Wasserstein Loss for Semantic Editing in the Latent Space of GANs

Perla Doubinsky, Nicolas Audebert, Michel Crucianu, Hervé Le Borgne

WASSERSTEIN LOSS FOR SEMANTIC EDITING IN THE LATENT SPACE OF GANS

Perla Doubinsky Nicolas Audebert Michel Crucianu Hervé Le Borgne

Preprint

ABSTRACT

The latent space of GANs contains rich semantics reflecting the training data. Different methods propose to learn edits in latent space corresponding to semantic attributes, thus allowing to modify generated images. Most supervised methods rely on the guidance of classifiers to produce such edits. However, classifiers can lead to out-of-distribution regions and be fooled by adversarial samples. We propose an alternative formulation based on the Wasserstein loss that avoids such problems, while maintaining performance on-par with classifier-based approaches. We demonstrate the effectiveness of our method on two datasets (digits and faces) using StyleGAN2.

Index Terms— GAN, Wasserstein distance, image edition

1. INTRODUCTION

GANs are known to encode the semantics of the training data in their latent space [1, 2, 3]. Moving the latent codes in certain directions results in changing specific semantic attributes in the generated images [1]. This ability makes GANs great tools to perform image editing, especially as it can be applied to real images through inversion methods [4].

The challenge is to identify the manipulations in the latent space that have the desired effect on one attribute without affecting others. To obtain such *disentangled* manipulations, existing supervised methods leverage the semantic knowledge learned by pretrained attribute classifiers operating either in the image domain (*image* classifiers) or directly in the latent domain (*latent* classifiers). The key idea is that manipulated latent codes (or the images they produce) shift the predictions to match the desired outcome [5, 6]. However, classifiers can easily be fooled [7], *e.g.* they can classify with high confidence out-of-distribution samples. As illustrated in Fig. 2a, the latent classifier of [6] steers latent codes outside the distribution resulting in edited images that are unrealistic. To address this issue, the authors employ an *ad hoc* $L2$-regularization to minimize the norm of the latent editing. While this fixes out-of-distribution edits, Fig. 2b shows that on MultiMNIST [8] this regularization produces adversarial samples [9] instead, *i.e.* the edited latent codes are correctly classified but the corresponding images remain unchanged. This is not surprising as changing the predicted class while minimizing the $L2$-norm of the edit precisely mimics the search for adversarial examples.

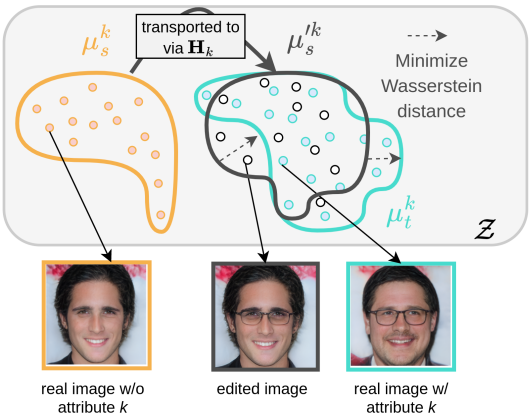


Fig. 1: **Method overview.** For each semantic attribute (*e.g.* "Glasses") we learn a mapping $\mathbf{H}_k$ that moves the distribution of latent codes lacking the attribute to the distribution of codes having that attribute. We enforce that each latent code is moved near a point that shares similar semantics, thus only changing that attribute. For identity preservation, the resulting distribution does not entirely match the target distribution.

To overcome these issues, we introduce a new formulation for learning semantic editing in the latent space, leading to a core solution that *does not* rely on classifiers.

From a global perspective, latent editing can be viewed as an optimal transport problem [10]. Given a distribution of latent codes sharing some semantics, we propose to transport it onto the distribution of latent codes that share the same semantics except for the attribute to be edited. Since the resulting images should not exhibit any other changes than the desired one, the initial points should be transported "close" to points sharing their semantics; that is, the transport should be optimal w.r.t. a cost representing the perceptual similarity. To achieve this, we learn transformations in latent space using the guidance of the Wasserstein loss with an Euclidean cost, which can be combined with a Wasserstein loss with a cost computed in the attribute space to enforce disentanglement.

