Preserving Image Properties Through Initializations in Diffusion Models

Jeffrey Zhang
jeff@revery.ai

Shao-Yu Chang
shaoyuc3@illinois.edu

Kedan Li
kedan@revery.ai

David Forsyth
daf@illinois.edu

Abstract

Retail photography imposes specific requirements on images. For instance, images may need uniform background colors, consistent model poses, centered products, and consistent lighting. Minor deviations from these standards impact a site's aesthetic appeal, making the images unsuitable for use. We show that Stable Diffusion methods, as currently applied, do not respect these requirements. The usual practice of training the denoiser with a very noisy image and starting inference with a sample of pure noise leads to inconsistent generated images during inference. This inconsistency occurs because it is easy to tell the difference between samples of the training and inference distributions. As a result, a network trained with centered retail product images with uniform backgrounds generates images with erratic backgrounds. The problem is easily fixed by initializing inference with samples from an approximation of noisy images. However, in using such an approximation, the joint distribution of text and noisy image at inference time still slightly differs from that at training time. This discrepancy is corrected by training the network with samples from the approximate noisy image distribution. Extensive experiments on real application data show significant qualitative and quantitative improvements in performance from adopting these procedures. Finally, our procedure can interact well with other control-based methods to further enhance the controllability of diffusion-based methods.

1. Introduction

Stable Diffusion [9] can generate high-quality, lifelike images, and has opened up numerous innovative applications. Examples include creating new art, style transfer between pictures, and generating high-resolution images from text. However many real applications require images to meet specific design requirements. For example, product images need consistent photographic standards so that retail websites maintain uniform aesthetic appeal and are "on-brand". Even minor deviations can make images unusable.

The essential requirements for our application mirror those of many other applications. Our application requires a text-to-image generator where: **(1) outputs reflect a garment description text accurately**; **(2) outputs are either a garment image or a human model wearing the described garment**; **(3) garments are not cropped, are centered, and appear on a white background**; **(4) human models are always depicted from foot to neck, stand in similar poses, and appear on a neutral background**; **(5) outputs have consistent professional lighting and shading**; and **(6) one text-to-image model can produce all desired images and does not produce others**. The first five requirements ensure images are "on-brand" and the last is for efficiency. Remarkably, Stable Diffusion [9] as currently practiced cannot meet these requirements, but quite simple changes result in a model that does.

We start from the observation that fine-tuning Stable Diffusion [9] with product images on neutral backgrounds *does not* produce a method that can generate product images on neutral backgrounds (Fig. 1). This unexpected effect is caused by a hiccup in the structure of the method. Current training methods [5, 11] form a weighted sum of noise and a base image (using a weight α), and a model is trained to denoise the noisy image. At inference, one assumes that a sufficiently noisy base image is indistinguishable from noise and that the denoising process can be started with pure noise. We show the assumption is true only for α much smaller than those used in current practice, meaning that the denoiser sees noticeably different distributions at train and test times. Training Stable Diffusion [9] for very small α is also challenging (see Supplementary).

A simple alternative is to initialize with a draw from a distribution that represents the training distribution reasonably well and is easily sampled. While training, some information about the original image can be recovered from the initial noisy sample, which at best is a blurry version of the original image. This means that a heavily noised sample from a mixture of principal components model is an acceptable approximation of the training distribution. We show significant improvement results from using this as an initial distribution *without* retraining Stable Diffusion [9] – for instance, erratic backgrounds are replaced by neutral backgrounds (see Supplementary). But these improvements highlight another initialization problem: the *joint* distribution between text and noisy image at training is misrepresented by both standard noise initialization and our initialization. We show that because our initialization is easily sampled, it can be used in fine-tuning the denoiser, leading to notable improvements in text-based control. Finally, we show that our initialization techniques are easily integrated with other controllability methods (e.g. ControlNet [14]) to provide more effective control for diffusion-based methods.

2. Related work

There is considerable research available on generating specific concepts or subjects using diffusion-based methods. However, to the best of our knowledge, we believe that we are the first to concentrate on creating specific image distributions for images.

2.1. Object preservation and harmonization

There are works that learn specific objects and generate variations of those objects faithfully. Dreambooth [10] fine-tunes a pretrained diffusion model to accurately generate new variations of a particular subject. Gal *et al*. [4] invert objects into pseudo-words to attain personalized text embeddings to create images of those objects. Other works have enabled editing in the forward pass of diffusion models without image-specific fine-tuning or inversion. InstructPix2Pix [2] takes an input image and generates a new image based on text instructions. Yang *et al*. [13] proposes an exemplar-based image editing model where the reference image is semantically transformed and harmonized into another image. Finally, Edward *et al*. [6] adapts the language models to generate specific objects by adding trainable parameters of the language embeddings to learn new concepts from a dataset. This allows the adaptation of new words and concepts by fine-tuning the newly added parameters instead of the entire generation model.

