

Happy People – Image Synthesis as Black-Box Optimization Problem in the Discrete Latent Space of Deep Generative Models

Steffen Jung^{1,2} Jan Christian Schwedhelm³ Claudia Schillings⁴ Margret Keuper^{1,2}

¹ Max Planck Institute for Informatics, Saarland Informatics Campus,
² University of Siegen, ³ University of Mannheim, ⁴ Free University of Berlin
{steffen.jung, keuper}@mpi-inf.mpg.de

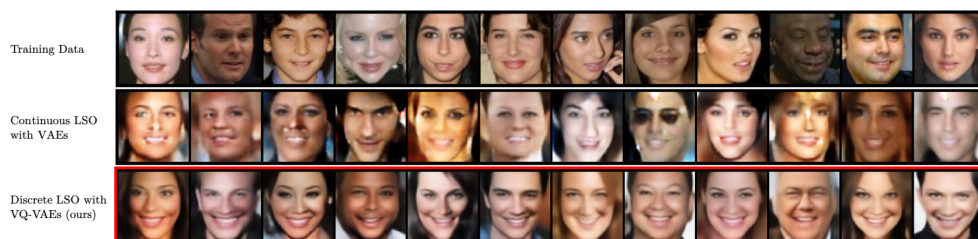


Figure 1. In contrast to previous latent space optimization (LSO) approaches (**middle**), our proposed method is able to synthesize smiling faces with high quality and less artifacts (**bottom**). Furthermore, the generated images have a significantly higher degree of smiling compared to the best points in the restricted training dataset (**top**, restricted to smiling degree ≤ 2) that was used to train the models. LSO is performed to maximize smiling degree, and FID to the target distribution (unseen smiling degrees 3 – 5) improves from 50.51 to 41.69.

Abstract

In recent years, optimization in the learned latent space of deep generative models has been successfully applied to black-box optimization problems such as drug design, image generation or neural architecture search. Existing models thereby leverage the ability of neural models to learn the data distribution from a limited amount of samples such that new samples from the distribution can be drawn. In this work, we propose a novel image generative approach that optimizes the generated sample with respect to a continuously quantifiable property. While we anticipate absolutely no practically meaningful application for the proposed framework, it is theoretically principled and allows to quickly propose samples at the mere boundary of the training data distribution. Specifically, we propose to use tree-based ensemble models as mathematical programs over the discrete latent space of vector quantized VAEs, which can be globally solved. Subsequent weighted retraining on these queries allows to induce a distribution shift. In lack of a practically relevant problem, we consider a visually appealing application: the generation of happily smiling faces (where the training distribution only contains less happy people) - and show the principled behavior of our approach in terms of improved FID and higher smile degree over baseline approaches.

1. Introduction

Many problems in science and engineering can be formulated as optimization of a costly-to-evaluate black-box function over high-dimensional or structured input domains. Notable examples are drug design, which typically requires expensive, time-consuming laboratory experiments for evaluation, neural architecture search, where every potential network solution with variable complexity must be trained and tested, or image synthesis, which can be framed as a black-box optimization problem whose objective function is a human judgment.

In the last few years, *latent space optimization* has been established [7], which tackles black-box optimization problems in a two-step procedure: First, a deep generative model (DGM) is trained on the input data, and second, standard optimization methods such as Bayesian optimization (BO) [2, 17] are used in the low-dimensional and continuous latent space learned by the DGM. Despite great successes in application fields such as chemical design [7, 11] and automatic machine learning [15], LSO lacks in performance if the training data of the DGM mainly consists of low-scoring points and the true global optimum lies far away from this data [20]. To address these weaknesses, Tripp et al. [20] have proposed a method to boost the efficiency and performance of LSO by iteratively retraining an encoder-decoder-based DGM (e.g., a Variational Autoencoder (VAE) [12])

on data points queried along the optimization trajectory and weighting those data points according to their objective value. This can be understood as an induced domain shift of a generative model.

