FID is a measure of similarity between two datasets of images. It was shown to correlate well with human judgement of visual quality and is most often used to evaluate the quality of samples of Generative Adversarial Networks. FID is calculated by computing the Fréchet distance between two Gaussians fitted to feature representations of the Inception network.

The formula for Calculates Frechet inception distance, which is used to access the quality of generated images, is:

$$FID = \|\mu - \mu_w\| + tr(\sum + \sum_w - 2(\sum \sum_w)^{\frac{1}{2}} \quad (8)$$

where $N(\mu, \sum)$ is the multivariate normal distribution estimated from Inception v3 (Szegedy et al., 2016) features calculated on real life images and $N(\mu_w, \sum_w)$ is the multivariate normal distribution estimated from Inception v3 features calculated on generated (fake) images. The metric was originally proposed in (Szegedy et al., 2016).

Using the default feature extraction (Inception v3 using the original weights from (Heusel et al., 2017)), the input is expected to be mini-batches of 3-channel RGB images of shape $(3 \times H \times W)$.

3. Related work

In large-scale machine learning, OT costs are primarily used as the loss to learn generative models. Wasserstein GANs introduced by (Arjovsky et al., 2017; Gulrajani et al., 2017) are the most popular examples of this approach. We refer to (Korotin et al., 2022; 2021a) for recent surveys of principles of WGANs. However, these models are out of scope of our paper since they only compute the OT cost but not OT plans or maps (4.3). To compute OT plans (or maps) is a more challenging problem, and only a limited number of scalable methods to solve it have been developed. We overview methods to compute OT plans (or maps) below. We emphasize that existing methods are designed only for strong OT formulation (2). Most of them search for a deterministic solution (1), i.e., for a map $T^\star$ rather than a stochastic plan $\pi^\star$, although $T^\star$ might not always exist. To compute the OT plan (map), (Lu et al., 2020; Xie et al., 2019) approach the primal formulation (1) or (2). Their methods imply using generative models and yield complex optimization objectives with several adversarial regularizers, e.g., they are used to enforce the boundary condition ($T_\# P = Q$). As a result, the methods are hard to setup since they require careful selection of hyperparameters. In contrast, methods based on the dual formulation (5) have simpler optimization procedures. Most of such methods are designed for OT with the quadratic cost, i.e., the Wasserstein-2 distance ($\mathbb{W}_2^2$). An evaluation of these methods is provided in (Korotin et al., 2021a). Below we mention their issues. Methods by (Taghvaei & Jalali, 2019; Makkuva et al., 2020; Korotin et al., 2021a) based on inputconvex neural networks (ICNNs, see (Amos et al., 2017)) have solid theoretical justification, but do not provide sufficient performance in practical large-scale problems. Methods based on entropy regularized OT (Genevay et al., 2016; Seguy et al., 2017), recover regularized OT plan that is biased from the true one, it is hard to sample from it or compute its density. According to (Korotin et al., 2021b), the best performing approach is $[MM : R]$, which is based on the maximin reformulation of (5). It recovers OT maps fairly well and has a good generative performance. The follow-up papers (Rout et al., 2021; Fan et al., 2021) test extensions of this approach for more general strong transport costs c(·, ·) and apply it to compute $\mathbb{W}_2$ barycenters (Korotin et al., 2022). Their key limitation is that it aims to recover a deterministic OT map T^* which might not exist

Therefore we solve reformulation task instead min task. It recovers OT maps fairly well and has a good generative performance. Min max formulation looks like this:

$$Cost(\mathbb{P}, \mathbb{Q}) = \sup_f \inf_T \mathcal{L}(f, T) \quad (9)$$

where $L(f, T)$ is next:

$$\mathcal{L}(f, T) = \int \frac{\|x - T(x)\|_2^2}{2} \, d\mathbb{P}(x) + \int f(y) \, d\mathbb{Y} - \int f(T(x)) \, d\mathbb{P}(x) \quad (10)$$

After our problem is reformulated, we applied UNet for T function and ResNet for f with initiated weights. During one training cycle we fix f terms in the loss function and at the same time looking for minimum of the T term, which is UNet for kind of generation. Having found the minimum of the first term, we fix it and start looking for the maximum of the function that is responsible for Reset for kind of discrimination. Then all the steps are repeated again until we find the saddle point.

4. Experiments and Results

	Min FID	Mean FID	Std FID
Iteration 1	35.0	43.9	9.5
Iteration 2	26.1	34.7	4.8
Iteration 3	37.9	44.6	2.1

Table 1. FID Scores

We have performed several iterations of stacking. And as you could see after about 10k epochs of the first iteration there is plateau and almost no improvement of FID metrics. Minimal FID at the first iteration is 34.9. The pictures of the bags obtained after the first iteration are quite similar to the real ones, although there are significant artifacts.