We apply our method in the latent space of StyleGAN2 to modify the number of digits and edit facial attributes. We compare quantitatively and qualitatively to the method of Yao *et al*. (LT) [6] that relies exclusively on a latent classifier. Without additional regularization, our method leads to realistic

edited images and achieves on-par disentanglement and identity preservation than a classifier-based method.

2. RELATED WORK

Early works on GANs have demonstrated that their latent space contains rich semantics that can be leveraged to control some properties of the generated data. Simply translating a latent code in a given direction can lead to the variation of a semantic attribute in the corresponding generated image [1, 2, 3, 12]. Latent semantic directions can be extracted from the latent space without supervision by performing PCA [2] or by singular value decomposition on the weights of the pretrained GAN [3, 12]. Supervised methods often employ classifiers to extract the directions. InterfaceGAN [1] introduces a framework to edit binary facial attributes. An SVM is trained in latent space to infer the hyperplane that best separates the positive vs. negative latent codes w.r.t. a semantic attribute. The vector orthogonal to the hyperplane then constitutes the editing direction. Later works aim at learning a direction specific to each latent code by passing the input code through an MLP or an affine layer that is trained with the guidance of a classifier. GuidedStyle [5] uses an attribute image classifier that classifies the images corresponding to the edited latent codes. The editing is correct if the classifier's predictions correspond to the desired change. Yao et al. [6] employ a similar objective but use a classifier trained directly in latent space. However, classifiers are unreliable [9, 7], potentially leading to images or latent codes that minimize the objective but do not correspond to the desired editing. Different from previous works, our core method does not rely on classifiers. Instead, we solve the problem using the optimal transport framework. To the best of our knowledge, this is the first work applying optimal transport for latent space editing.

3. WASSERSTEIN LOSS FOR GAN EDITING

Let G be a pretrained generator and $\mathcal{Z}$ its latent space such that $\mathbf{I} = G(\mathbf{z})$ where $\mathbf{z} \in \mathcal{Z}$ is a latent code and $\mathbf{I}$ the corresponding generated image. Suppose we have a collection of latent codes $\{\mathbf{z}^{(i)}\}_{i=1}^{N}$, where each code is associated with a set of binary semantic attributes $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, .., \mathbf{a}_K\} \in \{0, 1\}$. For a given attribute $\mathbf{a}_k$, we aim to learn an affine transform $\mathbf{H}_k$ in $\mathcal{Z}$,

$$\mathbf{z}'_k = \mathbf{z}_k + \alpha \cdot \mathbf{H}_k(\mathbf{z}),\ \alpha \in \mathbb{R} \tag{1}$$

such that only the attribute intensity $\mathbf{a}_k$ differs in the resulting image $\mathbf{I}' = G(\mathbf{z}')$, where α controls the strength of the change.

Let μ_s^k be the distribution of latent codes $\mathbf{z}_k$ that are negative with respect to the binary attribute $\mathbf{a}_k$ and μ_t^k the distribution of latent codes $\bar{\mathbf{z}}_k$ positive w.r.t. the attribute $\mathbf{a}_k$. To increase the intensity of the attribute $\mathbf{a}_k$ in the generated images, $\mathbf{H}_k$ should transport the distribution of edited latent codes $\mathbf{z}'_k$ denoted by $\mu_s'^k$ close to the distribution μ_t^k. However, the information encoding other attributes or properties should remain unchanged. The theory of optimal transport [10] introduces a framework to transport a distribution to another with a minimal cost. The Wasserstein distance between two distributions represents the minimal value of this cost. Thus, we propose to use this loss as supervision to learn $\mathbf{H}_k$ with a cost in latent space expressing similarity in image space. We call this model Latent Wasserstein (LW).