By building on [20], this paper demonstrates the applicability of such distribution shifts induced by weighted retraining on black-box optimization problems involving unstructured image data. In particular, we implement [20] on a generative model involving a discrete latent space, namely *Vector Quantized Variational Autoencoder (VQ-VAE)* [21] which constitute very powerful yet simple to train alternatives to VAEs. In contrast to VAEs, VQ-VAEs can generate images at high quality without overly smooth details. Classical black-box optimization approaches such as BO are typically based on Gaussian processes [16] or neural networks [18] and thus assume fully continuous input domains. To allow for targeted optimization within the discrete latent space, this work transfers the LSO process from continuous to categorical input domains by utilizing tree-based ensembles as surrogate objective model in the latent space, encoding their predictions as mathematical optimization programs and solving those programs deterministically using state-of-the-art global solvers, see Figure 2. As we show in Section 3, the proposed framework improves results compared to continuous LSO via standard VAEs, and generates high-quality images that have significantly higher objective function values than the training data.

2. Tree-based Vector Quantized Generator Optimization

Although learning representations with continuous variables has been the focus of many previous works [3, 4, 22], [21] has demonstrated that discrete representations learned by VQ-VAEs capture important features of the data with a more natural fit for images. Thus, we provide a solution to transfer the LSO from continuous to discrete latent spaces in order to leverage the expressiveness of VQ-VAEs to generate high-scoring images with good quality. Then, we apply weighted retraining [20] to induce a distribution shift in the generator distribution. These two steps can be alternated until convergence to train generators with desired target distributions.

2.1. Discrete latent variables in VQ-VAEs

The full VQ-VAE model [21] consists of a latent embedding space \mathcal{E} , an encoder e and a decoder d , where \mathcal{E} contains K learnable vectors $\mathcal{E}_i \in \mathbb{R}^D$, $i \in \{1, \dots, K\}$. The shared embedding space \mathcal{E} and the encoder e allow to represent each input image $x \in \mathbb{R}^{H \times W \times 3}$ as a grid $z \in \{1, \dots, K\}^{h \times w}$ of discrete latent variables, where the components

$$z_{i,j} = \arg \min_{l \in \{1, \dots, K\}} \|e(x)_{i,j,:} - \mathcal{E}_l\|$$

are obtained by vector quantization of the encoded image $e(x) \in \mathbb{R}^{h \times w \times D}$ w.r.t. the Euclidean distance. $H \times W$ and $h \times w$ describe the spatial extent of the input and latent representation, respectively.

2.2. Global Optimization in Discrete Latent Spaces

Our method includes optimization in the discrete latent space $z \in \{1, \dots, K\}^{h \times w}$ of a VQ-VAE, which is pre-trained on some input data \mathcal{X} . To this end, a latent objective model $h(\cdot)$ is constructed to approximate the black-box function $f(\cdot)$ at the output of the decoder $d(\cdot)$, i.e., $h(z) \approx f(d(z))$ for all $z \in \{1, \dots, K\}^{h \times w}$. h can be trained by using the encoder e and the learned embedding space \mathcal{E} , which together map every f -evaluated input image $x \in \mathcal{X}$ to a corresponding discrete latent representation $z = e(x) \in \mathcal{Z}_{\mathcal{X}} \subset \{1, \dots, K\}^{h \times w}$.