These studies primarily focus on preserving target objects and generally struggle to control non-target areas if no conditions exist in those areas. Unlike these approaches, our paper emphasizes stabilizing the image distributions throughout the diffusion process. Our proposed method can effectively preserve the properties of the entire image, not just the target objects.

2.2. User-defined controllablility

Many of the works mentioned above are text-guided, in which users provide a text prompt to control and edit images. However, language-guided manipulations often do not generate images satisfying users' requirements. CLIP features [8] leverage the representations of user-provided images to improve the diversity of the output results. Region-based image editing methods [1, 3] treat the task as a conditional inpainting task with a mask highlighting the regions of the images that needed to be edited while preserving non-target areas. To enhance task-specific control of diffusion models, ControlNet [14] adds an additional input condition (e.g. edge maps, segmentation masks, keypoints, etc.) alongside text prompts to manipulate image generation.

While these methods allow users to control target areas in the images by adding additional conditions, they still often fail to maintain image distributions. This failure is due to inconsistencies in the initialization process during training and inference. In Sec. 4.4, we show that combining our method with ControlNet [14] can strengthen the control abilities of diffusion models and stabilize the output (Fig. 5).

2.3. Image-to-Image Translation

Finally, Stable Diffusion [9] is often applied to image-to-image translation applications by choosing a starting image and generating variations from the image initialization. The resulting images have significant similarities to the initial image's colors and contours. On the other hand, other methods use DDIM Inversion [11] to find initial noise vectors to restore the original image during diffusion to apply image-to-image translation. In Tune-A-Video [12], the authors use DDIM Inversion to control the consistency of frames and the contours of objects. In Null-text Inversion [7], DDIM Inversion is used to create images similar in appearance to the original input, enabling users to edit specific words while preserving the objects of the original image.

In these works, reference images are used to generate variations of that reference image. In contrast, we demonstrate a sample from an approximate noisy image distribution significantly changes the behavior of diffusion-based methods because the method experiences similar samples from the training distribution at inference time. Furthermore, we show that using the right starting initialization during both training and inference is essential for consistently generating entire image distributions, not just for maintaining the features of a specific reference image.

3. Method

Current literature on Stable Diffusion [9] has assumptions at inference time that appear inconsequential, but we show these assumptions have significant consequences. We demonstrate substantial improvements in resolving these errors at inference time.

3.1. Background

Stable Diffusion [9] is trained to recover an image x_0 by denoising a noisy image x_t at timestep $t \in [0, T]$. At the t'th timestep, the denoiser is presented with

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{(1 - \alpha_t)} \epsilon_t \tag{1}$$

where $\epsilon_t \sim \mathcal{N}(0, I)$ and α_t is the cumulative product of scaling at each timestep t (refer to DDIM [11]). Following the training procedure from [11], a denoiser f predicts the added noise $\hat{\epsilon}_t = f(x_t, t, e; \Theta)$, where f is parameterized by Θ and takes in noisy image x_t, timestep t, and conditional encoding e. We write $\hat{x}_0$ for the predicted ground truth image derived from removing the predicted noise $\hat{\epsilon}_t$ from x_t. Our loss is

$$\mathcal{L} = \mathbb{E}[||\epsilon_t - \hat{\epsilon}_t||_2^2] \tag{2}$$

From Eq. 1, if we have the predicted $\hat{\epsilon}_t$ and the noisy image x_t, we can derive the predicted ground truth image $\hat{x}_0$ with

$$\hat{x}_0 = (x_t - \sqrt{(1-\alpha_t)}\epsilon_t)/\sqrt{\alpha_t} \tag{3}$$

3.2. Inference Assumption

During training, we have ground truth image $x_0 \sim P(images)$ and initial noisy image $x_T \sim P_T$ (from Eq. 1). However, at inference time, we do not have the ground truth image x_0 and must supply an alternative initialization x_{init}. Following [11], it is usual to argue that for sufficiently small α_T, x_T should be very similar to $\mathcal{N}(0, I)$. Hence, during inference, a common approach is to sample our initialization $x_{init} \sim \mathcal{N}(0, I)$.

