ChunkyGAN: Real Image Inversion via Segments

Adéla Šubrtová*[1], David Futschik*[1], Jan Čech[1],
Michal Lukáč[2], Eli Shechtman[2], and Daniel Sýkora[1]

[1] Czech Technical University in Prague,
Faculty of Electrical Engineering, Czech Republic
{subrtade,futscdav,cechj,sykorad}@fel.cvut.cz
[2] Adobe Research, USA
{lukac,elishe}@adobe.com

Abstract. We present ChunkyGAN—a novel paradigm for modeling and editing images using generative adversarial networks. Unlike previous techniques seeking a global latent representation of the input image, our approach subdivides the input image into a set of smaller components (chunks) specified either manually or automatically using a pre-trained segmentation network. For each chunk, the latent code of a generative network is estimated locally with greater accuracy thanks to a smaller number of constraints. Moreover, during the optimization of latent codes, segmentation can further be refined to improve matching quality. This process enables high-quality projection of the original image with spatial disentanglement that previous methods would find challenging to achieve. To demonstrate the advantage of our approach, we evaluated it quantitatively and also qualitatively in various image editing scenarios that benefit from the higher reconstruction quality and local nature of the approach. Our method is flexible enough to manipulate even out-of-domain images that would be hard to reconstruct using global techniques.

Keywords: StyleGAN, image inversion, segmentation, latent editing

1 Introduction

The increasing ability of GANs to generate images virtually indistinguishable from real photographs [14,12], has created a new paradigm for image editing. In this paradigm, one first estimates a latent code for the network that best reconstructs the input image [13,23], and then manipulates this latent code in specific ways to create particular variations of the input image. With a knowledge of which directions in latent space of a particular generator encode which properties of the output image, it is possible to perform high-level semantic editing of the appearance of the input photo while retaining the original visual features, e.g., adding more hair to a bald person while retaining their identity [22,18].

*joint first authors

Fig. 1. Real image manipulation examples created interactively using our method. The left-most images are the original photographs, the remaining columns show following edits: changing gaze direction, opening mouth, growing a beard and aging. *Source images: Shutterstock*

Due to the nature of adversarial training, a well-trained generator transforms any latent code drawn from the trained distribution into a plausible output, but mapping of an arbitrary in-domain image to a latent code might be difficult or even not possible. Existing methods address this by instead projecting into deeper spaces which makes accurate reconstruction easier, but weakens the original guarantee that every code maps to a plausible output, meaning that manipulated results may be out of domain and visually appear broken. This means there is an inherent trade-off between ease and accuracy of reconstruction, and quality of edited outputs [22], and existing methods perform on the spectrum of this trade-off. For example in StyleGAN2 [14], the original input code $z \in \mathbb{R}^{512}$ is transformed into a latent vector $\mathcal{W} \in \mathbb{R}^{512}$ which is easy to edit but difficult to reconstruct, whereas Abdal et al. [1] use $\mathcal{W}^+ \in \mathbb{R}^{18\times512}$ that has enough degrees of freedom to provide good reconstruction, but is more difficult to manipulate.

This issue becomes much more apparent when we examine examples that are in-domain, but far from typical. For example in the case of StyleGAN trained on a dataset of faces, we may consider human faces with unique features or accessories that do not appear in training datasets such as CelebA [15] or FFHQ [13], such as bindis, unusual glasses, heavy occlusions, etc. In these cases even techniques that have greater flexibility such as $\mathcal{S}$-space [23] usually fail.

The source of much of these difficulties are two underlying assumptions: that there exists a single latent code that exactly or almost exactly reconstructs the target image, and that the manifold of representative images is nearly convex with respect to finding such a latent code. But because the number of output pixels is much higher than the number of degrees of freedom in the latent space,

we may view the reconstruction problem as overdetermined, and although the aggregated reconstruction loss has local minima that can be found, a minimum for the entire image is not necessarily a minimum for all its regions. In practice, this means that the code retrieval problem is difficult and the solutions we arrive at are in effect suboptimal. In this paper we propose to resolve this difficulty by relaxing exactly these assumptions. We search not for a single latent code to represent the entire image, but rather a vector of latent codes, each corresponding to a segment of the image, such that when assembled they resemble the original image as closely as possible (see Fig. 2).