3.1. Wasserstein Distance

Let us define two discrete distributions:

$$\mu_s = \sum_{i=1}^{n_s} a_i \delta(x_i) \text{ and } \mu_t = \sum_{i=1}^{n_t} b_i \delta(y_i) \tag{2}$$

where $\delta(,)$ is the Dirac function and a_i, b_i the probability mass associated with each sample.
The Wasserstein distance between μ_s and μ_t is defined as:

$$\begin{aligned} W(\mu_s, \mu_t) &= \min \sum_{i,j} T_{i,j} c_{i,j} \\ \text{s.t. } T\mathbf{1}_{n_t} &= \mu_s, T^\top \mathbf{1}_{n_s} = \mu_t \end{aligned} \tag{3}$$

T is the transport matrix. $T_{i,j}$ represents how much probability mass must be transported from point x_i to point y_i and $c_{i,j}$ the cost of this transport. Estimating the Wasserstein distance is challenging in practice as it requires to solve the underlying optimal transport. The Wasserstein distance is usually estimated with the Sinkhorn divergence built on entropic regularization with debiasing terms [13, 14].

3.2. Core Method

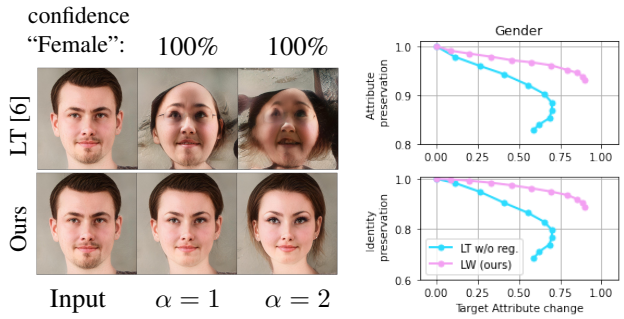
Our main objective is to minimize the Wasserstein loss between $\mu_s'^k$ and μ_t^k with a squared Euclidean cost function:

$$\begin{aligned} \mathcal{L}_{\text{edit}} = W\left(\mu_s'^k, \mu_t^k\right), \quad x_i = \mathbf{z}'^{(i)}_k \text{ and } y_j = \bar{\mathbf{z}}^{(j)}_k \\ c_{i,j} = \frac{1}{2}\|x_i - y_j\|^2 \end{aligned} \tag{4}$$

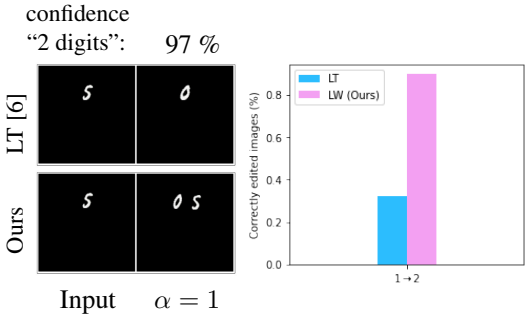
In Eq. (2), the probability mass of each sample is usually set uniformly across samples, *i.e.* $a_i = \frac{1}{n_s}$ and $b_i = \frac{1}{n_t}$ for all i. If there are biases in the collection of training latent codes, the representation of semantically similar samples may vary significantly between the $\mu_s'^k$ and μ_t^k [15]. In this case, we propose to weight the source samples according to the number of similar samples in the target distribution. More formally, we set $a_i = \frac{1}{n_t^A \times n_s^A}$ where n^A is the number of latent codes with the attribute combination A for a set of selected attributes.