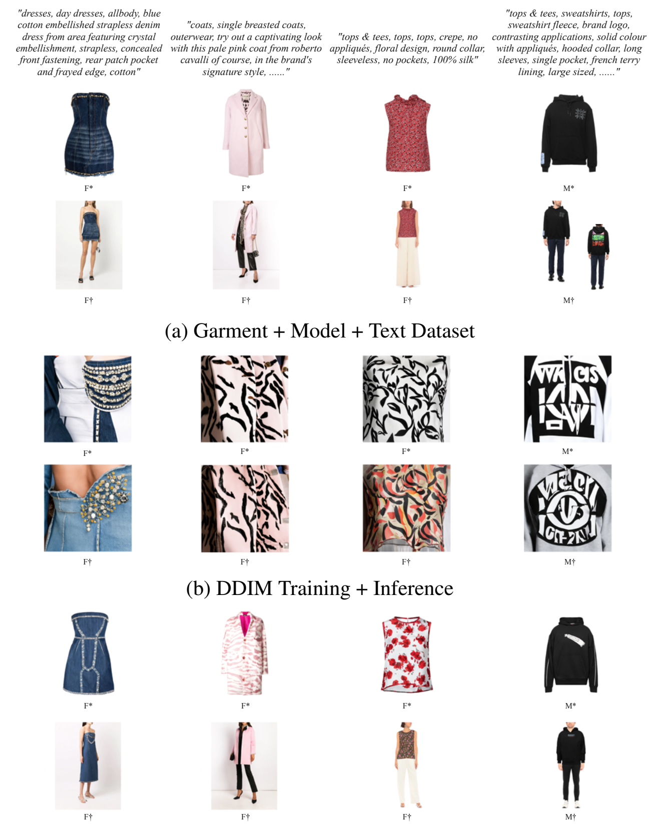
However, if α_T is not small enough, then x_T has information about x_0 that the network could recover and utilize for denoising. Pure noise as initialization may not behave as expected for two reasons: (1) denoising networks are trained on noticeably different data distributions than what they see at inference time and (2) denoising networks may extract information about x_0 from x_T to denoise x_T. If this is true, reliable inference procedures might use something other than $\mathcal{N}(0, I)$.

We show values of α_T in many pretrained models may indeed be too large. In Fig. 1, we compare the performance of a Stable Diffusion [9] fine-tuned to make images of models and garments on two different initializations. In Fig. 1b, we initialize $x_{init} \sim \mathcal{N}(0, I)$ during inference, and in Fig. 1c, we initialize with $x_{init} = x_T$ during inference using Eq. 1, where x_0 is the ground truth image from Fig. 1a. Different initializations have significantly different results, but more importantly, notice the network takes obvious hints from the ground truth image x_0 (in Fig. 1c).

For values of α_T that are not small enough, it is possible to reliably distinguish between samples from P_T and $\mathcal{N}(0, I)$ using elementary methods. For $x_{init} \sim \mathcal{N}(0, I)$ to be hard to distinguishable from $x_T \sim P_T$, we must have

$$\alpha_T < O(\frac{1}{d}), \tag{4}$$

where $d = H \times W \times C$ is the dimension of the sampled image. Meeting this constraint is difficult as $1/d$ is significantly smaller than current values of α_T. The key point is



(a) Garment + Model + Text Dataset

(b) DDIM Training + Inference

(c) DDIM Training + Control Inference ($x_{start} = x_0$)

Figure 1. Despite training on images with properties (1)-(5), normal diffusion-based training and inference lead to unexpected results. (a) shows sample sequences from our garment dataset. (b) shows standard fine-tuning and inference results with Stable Diffusion [9] do not generate the same distribution of images despite being trained on images from (a). The prompts are taken from training data, where we expect the best results. To show that this is not a training error, in (c), we set a control experiment by changing x_{start} (Eq. 10) to the training image shown in (a). The generated images match the training distribution, indicating that initialization information strongly influences results. (F*: "female garment, no person, white background"; M*: "male garment, no person, white background"; F†: "female person wearing garment"; M†: "male person wearing garment")

that spatial averages of images have strong properties and will be perceptible even at small α_T.