Since each latent code is then estimated for a much lower dimensional target, each of the regional subproblems become less overdetermined, which makes for an easier optimization problem. This in turn means that we can achieve much lower total error and thus more accurate reconstruction of the original. Besides superior accuracy and greater ability to generalize to the out-of-domain features, the segment-based nature of our method also allows for strictly localized edits, either based on segmentation generated automatically as a by-product of our method, or based on user-specified segments. Thanks to that property, visual content in different segments remains intact and thus helps retain the fidelity of the original photo. This leads to an interesting novel interactive scenario where the user adaptively applies individual local modifications in sequence to achieve a desired output that would normally be difficult to obtain using global manipulation techniques (see examples in Fig. 1). We demonstrate the power of our approach in various use cases that would be difficult to achieve using current state of the art. Moreover, a great advantage of our approach is that it does not replace previous methods but rather serves as a complementary part that, when plugged in, enables even better results than those produced by the technique applied in isolation.

2 Related Work

State-of-the-art approaches to finding suitable latent codes for the input image can be broadly split into two major categories: direct optimization and encoder-based techniques.

The first category takes into account the fact that the generator network is differentiable function on its own and thus gradient descent can be used to move from a real image into its latent code [17,10,11,24]. This typically leads to an inversion which is close to the original, however, since constraining the optimization to search across the manifold of naturally looking latent codes is nontrivial, the resulting projection is usually difficult to manipulate.

The other category relies on training an encoder which predicts the specific latent code given an image, using generated samples as training data [29,5]. Tov et al. [22] show that the encoder can learn to embed the real image into the natural manifold much closer than optimization methods, it does, however, often come at the cost of overall reconstruction quality, even considering multi-pass iterative techniques [3] or a modulation of StyleGAN weights [4,7].

Both of these approaches, therefore, are characterized by an important trade-off between faithfulness to the original image and the ability to perform editing operations on the projected latent code. Hybrid approach has also been proposed, such as the one by Zhu et al. [28], in which the direct optimization method is initialized by latent code proposed by a trained encoder, striking a better balance on the trade-off chart, however, the final result is far from ideal in either axis.

The trade-off itself is also not one dimensional. As the representation of the latent code turns into the final image via operations inside the generator network, it becomes easier to invert images into intermediate representations, at the cost of increased dimensionality, making editing more difficult. Recent work [30,25,11] tries to exploit this knowledge by imposing constraints like segmentation on relatively high-level, spatial representations, leading to solutions that can create high-quality inversions at the cost of restricting the set of possible edits.

Ling et al. [16] presented EditGAN that enables to edit images by altering their segmentation masks. In contrast to our technique EditGAN can only change shape and relative position of selected regions. There is no control over the content generated inside the edited area, and it is also challenging to perform global edits. Moreover, EditGAN uses only a single latent code with lower expressive power while relying on a pre-trained DatasetGAN model [27] that jointly generates images and their corresponding semantic segmentations. In our approach, each region have its own latent code, can be added on the fly at arbitrary locations and subsequently edited.

In StyleFlow, Abdal et al. [2] use continuous normalizing flows in the latent space that are conditioned by various attribute features. This enables edit disentanglement comparable to our approach that is, however, redeemed by lower reconstruction quality. Moreover, StyleFlow also requires pre-trained classifiers to find the disentangled attributes along which the edits are performed.

Roich et al. [21] propose that it is possible to fine-tune the generator network itself to improve the reconstruction quality while retaining the editability offered by a natural latent code. While their technique provides a well-rounded solution to both inversion accuracy and latent code editability, it requires fitting and storing per-image generator network, making it more resource-intensive and less suitable for downstream tasks.

In the earlier version of our method [9], segmentation-based inversion was developed for user-assisted local editing. In this extended version, we introduce joint optimization framework that enables automatic projection of the entire image while refining the shape of individual segments.

3 Our Approach

Our method accepts a real image I and reconstructs it as a vector of segmentation masks $S = \{S_i\}_{i=1}^{n}$, where pixel values range continuously from 0 to represent fully outside and 1 fully inside, and a vector of corresponding per-segment latent codes $X^I = \{X_i^I\}_{i=1}^{n}$. The masks are constrained so that they per-pixel sum up to 1. These latent codes are interpreted as images using a shared image

generator G^I and the output image is obtained by pixelwise linear blending, visualised in Fig. 2:

$$O(X^I, S) = \sum_{i=1}^{n} G^I(X_i^I) \cdot S_i. \tag{1}$$

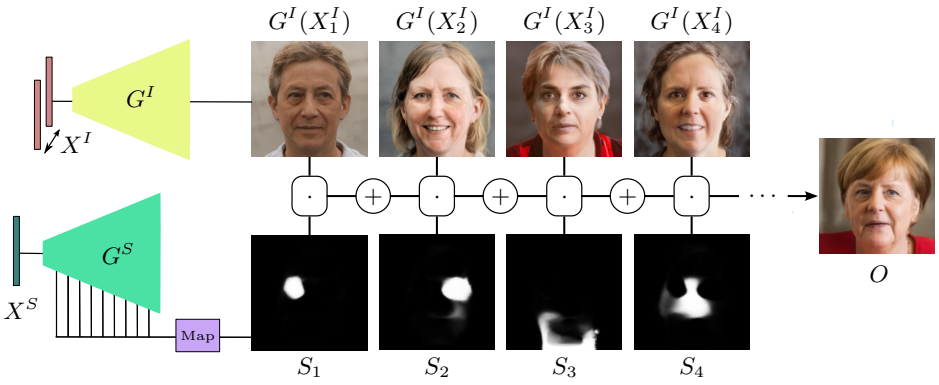

Fig. 2. ChunkyGAN flowchart—the output image O computed as a weighted combination of n images generated by a network G^I given a set of n latent codes X^I. Weights are specified by a set of n segmentation masks S that can be specified manually or generated automatically by a segmentation network G^S using a latent code X^S. *Source image: Raimond Spekking / CC BY-SA 4.0 (via Wikimedia Commons)*

This expression is trivially differentiable with respect to both S and X, and is optimized with respect to some dissimilarity measure between I and the composite O just like in a single-segment reconstruction scenario. Unless otherwise specified, in this paper we optimise with respect to the perceptual loss $\mathcal{L}_{\text{LPIPS}}$ of Zhang et al. [26].

Because the semantic segmentation is not universal and can vary dramatically between individual faces, it is necessary to optimize the masks as well. Optimizing them on a per-pixel basis would be memory intensive and would not take advantage of the domain knowledge we have for the problem. Therefore, we use a mask generator G^S to generate them from a segment latent code X^S, i.e., $S_i = G^S(X^S)_i$. In this work, we use a segment generator network based on DatasetGAN [27]. It consists of StyleGAN2 generator and a mapping network trained on a modest dataset (a few tens of images) of randomly generated StyleGAN2 images annotated by example based synthesis [8], using a single manually annotated image as exemplar.

To this end, the canonical form of our optimization problem is as follows:

$$\min_{X^S, X^I} \mathcal{L}_{\text{LPIPS}}\big(I, \sum_{i=1}^{n} G^I(X_i^I) \cdot G^S(X^S)_i\big) + \lambda_{reg} \sum_{i=1}^{n} \|X_i^I - X_\mu^I\|_2^2, \tag{2}$$

where the first term measures reconstruction loss and the second term penalizes dispersion among the latent codes, measured as sum of squared deviations from the mean code X^I_μ. Such regularization helps avoid mutually distant latent codes that do not produce realistic images. This is not typically a problem in the projection step, but during manipulation distant codes may diverge in appearance more quickly. This is caused by limitations in visual coherence in the pre-trained editing directions.

Our approach is orthogonal to the choice of the latent space of the X codes. In general it can be any combination of common latent spaces that allows compact encoding of the input image. In the case of StyleGAN [13,14], we consider $\mathcal{W}$, $\mathcal{W}^+$ [1], and $\mathcal{S}$-space [23], however, any previously published, potentially newly developed or a mixture of methods can be used. In fact, our method is a complementary extension that could help achieve better results regardless of the selected projection method.

In Fig. 3, we show an example of the optimization (per Equation 2) progression, starting from mean latent codes until convergence. Note that the segments tend to align with semantic facial features.

The processing speed of the optimization process relies on the number of segments and the number of optimization steps. When a joint multi-segment optimization with the DatasetGAN is performed the projection can take several minutes. However, during the interactive editing (as seen in our supplementary video), where segments are specified by the user one-by-one, the method runs at interactive rates on the GPU (a few seconds).

Fig. 3. Progression of the optimization. Images and color-coded segmentation maps for iterations 1, 5, 9, 15, 23, 37, 500. *Source image: Adobe Stock*

4 Evaluation

To validate our approach we performed two quantitatively and qualitatively evaluated experiments. In the first experiment we validate whether the projections produced by our method can reproduce target photos with greater fidelity when compared to standard projection techniques. In the second experiment

we demonstrate the ability of our approach to edit projected images by manipulating estimated latent codes and compare the fidelity of the resulting edits with standard techniques. Finally, we compare our approach with current optimization-based and encoder-based projection techniques.

