

Addressing degeneracies in latent interpolation for diffusion models

Erik Landolsi^[0000–0002–6639–1257] and Fredrik Kahl^[0000–0001–9835–3020]

Chalmers University of Technology, Göteborg, Sweden
{erik.landolsi,fredrik.kahl}@chalmers.se

Abstract. There is an increasing interest in using image-generating diffusion models for deep data augmentation and image morphing. In this context, it is useful to interpolate between latents produced by inverting a set of input images, in order to generate new images representing some mixture of the inputs. We observe that such interpolation can easily lead to degenerate results when the number of inputs is large. We analyze the cause of this effect theoretically and experimentally, and suggest a suitable remedy. The suggested approach is a relatively simple normalization scheme that is easy to use whenever interpolation between latents is needed. We measure image quality using FID and CLIP embedding distance and show experimentally that baseline interpolation methods lead to a drop in quality metrics long before the degeneration issue is clearly visible. In contrast, our method significantly reduces the degeneration effect and leads to improved quality metrics also in non-degenerate situations.

Keywords: Diffusion models · Text-to-image models · Image interpolation

1 Introduction

Over the last few years, diffusion models have shown remarkable performance in image generation [2, 20, 23]. This has sparked an interest in using diffusion models for data augmentation or fully synthetic data generation in label-constrained tasks [1, 25, 28]. Such tasks often involve adapting a diffusion model to create images with a similar appearance as a set of input images (the scarce available training data). Approaches in this direction sometimes involve some sort of interpolation in latent space, e.g. computing the centroid of a set of input examples [24] or blending inverted inputs with noise [12].

Another line of research around diffusion models involves image morphing, with the goal of producing visually pleasing smooth transitions between two images [30, 31]. Such methods could be extended to smooth interpolation between a larger set of images. The input images would then define a manifold in the latent space, from which new images could be generated from any choice of interpolation coefficients mixing the original images.

A common operation in the mentioned methods is interpolation between diffusion model latents. That is, given a set of input images $\{\mathbf{x}_n\}$ with latents

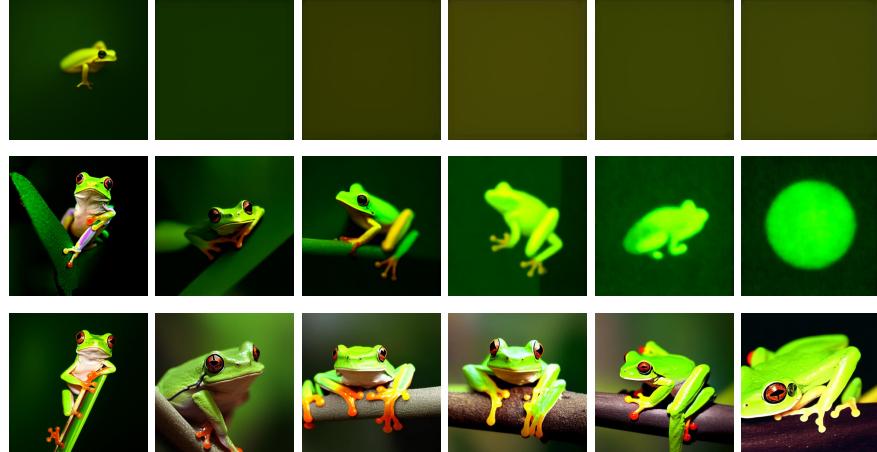


Fig. 1. Images generated from *centroids* of N latents obtained from N input images of ImageNet class “tree frog” for $N = 2, 8, 32, 48, 64, 96$. Top row: Linear interpolation. Middle row: Linear interpolation with fixed normalization. Bottom row: channel-wise mean adjustment (our suggested method).

$\{\mathbf{z}_n\}$ and a corresponding set of mixing weights $\{w_n : w_n > 0, \sum w_n = 1\}$, finding a mixed latent representing a meaningful mixture of the input images. We note that naive methods for computing such mixtures can easily lead to degenerate results. As a motivating example, in Figure 1, a set of latents was produced from N images using DDIM inversion [27]. The latent centroids were then computed and used to generate new images. The top row shows results using regular linear interpolation, which quickly breaks down due to the norm of the interpolated latent being too low [24]. A simple fix would be to normalize the latent centroid to a suitable norm level. As shown in the middle row, this works well when N is small, but the results are still degenerate when N is large.

In this paper, we deep-dive into the issue of degenerate output from diffusion model latent interpolation. After presenting related work in Section 2, we review baseline interpolation methods in Section 3. In Section 4, we diagnose the degeneracy illustrated in Figure 1, showing when and why it appears. In Section 5, we examine alternative normalization schemes as a potential remedy. Finally, in Section 6, we show experimentally that such schemes can measurably improve the quality of generated images even for smaller N , where the issue is not as obvious as in the high N examples in Figure 1.

2 Related work

Our work builds on text-to-image diffusion models [9, 27], specifically latent diffusion models in the Stable Diffusion family [5, 21]. We are not aware of any prior work analyzing degenerate output from latent interpolation in such models.