Set $\mathbf{a} = \frac{1}{d}\mathbf{1}$ where $\mathbf{1}$ is the 1-vector of size d. Then, we can write

$$\mathbb{E}_{P_T}[\mathbf{a} \cdot x_T] = \sqrt{\alpha_T}\mathbb{E}_{P(images)}[\mathbf{a} \cdot x_0] = \sqrt{\alpha_T}\mu \neq 0, \tag{5}$$

where $\mu = \mathbb{E}_{P(images)}[\mathbf{a} \cdot x_0]$ is the average of images. Simple experiments show that μ is non-zero, and different sets of images can have different values of μ (e.g., white background, people in similar poses, etc.). From Eq. 5, $\mathbf{a} \cdot x_T$ has mean $\mu_T = \sqrt{\alpha_T}\mu$ and variance $\sigma_T^2 = \alpha_T\sigma^2 +$

$\frac{(1-\alpha_T)}{d}$. But for $x_{init} \sim \mathcal{N}(0, I)$, $\mathbf{a} \cdot x_{init}$ has mean 0 and $\sigma^2 = 1/d$.

Hence, the network is presented with samples from two distributions, $\mathcal{N}(\mu_T, \sigma_T^2)$ during training and $\mathcal{N}(0, \sigma^2)$ during inference. For the distributions to be difficult to distinguish, we want μ_T/σ and μ_T/σ_T to be small. We have

$$(\frac{\mu_T}{\sigma})^2 = d\alpha_T(\mu)^2, \tag{6}$$

and

$$(\frac{\mu_T}{\sigma_T})^2 = \frac{(\sqrt{\alpha_T}\mu)^2}{\alpha_T\sigma^2 + (1-\alpha_T)\frac{1}{d}} = \frac{d\alpha_T\mu^2}{\alpha_T(\sigma^2 d - 1) + 1} \tag{7}$$

Consequently, both Eq. 6 and 7 are small if α_T is less than $1/d$ or smaller, giving us Eq. 4.

Practical numbers are $d = 64 \times 64 \times 4 = 16384$ and $\alpha_T = 0.0047$; but $0.0047 \not< 0.000061$, hence α_T is not in the right range (Eq. 4). This indicates that a denoiser network f could tell the difference between the initialization samples used in training and those used at inference. If it can tell the difference, different behaviors between training and inference are possible. Fig. 1 demonstrates the network actually behaves differently for samples with these different distributions.

One solution is to train with a smaller α_T. However, training with smaller α_T requires scaling down α_t for many timesteps, which leads to more difficult training as more noise is added (see Supplementary). We show that approximating $P(images)$ offers a more efficient and reliable solution.

3.3. PCA-K Offset Inference

Our procedure **"PCA-K Offset Inference"** initializes inference with:

$$x_{init} = \sqrt{\alpha_T}x_{start} + \sqrt{1-\alpha_T}\epsilon_T, \tag{8}$$

where x_{start} is sampled from a distribution Q that approximates $P(images)$. Q does not need to be a particularly strong approximation of $P(images)$ because a large magnitude of noise is added in x_T. Because this noise is i.i.d. Gaussian noise, we expect that the information the network can extract about x_{start} from x_{init} is a heavily smoothed version of x_{start}. Hence, we need a distribution model that is easy to sample, reasonably approximates blurry images, and can handle multiple classes.

PCA-K Offset Inference uses a mixture of normals for Q, where each normal is derived from Principal Component Analysis (PCA) of images of its class c. PCA is known to be an effective description of blurred images. K represents the number of principal components. For each class c in our dataset, we model an image as

$$x_R^c = \mu^c + \sum_{i=1}^{K} \xi_i \mathbf{p}_i \tag{9}$$

where $\xi_i \sim N(0, \lambda_i)$, $\mathbf{p}_i$ are orthonormal principal components, and μ_c is the mean image of class c. Write x_R for a random image drawn from an R principal component model. Then setting $x_{start} = x_R^c$, our initialization becomes:

$$x_{init} = \sqrt{\alpha_T}x_R^c + \sqrt{1-\alpha_T}\epsilon_T \tag{10}$$

We call this very useful case where $R = 0$, x_{start} is the class mean μ^c, **"Mean Offset Inference"**.

3.4. PCA-K Offset Training

While PCA-K Offset Inference allows the inference procedure to mimic the training procedure much more closely, we still expect operating conditions to differ. Our approximation may not exactly match the distribution used in training. In particular, our approximation may not preserve the delicate relationship between the ground truth image x_0 and the text encoding e supplied during training. Fig. 1 shows improvements by using an approximate distribution during inference time that was trained as usual, but further improvements are available. At training, the network sees a noisy version of an image and models a complex relationship between image and text encoding. But at inference, our network will be presented with a text encoding e and a sample x_{start} from the initial distribution, which has not been conditioned on e. We cannot guarantee an approximate distribution can preserve those relationships. We can, however, use the same approximate distribution during training so that the network experiences the same distribution at train time and test time. This is a significant advantage of a Q that is easy to sample.