4.1 Fidelity of projected images

To quantitatively evaluate fidelity of projected images we took the first 100 images from CelebA dataset [15] excluding blurred images and those with people wearing additional props such as hats or glasses. We then projected all those images globally into $\mathcal{W}$, $\mathcal{W}^+$, $\mathcal{S}$-space, and also locally using our method. When using $\mathcal{W}^+$, we show both cases, with ($\lambda_{reg} = 1$) and without ($\lambda_{reg} = 0$) the regularization. For all projections we measured the LPIPS, identity (measured as cosine distance between ArcFace descriptors [6]), and L_2 loss with respect to the original target photos.

Projection	LPIPS	Identity	L_2
$\mathcal{W}$	0.4190 ± 0.0363	0.1745 ± 0.1328	0.0725 ± 0.0699
Ours in $\mathcal{W}$	0.3697 ± 0.0396	0.1384 ± 0.1117	0.0481 ± 0.0289
$\mathcal{W}^+$	0.3675 ± 0.0387	0.1195 ± 0.1047	0.0436 ± 0.0623
Ours in $\mathcal{W}^+$	**0.3194 ± 0.0365**	0.0937 ± 0.0855	**0.0207 ± 0.0151**
Ours in $\mathcal{W}^+$ reg.	0.3330 ± 0.0350	**0.0894 ± 0.074**	0.0217 ± 0.0130
$\mathcal{S}$	0.3577 ± 0.0397	0.1070 ± 0.0965	0.0328 ± 0.0188
Ours in $\mathcal{S}$	0.3572 ± 0.0401	0.1053 ± 0.0928	0.0319 ± 0.0187

Table 1. Projection fidelity. Losses were measured between the projected and the original image for each of the projection methods. Each cell reports the loss averaged over the CelebA subset along with the standard deviation. Our method significantly outperforms the baseline methods in all latent spaces for all losses.

The resulting numbers are shown in Table 1 which shows losses averaged over all 100 images with corresponding standard deviations. Those confirm that on average our method outperforms global projection methods significantly. This fact is visually apparent from scatter plots shown in Fig. 4 where each point corresponds to an image and its coordinates encode the LPIPS losses for the global and the segmented projection respectively. Red line depicts the margin where losses for both projection methods are equal.

Since the best projection is achieved by our method in $\mathcal{W}^+$, we select $\mathcal{W}^+$ as the default space for our method. The regularization slightly decreases the projection fidelity in terms of LPIPS, but improves the identity and editability, which is discussed in Sec. 4.2.

Because differences between the evaluated methods are difficult to observe in a typical case, we have for the purposes of qualitative evaluation of projection fidelity deliberately pre-selected a subset of hard-to-project images. Specifically, these were images that contain features uncommon in the standard datasets, e.g.

bindis, face masks, asymmetric glasses, or occluded faces. For those examples all compared methods were initialized equally (using mean latent vector) and the corresponding projection results are presented in Figure 5. It is apparent that thanks to greater flexibility of our approach, more realistic projections can be achieved when compared to standard techniques. Moreover, a workable inversion can be obtained even on out-of-domain images as shown in Fig. 5 (two bottom rows).

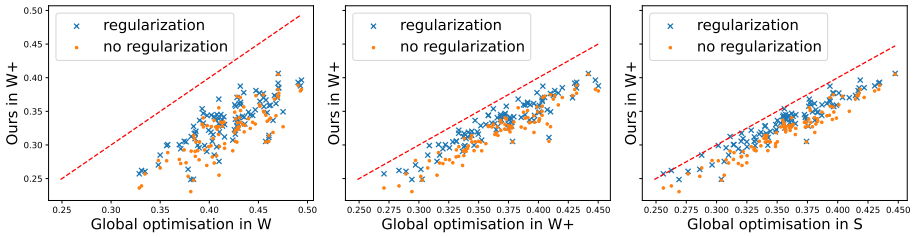

Fig. 4. Projection fidelity – scatter plots. Our method is compared with global projections ($\mathcal{W}, \mathcal{W}^+, \mathcal{S}$-space). X and Y axis represent the LPIPS loss between the original image and the image projected globally and projected by our method in $\mathcal{W}^+$ respectively. Each point corresponds to one image from the CelebA subset, in blue and in orange with and without the regularization respectively. The red line delineates the equal LPIPS losses. Our method improves projection for all images in all tested latent spaces. The regularization slightly decreases the projection fidelity, but remains still better than global methods.

4.2 Editability of projected images

Quantitative evaluation of editability was performed on the same set of CelebA images used for evaluation of projection fidelity. We pre-selected 4 semantic directions (gender, smile, age, and beard), changed all latent codes X in the same direction with the same magnitude, and finally measured the effect of the edits on identity.

