Counterfactual Emotion Generation via Sequential Monte Carlo Guided Latent Diffusion Models

Arthur Paing **Côme Rochas**

December 14, 2025

Abstract

The generation of counterfactual explanations for high-dimensional data, such as images, remains a challenging task in computer vision and interpretability. Our goal is to modify a high-level attribute of an instance, such as the emotion of a face, while preserving its other intrinsic characteristics (identity and background mostly). While Generative Adversarial Networks (GANs) have historically dominated this field, Diffusion Models have recently emerged as a superior alternative due to their training stability and mode coverage. In this work, we propose a framework for counterfactual emotion generation using Latent Diffusion Models (LDM) with a Sequential Monte Carlo (SMC) inference strategy rather than classifier-guided generation. We demonstrate that this approach allows for robust editing of facial expressions on the CelebA-HQ dataset, effectively balancing the trade-off between target class fidelity and identity preservation.

1 Introduction

Deep generative models have revolutionized the way we synthesize and manipulate visual data. A particularly impactful application of these models is the generation of *counterfactuals*. In the context of facial analysis, a counterfactual query asks: "What would this person look like if they were smiling instead of ?", assuming all other factors such as lighting, pose, or identity remained constant. This capability is not only relevant for creative applications in entertainment and media but serves as a crucial tool for AI interpretability and bias mitigation.

Historically, Generative Adversarial Networks (GANs) have been the state-of-the-art for such tasks. However, GANs suffer from well-known limitations, notably mode collapse and training instability. Furthermore, projecting a real image back into the GAN's latent space (GAN inversion) to edit it is often imperfect, leading to artifacts or loss of identity.

Recently, Denoising Diffusion Probabilistic Models (DDPMs) [1] have surpassed GANs in image synthesis quality. Diffusion models learn to reverse a gradual noising process, generating data from pure Gaussian noise. Their probabilistic nature allows for flexible conditioning mechanisms. To scale these models to high-resolution images, Latent Diffusion Models (LDMs) operate in the compressed latent space of a pre-trained autoencoder, significantly reducing computational costs.

In this project, we tackle the problem of **Classifier-Guided Counterfactual Generation**. We aim to take a real input image of a face and transform its emotional expression to a target class using a trained LDM.

The main interest of this paper, other than using Denoising Diffusion Probabilistic Models for counterfactual generation, is the implementation of a guidance mechanism based on Sequential Monte Carlo (SMC) methods [3]. Rather than relying on the gradients of a classifier to shift the mean of the diffusion process (as in standard Classifier Guidance [2]), we maintain a population of particles (latent hypotheses). After a fixed number of denoising step, we evaluate these particles using a trained emotion classifier and resample them, favoring those that align with the target emotion. This approach, often referred to as "filtering" or "Feynman-Kac steering", provides a flexible way to impose constraints without requiring the classifier to be differentiable or noise-aware.

2 Model Architecture and Theory

Our model consists of three distinct components that will each be described in detail. The model features a Variational Autoencoder (VAE) to encode and decode images in a latent space used throughout the model, a UNet-based Diffusion Model trained on these latents and used for image generation and counterfactual sampling, and an Emotion Classifier used to guide the generation during the SMC inference steps.

2.1 Visualization of the Architecture

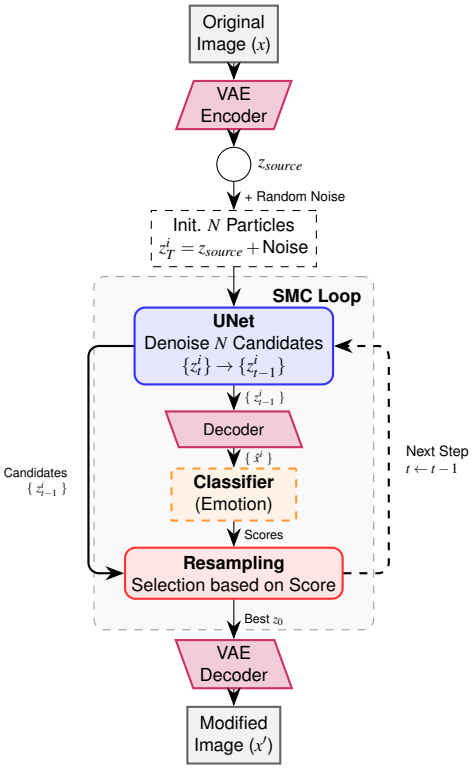

Figure 1: Model Architecture (for an inference step of 1)

We will now focus on a more specific description of each of these components, before giving more detail about the Sequential Monte Carlo guided inference.