We use x_{init} from Eq. 8 in place of the initialization x_T in Eq. 1 to resemble the desired start point for image x_0. This allows the network to train and infer from the same distribution for the first step of the diffusion process:

$$x_T^{new} = \sqrt{\alpha_T}x_{start} + \sqrt{1-\alpha_T}\epsilon_T \tag{11}$$

First, we want the denoiser to recover the ground truth image x_0 from x_T^{new}. So, we alter the noise objective to

$$\epsilon_T^{new} = (x_T^{new} - x_0\sqrt{\alpha_T})/(\sqrt{1-\alpha_T}) \tag{12}$$

Second, we want to skip timesteps (for computation efficiency, as in DDIM's [11]), so this change must be applied to multiple timesteps. Let S be the number of skips per timestep. Then, we apply the mean offset training to all timesteps within the first skip step to guarantee the first skip step is trained with the mean offset initialization. Hence, we alter Eq. 1 and Eq. 12 for $t \geq T - S$ to

$$x_t^{new} = \sqrt{\alpha_t}x_{start} + \sqrt{1-\alpha_t}\epsilon_t \tag{13}$$

and

$$\epsilon_t^{new} = (x_t^{new} - x_0\sqrt{\alpha_t})/(\sqrt{1-\alpha_t}) \tag{14}$$

Thus, our final loss is a combination of Eq. 2 and the new noise objective from Eq. 14:

$$\mathcal{L}_{new} = \begin{cases} \mathbb{E}[||\epsilon_t - \hat{\epsilon}_t||_2^2] & \text{if } t < T - S \\ \mathbb{E}[||\epsilon_t^{new} - \hat{\epsilon}_t||_2^2] & \text{if } t \geq T - S \end{cases} \tag{15}$$

During training, we project a ground truth image x_0 with class c into x_K^c and set $x_{start} = x_K^c$ from Eq. 9, giving our proposed **"PCA-K Offset Training"** procedure. Setting $K = 0$ gives us $x_K = \mu^c$, which is simply just initializing $x_{start} = \mu^c$ and is our **"Mean Offset Training"** procedure. We find Mean Offset Training works best compared to higher K values and is much simpler in practice (see Supplementary). As a result, we use Mean Offset Training for results in the main text for the sake of simplicity.

4. Experiments and Results

In these experiments, we use data collected from retailers that follow properties specified in Sec. 1 (details in Sec. 4.1). We show PCA-K Offset Inference behaves better because the initialization is similar to the training distribution in the initial denoising timesteps (Sec. 4.2). We show sampling a bad initialization from PCA-K can damage the relationship between text and initialization during inference because the operating conditions are still different (Sec. 4.2 and Fig. 3). We show that incorporating our PCA-K Offset Training procedure fixes this issue (Sec. 4.3 and Fig. 4). Finally, we show other control methods experience the same initialization problem, and our procedures can be easily combined with other methods to provide further control in generation (Sec. 4.4).

4.1. Dataset

We collect over a million image pairs of retailer garment, garment on model, and garment text description triplets (Fig. 1a). We are given one text prompt corresponding to a garment image and a model wearing that garment. All training data triplets satisfy **properties (1)-(5)** described in Sec. 1. Our task is to generate images of garments and fashion models wearing garments that satisfy all properties (1)-(6). To distinguish between generating garments and models wearing garments, we prepend the caption with "male/female garment, no person, white background" (denoted with **M*** and **F***, respectively) for generating garments and "male/female person wearing garment" for generating human models (denoted with **M†** and **F†**, respectively).

For inference, we collect 24 different freeform text descriptions of garments by asking fashion designers to describe diverse garment descriptions (see Supplementary). These are fictional garment descriptions used to test the generalizability of our text-to-image model.