Since the effect of unit strength manipulation along a pre-trained semantic direction can differ among latent spaces and the use of global/local projection, we calibrate the changes to make sure the effect on the manipulated image is equal. To do that we use an image classifier for each semantic direction. For each space and method, we measure image classifier responses while spanning the latent edit strength along a semantic direction for the entire dataset. We use linear regression to find the rate of change of the classifier response to the edit strength, and adjust the edit strength to be equal for all tested methods.

Table 2 shows a quantitative evaluation of the identity loss between the projected and edited images. It is apparent that the identity losses are the best for our method with the regularization engaged since regularization pushes the codes

Fig. 5. Qualitative assessment of projection fidelity on hard examples. All images were projected with regularization. For more examples refer to the supplementary material. *Source images: Adobe Stock*

	(a)				(b)			
	gender	smile	age	beard	gender	smile	age	beard
$\mathcal{W}$	0.169	0.022	0.07	0.279	0.249	0.18	0.191	0.328
$\mathcal{W}^+$	0.209	0.02	0.095	0.296	0.256	0.128	0.171	0.325
Ours in $\mathcal{W}^+$	0.298	0.049	0.151	0.312	0.325	0.125	0.203	0.333
Ours in $\mathcal{W}^+$ reg.	**0.126**	**0.018**	**0.069**	**0.091**	**0.169**	**0.099**	**0.129**	**0.144**

Table 2. Identity preservation during editing. Identity loss was computed between the projected and the edited images (a), and between the original and the edited images (b). Our method with regularization outperforms all other methods.

of all segment images towards latent areas where the linear latent manipulation works better. The results confirm that our method keeps the identity consistent during editing.

Regarding the reconstruction-editability trade-off [22], latent code regularization is essential in order to perform realistic edits. While our method without regularization achieves better results in projection fidelity it performs poorly during editing. By adding the regularization term, projection fidelity slightly deteriorates, but the identity preservation during edits improves by a large margin. The editability can be observed during the classifier-based calibration; methods without regularization need much stronger edits in order to achieve the same editing effect.

For the qualitative evaluation we pre-selected images and directions (age and yaw) that would cause difficulties to standard techniques, i.e., the identity is not well preserved during editing. During the yaw manipulation using our method the segmentation masks were edited as well (the segmentation latent code was manipulated automatically in the same way as the images) to adjust the segments geometrically. Results are presented in Fig. 6 and 7. It is clearly visible that our method keeps the identity better. Fig. 7, a man wearing a mask is especially challenging. The global techniques are unable to project the image properly. Our method projects the image faithfully and moreover, the global edits still work. Note that these results were achieved fully automatically, neither manual adjustment of the segmentation partitioning nor any post-processing were applied for images in Fig. 6 and 7.

4.3 Comparison with current state-of-the-art

To demonstrate how our approach compares to current state-of-the-art in the optimization-based and encoder-based techniques we performed various qualitative experiments seen in Figures 7 and 8. When compared to current best approaches based on optimization (Pivotal Tuning [21] and StyleFlow [2]), our method achieves better or comparable projection quality while still being able to deliver compelling edits (c.f. Fig. 7). Our method also outperforms encoder-based techniques (HyperStyle [4], ReStyle [3], pSp [20], and e4e [22]) with respect to the projection fidelity namely thanks to its ability to reproduce small details that are usually omitted by encoders (c.f. Fig. 8).