2.2 Latent Space Compression (VAE)

To make training computationally feasible on available hardware (e.g. Kaggle GPUs), we use a pre-trained AutoencoderKL, specifically `stabilityai/sd-vae-ft-ema`. This KL-regularized Variational Autoencoder (VAE) was fine-tuned with Exponential Moving Average (EMA) weights to map inputs to a compressed latent space with high reconstruction fidelity.

Given an image $x \in \mathbb{R}^{256\times256\times3}$, the encoder $\mathscr{E}$ maps it to a latent code $z = \mathscr{E}(x)$, where $z \in \mathbb{R}^{4\times32\times32}$.

$$z \sim \mathcal{N}(\mu(x), \sigma(x))$$

The decoder $\mathscr{D}$ reconstructs the image: $\hat{x} = \mathscr{D}(z)$. This reduces the dimensionality of the generative space and allows the diffusion model to focus on the semantic structure of faces rather than low-level pixel statistics. This sharply improves the realism of the generated images and enables working with high resolution images.

2.3 Diffusion Process

We employ a **UNet2DModel** architecture for latent diffusion.

2.3.1 Model Inputs and Outputs

At each denoising timestep t, the model receives:

- the noisy latent $z_t \in \mathbb{R}^{4\times32\times32}$,
- a timestep embedding encoding the diffusion step t.

The UNet outputs a tensor of the same shape, representing the predicted noise $\varepsilon_\theta(z_t, t)$.

Predicting noise instead of directly predicting the clean latent corresponds to the formulation used in DDPM and Stable Diffusion.

2.3.2 Model Architecture

The UNet follows the classical encoder–bottleneck–decoder structure:

- A downsampling path that progressively reduces spatial resolution while increasing channel depth, capturing coarse structure.
- A bottleneck block encoding global context.
- A symmetric upsampling path that reconstructs high-resolution details.
- Skip connections that pass fine-grained information from earlier layers to later ones.

Here is a breakdown of the architecture :

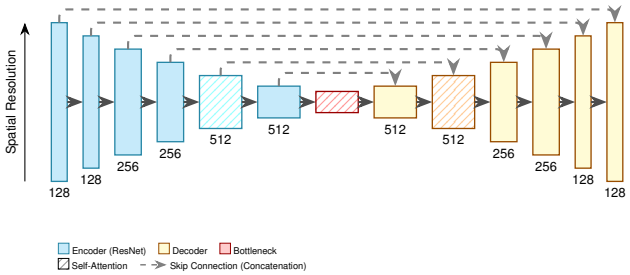

Figure 2: UNet Model breakdown

This architecture allows the model to represent both the global semantic layout of faces (shape, pose) and local details (eyes, mouth, textures). Our model contains approximately **113M trainable parameters**, similar to the base UNet used in Stable Diffusion v1.5, which is typically sufficient to learn the distribution of human faces when trained on a dataset of adequate scale.

2.4 Emotion Classifier

To guide the generation, we utilized a pre-trained classifier $\mathscr{C}_\phi$ from Hugging Face `dima806/facial_emotions_image_detection`.

- **Architecture:** A Vision Transformer (ViT) model (specifically based on `vit-base-patch16-224`).
- **Classes:** 7 emotional categories (Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise).
- **Training:** The classifier was pre-trained on ImageNet-21k and fine-tuned on the FER-2013 dataset. We used the model in inference mode (frozen weights).

The Emotion Classifier is used during the SMC inference steps, when the latents from a given step t are momentarily decoded back into the image space to be scored by the Classifier as the basis for the resampling step.

2.5 SMC Guided Inference [3]

This is the core interest of our study and is based on a paper published by *Singhal et al.*. Standard generation samples $z_{t-1} \sim p_\theta(z_{t-1}|z_t)$ at each step. To force the image to possess a specific emotion y_{target} while staying close to the original identity, we use a particle filter.
Let N be the number of particles. At each timestep t (going from T to 1):

1. **Propose:** We generate N potential candidates for the next step using the diffusion model's reverse dynamics.
2. **Weight:** We assign a weight w^i to each particle z^i_{t-1}. The weight depends on the score assigned by the classifier to each particle for the target emotion :

$$w^i \propto P_{\text{classif}}(y_{\text{target}} \mid \mathscr{D}(z^i_{t-1}))$$

3. **Resample:** We select particles for the next step based on these weights (we used multinomial resampling). This effectively discards trajectories that do not resemble the target emotion and duplicates promising ones.

The procedure is summarized in Algorithm 1 below and illustrated in the appendix.