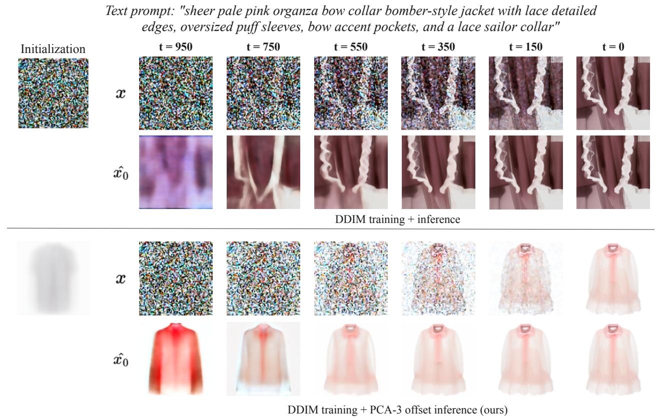


Figure 2. Intermediate outputs for a $S = 20$ DDIM training process are visualized to show the first step of the diffusion process is out of distribution for standard DDIM training + inference. The top two rows show the intermediate outputs when initializing with noise (DDIM inference). The bottom two rows show the intermediate outputs when projecting a gray sweater with PCA-3 Offset Inference. Rows 1 and 3 show the noisy image x_t and rows 2 and 4 show the predicted $\hat{x_0}_t$ at each time step t. We can see from row 2 that the first predicted $\hat{x_0}$ introduces a dark, non-uniform background that is propagated throughout the process, whereas in row 4, the predicted $\hat{x_0}$ is already close to the desired distribution, making the diffusion process is much more stable.

4.2. DDIM Finetuning with PCA-K Offset Inference

To set a baseline, we fine-tune sd-v1.5 [9] on our dataset and show results from different initializations in Fig. 1. We fine-tune for 50,000 steps on a batch size 16 and learning rate 1e-5. We use a set total number of timesteps $T = 1000$ for training and skip timesteps $S = 50$ for inference (i.e., 20 total timesteps for inference). We indicate the standard training and inference procedure from [11] as **DDIM training** and **DDIM inference**, respectively.

Different inference initializations have qualitatively different effects. Fig. 1b shows that if we initialize with noise (DDIM inference), none of the generated images respect **properties (1)-(6)** despite being fine-tuned on images with those properties. These images are not curated; generated images hardly ever show isolated garments or models. Furthermore, the text prompts used for inference *were taken from the training data*, where we expect the best behavior. To show that this is not a training bug, we set x_{start} in Eq. 8 to the ground truth dataset images shown in Fig. 1a. The fine-tuned network can now generate images that satisfy all our desired image properties (Fig. 1c). This indicates that x_{start} in the initialization heavily impacts the denoising process, and the assumption that $\mathcal{N}(0, I)$ is close enough to P_T has clear implications during inference (using the actual ground truth is not the issue here; below and Fig. 3).