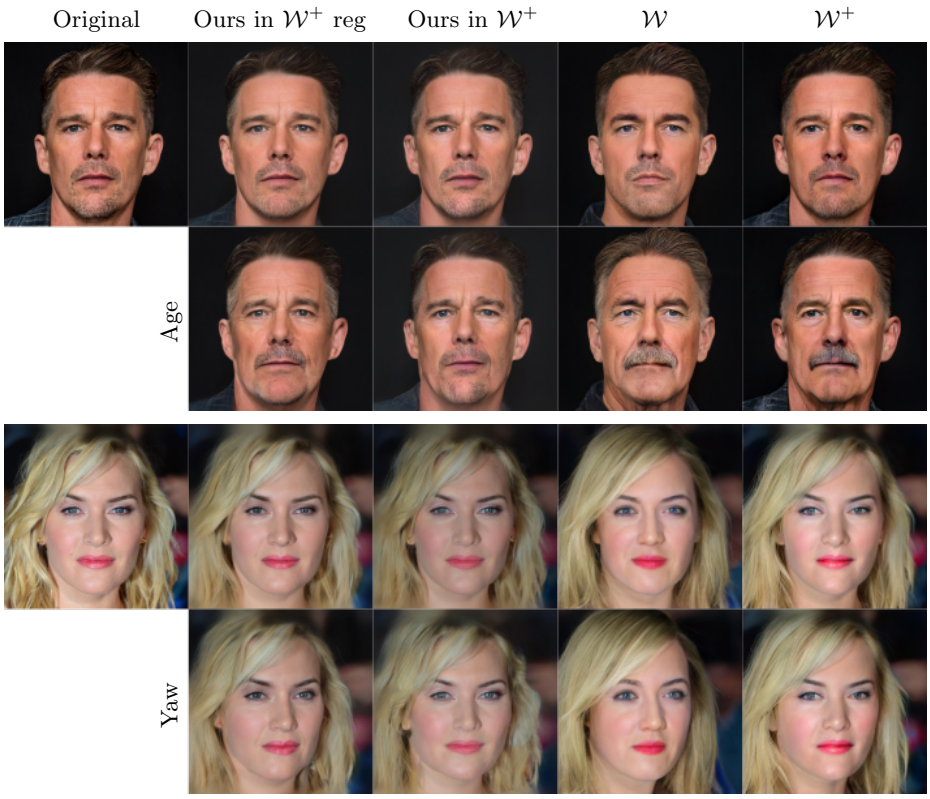

Fig. 6. Global edits with the same effective strength. For our methods the latent codes of all segments were manipulated equally. *Source images: Mingle Media TV (Kate Winslet), Neil Grabowsky / Montclair Film (Ethan Hawke)*

5 Applications

Aside from the fully automatic solution proposed in Section 3 our framework can also be extended to allow for interactive step-by-step manipulation in a few different ways. To facilitate this, we define the notion of a static mask S^X which defines an area of the image which is not changed during the optimisation. In terms of our objective function, this creates a mixed composite:

$$O(X^I, S, S^X, I) = S^X \cdot I + (\mathbf{1} - S^X) \cdot \sum_{i=1}^{n} G^I(X_i^I) \cdot S_i \tag{3}$$

In practice, for edits with small spatial extent it is often sufficient to reduce the number of segments being optimized to one, in which case there is no need to optimize S_i.

Using this static mask, instead of generating segment masks automatically, we allow the user to manually specify the region of interest. The user then runs the projection, edits the latent code, and produces an intermediate composite O which can then become a new I for next iteration. This user-driven iterative scheme is shown in Fig. 9. Such a workflow is intuitive for users as they can specify what they want to change, overview the resulting composition, and then possibly go back and revise their requirements by making additional changes in different regions.

Fig. 7. Challenging global edits. The first row depicts the original and the projected images using our approach with and without regularization, Pivotal Tuning [21], StyleFlow [2], $\mathcal{W}$ and $\mathcal{W}^+$ [1]. The remaining two rows show resulting global edits of age. *Source image: BlochWorld*

When making the composite O from edited image, even when edits of X are consistent, continuity around boundaries may no longer be guaranteed. Small discrepancies are suppressed automatically thanks to blending with soft masks. When the edit produces more notable global color shift we use Poisson image editing [19] to alleviate them. In most challenging scenario segment boundaries may start to interfere with newly synthesized salient features. In this case continuity can be enforced using a slightly modified version of our segmentation-based approach that will act as semantically meaningful hole-filling as illustrated in Fig. 10.

Suppose we have a photo of a person (Fig. 10a) and the aim is to add glasses. We select a loose region S_1 around eyes (Fig. 10b) and run the local projection to get latent code X_1 that reproduces the original image within S_1 (Fig. 10b). Then we manipulate X_1 to add glasses, however, as visible in Fig. 10c the shape of S_1 is insufficient to encompass newly added content. To fix this discrepancy we let the user specify correction mask S_2 with two connected components (Fig. 10d) and refine X_1 to obtain a new code X_2 that will match the content within S_2 (green region). From the image generated by X_2 we then use the dark part that lies inside S_2 to make the final composite (Fig. 10e). The X_2 code in fact generates a semantically meaningful hole-filling that completes the missing part of glasses.

6 Limitations

While the multi-segment reconstruction is remarkably robust, and segmented editing produces superior results for spatially limited edits, we can experience incoherence between segments for global edits (e.g. age, yaw) with high strength.