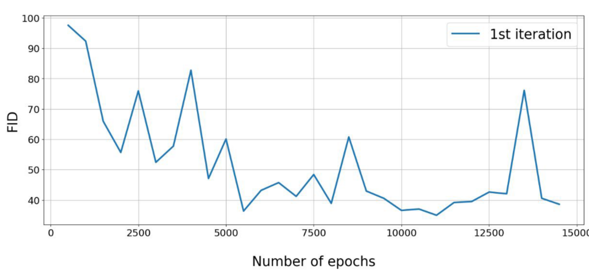


Figure 3. FID metric of the first iteration. Shoes to Bags minimal FID is 34.9.

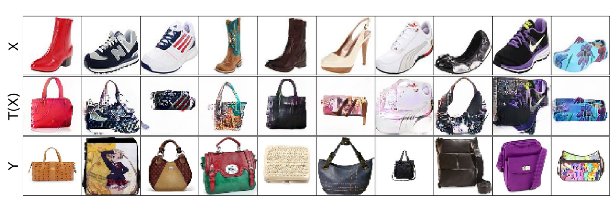


Figure 4. Result of first iteration. Shoes to Bags min FID is 34.9

We then proceeded to the second iteration and succeeded here by achieving a smaller FID - 26.2. The FID metric at first stacking behaves more smoothly without such peaks, but keep oscillate. Here we have received a significant improvement in style-transfer, because the resulting images of bags have fewer artifacts and began to look more realistic.

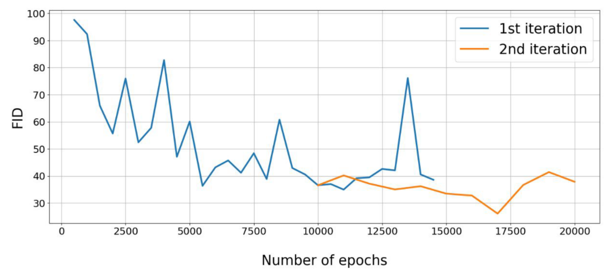


Figure 5. Result of second iteration. Bags from shoes to Bags min FID is 26.2 (orange line)

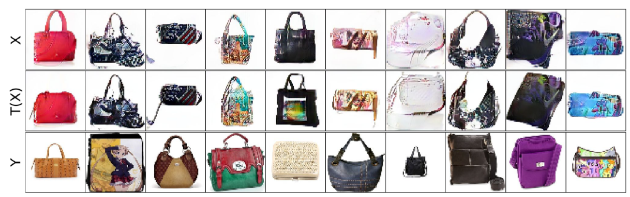


Figure 6. Bags from shoes to Bags min FID is 26.2

The last third iteration have almost no change in FID metrics, and beyond that it showed a worse result than the previous iterations. If we look at the resulting images 8, we will see that the images obtained in the previous iteration almost do not differ from the images from the last iteration

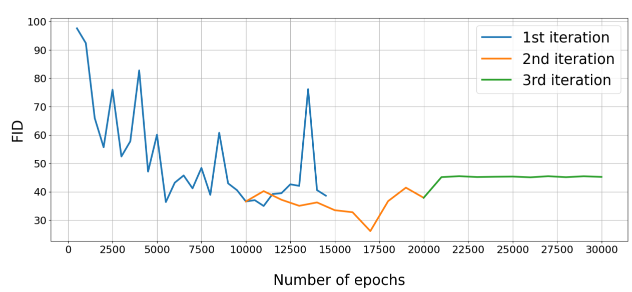


Figure 7. The main results of the work. The graph shows the dependence of the number of epochs on the FiD metric.Bags from the first stack to Bags min FID is 38.9

5. Conclusion

On zero levels of stacking the FID is equal to 34.9. With one level of stacking min FID is 26.2, which is decreased compared to previous results, that gives us a good mapping

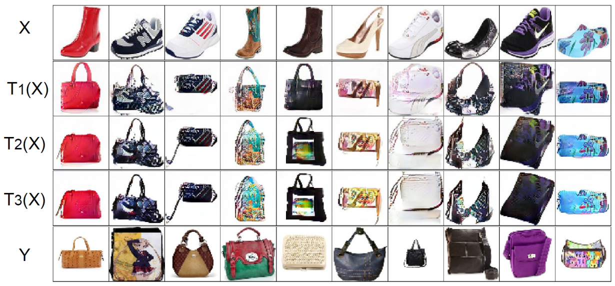


Figure 8. The main results of the work.Bags from the first stack to Bags min FID is 38.9

as an output. And finally two levels of stacking result in FID 38,9, which is much higher than the one level of stacking. Final conslusions regarding this experiment can’t be made, since it is required explorations with bigger amount of lavels.

References

Amos, B., Xu, L., and Kolter, J. Z. Input convex neural networks. In *International Conference on Machine Learning*, pp. 146–155. PMLR, 2017.