When dealing with categorical feature spaces such as $\{1, \dots, K\}^{h \times w}$, tree-based ensemble models like random forests [1] or gradient-boosting trees [5] are popular choices for h , since they naturally support various data types. Thus, we train a decision tree ensemble model to predict the objective value of a given image sample from its discrete latent representation. Interestingly, [19] propose an intricate approach that allows to encode the trained tree-based model with a mixed-integer optimization (MIO) formulation which allows the optimization of the discrete latent code w.r.t. the objective value, the tree-based model has been trained to predict. Please refer to [19] for the theoretical proof. By solving the resulting MIO program deterministically using a global solver [13], h can be optimized to determine the next latent query point \tilde{z} . During latent optimization, the solution \tilde{z} has to be restricted to stay sufficiently close to latent representations of the training data. Otherwise, without any constraints on the latent variables, the decoded version $\tilde{x} = d(\tilde{z})$ of the query point \tilde{z} most likely has bad quality for the following reason: values of those latent variables not having an impact on image regions that determine the objective function value can arbitrarily be chosen in the optimization process, which may lead to highly distorted image features. We address this problem as follows: First, a single training sample $x_r \in \mathcal{X}$ is randomly drawn and mapped to its latent representation z_r . Second, only those $t \in \mathbb{N}$ latent variables having the highest feature importances under the trained tree-based model are free to be globally optimized, while the remaining $hw - t$ variables are fixed by being set to the respective entries of z_r . Taking into account the optimization result \tilde{z} together with the corresponding objective function evaluation $f(\tilde{x})$ of its decoded version $\tilde{x} = d(\tilde{z})$, h is refit after every iteration.

2.3. Weighted Retraining

Subsection 2.2 introduces a technique to perform LSO in discrete latent spaces. However, identified in [20], the un-

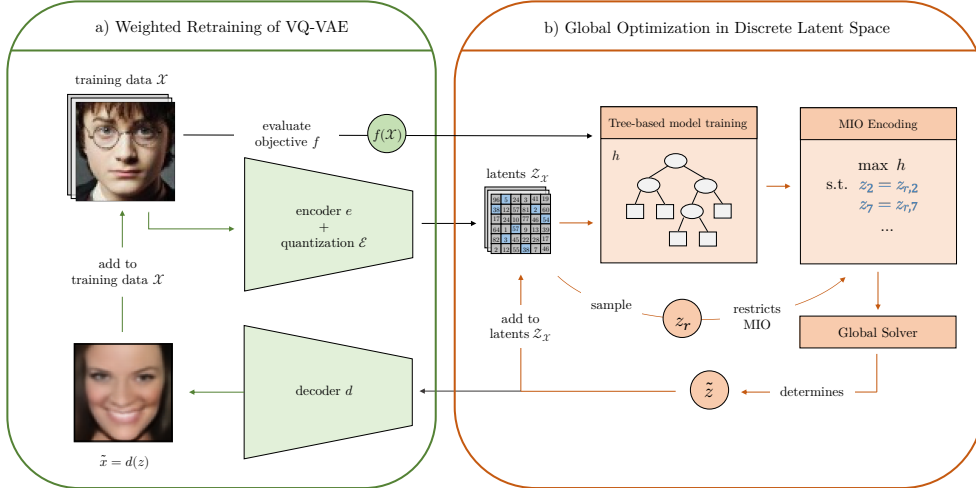


Figure 2. Proposed Framework. **a)** The encoder and decoder of a VQ-VAE are initially pre-trained and periodically fine-tuned on weighted data \mathcal{X} . Images are encoded in a discrete latent space. **b)** Optimization is performed in the discrete latent space: a tree-based ensemble model is learned from the latent training data representations $\mathcal{Z}_{\mathcal{X}}$, encoded as a constrained mixed integer optimization problem and globally solved to determine the next (optimal) query point \tilde{z} according to the black-box evaluation function. This procedure is repeated r times before the VQ-VAE is fine-tuned on the queried data points. For this, the corresponding decoded images $\tilde{x} = d(\tilde{z})$ are added to the training data and the weight for each training image is updated.

derlying DGM does not necessarily learn a latent space that is amenable to efficient optimization of the objective function, especially in cases where the global optimum is far away from the training data. To resolve this issue, we follow [20] and weight the training data points $\mathcal{X} = \{x_i\}_{i=1}^N$ according to their objective values $\{f(x_i)\}_{i=1}^N$: the higher a value $f(x)$, the more probability mass the training distribution should place on the corresponding input point x . The weighting scheme requires assigning a weight $w(x_i) \in \mathbb{R}$ with $\sum_{i=1}^N w(x_i) = 1$ for all $x_i \in \mathcal{X}$ using some weight function w . In this work, we adopt the rank-based function introduced in [20].