3.3. Enforced Disentanglement

To ensure that the transported latent codes share the same attributes as the initial ones, we propose to minimize the Wasserstein loss between $\mu_s'^k$ and μ_s^k:



(a) FFHQ [11], 'Male' $\rightarrow$ 'Female'



(b) MultiMNIST [8], '1 digit' $\rightarrow$ '2 digits'

Fig. 2: **Failure cases of a classifier-based method.** LT [6] learns edits in latent space under the guidance of a latent classifier. (a) On FFHQ: without $L2$-regularization on the edited codes, the edited images are unrealistic (as shown in the qualitative result on the left) before reaching the desired editing. The classifier leads to out-of-distribution regions as it allocates high confidence to regions larger than that of the training samples [7]. The quantitative analysis on attribute and identity preservation shows highly degraded results. (b) On MultiMNIST: the edited images remain unchanged (no digit is being added) while the classifier indicates the opposite (predicts 2 digits with high confidence). The classifier leads to regions close to the decision boundaries where there are adversarial samples. The quantitative analysis shows that only 32% of images are correctly edited.

$$\mathcal{L}_{\text{edit}} = W\left(\mu_s'^k, \mu_s^k\right), \quad x_i = \mathbf{z}'^{(i)}_k \text{ and } y_j = \mathbf{z}^{(j)}_k$$
$$c_{i,j} = \frac{1}{2}\sum_{l \neq k}(1 - \gamma_{lk})\|\mathbf{F}_l(x_i) - \mathbf{F}_l(y_j)\|_2^2 \quad (5)$$

In contrast to the previous objective, we employ a cost computed in the *attribute* space. We follow the cost defined in [6], where $\mathbf{F}_l$ is a latent classifier trained to predict $\mathbf{a}_l$ given a latent code $\mathbf{z}$. The term γ_{lk} represents the absolute correlation between attribute $\mathbf{a}_l$ and $\mathbf{a}_k$ and is used to avoid disentangling naturally correlated attributes (*e.g.* "Smile" and "High Cheekbones"). Although we use the cost introduced in [6], our constraint is a more relaxed constraint since we operate on the distribution.

The final objective to minimize is then $\mathcal{L} = \mathcal{L}_{\text{edit}} + \lambda\mathcal{L}_{\text{pres}}$ where λ allows to balance the two losses.

4. EXPERIMENTS

4.1. Implementation Details

We present two editing applications: facial attributes on FFHQ/CelebAHQ and number of digits on MultiMNIST[8], consisting of images with 1 to 4 MNIST digits. We apply the editing in the latent space of StyleGAN2 [11] pretrained on FFHQ resp. MultiMNIST. For the training data, we employ latent codes corresponding to real images previously projected in latent space using the pSp encoder [4], that projects the images into the $\mathcal{W}+$ latent space. We employ respectively the $30K$ labeled 1024×1024 CelebAHQ images [16] for face editing and $25K$ 128×128 MultiMNIST images. To learn a transformation, we use the implementation of the Wasserstein loss provided by the GeomLoss [13] library. We set the batch size as the minimum between the number of samples in the source and target distributions, and drop the last batch if it causes a strong imbalance between both. We use Adam optimizer with a learning rate of 0.001. To avoid overfitting the target distribution, we perform early stopping on a hold-out validation set. As CelebAHQ contains various biases, we weight the samples and use the disentanglement loss. Optimal value for λ is 1 for all considered attributes except for "Glasses" ($\lambda = 15$). The cost is computed on all 40 attributes of CelebA [17]. Samples are weighted based on the most common attributes.

4.2. Metrics

We use three metrics [6] to evaluate the different methods. The *target attribute change* rate indicates the percentage of images for which the target attribute is indeed modified. The *attribute preservation* rate corresponds to the average number of attributes, apart from the target attribute, that are preserved. The aforementioned metrics are computed by running pretrained attribute image predictors before and after the editing (for a given α) and finding which attributes have changed. An attribute is considered present if the probability is greater than 0.5. We also compute the *identity preservation* rate as the average of the cosine similarities between ArcFace [18] features of input and edited images. All metrics are evaluated on $1,000$ images from FFHQ. The attribute and identity preservation rates are reported against the target change for 10 values of $\alpha \in [1 \cdot d, 2 \cdot d]$ where d is chosen such that the target change