Algorithm 1 SMC-Guided img2img Inference (heuristic)

Require: UNet ε_θ, scheduler STEP/ADDNOISE, VAE $(\mathscr{E},\mathscr{D})$, emotion classifier $\mathscr{C}$, target label y, N particles, K steps, resample interval R, guidance exponent γ, noise timestep t_{start}

$z_0 \leftarrow \mathscr{E}(x_{\text{src}})$ (encode source)
Sample $\varepsilon \sim \mathscr{N}(0,I)$
Set $z \leftarrow \text{ADDNOISE}(z_0, \varepsilon, t_{\text{start}})$
Initialize N particles: $z^1,\dots,z^N \leftarrow z$ (duplicate)
for $i = 1,\dots,K$ **do**
  for $j = 1,\dots,N$ **do**
    $\hat{\varepsilon}^j \leftarrow \varepsilon_\theta(z^j, t_i)$
    $z^j \leftarrow \text{STEP}(z^j, \hat{\varepsilon}^j, t_i)$ (one reverse step)
  end for
  if $i \bmod R = 0$ **then**
    $\hat{x}^j \leftarrow \mathscr{D}(z^j)$ for all j (decode)
    $s^j \leftarrow \mathscr{C}(\hat{x}^j)[y]$ for all j (scores)
    $w^j \leftarrow (s^j)^\gamma$; normalize w (weights)
    Resample $\{z^j\}_{j=1}^N$ with replacement using weights w
  end if
end for
return $\arg\max_j \mathscr{C}(\mathscr{D}(z^j))[y]$ (best particle)

During training, the diffusion model is trained using a fixed number of timesteps (1000 in our setting). At inference time, the reverse diffusion process operates on a reduced set of timesteps, whose number is controlled by the number of inference steps. This parameter determines how many denoising updates are performed to transform a noisy latent into a realistic image.

Starting the reverse process from pure noise would completely alter the structure of the source face. To preserve identity and facial structure, we introduce a parameter $noise\_strength \in [0,1]$, which controls the noise level at which the reverse diffusion is initialized. A lower value of $noise\_strength$ corresponds to starting the reverse process from a less corrupted version of the original latent, thereby limiting undesired structural changes.

In practice, we set $noise\_strength = 0.23$ and use 250 inference timesteps. This configuration results in approximately 60 effective reverse diffusion steps for each particle. During inference, resampling is performed after every reverse diffusion step in order to strongly favor particles that maximize the target emotion score.

3 Final Training and Results

3.1 Dataset and Experimental Setup

We conducted our experiments on the CelebA-HQ dataset, resized to a resolution of 256×256, which contains approximately 30,000 high-quality face images. Since emotion guidance relies on a pre-trained emotion classifier, no emotion annotations were required for training. CelebA-HQ is particularly well suited to our task, as it offers a large diversity of facial identities, expressions, and visual conditions, along with sufficient resolution to preserve fine-grained facial details.

The UNet denoising model was trained from scratch on CelebA-HQ using a standard Kaggle GPU (Tesla P100). During training, all images were first encoded into the latent space using a frozen VAE, and learning was performed exclusively in this latent domain. Optimization was carried out using the AdamW optimizer with a constant learning rate of 10^{-4} and mixed-precision (FP16) training. The model was trained for 89 epochs, which was sufficient to ensure convergence of the denoising objective.

3.2 Training Monitoring and Convergence

We monitored the training loss of the UNet and the generation of unconditioned samples using the Weights & Biases (WandB) platform.

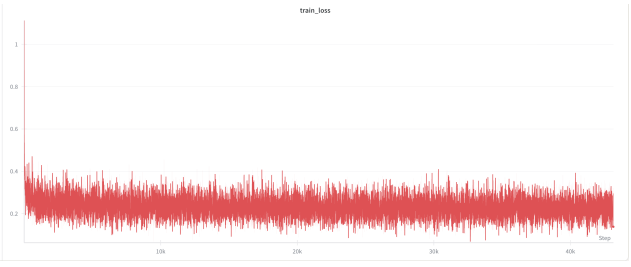

Figure 3: Training Loss of the Latent Diffusion Model over epochs.

The loss decreases slowly but steadily as the training advances, indicating the model learns to denoise effectively. As expected for the training of a large size Diffusion Model, realistic results can only be generated after a very large number of training steps.