To propagate information on new points acquired during the iterative LSO process to the VQ-VAE, where it could potentially help to uncover new promising regions that an optimization algorithm can exploit, the VQ-VAE is periodically fine-tuned after every $r \in \mathbb{N}$ LSO iterations.

3. Experiments

Here, we provide an empirical evaluation of the proposed discrete LSO with weighted retraining, as introduced in Section 2. First, we define a challenging optimization task using the famous face dataset CelebA [14]. Based on this image synthesis task, we evaluate our methods’s ability to compete with continuous LSO via standard VAEs [20]. For the underlying VQ-VAE, we use the CNN encoder-decoder architecture introduced in [21]. Please refer to the appendix for more details.

Optimization task. We employ CelebA 64×64 for an

image synthesis task which can be viewed as black-box optimization problem. Our goal is to generate smiling faces by optimizing for the respective attribute degree in the space of colored 64×64 face images. To represent the smiling attribute on a continuous scale, we make use of the extended CelebA-Dialog dataset [10], which includes fine-grained labels for five selected CelebA attributes that cannot be accurately described by binary labels. In particular, the smiling attribute is divided into six levels (0 – 5) that describe the degree in ascending order. For unseen face images, we estimate the degree of smiling by using a probability-weighted average $\hat{f}_{\text{smile}} : \mathbb{R}^{64 \times 64 \times 3} \rightarrow [0, 5]$ of class predictions coming from the ResNet-50 [8] classifier used in the official implementation of [10], which is pre-trained on CelebA-Dialog.

We discard points with high objective values from the training data to make the problem more challenging and to represent the situation where the optimum (degree 5) lies far outside the training distribution. Specifically, *only images with a smiling degree ≤ 2 are kept in the training set.*

Implementation details. Throughout this work, we assume a query budget of 500 and a retrain frequency of $r = 5$. For the underlying VQ-VAE, we use the CNN encoder-decoder architecture as in [21]. We assume a discrete latent space of size 8×8 (64 latent variables in total). Moreover, we consider $K = 256$ embedding vectors with a dimensionality of $D = 64$ each. The tree-based model h is chosen to be an ensemble of 800 gradient-boosted regression tree and an interaction depth of 2, following [19]. The

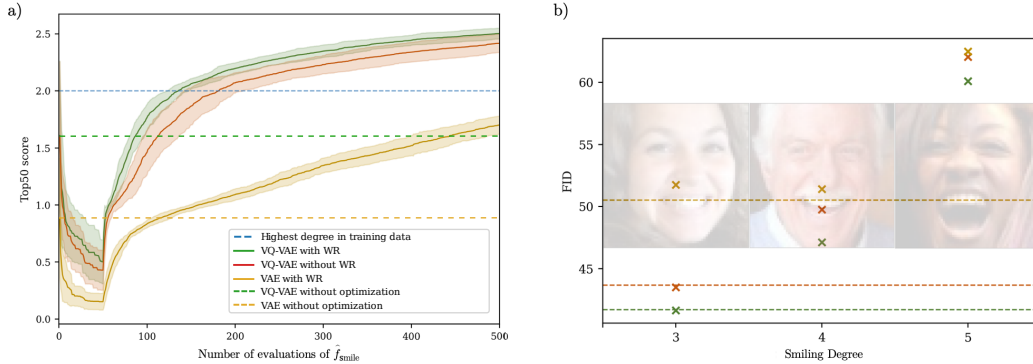


Figure 3. Quantitative comparison between discrete and continuous LSO variants: VQ-VAE with weighted retraining (**green**), VQ-VAE without weighted retraining (**red**), and VAE with weighted retraining (**yellow**). Evaluation metrics: **a)** Top50 score function, where the blue dashed line depicts the best smiling degree in the training data. VQ-VAE as well as weighted retraining improve the degree of smiling substantially. **b)** FID scores, where the dashed lines show the respective scores of testing against the subset of CelebA images having an unseen smiling degree (3 – 5), and the markers show testing against each training data subset with the corresponding unseen smiling degree (3, 4, and 5; background images depict training data samples for these smiling degrees). In all cases, VQ-VAE with weighted retraining achieves the best result.

minimum number of samples in one leaf equals 20, and the maximum number of leaves per tree is set to 5.