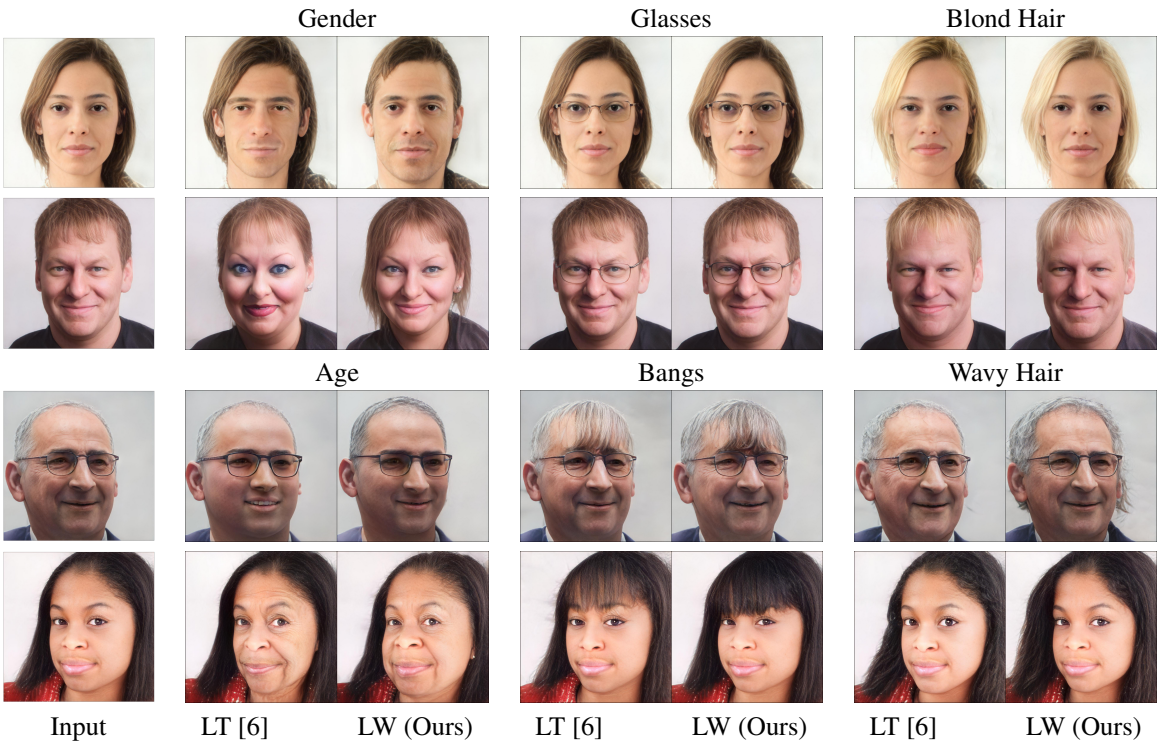


Fig. 3: **Qualitative results for facial attribute editing**. We report the editing results for $\alpha = \pm 2$. We observe that our approach better preserves identity and some facial attributes (*e.g.* expression, absence of makeup) compared to LT.

for a given α is comparable between the different methods. In tables, we report the mean over all values of α.

4.3. Facial Attribute Editing

We present a quantitative and qualitative comparison with Latent Transformer (LT) [6] that relies on the guidance of a latent classifier. In addition to the classification objective, the authors introduce a disentanglement loss and an $L2$-regularization on the norm of the transformation. The latter is used to enforce identity preservation but is also critical to obtain latent codes corresponding to realistic images. The comparison is conducted on common attributes ("Glasses", "Gender", "Smile", "Age") and rarer attributes chosen based on their representation and the performances of the image classifiers ("Pale Skin", "Bangs", "Blond Hair", "Wavy Hair"). Quantitative results from Fig. 4 show that our results are on par with LT with occasionally slightly lower attribute preservation ("Gender") but generally higher identity preservation ("Gender", "Age", "Blond Hair"). Note that this is surprising since we do not explicitly enforce identity preservation. Qualitative results in Fig. 3 showcase some advantages of our method. Nose, lips and eyes shape are much better preserved for "Gender" and "Age". LT also produces "cartoonish" edits for these attributes while ours remains naturalistic. LT 'Gender' editing is also heavily entangled with 'Makeup' while LW adds nearly none. We provide additional qualitative results in the supplementary.