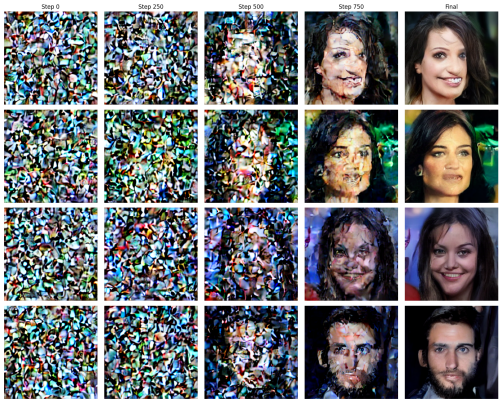

Figure 4: Samples generated from pure noise to monitor training

After a few epochs, the quality of the samples improves very slowly. These images were generated after epoch 79. Because our goal is to denoise partially corrupted images, we do not need our model to achieve perfect realism on images generated from pure noise.

3.3 Hyperparameter Tuning for Inference

After training the UNet model, we focused on the guided generation mechanism which depends on the following parameters:

- **Target emotion:** The specific emotional class the algorithm aims to maximize for the generated image.
- **Number of particles:** The number of candidate images maintained at each timestep t during the inference process.
- **Number of steps:** The total number of denoising iterations defined by the scheduler. This determines the temporal resolution of the reverse diffusion process.
- **Guidance scale:** The exponent γ applied to the classifier probabilities during the weighting step ($w \propto P(y \mid z)^{\gamma}$). A higher scale enforces the target emotion more strictly (making the selection "greedier"), while a lower scale preserves more diversity among particles.

- **Noise strength:** A parameter $\in [0, 1]$ determining the starting point of the inference. A value of 1.0 initiates guided generation from pure Gaussian noise, while a lower value (e.g., 0.1) starts from a partially noisy version of the source image, thereby preserving more of its original structural identity.
- **Resample interval:** The frequency at which the resampling step is triggered. An interval of 1 means particles are re-weighted and resampled at every denoising step, whereas a higher interval reduces the frequency of classifier interventions.

We have studied the effects of these parameters on the quality of the final image.

3.4 Quantitative Evaluation

To quantitatively assess the behavior of our SMC-guided diffusion framework, we conducted a controlled ablation study on three key hyperparameters: the *guidance scale*, the *noise strength*, and the number of *inference steps*. For each parameter, four representative values were tested while keeping all other variables fixed (when not tested : guidance scale was 5, noise strength was 0.2, inference steps was 250). All experiments were run with target emotion $happy$, 20 particles and a resample interval of 1. The goal was to evaluate the realism of the images created, and the accuracy of the emotion generation.

Evaluation metrics. We report two complementary metrics:

- **Target Accuracy (Emotion Score):** the average probability assigned to the target emotion by a pre-trained emotion classifier.
- **Realism Score (Face Probability):** the average face detection confidence obtained using MTCNN, serving as a proxy for facial realism and structural coherence.

Expected Trade-offs :

- **Guidance scale :** Increasing the guidance scale yields short-term emotion score improvements, but reduces the variety of explored trajectories. Overall, a too high guidance scale could prevent reaching a higher scores.
- **Noise strength :** controls the level of corruption applied to the source latent before reverse diffusion. Increasing the noise enables to generate faces more different from the source image. It allows better emotion control, but may remove some characteristic traits.
- **Number of inference steps :** Increasing the number of inference steps allows a finer control of the emotions through more resampling without diminishing the resampling rate too much. However, it also augments the inference time.

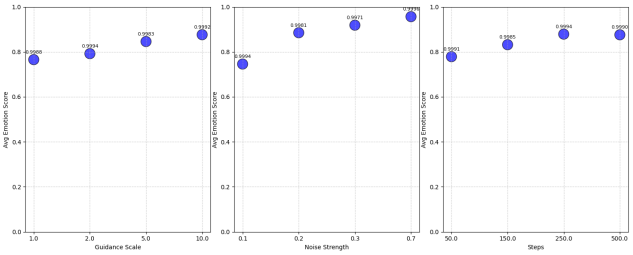

Figure 5: Configuration benchmarks

On these graphs, the vertical axis corresponds to the average emotion score, computed by evaluating how accurately the target emotion was generated across 10 samples. The size of each marker represents the average realism score measured on the same set of samples.

All tested configurations achieved a realism score above 0.99, indicating that the generated faces were consistently realistic enough to be successfully detected. While this confirms the overall visual validity of the results, it also suggests that the chosen realism metric lacks sufficient discriminative power to meaningfully differentiate between configurations.

Furthermore, we observe that increasing each of the tested parameters generally leads to improved emotion generation accuracy. In the case of the number of inference steps, a trade-off must be found to ensure reasonable inference time, which we evaluated at around 250 steps. For the guidance scale, higher values consistently yield better emotion scores. However, increasing the noise strength beyond moderate values tends to alter the facial structure, producing images that no longer closely resemble the original identities.