Continuous LSO with VAEs is implemented by a neural network as latent objective model, as in [18]. Motivated by [6], local optimization in the continuous latent space is constrained using ex-post density estimation via Gaussian Mixture Models. Optimization is carried out using the SLSQP algorithm [13].

Evaluation metrics. Following common practice for BO [17, 20], we show the worst of the 10 and 50 best novel smiling degree predictions obtained up until query $m = 1, \dots, 500$, which is denoted as Top10 and Top50 score function, respectively. To ensure statistical significance, the mean \pm one standard deviation across 20 runs with different random seeds is reported. Furthermore, we use FID scores [9] as quantitative assessment of the quality of generated images. Since our goal is to find faces that have a higher smiling degree than the best point in the training data, we compute FID scores between all 10,000 generated images from the 20 runs and the subset of CelebA faces having a smiling degree between 3 and 5.

Results on smiling face synthesis. Figure 3 presents quantitative comparison between our approach and continuous LSO via standard VAEs. Corresponding visual results are shown in Figure 1. Even if weighted retraining is applied, the Top50 scores resulting from LSO with VAEs are strictly below the value 2.0 that corresponds to the best training images (mean final score: 1.70). Even without weighted retraining of the VQ-VAE, our method outperforms VAEs by a large margin (mean final score: 2.42). Weighted retraining further improves the final Top50 score from 2.42 to 2.50 on average.

Moreover, the FID results show that our method success-

fully leverages the expressiveness of VQ-VAEs to generate images with higher quality compared to standard VAEs. The FID score can be decreased by 21%, from 50.51 to 41.69. Again, weighted retraining hereby has a positive effect and leads to a significant improvement of 5%. In addition, we tested against three separate subsets containing all images having a smiling degree of 3, 4, and 5, respectively, to detect potential biases among the generated faces. However, for all considered LSO variants, the results are as expected: higher smiling degrees in the target distribution lead to higher FID scores. Overall, the level of FID scores is relatively high, which can likely be attributed to the following reason: the target distribution (degrees 3 – 5) is different from the distribution the VQ-VAE is pre-trained on (degrees 0 – 2), which prevents direct comparison to other generative models that are explicitly designed and trained to generate samples that resemble the given training data.

4. Conclusion

We propose a method for efficient black-box optimization in the discrete latent space of VQ-VAEs, which combines (i) choosing a tree-based ensemble as latent objective model, (ii) encoding its predictions as an MIO problem that is solved globally to determine the next query point and (iii) iteratively fine-tuning the underlying VQ-VAE on weighted data. With the challenging task of generating smiling faces that are not contained in the training distribution, we demonstrate that our method notably outperforms continuous LSO with VAEs in terms of both image quality and optimization of the objective function. To the best of our knowledge, this is the first work that successfully applies LSO in discrete latent spaces on image synthesis tasks.

