Author response: ‘Deep Automodulators’ NeurIPS: #6295

We thank the reviewers for their insights. We summarize the overall response as positive. The few criticisms presented by R1 and R2 are based on some misunderstandings. We will use page 9 of the camera-ready version to clarify these points and to accommodate the block diagram requested by R3. We address the concerns in the order received.

R1: There might be a misunderstanding here. The sole concern of R1 is that our model is an *“extension of the idea in BEGAN to be able to generate fused samples from an arbitrary combination of several images, allowing instantaneous ‘style-mixing’.”* However, the provided references [1, 2] do not show that BEGAN could do such a thing. We are happy to outline some *fundamental differences to BEGAN*, and will add the same discussion and the citation (in Sec. 2). First, although both models introduce extra paths from latent code to each decoder layer, the similarity ends there. In our model, due to the *modulation*, you can apply latent code $\boldsymbol{z}_1$ at certain layers of the decoder and $\boldsymbol{z}_2 \neq \boldsymbol{z}_1$ at other layers. This enables style-mixing, enforcement of scale-specific invariances, and scale-specific attribute modifications. BEGAN can do none of this, because it connects the one and the same latent code to each of the decoder layers. Unlike in our paper, their skip connections *only* improve the reconstruction quality of the single image. Second, our model (following AGE [43]) consists of nothing but the encoder and decoder, whereas BEGAN *also* contains a discriminator (in your ref. [1], see Sec. 3.5). All else being equal, this makes their model nearly 50% larger.

R2: We were puzzled by this review. The criticism of our approach amounts to the claim that *“Compared with the results of the automatic encoder, the experimental results show that the improvement of this method is very limited.”* We are not sure what the “the automatic encoder” refers to in this context. If it means ‘autoencoder’, then we guess this refers to Table 2a. If so, then it is *not at all* the case that the table shows the *“improvement is very limited”*. Table 2a shows our model to be clearly superior to all the baselines, except for the FID of B-Pioneer [17]. Perhaps there was confusion by the fact that, *in Table 2a, the B-Pioneer refers to the classic architecture that cannot do style-mixing* and was only included for reference? In any case, we will clarify this in the table. Our model can do style-mixing and leverage scale-specific invariances and attribute modification. The classic architectures cannot. In the table, you can only compare our model directly against the VAE-AdaIn and WAE-AdaIn.

R3: Thank you for good questions and suggestions. We agree that the presented idea could be further improved to make it even more elegant. Yet, without discriminators, our model is already architecturally simpler than a corresponding GAN-based encoder model (such as BEGAN, see above) would be. Regarding comparison to GANs, we will follow your advice and add an experiment evaluating the FIDs of 10,000 mixed latent codes with FFHQ-trained StyleGAN-projections and our model, as a function of optimization time (as in Fig. 15).

R3 (1): *“If latent code is normalized to a unit sphere, why does KL loss to unnormalized N(0,I) still make sense?”* It turns out one can derive the analytical solution for the KL divergence between an empirical distribution of code vectors normalized to unit hypersphere and the unnormalized unit Gaussian with diagonal covariance. See, *e.g.*, Eq. (7) in [43] or (corrected) Eq. (1) in [16]. You can think of it intuitively as follows: If this loss term is at minimum, where are the points on the unit sphere? They must be uniformly distributed. The larger the loss, the less uniform is the distribution.

R3 (2): Interestingly, powerful scale-specific editing is not only about encoding the image in the ‘appropriate point’ in latent space but about the smoothness of the space overall, and the KL divergence minimization encourages the posterior to lie on such a smooth well-behaved manifold. In absence of enforcing this, the path between two latent codes could contain points that decode into low-quality images. This would prevent finding latent vectors that correspond to specific image attributes in latent space (Fig. 4b) or interpolating between images (supplement Fig. 12).

R3 (3) (and question on $d_{\cos}$): For ablation of the KL gap, see [17]. The code rec. loss (d_{cos}) was studied in [43] (the paper uses L2, but *cos* in the implementation [44]). The code loss is explicitly applied only to the decoder because although the encoder would indeed like to increase it, it already achieves that effect by pushing the codes of generated samples away from Gaussian by the KL terms. For clarity, however, we will summarize these points in the paper.

R3 (4): The KL loss is also evaluated on original (training) images, see $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ in Eq. (1) and L113 for $\boldsymbol{x}$.

R4: Thank you for your encouraging comments! We agree that although a high level of disentanglement is evidently achieved in experiments, certain mixture artifacts indicate that the “disentanglement of factor/scales of variation” is not always perfectly achieved. On page 9, we will add more discussion on this and the prospects for more thorough evaluation. We focused on generalizability and a wide range of experiments rather than exhaustive optimization (such as playing with learning rates and latent space truncation as in StyleGAN [24, 25]).

R4: For the general representation learning power of the architecture, we fully agree that measuring performance on auxiliary downstream tasks would be valuable follow-up work. However, we suggest you consider that we do more than *“scope the task to. . . style-mixing”*. After all, we *also* do evaluate reconstruction, random sampling, interpolation (supplement Fig. 12), and attribute modification.