Various forms of latent space interpolation are most commonly found in morphing methods, and already the original DDPM paper [9] showed examples of this. A more recent example [30] first run textual inversion [6] on the inputs. The resulting text embeddings were interpolated linearly and the noisy latents using SLERP. The authors reached improved results by also performing a low-rank adaptation of the diffusion model. Finally, they also proposed a perceptually uniform sampling to ensure a smooth transition between the inputs. Another work [31] used similar interpolation of text conditions and embeddings, but introduced an interpolation also on the attention maps. Both these works only considered two inputs, and the authors did not identify or address the degeneracy that is our focus. Note that the latter method [31] includes an adaptive instance normalization (AdaIN) that is similar to our suggested channel-wise mean adjustment. However, they introduce it in an ad-hoc fashion, without any deeper motivation, whereas we provide a detailed analysis and show that such normalization is key to avoid degenerate results for large N .

Our work is also related to diffusion model inversion. The degeneration issue appears using inverted latents, and improved inversion techniques may be a competing way of fixing it. In this paper, we rely mostly on the DDIM inversion introduced in the original DDIM paper [27]. Since then, null-text inversion [17] has been suggested as a way of inverting diffusion models including classifier-free guidance by optimizing the unconditional embeddings. A later work [16] provided a related but faster method by avoiding the costly optimization. Another line of work studied fixed-point methods for inverting diffusion models [15, 18]. The recent ReNoise method [7] included a fixed-point inversion and additional loss terms aimed at better shaping the noise statistics. While improved inversion methods could potentially serve as competing remedies for degenerate centroids, none of the mentioned papers identified or studied this specific issue.

One related work [24] studied interpolation paths in the latent space. The authors first noted that it is critical to maintain a well-scaled latent norm. They then constructed interpolation paths between latents by minimizing a likelihood-related measure, preferring paths passing through areas where the latents have a typical norm value. Their interpolation method induces a metric in the latent space that they used also for computing centroids of up to 5 examples. These centroids were then used in a deep data augmentation pipeline for label-constrained recognition problems. However, they failed to notice that such centroids can become degenerate as the number of inputs increases, and they do not provide any analysis or remedy relating to this effect.

3 Latent space interpolation

3.1 Latent diffusion models

Denoising diffusion probabilistic models [9] model the distribution of a random variable \mathbf{x}_0 by transforming it into a tractable distribution (noise) over timesteps $t \in \{0, \dots, T\}$. A *latent diffusion model* performs the diffusion process on a latent variable \mathbf{z}_t of lower dimensionality than \mathbf{x}_t . We use the *Stable Diffusion* (SD) [21]

family of models due to their public availability and wide use in prior work. To make sure that our findings are not just based on implementation curiosities in a single version, we use two distinct SD versions (1.5 and 3.5). In SD 1.5, the noise estimation is implemented using a U-net [22], and the conditioning input is a CLIP embedding [19] of a text prompt. The forward noise model can be written as $\mathbf{z}_t = \sqrt{\alpha_t} \mathbf{z}_0 + \sqrt{1 - \alpha_t} \epsilon$ and the latent variable is a 4-channel feature map with a resolution 8 times lower than the images.

In SD 3.5 [5], the U-net is replaced with a transformer model, and the latent feature dimensionality is increased to 16. Furthermore, the noise model is changed into a rectified flow model [14], where the data is transformed to noise using a direct linear path according to $\mathbf{z}_t = (1 - t) \mathbf{z}_0 + t \epsilon$.

3.2 Interpolation notation

Let $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$ be an ordered set of input latents at time $t = T$ (dropping the t subscript for brevity) and let $\mathbf{w} = \{w_1, \dots, w_N\}$ be a corresponding ordered set of weights with $w_n > 0$ and $\sum_n w_n = 1$. Furthermore, let $\mathbf{z}' = f(\mathbf{Z}, \mathbf{w})$ denote an interpolation operation mixing the \mathbf{z}_n using weights w_n . A desired property of any f is that if any $w_n = 1$, then $f(\mathbf{Z}, \mathbf{w}) = \mathbf{z}_n$, such that inputs are reproduced exactly. Finally, for compactness, let $f(\mathbf{Z}) = f(\mathbf{Z}, \{\frac{1}{N}, \dots, \frac{1}{N}\})$ denote a centroid computation.

3.3 Baseline interpolation options

Linear interpolation. The most direct baseline option is basic linear interpolation or *convex combination* according to

$$\mathbf{z}' = f_{\text{LIN}}(\mathbf{Z}, \mathbf{w}) \triangleq \sum_n w_n \mathbf{z}_n. \quad (1)$$

As noted in prior work [24] and illustrated in Figures 1-2, this often produces output with a significantly lower norm than the inputs, leading to washed-out images with a severe lack of detail.

Fixed normalization. The norm of randomly sampled $\mathbf{z} \sim \mathcal{N}(0, I)$ is sharply distributed around \sqrt{L} , where L is the dimensionality of \mathbf{z} [24]. A simple attempt to fix the low norms produced by linear interpolation would be to normalize the interpolated latent to the typical norm \sqrt{L} using

$$\mathbf{z}' = f_{\text{FIX}}(\mathbf{Z}, \mathbf{w}) = \frac{\sqrt{L}}{\|\sum_n w_n \mathbf{z}_n\|} \sum_n w_n \mathbf{z}_n. \quad (2)$$

However, as illustrated in Figure 2, this breaks the desired input reproduction property mentioned in Section 3.2, since $f_{\text{FIX}}(\mathbf{Z}, \{1, 0, 0, \dots\}) = \sqrt{L} \|\mathbf{z}_1\|^{-1} \mathbf{z}_1$ which is not equal to \mathbf{z}_1 in the general case. Therefore, f_{FIX} may not be suitable as a general interpolation method.