Table 1: Quantitative results for the attributes "Gender" (G), "Age" (A) and "Pale Skin" (PS). We compare the classifier loss approach (LT) with our Wasserstein loss approach (LW). Setting (*) is the "core" method, w/o any regularization.

Method	Attr. preservation			Id. preservation		
	G	A	PS	G	A	PS
LT$^{(*)}$	0.92	0.95	0.86	**0.94**	0.96	0.96
LW$^{(*)}$	**0.95**	**0.96**	**0.96**	**0.94**	**0.97**	**0.98**
LT	**0.98**	**0.98**	**0.98**	0.95	0.96	**0.98**
LW	0.97	**0.98**	0.97	**0.96**	**0.97**	**0.98**

Classifier vs. Wasserstein. We evaluate the ability of both methods to achieve disentangled and identity preserving editing without any explicit constraint. We denote by LT$^{(*)}$ the latent transformer of Yao *et al*. [6] trained without the disentanglement loss nor the $L2$-regularization. In Table 1 (top), we compare it to our model trained without the disentanglement loss ($\lambda = 0$), denoted by LW$^{(*)}$. The Wasserstein baseline outperforms the classifier baseline both regarding disentanglement and identity preservation. As shown in the qualitative results presented in Fig. 5, the latter produces highly entangled edits (*e.g.* with the attribute "Smile") and alters the identity. Without enforcing it explicitly, the Wasserstein approach already exhibits a good disentanglement ability and the identity

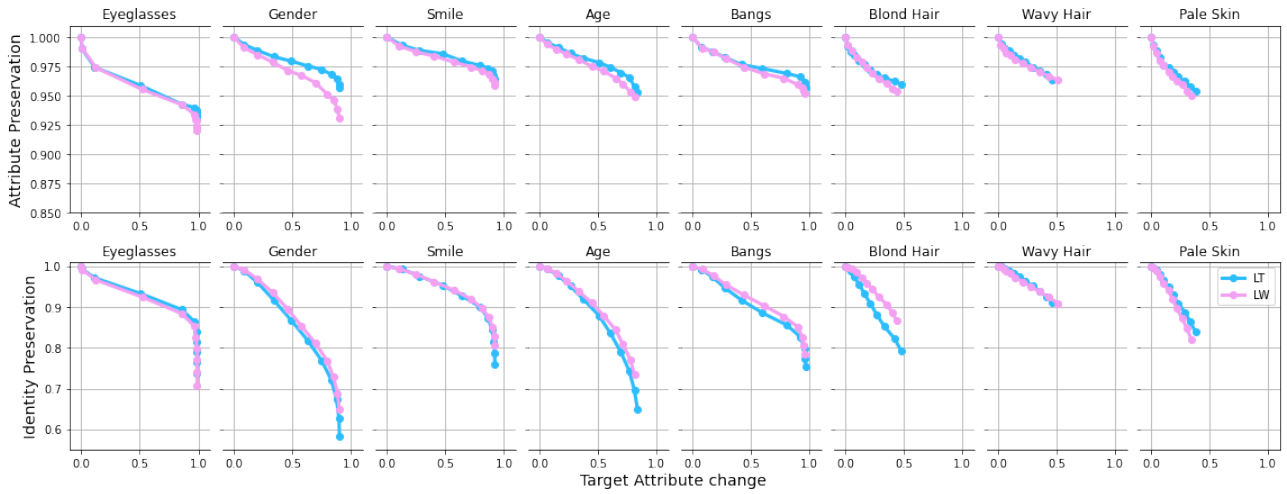


Fig. 4: **Quantitative results for facial attribute editing**. We report the attribute preservation rate (computed on all other attributes indicated here) and the identity preservation rate for different values of α (points of the curves). The x-axis is the ratio of images (among all test images) for which the target attribute is successfully flipped.