Fig. 8. Projection fidelity of our method with respect to the current state-of-the-art in encoder-based techniques: HyperStyle [4], ReStyle [3], pSp [20], and e4e [22]. *Source images: Ayush Kejriwal (bindi), BlochWorld (face mask)*

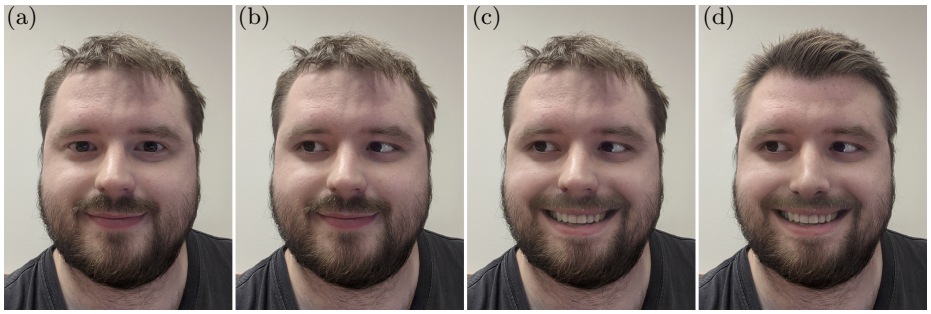

Fig. 9. Examples of local layered edits applied subsequently on a real photograph (a): changing gaze direction (b), adding smile (c), changing haircut and nose shape (d).

The reason for this is that the editing directions are local linear approximations of the property of interest on the latent manifold, and for higher edit strength this linearity assumption no longer applies. This issue is present also in single-code editing, where it may cause loss of identity which may be in some scenarios more tolerable. With multiple segments however, this is highlighted as a greater change resulting in individual segments to lose identity in different ways and therefore gives rise to incoherence. It only occurs in editing and not in reconstruction because in reconstruction the input image provides effective supervision to maintain coherence between segments.

The incoherence does not usually occur for easy-to-invert images and moderate edits, as seen in Fig. 6, but can be spotted in harder examples with a challenging global edit, as e.g., in Fig. 7 in Age+ of our method with regularization. Nevertheless, the small artifact on the mask shape, can be interactively removed by the hole-filling method demonstrated in Fig. 10.

As another option, this issue could be addressed by formulating and imposing an explicit segment coherence measure during editing, which can be done either

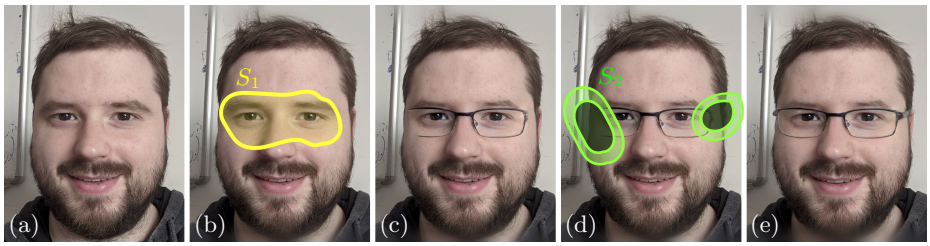

Fig. 10. Enforcing continuity of inconsistent edits—a photo of a person to which we would like to add glasses (a), user-specified segmentation mask S_1 with a projection X_1 matching the original image (b), manipulating X_1 generates glasses that do not fit the shape of S_1 (c), a new mask S_2 is marked encompassing two discontinuous parts (d), a composite with a projected region S_2 where the new latent code X_2 is refined from X_1 to produce the dark region inside S_2 (e).

locally, by measuring agreement between segments in their regions of overlap, or globally by e.g. an adversarial loss. Alternatively, instead of linear directions, one might train a separate model to explicitly encode a higher-order approximation of identity-preserving edit direction, which has the potential to also benefit vanilla methods under high edit strength.

7 Conclusion

We presented a new technique for image reconstruction and editing based on generative adversarial networks that subdivides the input image into a set of segments for which the corresponding latent vectors are retrieved separately. By so decomposing the problem, we facilitate more accurate reconstructions that better preserve the identity and visual appearance of facial images, especially in more challenging cases that are difficult to handle using state-of-the-art techniques.

We demonstrated the utility of this technique for both the base project-and-edit scenario as well as novel interactive sequential editing applications. As our approach provides measurable improvements while being easily combined with other techniques, we anticipate it will find a place in modern image editing tools.