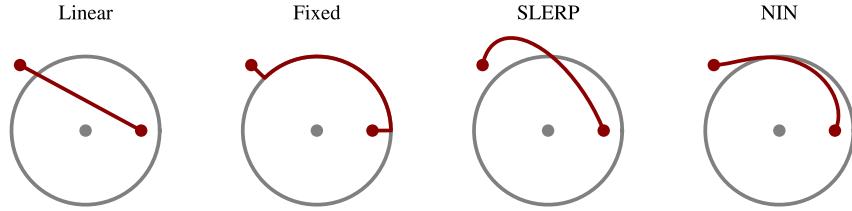


Fig. 2. Interpolation paths produced by the methods from Section 3.3 for a 2D toy example. The gray circle represents latents with norm \sqrt{L} , which is where randomly sampled latents are typically located.

SLERP. Spherical linear interpolation (SLERP) [26] is a way of interpolating between two points on a unit sphere according to

$$\text{slerp}([\mathbf{p}_1, \mathbf{p}_2], t) = \frac{\sin((1-t)\theta)}{\sin\theta} \mathbf{p}_1 + \frac{\sin(t\theta)}{\sin\theta} \mathbf{p}_2 \quad (3)$$

where $\theta = \cos^{-1}(\mathbf{p}_1^T \mathbf{p}_2)$ is the angle between \mathbf{p}_1 and \mathbf{p}_2 . If the inputs are not on the unit sphere, this is often handled by a normalization when computing θ , letting $\theta = \cos^{-1}(\mathbf{p}_1^T \mathbf{p}_2 / \|\mathbf{p}_1\| \|\mathbf{p}_2\|)$ [29, 30]. However, this departs from the original SLERP formulation and can lead to unexpected results. In the example in Figure 2, the path initially moves away from the circle representing the typical norm value near the original inputs. This leads to latent norms outside the input norm range, which is undesirable since generated image quality tends to deteriorate when the norm departs from the nominal value [24]. Furthermore, in order to generalize SLERP to multiple inputs, iterative methods are required [3]. For these reasons, we will not consider SLERP further in this paper.

Normalization to interpolated norms. In [12], it was suggested to instead set the norm of an interpolated latent to the linearly interpolated norms of the inputs, i.e. letting

$$\mathbf{z}' = f_{\text{NIN}}(\mathbf{Z}, \mathbf{w}) = \frac{\sum_n w_n \|\mathbf{z}_n\|}{\left\| \sum_n w_n \mathbf{z}_n \right\|} \sum_n w_n \mathbf{z}_n \quad (4)$$

This operation fulfills the input reproduction property mentioned in Section 3.2 and does not suffer from the norm overshoots that can happen using SLERP (see Figure 2).

4 Degenerate interpolation output

Using f_{FIX} from Eq. 2 or f_{NIN} from Eq. 4, one might expect interpolated latents to be free from issues caused by lacking normalization. However, as illustrated in Figure 1, as the number of inputs included in an interpolation operation grows, the output can be degenerate even though the latent norm should now be well-behaved. In this section, we analyze the cause of this phenomenon.

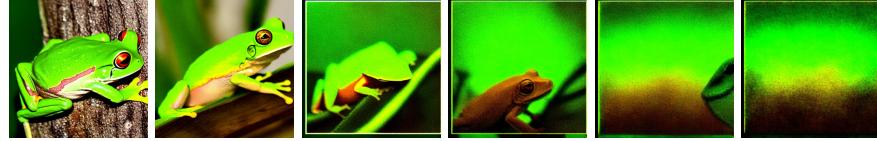


Fig. 3. Images generated from centroids created using the recipe in Section 4.1. ImageNet class “tree frog”, $N = 2, 8, 32, 48, 64, 96$, fixed normalization.

4.1 Initial investigation

Let us first consider an ideal case of N i.i.d. latents $\mathbf{z}_n \sim \mathcal{N}(0, I)$ and their normalized average $\mathbf{z}' = f_{\text{FIX}}(\mathbf{Z}, \mathbf{w})$. The weighted sum of i.i.d. normally distributed variables is also normally distributed with adjusted variance. After normalization, \mathbf{z}' is thus uniformly distributed on a hypersphere with radius \sqrt{L} , regardless of N . In this ideal case, we do not see any degeneration effect as N grows. Hence, we can conclude that our inverted latents \mathbf{z}_n of real images are not i.i.d. $\mathcal{N}(0, I)$.

In our examples, the \mathbf{z}_n are drawn from the same ImageNet class, and that class might not align perfectly with a text concept in the diffusion model. Therefore, inverted latents might be clustered in the latent space, breaking the normality assumption. To check if the issue is caused by such a misalignment, we could instead construct examples \mathbf{z}_n using the following procedure:

1. Draw a random initial $\epsilon \sim \mathcal{N}(0, I)$, feed it through the diffusion model to create an image \mathbf{x} using the prompt *a photo of a [class name]*.
2. Perturb \mathbf{x} by adding a small amount of Gaussian noise, in order to create a new image that is not a pixel-perfect actual diffusion model output.
3. Run a diffusion inversion procedure on the perturbed \mathbf{x} to create an inverted noisy latent \mathbf{z} .