References

- [1] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. [2](#)
- [2] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, 2010. [1](#)
- [3] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016. [2](#)
- [4] Emily L. Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *CoRR*, abs/1611.06430, 2016. [2](#)
- [5] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. [2](#)
- [6] P. Ghosh*, M. S. M. Sajjadi*, A. Vergari, M. J. Black, and B. Schölkopf. From variational to deterministic autoencoders. In *8th International Conference on Learning Representations (ICLR)*, Apr. 2020. *equal contribution. [4](#), [6](#)
- [7] Rafael Gómez-Bombarelli, David Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *CoRR*, abs/1610.02415, 2016. [1](#)
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. [3](#)
- [9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. [4](#)
- [10] Yuming Jiang, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu. Talk-to-edit: Fine-grained facial editing via dialog. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. [3](#)
- [11] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation, 2019. [1](#)
- [12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014. [1](#)
- [13] D. Kraft. *A software package for sequential quadratic programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988. [2](#), [4](#), [6](#)
- [14] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. [3](#)
- [15] Xiaoyu Lu, Javier Gonzalez, Zhenwen Dai, and Neil Lawrence. Structured variationally auto-encoded optimization. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3267–3275. PMLR, 10–15 Jul 2018. [1](#)
- [16] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003. [2](#)
- [17] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016. [1](#), [4](#)
- [18] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable bayesian optimization using deep neural networks. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2171–2180, Lille, France, 07–09 Jul 2015. PMLR. [2](#), [4](#), [6](#)
- [19] Alexander Thebelt, Jan Kronqvist, Miten Mistry, Robert M. Lee, Nathan Sudermann-Merx, and Ruth Misener. Entmoot: A framework for optimization over ensemble tree models. *Computers & Chemical Engineering*, 151:107343, Aug 2021. [2](#), [3](#), [6](#)
- [20] Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining, 2020. [1](#), [2](#), [3](#), [4](#)
- [21] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. [2](#), [3](#), [6](#)
- [22] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, dec 2010. [2](#)
- [23] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016. [9](#)

A. Appendix

A.1. Architectures and Implementation

For the underlying VQ-VAE, we use the CNN encoder-decoder architecture as introduced in [21]. We assume a discrete latent space of size 8×8 , which results in 64 latent variables in total. Moreover, we consider $K = 256$ embedding vectors with a dimensionality of $D = 64$ each. The tree-based model h is chosen to be a gradient-boosting regression tree with an ensemble of 800 decision trees and an interaction depth of 2, following [19]. Furthermore, the minimum number of samples in one leaf equals 20, and the maximum number of leaves per tree is set to 5. During LSO, we want to preserve as much image quality as possible. Therefore, we will not optimize over all 64 latent variables but only over those that are most important for the objective value we are optimizing for. Thus, we leverage the feature importance estimated by the decision tree ensemble to determine latent variables to optimize. An ablation of the number of latent variables used for optimization is given in Section A.2.

Continuous LSO with VAEs is implemented by a neural network as latent objective model, as in [18]. Motivated by [6], local optimization in the continuous latent space is constrained using ex-post density estimation via Gaussian Mixture Models, and new query points are restricted to stay sufficiently close to the estimated distribution. Optimization is carried out using the SLSQP algorithm [13].

A.2. Ablation study

During LSO, we want to preserve as much image quality as possible. Therefore, we will not optimize over all 64 latent variables but only over those that are most important for the objective value we are optimizing for. Here, we measure the impact of the number t of free LSO variables on the performance of our proposed method. For this purpose, we test 4, 8, and 16 free variables for 20 optimization runs each. FID scores as well as final Top10 and Top50 scores after 500 optimization iterations are reported in Table 1.

t	FID	Top10	Top50
4	42.69	2.72 ± 0.07	2.42 ± 0.04
8	41.69	2.85 ± 0.07	2.50 ± 0.05
16	43.70	2.88 ± 0.06	2.58 ± 0.02

Table 1. Ablation study results. Top10 and Top50 scores are reported as mean final score \pm one standard deviation after reaching the query budget of 500.

As we can observe, choosing $t = 8$ hits a sweet spot: while generated faces have significantly lower quality if t is increased (FID: 41.69 vs. 43.70), a reduction of t leads to lower smiling degrees (Top10: 2.85 vs. 2.72). We set t to 8 for all experiments.

A.3. Details on VQ-VAE

Stage	Layer	Input Shape (H_0, W_0, D_0)	Output Shape (H_1, W_1, D_1)	# Filters K	Kernel Size F	Padding P	Stride S
1	Conv Leaky ReLu	(64, 64, 3)	(32, 32, 64)	64	4	1	2
2	Conv Leaky ReLu	(32, 32, 64)	(16, 16, 128)	128	4	1	2
3	Conv Leaky ReLu	(16, 16, 128)	(8, 8, 256)	256	4	1	2
4	Conv Leaky ReLu	(8, 8, 256)	(8, 8, 256)	256	3	1	1
5	Residual Block	(8, 8, 256)	(8, 8, 256)				
6	Residual Block	(8, 8, 256)	(8, 8, 256)				
7	Conv Leaky ReLu	(8, 8, 256)	(8, 8, 64)	64	1	0	1