Acknowledgments. We thank the anonymous reviewers for their valuable feedback and insightful comments. We are also grateful to Jakub Javora for creating some of the interactive editing examples. This research was supported by Adobe, the Grant Agency of the Czech Technical University in Prague, grants No. SGS19/179/OHK3/3T/13 and No. SGS20/171/OHK3/3T/13, and by the Research Center for Informatics, grant No. CZ.02.1.01/0.0/0.0/16_019/0000765.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2StyleGAN: How to embed images into the StyleGAN latent space? In: Proceedings of IEEE International Conference on Computer Vision (2019)
2. Abdal, R., Zhu, P., Mitra, N.J., Wonka, P.: StyleFlow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. ACM Transactions on Graphics **40**(3), 21 (2021)
3. Alaluf, Y., Patashnik, O., Cohen-Or, D.: ReStyle: A residual-based StyleGAN encoder via iterative refinement. In: Proceedings of IEEE International Conference on Computer Vision. pp. 6711–6720 (2021)
4. Alaluf, Y., Tov, O., Mokady, R., Gal, R., Bermano, A.H.: HyperStyle: StyleGAN inversion with hypernetworks for real image editing. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 18511–18521 (2022)
5. Bau, D., Zhu, J.Y., Wulff, J., Peebles, W.S., Zhou, B., Strobelt, H., Torralba, A.: Seeing what a GAN cannot generate. In: Proceedings of IEEE International Conference on Computer Vision. pp. 4501–4510 (2019)
6. Deng, J., Guo, J., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition pp. 4685–4694 (2019)
7. Dinh, T.M., Tran, A.T., Nguyen, R., Hua, B.S.: HyperInverter: Improving stylegan inversion via hypernetwork. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 11389–11398 (2022)
8. Fišer, J., Jamriška, O., Simons, D., Shechtman, E., Lu, J., Asente, P., Lukáč, M., Sýkora, D.: Example-based synthesis of stylized facial animations. ACM Transactions on Graphics **36**(4), 155 (2017)
9. Futschik, D., Lukáč, M., Shechtman, E., Sýkora, D.: Real image inversion via segments. In: arXiv. No. 2110.06269 (2021)
10. Huh, M., Zhang, R., Zhu, J.Y., Paris, S., Hertzmann, A.: Transforming and projecting images into class-conditional generative networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 17–34 (2020)
11. Kang, K., Kim, S., Cho, S.: GAN inversion for out-of-range images with geometric transformations. In: Proceedings of IEEE International Conference on Computer Vision. pp. 13941–13949 (2021)
12. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Proceedings of Conference on Neural Information Processing Systems (2021)
13. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)
14. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 8107–8116 (2020)
15. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 5549–5558 (2020)
16. Ling, H., Kreis, K., Li, D., Kim, S.W., Torralba, A., Fidler, S.: EditGAN: High-precision semantic image editing. In: Proceedings of Conference on Neural Information Processing Systems (2021)

17. Lipton, Z.C., Tripathi, S.: Precise recovery of latent vectors from generative adversarial networks. In: Proceedings of International Conference on Learning Representations (2017)
18. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: StyleCLIP: Text-driven manipulation of StyleGAN imagery. In: Proceedings of IEEE International Conference on Computer Vision. pp. 2085–2094 (2021)
19. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Transactions on Graphics **22**(3), 313–318 (2003)
20. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: A stylegan encoder for image-to-image translation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 2288–2296 (2021)
21. Roich, D., Mokady, R., Bermano, A.H., Cohen-Or, D.: Pivotal tuning for latent-based editing of real images. In: arXiv. No. 2106.05744 (2021)
22. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for StyleGAN image manipulation. ACM Transactions on Graphics **40**(4), 133 (2021)
23. Wu, Z., Lischinski, D., Shechtman, E.: Stylespace analysis: Disentangled controls for StyleGAN image generation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 12863–12872 (2021)
24. Xu, Y., Du, Y., Xiao, W., Xu, X., He, S.: From continuity to editability: Inverting GANs with consecutive images. In: Proceedings of IEEE International Conference on Computer Vision. pp. 13910–13918 (2021)
25. Yao, X., Newson, A., Gousseau, Y., Hellier, P.: Feature-style encoder for style-based GAN inversion. In: arXiv. No. 2202.02183 (2022)
26. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 586–595 (2018)
27. Zhang, Y., Ling, H., Gao, J., Yin, K., Lafleche, J.F., Barriuso, A., Torralba, A., Fidler, S.: DatasetGAN: Efficient labeled data factory with minimal human effort. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 10145–10155 (2021)
28. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain GAN inversion for real image editing. In: Proceedings of European Conference on Computer Vision. pp. 592–608 (2020)
29. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: Proceedings of European Conference on Computer Vision. pp. 597–613 (2016)
30. Zhu, P., Abdal, R., Femiani, J., Wonka, P.: Barbershop: GAN-based image compositing using segmentation masks. ACM Transactions on Graphics **40**(6), 215 (2021)