Running this N times produces N i.i.d. latents \mathbf{z}_n where there should be no clustering in the latent space caused by misalignment between input data classes and the diffusion model. Figure 3 shows an example of images generated from centroids computed from such latents. We see that the degeneration issue remains, showing that the issue is not caused by the mentioned potential misalignment.

Hence, we can conclude that inverted latents do not in general follow the statistics of random samples drawn from a normal distribution. In fact, we have noted that the channel-wise mean values of inverted latents often have a small bias that gets further amplified by the latent normalization.

This effect is studied in more detail in the following subsections.

4.2 Bias amplification

To see how a small latent bias is amplified, consider a simplified case with latents $\mathbf{z}_n = \mathbf{d} + \mathbf{e}_n$ consisting of i.i.d. noise terms $\mathbf{e}_n \sim \mathcal{N}(0, I)$ perturbed by a small common deterministic term \mathbf{d} . We now consider what happens to the normalized

mean value $\mathbf{z}' = f_{\text{FIX}}(\mathbf{Z})$ compared to the ideal unperturbed $\mathbf{e}' = f_{\text{FIX}}(\mathbf{E})$, where $\mathbf{E} = \{\mathbf{e}_1, \dots, \mathbf{e}_N\}$. Let α be the amplification factor in $f_{\text{FIX}}(\mathbf{Z})$, i.e.

$$\alpha = \frac{\sqrt{L}}{\left\| \sum_n \frac{1}{N} \mathbf{z}_n \right\|} = \frac{\sqrt{L}}{\left\| \mathbf{d} + \frac{1}{N} \sum_n \mathbf{e}_n \right\|}. \quad (5)$$

For \mathbf{z}' , we can write

$$\mathbf{z}' = \alpha \sum_n \frac{1}{N} \mathbf{z}_n = \alpha \mathbf{d} + \alpha \frac{1}{N} \sum_n \mathbf{e}_n. \quad (6)$$

To understand the amplification factor α , note that since \mathbf{e}_n are i.i.d. Gaussians with $\mathbb{E}[\|\mathbf{e}_n\|^2] = L$, then $\mathbb{E}\left[\left\| \frac{1}{N} \sum_n \mathbf{e}_n \right\|^2\right] = \frac{L}{N}$. Since L is large, $\|\mathbf{e}_n\|$ is sharply distributed, and $\left\| \frac{1}{N} \sum_n \mathbf{e}_n \right\|$ even more so. We can therefore approximate this norm with a fixed value, letting

$$\left\| \frac{1}{N} \sum_n \mathbf{e}_n \right\| \approx \sqrt{\frac{L}{N}}. \quad (7)$$

If \mathbf{d} is small, its contribution to α is negligible. Consider for example the case where $\mathbf{d} = b$ is a fixed small constant. Then $\|\mathbf{d}\| = b\sqrt{L}$, and \mathbf{d} is negligible in the α denominator if $b^2 \ll \frac{1}{N}$, which is a reasonable assumption in the situations studied in this paper. We can then approximate α as

$$\alpha \approx \frac{\sqrt{L}}{\left\| \frac{1}{N} \sum_n \mathbf{e}_n \right\|} \approx \sqrt{N}, \quad (8)$$

leading to the final approximation

$$\mathbf{z}' \approx \sqrt{N} \mathbf{d} + \mathbf{e}'. \quad (9)$$

In other words, any small common bias in the latents will be amplified by approximately \sqrt{N} in the normalization. Although this analysis used f_{FIX} , similar behavior can be expected also using f_{NIN} , since all $\|\mathbf{z}_n\| \approx \sqrt{L}$.

In Figure 4, we show an example where the measured channel-wise mean for a few examples are plotted against N . There are a few outliers, but the general trend is that the bias grows roughly linearly in \sqrt{N} , as predicted by our theory.

4.3 The origin of latent bias

It has previously been shown [13] that common implementations of diffusion model schedulers are flawed, in the sense that the latent \mathbf{z}_t does not reach zero terminal SNR at $t = T$. For e.g. the DDIM scheduler in the Hugging Face Diffusers implementation of SD 1.5, at $t = T$ the latent $\mathbf{z}_T = \sqrt{\alpha_T} \mathbf{z}_0 + \sqrt{1 - \alpha_T} \epsilon$ with $\alpha_T = 0.0047$. Even though α_T may appear negligible, $\sqrt{\alpha_T} \approx 0.07$, which is a significant number. This could certainly be a source of a deterministic trace signal in \mathbf{z}_T . In SD 3.5, this non-zero terminal SNR issue has been fixed. However, as illustrated in Figure 5, the degeneration issue remains. We hypothesize that trace amounts of latent bias could have several origins, including using imperfect schedulers and training imperfections in the noise estimation model.

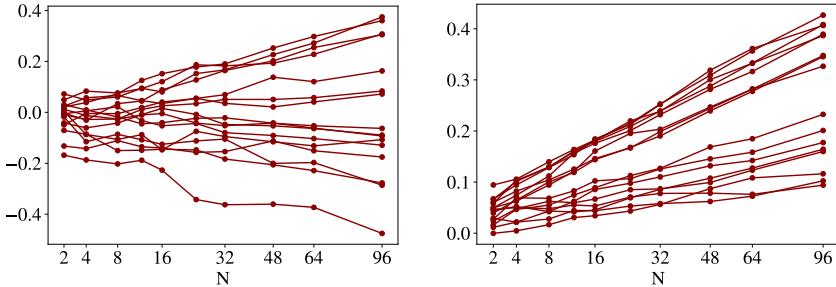


Fig. 4. Channel-wise mean of the first two channels of f_{FIX} centroids computed from N inverted input images, plotted for 8 selected ImageNet classes. Left: SD 1.5, right: SD 3.5. The x-axes have a square-root scale to illustrate the linear dependence on \sqrt{N} .