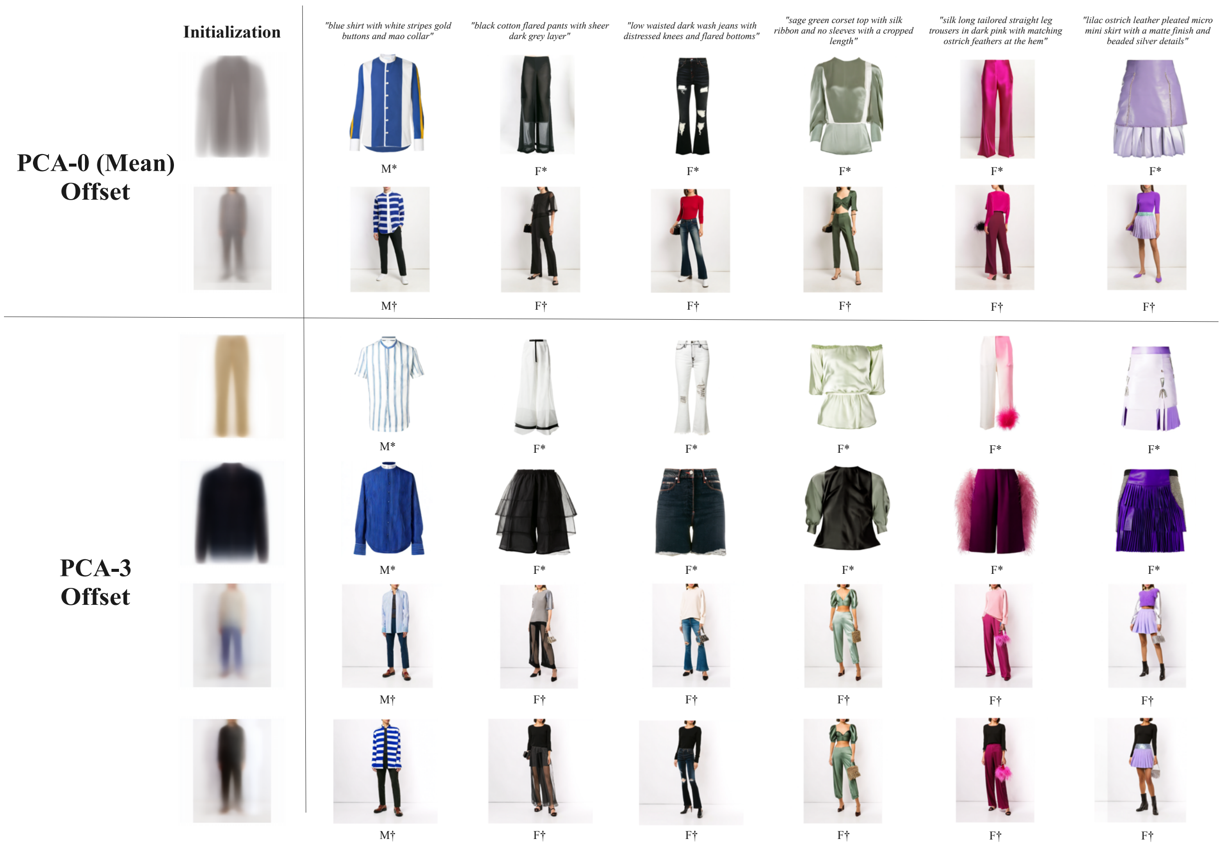


Figure 3. We show applying our PCA-0 and PCA-3 Offset Inference on DDIM Training can significantly improve generating desired image properties but is strongly biased by the sampled initialization x_{start}. This leads to some undesirable artifacts. Rows 1 and 2 show PCA-0 occasionally generates non-white backgrounds for garments due to faint sleeves in the mean image - violating **property (3)**. In rows 3-6, generated images are strongly influenced by the color and shape of x_{start} and "...black cotton flared pants..." are generated to be white, "...tailored straight leg trousers..." are generated as shorts, etc. This violates (**property (1)**) as the generated images do not respect the text and further indicate that x_{start} strongly influences the denoiser. Descriptions are freeform text from fashion designers.

Visualizing the mechanism by displaying intermediate time steps in generation helps to understand the impact of different initializations better. In Fig. 2, we compare the results between random noise initialization (DDIM training + inference) and initializing with a gray garment from our dataset projected to 3 PCA components (PCA-3 Offset Training + Inference) on freeform text from fashion designers. We see that during the first few steps of the diffusion process, a random noise input will predict a x_0 that is very different from our desired image distribution. This mistake is not corrected in later timesteps and is accumulated throughout the denoising process. This is because the denoiser is not trained to denoise an image from a non-white background. If we initialize with a PCA-projected garment image with a white background, then the training data distribution is maintained in all intermediate steps.

PCA-K Inference fixes distribution issues, but initializations strongly affect text control in generations. We apply PCA-K Inference on the fine-tuned model and show image distribution problems are mitigated in Fig. 3, but the starting point can bias the type of images generated. Notice the initialization alters the shape and color of the garment generated because the network f is trained to take hints from this initialization. The lighter initialization in row 3 generates more light colors, while row 4 shows much darker and boxier generations that adhere to the color and shape of the initialization. Notice that the generated garments do not respect the text ("black cotton flared pants..." are generated as white in the third column, "...straight leg trousers..." are generated as shorts in the sixth column, etc.), thus not satisfying **property (1)**. We try to apply a more neutral initialization by setting x_{start} to garment and model means. However, Fig. 3 row 1 shows artifacts in the background due to the faint sleeve of the mean garment image.

The denoiser clearly takes hints from the initialization when generating images. This further substantiates that x_0 has a tremendous weight during training. The denoiser heavily relies on cues from x_0 to denoise the image because

(a) DDIM Training + Mean Offset Inference

(b) Mean Offset Training + Inference

Figure 4. Using Mean Offset Training and Mean Offset Inference provides better text control because the relationship between initialization and text is preserved during training and inference. We apply two class mean initialization for garments and models and intentionally swap the means to test the effect of different initializations during inference. Figure (a) shows garment and model results DDIM Training + Mean Offset Inference that violate various properties. Figure (b) shows Mean Offset Training + Inference results satisfy all desired properties. Red boxes highlight generation errors in (a) and green boxes show they are fixed in (b). Red solid borders show artifacts that shouldn't exist and don't fully respect the text ((a) fails **property (1)**). Red dashed borders show generated models instead of garments, as specified by the text, and the person is not in the proper pose ((a) fails **properties (1) and (4)**). Red dotted borders show non-white backgrounds or cropped garments/models ((a) fails **property (3)**).

x_0 strongly correlates with the encoding text e. Unfortunately, x_{start} is sampled from a distribution Q that is independent of the text prompt. As a result, the word control for denoising worsens when we sample from Q.