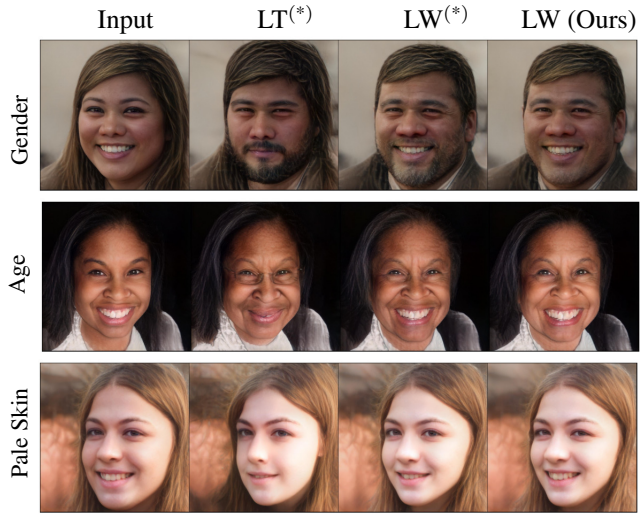


Fig. 5: Qualitative comparison between classifier-based edits $\left(\text{LT}^{(*)}\right)$ and our Wasserstein-based approach without any constraint $\left(\text{LW}^{(*)}\right)$ vs with the disentanglement constraint (LW).

is also well-preserved. These abilities can be explained as the Euclidean cost in employed in Eq. (4) fairly reflects the perceptual distance in image space.

Disentanglement Constraint. We study the influence of adding the disentanglement constraint from Eq. (5). As shown in Table 1, we improve attribute preservation. Qualitatively, the results are also improved as shown in Fig. 5. "Gender" is no longer heavily entangled with "Beard" (1st row) and the slight entanglement with "Smile" is removed. As shown in Fig. 2 (left), when the disentanglement constraint is used in the classifier-based approach, the edited images are unrealistic. The attribute and identity preservation curves show atypical behavior as image classifiers are disrupted by such images. As the decision boundaries of classifiers cover areas that are larger than the area of training samples, latent codes which are far away from the training distribution can still minimize the classification objective. The $L2$-regularization in [6] enforcing that the edited latent codes remain close to the initial ones is thus necessary to circumvent this limitation. Our method does not require any regularization to produce realistic edits, since our main objective enforces closeness to the target distribution.

4.4. Editing the Number of Objects

The $L2$-regularization in conjunction with the classification objective is similar to the formulation employed to produce adversarial examples [9]. While this rarely occurs on faces, this plagues editing performance on MultiMNIST. The quantitative results in Table 2 show that for a target change of 100% according to the latent classifier, the image classifier indicates significantly lower target changes for LT. In other words, the latent classifier predicts that the number of digits has increased while it has stayed the same in the image, undermining the goal of an editing method. In contrast, our method has a high editing effect and actually adds digits in the edited images. Qualitative results are provided in Fig. 6.

5. CONCLUSION

We present a new method to learn semantic editing in the latent space of GANs, that proposes to model the problem as an optimal transport problem. We look for transformations that transport a collection of latent codes to the most semantically similar points in the distribution of latent codes with the desired semantic. We use the squared Euclidean distance in

Table 2: Quantitative results for the manipulations "adding one digit in an image containing n digits, for $n = 1, 2, 3$" in real images from MultiMNIST [8]. Given a target change rate of 100% according to a latent classifier, we report the *actual* change rate as measured by an image classifier. Higher values indicate a lower rate of adversarial samples.