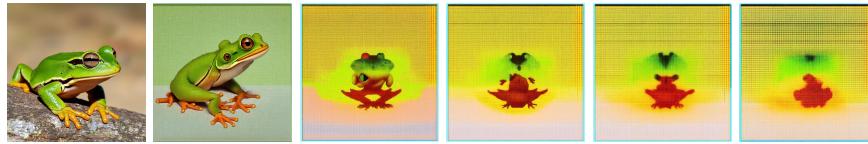


Fig. 5. Images generated from centroids created using the recipe from Section 4.1, to ensure i.i.d. examples, using SD 3.5 to avoid the non-zero terminal SNR issue. $N = 2, 8, 32, 48, 64, 96$ (increasing to the right), fixed normalization.

4.4 The effect of latent bias

To study the effect of unexpected latent statistics, we show a qualitative example with a set of images produced from the same latent noise, but where a part of the latent was perturbed by a constant offset. Specifically, in the top quarter of the image, channel 0 of the latent was offset by $-b$ and channel 1 by b , such that the global mean remained unchanged. The modified latent was then fed to the diffusion model, and the generated images are shown in the top row of Figure 6.

As a comparison, the bottom row of Figure 6 shows the effect of instead adding the same offsets to the latent at timestep 0, i.e. after the diffusion process but before decoding it into an image. In this case, only the image part corresponding to the modified latent part changes, and the effect is hardly noticeable until the offset is quite large. This indicates that the degeneration issue is not caused by latent codes getting shifted outside of their valid domain, but rather by the noise prediction model being sensitive to unexpected input statistics.

4.5 Alternative inversion procedures

Since the degenerate outputs are likely caused by inverted latents \mathbf{z}_n having unexpected statistics, one class of remedies could be to attempt improved inversion methods. Diffusion model inversion is a research direction on its own, and we will only touch briefly upon this direction here.

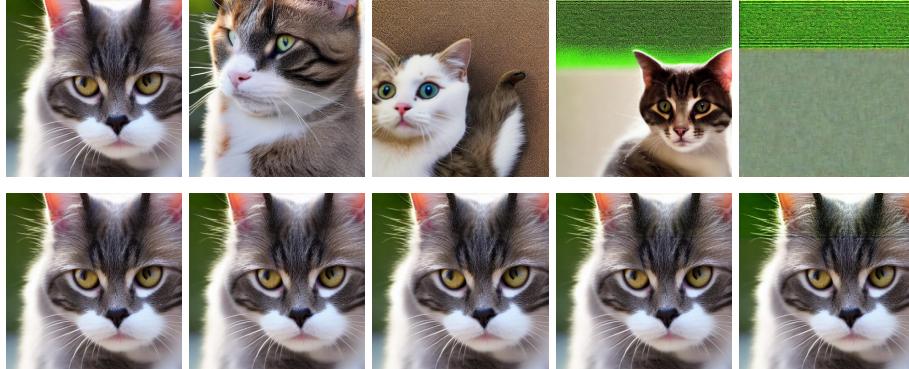


Fig. 6. Illustrating the effect of latent mean value offsets. Top row: offsets at timestep $t = T$. Bottom row: offsets at timestep $t = 0$. Columns: offsets $b = 0.0, 0.1, 0.2, 0.4, 0.8$.

First, note that DDIM inversion is an approximation of the forward process that works best without classifier-free guidance and with a large number of diffusion timesteps. We confirmed that the degeneracy issue persists also under such conditions. We also considered *null-text inversion* [17] as an example of an improved inversion procedure supporting classifier-free guidance. Also in this case, the degeneracy issue persists. More details and qualitative examples can be found in Appendix B.

5 Mean-adjusted interpolation

To summarize from last section, the degenerate outputs are likely caused by small biases from imperfections in the noise estimation model that are amplified by the necessary norm adjustment.

In order to find a suitable remedy, we first acknowledge that inverted latents \mathbf{z} may not be correctly modeled as pure noise. Instead, we suggest to model them as a sum of a deterministic part \mathbf{d} and a noise part \mathbf{e} , letting $\mathbf{z} = \mathbf{d} + \mathbf{e}$. Given some method for decomposing \mathbf{z} into \mathbf{d} and \mathbf{e} terms, we can treat \mathbf{d} and \mathbf{e} differently. The noise part \mathbf{e} is the main term, where norm adjustment is absolutely critical. Therefore, it makes sense to interpolate \mathbf{e} using one of the norm-adjustment schemes in Section 3.2, while regular linear interpolation might suffice for \mathbf{d} to avoid amplifying the bias. The final interpolated \mathbf{z}' could then be the sum of the interpolated deterministic and noise parts. In other words, let $\mathbf{d}' = f_{\text{LIN}}(\mathbf{D}, \mathbf{w})$, $\mathbf{e}' = f_*(\mathbf{E}, \mathbf{w})$, and $\mathbf{z}' = \mathbf{d}' + \mathbf{e}'$, where f_* could be any norm-adjusted interpolation. We will consider using f_{FIX} or f_{NIN} as f_* .