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

Backhoff-Veraguas, J., Beiglböck, M., and Pammer, G. Existence, duality, and cyclical monotonicity for weak transport costs. *Calculus of Variations and Partial Differential Equations*, 58(6):203, 2019.

Fan, J., Liu, S., Ma, S., Chen, Y., and Zhou, H. Scalable computation of monge maps with general costs. *arXiv preprint arXiv:2106.03812*, pp. 4, 2021.

Genevay, A., Cuturi, M., Peyré, G., and Bach, F. Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29, 2016.

Gozlan, N., Roberto, C., Samson, P.-M., and Tetali, P. Kantorovich duality for general transport costs and applications. *Journal of Functional Analysis*, 273(11):3327–3405, 2017.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

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.

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

Korotin, A., Li, L., Genevay, A., Solomon, J. M., Filippov, A., and Burnaev, E. Do neural optimal transport solvers work? a continuous wasserstein-2 benchmark. *Advances in Neural Information Processing Systems*, 34:14593–14605, 2021a.

Korotin, A., Li, L., Solomon, J., and Burnaev, E. Continuous wasserstein-2 barycenter estimation without minimax optimization. *arXiv preprint arXiv:2102.01752*, 2021b.

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

Liu, M.-Y., Breuel, T., and Kautz, J. Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, 30, 2017.

Lu, G., Zhou, Z., Shen, J., Chen, C., Zhang, W., and Yu, Y. Large-scale optimal transport via adversarial training with cycle-consistency. *arXiv preprint arXiv:2003.06635*, 2020.

Makkuva, A., Taghvaei, A., Oh, S., and Lee, J. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pp. 6672–6681. PMLR, 2020.

Petzka, H., Fischer, A., and Lukovnicov, D. On the regularization of wasserstein gans. *arXiv preprint arXiv:1709.08894*, 2017.

Rout, L., Korotin, A., and Burnaev, E. Generative modeling with optimal transport maps. *arXiv preprint arXiv:2110.02999*, 2021.

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.

Seguy, V., Damodaran, B. B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. Large-scale optimal transport and mapping estimation. *arXiv preprint arXiv:1711.02283*, 2017.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Taghvaei, A. and Jalali, A. 2-wasserstein approximation via restricted convex potentials with application to improved training for gans. *arXiv preprint arXiv:1902.07197*, 2019.

Villani, C. Optimal transport, old and new. notes for the 2005 saint-flour summer school. *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer*, 3, 2008.

Xie, Y., Chen, M., Jiang, H., Zhao, T., and Zha, H. On scalable and efficient computation of large scale optimal transport. In *International Conference on Machine Learning*, pp. 6882–6892. PMLR, 2019.

A. Team member's contributions

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

Daniil Panov (25% of work)

- Coding a part of stacking section
- Experiments on the first iteration of the stacking
- Preparing the GitHub Repo
- General reviewing and editing of the report
- Preparation of the project presentation

Nikita Bogdanov (25% of work)

- Coding a part of stacking section
- Experiments on the all iteration of the stacking
- Contributing to the GitHub Repo
- Editing of the experiments and Results and reviewing other sections
- Preparation of the project presentation

Nikolay Kashin (17% of work)

- Participating in group discussions
- Editing of the experiments and Results and reviewing other sections
- Preparation of the project presentation

Anastasia Gavrish (16.5% of work)

- Participating in group discussions
- Editing of the introduction section, related work section and reviewing other sections

Nikita Vasilev (16.5% of work)

- Participating in group discussions
- Editing of the introduction section, related work section and reviewing other sections

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.

   - [x] Yes.
   - [ ] No.
   - [ ] Not applicable.

   General comment: If the answer is **yes**, students must explicitly clarify the stacking piplene was implemented. We deeply investigate the ready code and add around 20%.

   Students' comment: None

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

   - [x] 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.

   - [x] Yes.
   - [ ] No.
   - [ ] Not applicable.

   Students' comment: None

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

   - [x] Yes.
   - [ ] No.
   - [ ] Not applicable.

   Students' comment: None

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

   - [x] 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.

   - [x] 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.

   - [x] 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.
   - [x] Not applicable.

   Students' comment: We were suggested not to modify hyper-parameters of the NN. Thus, creating and evaluation of the stacking pipline was done.

9. The exact number of evaluation runs is included.

   - [x] Yes.
   - [ ] No.
   - [ ] Not applicable.

   Students' comment: None

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

    - [x] 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.

    - [x] Yes.
    - [ ] No.
    - [ ] Not applicable.

    Students' comment: None

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

    - [x] Yes.
    - [ ] No.
    - [ ] Not applicable.

    Students' comment: None

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

- [x] Yes.
- [ ] No.
- [ ] Not applicable.

Students’ comment: None