Method	Target change rate		
	1→2	2→3	3→ 4
LT	0.32	0.31	0.64
LW (ours)	**0.90**	**0.95**	**0.99**

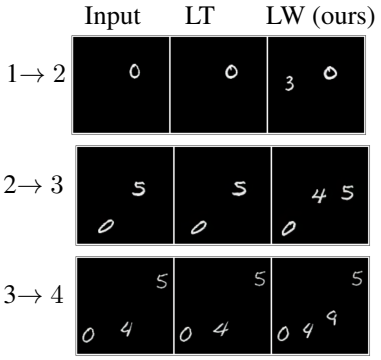


Fig. 6: Qualitative results for number of objects editing. Our method adds a digit while LT fails to add one.

latent space as a cost function as it fairly reflects the perceptual distances in image space. This formulation readily produces almost totally disentangled editing whereas classifier-based methods require an explicit disentanglement constraint. To achieve even more disentangled editing, we introduce an explicit loss enforcing the transported codes to remain close to the distribution of initial codes. This loss is also formulated with optimal transport but using a semantic cost computed in *attribute* space. On the task of facial attribute editing on CelebA/FFHQ, our method is competitive with a state-of-the-art classifier-based method without requiring an additional constraint to ensure that the obtained images are realistic. Our method also alleviates other issues from using classifiers, such as the sensitivity to adversarial examples as we illustrate on the editing of the number of digits in MultiMNIST images.

Our method achieves particularly strong identity preservation performances when editing facial attributes. This is unexpected as there is no explicit constraint to do so, and the train and target distributions contain different identities. We attribute this ability to a combination of early stopping, that prevents us from overfitting our edited codes on the target distribution, and of the inductive bias of the model, which defines edits as simple affine transformations in the latent space, acting as a regularization.

While the Wasserstein loss based on the latent Euclidean distance results in state-of-the-art editing performances, it does not perfectly reflect the perceptual distance in image space. This could explain why some edits are not totally disentangled. As an extension of this work, we believe that performances could be further improved by using a cost based on the perceptual LPIPS metric [19] or an equivalent proxy learned in the latent space to reduce computation time.

References

[1] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou, "InterFaceGAN: Interpreting the disentangled face representation learned by GANs," *TPAMI*, 2020.

[2] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris, "GANSpace: Discovering interpretable GAN controls," in *NeurIPS*, 2020.

[3] Yujun Shen and Bolei Zhou, "Closed-form factorization of latent semantics in GANs," in *CVPR*, 2021.

[4] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or, "Encoding in style: a StyleGAN encoder for image-to-image translation," in *CVPR*, 2021.

[5] Xianxu Hou, Xiaokang Zhang, Hanbang Liang, Linlin Shen, Zhihui Lai, and Jun Wan, "Guidedstyle: Attribute knowledge guided style manipulation for semantic face editing," *Neural Networks*, vol. 145, pp. 209–220, 2022.

[6] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier, "A latent transformer for disentangled face editing in images and videos," in *ICCV*, 2021.

[7] Anh Nguyen, Jason Yosinski, and Jeff Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *CVPR*, 2015.

[8] Shao-Hua Sun, "Multi-digit MNIST for few-shot learning," GitHub repository, 2019.

[9] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[10] Cédric Villani, *Optimal Transport: Old and New*, Springer Berlin Heidelberg, 2008.

[11] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila, "Analyzing and improving the image quality of StyleGAN," in *CVPR*, 2020.

[12] Nurit Spingarn, Ron Banner, and Tomer Michaeli, "GAN "steerability" without optimization," in *ICLR*, 2021.

[13] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré, "Interpolating between optimal transport and MMD using sinkhorn divergences," in *AISTATS*, 2019.

[14] Marco Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *NeurIPS*, 2013.

[15] Perla Doubinsky, Nicolas Audebert, Michel Crucianu, and Hervé Le Borgne, "Multi-attribute balanced sam-

pling for disentangled GAN controls," *Pattern Recognition Letters*, vol. 162, pp. 56–62, 2022.

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *ICLR*, 2018.

[17] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015.

[18] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[19] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.