Note that the desired input reproduction property from Section 3.2 is fulfilled if using f_{NIN} , since if $\mathbf{w} = \{1, 0, \dots\}$, then $\mathbf{z}' = \mathbf{d}_1 + \|\mathbf{e}_1\|/\|\mathbf{e}_1\| \mathbf{e}_1 = \mathbf{z}_1$. Also note that this holds regardless of the method used for estimating \mathbf{d} and \mathbf{e} from \mathbf{z} . However, this property does not hold if we use f_{FIX} instead of f_{NIN} .

What remains to determine is a method for estimating an approximate split of \mathbf{z} into terms \mathbf{d} and \mathbf{e} . Some options could be:

1. Approximate $\mathbf{d} = 0$. This represents a baseline choice and is equivalent to simply using f_{NIN} or f_{FIX} from Section 3.3 directly.
2. Approximate \mathbf{d} as the mean value of \mathbf{z} over all channels and spatial dimensions.
3. Approximate \mathbf{d} using the mean value of each feature channel in \mathbf{z} separately, i.e. let \mathbf{d} be a constant signal over spatial dimensions but with a distinct value per feature channel.
4. Approximate \mathbf{d} using a low-pass-filtered version of \mathbf{z} .

We aim for a simple normalization scheme that can easily be integrated into other methods and opt for comparing options (2) and (3), keeping option (1) as a baseline choice. More advanced options are left as future research.

6 Experimental results

In this section, we examine experimentally whether the suggested normalization procedures reduce degeneracies. All evaluation is done on images from ImageNet, due to its wide availability. For more experimental details, see Appendix A.

Evaluation metrics. To measure image quality, we use two common measures; the FID metric [8] and CLIP [19] embedding distance. We acknowledge that the FID metric has been criticized for not always aligning well with human assessment and for being sensitive to the choice of resampling operations and number of examples, and that alternatives have been suggested [4, 10, 11]. However, as we are interested in quantifying grave degradations rather than precisely comparing the quality of competing high-aesthetic outputs, we opt for using the original metric due to its wide availability. For the CLIP distance, the CLIP embedding of images produced from latent centroids are compared to the mean image embedding of all training examples for the class using the cos distance.

Mean adjustment results. Quality metrics for the compared methods are shown in Figure 7. Here, we let `fix` and `nin` denote norm adjustment according to f_{FIX} and f_{NIN} from Section 3.3, while the suffixes `/0`, `/m` and `/chm` denote the choice of mean adjustment (none, global mean, or channel-wise mean) according to Section 5. The figure shows that the quality drops dramatically as N grows using baseline methods. The mean-adjusted options are slightly better, while the channel-wise mean adjustment options provide a significant remedy. These overall trends are similar between the FID and CLIP measures. Some qualitative examples of centroid images produced using these methods can be found in Figure 8-9, with one more example in Appendix C. In Figure 9, all examples are produced using the same N . Not all examples are degenerate using the baseline option at this N , but also for the non-degenerate examples, the apparent visual quality is better with the mean-adjusted method. We do note that the results are not always perfect. There is a measurable drop in quality also for the `chm`

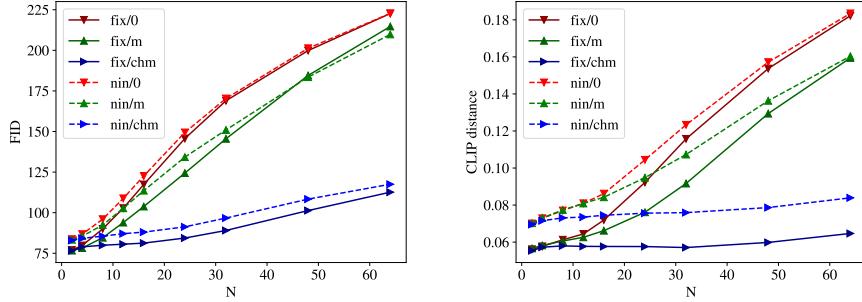


Fig. 7. Image quality of images produced from centroids of N noisy diffusion model latents, measured using the FID metric and cos distance in CLIP embedding space.

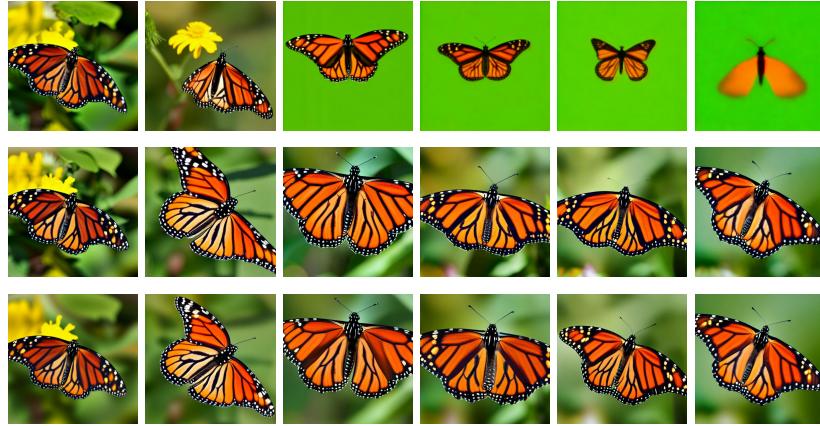


Fig. 8. Images generated from centroids of latents obtained from N input images of ImageNet class ‘‘monarch butterfly’’ for $N = 2, 8, 32, 48, 64, 96$, SD 1.5. Top row: Fixed normalization. Middle row: **fix/chm**. Bottom row: **nin/chm**.

methods as N grows sufficiently large. Qualitatively, this often manifests in over-saturated colors and loss of detail, as visible in the high N examples in Figure 1.