Table 2. Encoder architecture of VQ-VAE

Stage	Layer	Input Shape (H_0, W_0, D_0)	Output Shape (H_1, W_1, D_1)	# Filters K	Kernel Size F	Padding P	Stride S	O-Padding O
1	Conv Leaky ReLu	(8, 8, 64)	(8, 8, 256)	256	3	1	1	
2	Residual Block	(8, 8, 256)	(8, 8, 256)					
3	Residual Block	(8, 8, 256)	(8, 8, 256)					
4	ConvT Leaky ReLu	(8, 8, 256)	(16, 16, 128)	128	4	1	2	0
5	ConvT Leaky ReLu	(16, 16, 128)	(32, 32, 64)	64	4	1	2	0
6	ConvT	(32, 32, 64)	(64, 64, 3)	3	4	1	2	0

Table 3. Decoder architecture of VQ-VAE

Hyperparameter	Value
Latent Dimension	64
Embedding Dimension D	64
# Categories K	256
Batch Size	128
# Epochs	70
Optimizer	Adam
Learning Rate	0.001
β	0.25

Table 4. Training hyperparameters of VQ-VAE

A.4. Details on VAE

Stage	Layer	Input Shape (H_0, W_0, D_0)	Output Shape (H_1, W_1, D_1)	# Filters K	Kernel Size F	Padding P	Stride S
1	Conv Leaky ReLu BatchNorm	(64, 64, 3)	(32, 32, 32)	32	3	1	2
2	Conv Leaky ReLu BatchNorm	(32, 32, 32)	(16, 16, 64)	64	3	1	2
3	Conv Leaky ReLu BatchNorm	(16, 16, 64)	(8, 8, 128)	128	3	1	2
4	Conv Leaky ReLu BatchNorm	(8, 8, 128)	(4, 4, 256)	256	3	1	2
5	Conv Leaky ReLu BatchNorm	(4, 4, 256)	(2, 2, 512)	512	3	1	2
6	Flatten Dense	(2, 2, 512) 2048	2048 128				

Table 5. Encoder architecture of VAE

Stage	Layer	Input Shape (H_0, W_0, D_0)	Output Shape (H_1, W_1, D_1)	# Filters K	Kernel Size F	Padding P	Stride S	O-Padding O
1	Dense Leaky ReLu BatchNorm	64	2048					
2	Unflatten ConvT Leaky ReLu BatchNorm	2048 (2, 2, 512)	(2, 2, 512) (4, 4, 256)	256	3	1	2	1
3	ConvT Leaky ReLu BatchNorm	(4, 4, 256)	(8, 8, 128)	128	3	1	2	1
4	ConvT Leaky ReLu BatchNorm	(8, 8, 128)	(16, 16, 64)	64	3	1	2	1
5	ConvT Leaky ReLu BatchNorm	(16, 16, 64)	(32, 32, 32)	32	3	1	2	1
6	ConvT Leaky ReLu BatchNorm	(32, 32, 32)	(64, 64, 32)	32	3	1	2	1
7	Conv	(64, 64, 32)	(64, 64, 3)	3	3	1	1	

Table 6. Decoder architecture of VAE

Hyperparameter	Value
Latent Dimension	64
Batch Size	64
# Epochs	30
Optimizer	Adam
Learning Rate	0.0001
β_{start}	10^{-6}
β_{final}	0.215
β Warm-up	5000
β Step	1.1
β Step Frequency	50

Table 7. Training hyperparameters of VAE

A.5. Details on dataset

We focus on the facial part of CelebA images by using pre-trained Multitask Cascaded Convolutional Networks (MTCNN) [23] to localize faces in the input plane, and cropping images according to the predicted bounding box. Furthermore, the data is resized to 64×64 via resampling using pixel area relation.