Method	CLIP similarity(↑)	Score scaled by GT(↑)
Ground Truth Garments	0.294	1.0
DDIM Training + DDIM Inference	0.2542	0.865
DDIM Training + Mean Offset Inference (Ours)	0.2761	0.939
Mean offset Training + Inference (Ours)	**0.2812**	**0.956**

Table 1. We use CLIP similarity [8] (higher is better) between images and text to show our methods generate images that respect text better (**property (1)**). We compare with the ground truth image from our dataset to set a baseline. Notice our Mean Offset Inference easily outperforms standard DDIM training and inference. Furthermore, incorporating DDIM Training + Inference further improves performance, indicating that a better text-to-generation relationship is preserved (for property (1)).

4.3. PCA-K Offset Training

By incorporating PCA-K Offset Training, we alter the training procedure to have consistent initialization and text pairings during training. We test Mean Offset Training (PCA-0) with average garment and average model initializations. Training hyperparameters are identical to the fine-tuned model in Sec. 4.2. Results for PCA-K ($K > 0$) are shown in Supplementary.

Qualitatively, Mean Offset Training generates images that respect the text better than standard training and satisfies all desired properties (1)-(6). Fig. 4 shows the difference between using DDIM training + Mean Offset Inference (Fig. 4a) and Mean Offset Training + Inference (Fig. 4b). While only Mean Offset Inference significantly helps generate our desired image properties, it occasionally produces artifacts (not pure white backgrounds), does not accurately follow the text (generates sleeves when there shouldn't be sleeves), and crops the garments and models. Incorporating mean offset initialization in training resolves these issues and generates our desired image distribution (see results with all 24 text prompts in the Supplementary).

Quantitatively, Mean Offset Training + Inference generates more accurate images as demonstrated by running CLIP similarities [8] between generated garments and text prompts in Table 1. Notice the 10.6% improvement from DDIM Training + DDIM Inference to Mean Offset Training + Inference. While CLIP similarity is not a perfect representation of similarity, the improvement is significant. The learned text-to-initialization relationship is better preserved because the same initialization distribution is used during training and inference.

4.4. Application to ControlNet

We apply our method to ControlNet [14] for a different task of virtual try-on using our dataset. We show image distribution issues persist in this new task due to noise initialization during inference, but can be fixed with our PCA-K Offset Training + Inference. For this task, we are given a garment as control and train a denoiser to generate a realistic person wearing that garment. To train, we mask the

Figure 5. We adapt ControlNet [14] to take a garment condition to generate models wearing garments. We display three seeds for the same control to show that vanilla ControlNet (DDIM Training + Inference) consistently produces out-of-distribution results (violating **properties (4) and (5)**), whereas ControlNet with Mean Offset Training + Inference (Ours) perfectly preserves the desired training distribution.

region of a garment from a person in our dataset and adapt ControlNet [14] to take the masked garment image as the condition to generate the remaining image. The left column of Fig. 5 shows the effect of training and running ControlNet without any modification to the initialization procedure, and background and lighting properties are not preserved (**properties (4) and (5)**). The right column of Fig. 5 shows that applying our Mean Offset Training + Inference preserves desired generated image properties.

5. Discussion

We believe our approximate initialization distribution has broad applications, not just limited to fashion retail images. Situations where images follow structural requirements could benefit from our training and inference procedure. Additionally, because sampling initialization can bias the inference (Fig. 3), we intend to investigate using CCA to build relationships between PCA-K initializations and text features.

6. Conclusion

Our research indicates that existing training and inference procedures for diffusion-based methods are problematic and cannot preserve certain image distributions. We uncover that the assumption of employing random noise as the starting point may significantly affect the way images are generated. More importantly, we show the current training procedure is largely biased by its initialization, but can be mitigated by adopting our PCA-K Offset Training + Inference. Finally, we demonstrate that our work is orthogonal to other manipulation methods, such as ControlNet [14], and can be combined to enable greater control of diffusion-based image generation.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218, June 2022. 2

[2] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. November 2022. 2

[3] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. 10 2022. 2

[4] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. 2

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. 1

[6] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 2

[7] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022. 2

[8] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. 04 2022. 2, 7

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1, 2, 3, 5

[10] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022. 2

[11] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, October 2020. 1, 2, 3, 4, 5

[12] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *arXiv preprint arXiv:2212.11565*, 2022. 2

[13] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. 11 2022. 2

[14] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2, 7, 8