Figure 7 also shows that using f_{FIX} produces slightly better quality metrics than f_{NIN} most of the time, both with and without any mean adjustment in place. This was surprising to us. We hypothesize that the reason for this behavior is that the computed latent centroids are often far enough away from the input \mathbf{z}_n to render the norms of the inputs non-representative as suitable normalization targets for the centroid. Using the nominal norm \sqrt{L} seems to be a slightly better choice for such latents. However, this difference is significantly smaller than the difference caused by the degeneracy issue, and the apparent visual quality is often similar (as in Figure 8). We therefore consider both **fix/chm** and **nin/chm** to be reasonable options.



Fig. 9. Images generated from centroids of latents obtained from $N = 12$ input images of ImageNet classes “boa constrictor”, “crate”, “hourglass”, “oxygen mask”, “shark” using SD 3.5. Top row: Fixed normalization. Bottom row: `nin/chm`.

7 Discussion

We have identified and dissected a problem with interpolating diffusion model latents, and suggested a modified normalization scheme that offers a significant remedy. We consider the work foundational in nature, aiming at a better understanding of the intricacies of diffusion models and their latent spaces. As such, the work is not tied to a particular application. However, the most direct applications would be methods for image morphing using diffusion models. Fixing the degeneracies opens up for generalizing such methods to many inputs, producing latent space manifolds that can be traversed by adjusting the interpolation weights. Other potential applications are in deep data augmentation for label-constrained learning problems. Exploring such applications is a direction for future research.

One surprising discovery was that f_{FIX} leads to slightly better quality metrics than f_{NIN} . Recall that the motivation behind f_{NIN} was to ensure interpolation paths without discontinuities close to the original \mathbf{z}_n . It is possible to construct other interpolation paths that are continuous close to the original \mathbf{z}_n while approaching the nominal value \sqrt{L} when we get sufficiently far away from the original inputs. One example in this direction is the norm-aware optimization suggested by Samuel et al. [24], but that method relies on a cumbersome iterative procedure. Examining and evaluating more convenient options that can be formulated in closed form is also a subject for future research.

Acknowledgements. The research work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, and the computations by the National Academic Infrastructure for Supercomputing in Sweden (NAIIS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., Fleet, D.J.: Synthetic data from diffusion models improves imagenet classification. arXiv preprint arXiv:2304.08466 (2023)
2. Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., et al.: ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. arXiv preprint arXiv:2211.01324 (2022)
3. Buss, S.R., Fillmore, J.P.: Spherical averages and applications to spherical splines and interpolation. ACM Transactions on Graphics (TOG) **20**(2), 95–126 (2001)
4. Chong, M.J., Forsyth, D.: Effectively unbiased fid and inception score and where to find them. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6070–6079 (2020)
5. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first International Conference on Machine Learning (2024)
6. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. arXiv preprint arXiv:2208.01618 (2022)
7. Garibi, D., Patashnik, O., Voynov, A., Averbuch-Elor, H., Cohen-Or, D.: Renoise: Real image inversion through iterative noising. arXiv preprint arXiv:2403.14602 (2024)
8. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
9. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
10. Jayasumana, S., Ramalingam, S., Veit, A., Glasner, D., Chakrabarti, A., Kumar, S.: Rethinking fid: Towards a better evaluation metric for image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9307–9315 (2024)
11. Khanna, S., Liu, P., Zhou, L., Meng, C., Rombach, R., Burke, M., Lobell, D.B., Ermon, S.: Diffusionsat: A generative foundation model for satellite imagery. In: The Twelfth International Conference on Learning Representations (2023)
12. Landolsi, E., Kahl, F.: Tiny models from tiny data: Textual and null-text inversion for few-shot distillation. arXiv preprint arXiv:2406.03146 (2024)
13. Lin, S., Liu, B., Li, J., Yang, X.: Common diffusion noise schedules and sample steps are flawed. arXiv preprint arXiv:2305.08891 (2023)
14. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
15. Meiri, B., Samuel, D., Darshan, N., Chechik, G., Avidan, S., Ben-Ari, R.: Fixed-point inversion for text-to-image diffusion models. arXiv preprint arXiv:2312.12540 (2023)
16. Miyake, D., Iohara, A., Saito, Y., Tanaka, T.: Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. arXiv preprint arXiv:2305.16807 (2023)
17. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6038–6047 (2023)