Based on these observations, we selected a noise strength of 0.23, 250 inference steps, and a guidance scale of 5 for our final configuration.

3.5 Final results

Here are some visual representations of the results of changing the emotion of a face using the fully trained model.

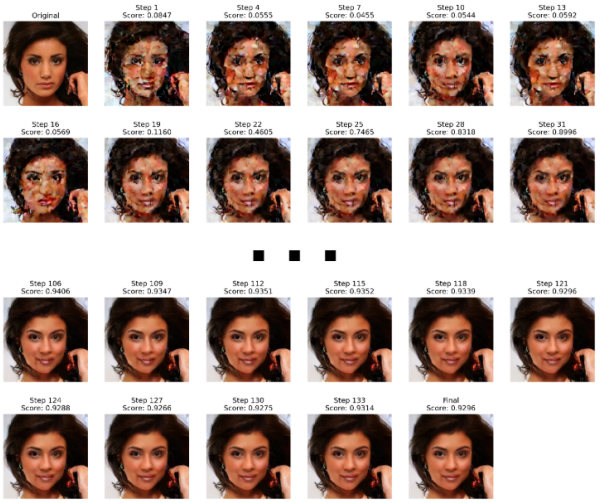

Figure 6: Counterfactual generation towards the class "Happy" for an image from CelebA

Figure 7: Counterfactual emotion generation towards classes "Happy", "Sad" and "Angry" for the same image from CelebA

4 Discussion

The SMC approach proves to be a flexible alternative to gradient-based guidance. It allows us to plug in any classifier without needing its gradients, which is advantageous for black-box models or non-differentiable metrics. However, the method is computationally expensive. Generating a single image with N particles requires N times more forward passes through the UNet and the Classifier than standard sampling. We also noticed that most of the inference time was due to the conversion of latents back into the image space at every resampling step (for scoring with the Emotion Classifier). In order to achieve faster inference, future work could investigate training the emotion classifier on the latents directly, or using more efficient sampling strategies like Importance Sampling to reduce the particle count.

We have also studied the possibility to add en extra parameter $leash$ to the SMC Generation step. This feature was used after each inference step, as a coefficient to average the new latent with the original uncorrupted latent. The $leash$ was intended to ensure candidates from any given time step would maintain identity by preventing an overly disruptive denoising. This lead was however abandoned when we realized this parameter was somewhat redundant if we consider its effect can be reproduced by simultaneously adjusting the guidance scale, noise strength and resample interval during generation.

In order to more thoroughly evaluate the impact of our hyperparameters, three metrics should be considered: target emotion accuracy, image realism, and facial similarity after modification. While the target emotion accuracy is properly assessed in our experiments, the two remaining metrics could be further improved or extended. For instance, image realism could be evaluated using the Fréchet Inception Distance (FID), and facial similarity could be measured by embedding the original and modified images into a shared feature space and computing their cosine similarity.

Finally, it seems the model reconstructs poorer quality images when the input image we wish to modify has some unique features which are underrepresented in the CelebA Dataset (glasses, beards, open mouth, ...). The model could gain in precision and generalization if trained on a larger Dataset to take into account these features.

5 Conclusion

In this work, we successfully implemented a Latent Diffusion Model and applied Sequential Monte Carlo guidance to generate emotional counterfactuals. Our results demonstrate that it is possible to modify high-level semantic attributes of a face while maintaining a coherent identity. This confirms the potential of sampling-based guidance methods for controlled generation tasks in computer vision.

The code to our experiment can be found here.

References

[1] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[2] P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.

[3] R. Singhal, Z. Horvitz, R. Teehan, M. Ren, Z. Yu, K. McKeown, and R. Ranganath. A general framework for inference-time scaling and steering of diffusion models. *ICML*, 2025.

Appendix : Illustration of SMC

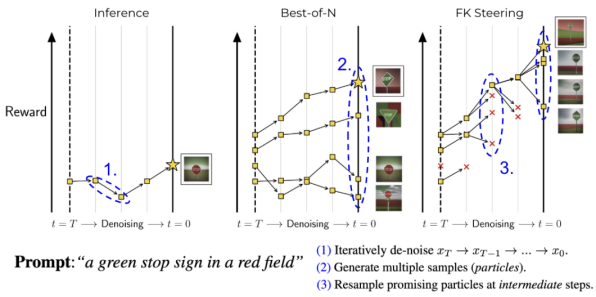

Figure 8: Graphical representation of the SMC mechanism. This visualization is drawn directly from [3].