18. Pan, Z., Gherardi, R., Xie, X., Huang, S.: Effective real image editing with accelerated iterative diffusion inversion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15912–15921 (2023)
19. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
20. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
21. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
22. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)
23. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems **35**, 36479–36494 (2022)
24. Samuel, D., Ben-Ari, R., Darshan, N., Maron, H., Chechik, G.: Norm-guided latent space exploration for text-to-image generation. Advances in Neural Information Processing Systems **36** (2024)
25. Sariyildiz, M.B., Alahari, K., Larlus, D., Kalantidis, Y.: Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8011–8021 (2023)
26. Shoemake, K.: Animating rotation with quaternion curves. In: Proceedings of the 12th annual conference on Computer graphics and interactive techniques. pp. 245–254 (1985)
27. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
28. Trabucco, B., Doherty, K., Gurinas, M., Salakhutdinov, R.: Effective data augmentation with diffusion models. In: Proceedings of the International Conference of Learning Representations (2024)
29. Wang, Y., Schiff, Y., Gokaslan, A., Pan, W., Wang, F., De Sa, C., Kuleshov, V.: Infodiffusion: Representation learning using information maximizing diffusion models. In: International Conference on Machine Learning. pp. 36336–36354. PMLR (2023)
30. Yang, Z., Yu, Z., Xu, Z., Singh, J., Zhang, J., Campbell, D., Tu, P., Hartley, R.: Impus: Image morphing with perceptually-uniform sampling using diffusion models. arXiv preprint arXiv:2311.06792 (2023)
31. Zhang, K., Zhou, Y., Xu, X., Dai, B., Pan, X.: Diffmorpher: Unleashing the capability of diffusion models for image morphing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7912–7921 (2024)

A Additional experimental details

All experiments were implemented in PyTorch with the HuggingFace Diffusers library. Except where otherwise stated, examples and results were produced using Stable Diffusion 1.5. Initial preliminary experiments showed similar effects across versions 1.4, 1.5, 2.1 and 3.5.

All experiments used the default classifier-free guidance values of 7.5 for SD 1.5 and 4.5 for SD 3.5, except where stated that no guidance was used. Experiments using SD 3.5 used the *medium* model size.

The null-text inversion experiments were run using 32-bit float precision, due to the use of optimization using backpropagation. The high-iteration DDIM-inversion experiments with SD 1.5 also used 32-bit precision in order to get good numerical conditions. The high-iteration DDIM inversion experiments with SD 3.5 used 16-bit precision due to GPU memory constraints (we wanted all experiments to be runnable using 24 Gb of VRAM). All other experiments were run with 16-bit float precision in the image generation but 32-bit precision in the interpolation operations in order to ensure that the interpolation was not affected negatively by the limited precision. Finally, we note that the observed bias (that is amplified by the normalization), is significantly larger than the 16-bit floating-point epsilon.

The final results in Section 6 were produced by comparing 1000 computed centroids from up to 64 examples of one class from the training split of ImageNet, averaged over 10 randomly drawn classes. This experiment required around 300 GPU-hours using NVidia A40 GPUs. The additional compute required for the suggested normalization was negligible compared to running the diffusion model.

B Alternative inversion procedures

To see if the degenerate behavior persists across more inversion procedures, we run two qualitative experiments. The first experiment used DDIM inversion with 500 iterations and without classifier-free guidance. This experiment was run using both SD 1.5 and 3.5, where the 3.5 results are also free from the non-terminal SNR issue mentioned in Section 4.3. A qualitative example is shown in Figure 10. As expected, the image quality is worse than when using guidance, but the point here is to show that results are still degenerate for large N .

The second experiment used null-text inversion [17]. Latents were interpolated using f_{FIX} and unconditional embeddings using f_{LIN} , following prior work using linear interpolation for conditioning inputs [30, 31]. A qualitative example is shown in Figure 11, where the degeneration issue clearly still exists.

C Additional qualitative example

Figure 12 shows an additional example with images generated from centroids computed using increasing N , this time using SD 3.5. Similarly to Figure 8, the degeneracy is greatly reduced using channel-wise mean adjustment, and the difference between the `nin` and `fix` is marginal in comparison.

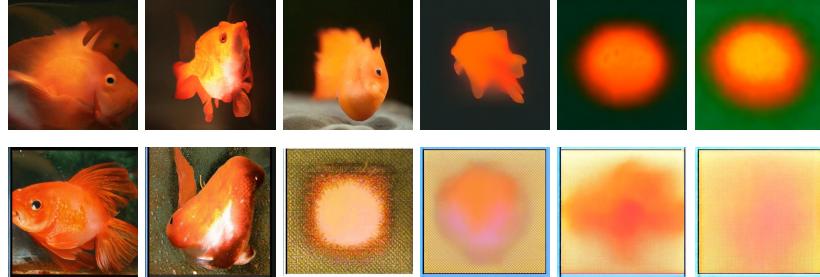


Fig. 10. Images generated from centroids using 500 diffusion timesteps without classifier-free guidance, showing that the degeneration issue persists also under more ideal DDIM inversion conditions. $N = 2, 4, 8, 16, 32, 64$ (increasing to the right), fixed normalization, ImageNet class “goldfish”. Top row: SD 1.5, bottom row: SD 3.5.

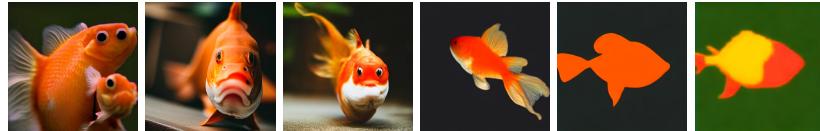


Fig. 11. Images generated from centroids computed using null-text inversion, showing that the degeneration issue still persists. $N = 2, 4, 8, 16, 32, 64$ (increasing to the right), fixed normalization, ImageNet class “goldfish”.



Fig. 12. Images generated from centroids of latents obtained from N input images of ImageNet class “baboon” using SD 3.5. $N = 2, 8, 32, 48, 64, 96$. Top row: Fixed normalization. Middle row: `fix/chm`. Bottom row: `nin/chm`.