

Diffusion Classifier Guidance for Non-robust Classifiers

Philipp Vaeth^{1,2}[0000–0002–8247–7907] (✉),
Dibyanshu Kumar¹[0009–0007–2542–4781],
Benjamin Paassen²[0000–0002–3899–2450], and
Magda Gregorová¹[0000–0002–1285–8130]

¹ Center for Artificial Intelligence and Robotics, Technical University of Applied Sciences Würzburg-Schweinfurt, Franz-Horn-Straße 2, Würzburg, Germany
{philipp.vaeth,magda.gregorova}@thws.de, kumardibyanshu05@gmail.com
² Bielefeld University, Universitätsstraße 25, Bielefeld, Germany
bpaassen@techfak.uni-bielefeld.de

Abstract. Classifier guidance is intended to steer a diffusion process such that a given classifier reliably recognizes the generated data point as a certain class. However, most classifier guidance approaches are restricted to robust classifiers, which were specifically trained on the noise of the diffusion forward process. We extend classifier guidance to work with general, non-robust, classifiers that were trained without noise. We analyze the sensitivity of both non-robust and robust classifiers to noise of the diffusion process on the standard CelebA data set, the specialized SportBalls data set and the high-dimensional real-world CelebA-HQ data set. Our findings reveal that non-robust classifiers exhibit significant accuracy degradation under noisy conditions, leading to unstable guidance gradients. To mitigate these issues, we propose a method that utilizes one-step denoised image predictions and implements stabilization techniques inspired by stochastic optimization methods, such as exponential moving averages. Experimental results demonstrate that our approach improves the stability of classifier guidance while maintaining sample diversity and visual quality. This work contributes to advancing conditional sampling techniques in generative models, enabling a broader range of classifiers to be used as guidance classifiers.

Keywords: DDPM · Diffusion Models · Conditional Sampling · Classifier Guidance · Gradient Guidance.

Reproducibility: The code, the trained model weights and the supplementary material to reproduce the results is available at <https://github.com/philippvaeth/nrCG>.

1 Introduction

Denosing diffusion probabilistic models (DDPM) [9] are state of the art generative models, modelling an intractable data distribution $\mathbf{x}_0 \sim p_{\text{data}}$ via a learned latent variable model $p_{\theta}(\mathbf{x}_0) = \int p(\mathbf{x}_T) \prod_{t=1}^T p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t) d\mathbf{x}_{1:T}$. Through a Markov chain Gaussian forward process ($\mathbf{x}_0 \rightarrow \mathbf{x}_T$) with noising transitions $q(\mathbf{x}_t | \mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$, the data \mathbf{x}_0 is progressively noised with a pre-defined variance schedule β_1, \dots, β_T . The Gaussian Markov reverse process ($\mathbf{x}_T \rightarrow \mathbf{x}_0$) with learned denoising steps $p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t)$ reverses the forward process from random noise $\mathbf{x}_T \sim \mathcal{N}(0, I)$ to produce samples following the data distribution $p_{\theta} \approx p_{\text{data}}$.

A special property of this type of generative model is the iterative sampling procedure where conditional information can be added without the need for training a specific conditional model through a procedure known as *classifier guidance* [18,4]. For an unconditionally trained DDPM $p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_{\theta}(\mathbf{x}_t), \Sigma_t(\mathbf{x}_t))$, the mean $\mu_{\theta}(\mathbf{x}_t)$ of the transitions can be shifted by the gradients of a classifier trained over the noisy data \mathbf{x}_t as:

$$\mu_{\theta}(\mathbf{x}_t)' = \mu_{\theta}(\mathbf{x}_t) + s \Sigma_t(\mathbf{x}_t) \nabla_{\mathbf{x}_t} \log p_{\text{cl}}(y | \mathbf{x}_t) \quad , \quad (1)$$

where s is a gradient scaling factor controlling the strength of the classifier guidance, and $\mu_{\theta}(\mathbf{x}_t)'$ is the new mean of the reverse transition used for conditionally sampling the previous sample \mathbf{x}_{t-1} .

Classifier guidance is commonly used to add conditional information during the diffusion reverse process (e.g., in explainability [1], in protein design [7] and in molecular design [21]). The main limitation of classifier guidance is that the classifier needs to be robust to noise similar to that added during the diffusion forward process [4] so that the gradients $\nabla_{\mathbf{x}_t} \log p_{\text{cl}}(y | \mathbf{x}_t)$ in equation 1 are meaningful. This requires training a guidance classifier for each specific diffusion model and re-training it if the desired conditioning changes or if the diffusion forward process definition changes. Extending classifier guidance to classifiers not trained over the specific DDPM noise (non-robust classifiers) remains a challenge.

A previously proposed solution is to let the classifier decide on a one-step denoised image from the diffusion model instead of the noisy images directly, referred to as $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction [2,1,20]. We introduce the $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction in detail in section 2.4 (equation 5), and show that it is not enough to solve the challenge of non-robust classifier guidance. Based on a detailed analysis of the classifier gradients including the $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction, we propose in section 2.5 to leverage methods from stochastic optimization to additionally stabilize the non-robust guidance gradients further, bridging the gap to the performance of robust classifier guidance. Finally, we transfer our proposed stabilization method to the diffusion reverse process in section 3 and show that the stabilization enables the use of non-robust classifiers for guided sampling. In summary, we provide a detailed analysis of how non-robust and robust classifiers behave during the diffusion forward process, and propose a guidance stabilization technique that allows non-robust classifiers to be used effectively for guidance in the diffusion reverse process.

2 Diffusion forward process

We start our analysis by comparing the classifier accuracy over different levels of noisy data in section 2.1. We then showcase how the logits (section 2.2) and gradients (section 2.3) of the classifier behave over time t for similar inputs. Finally in section 2.5, we analyze how the $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction (equation 5) influences the gradients of non-robust classifiers and, as a result, propose stabilization techniques to further improve non-robust classifier guidance. We conclude section 2 by recommending a stabilization technique for non-robust classifier guidance and test this on the reverse diffusion process in section 3.

For our analysis, we train two standard MobileNetV3 [11] classifiers on the CelebA [14] data set with an image size of 64x64 (details in section 3) to detect the binary attribute *female*: (1) a **non-robust** classifier trained on the original non-noisy data and (2) a **robust classifier** trained on data augmented by the forward noising process of the diffusion model. In detail, for a standard training batch of n images, we draw n time steps from a discrete uniform distribution $t \sim \mathcal{U}\{0, T\}$ and run the diffusion forward process for each image as:

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}) \quad , \quad (2)$$

with $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$ and $\alpha_t := 1 - \beta_t$ [9]. An increasing β_t noise schedule therefore corresponds to progressively noisier samples \mathbf{x}_t for a higher t .

For our diffusion model, we train a standard DDPM [9] with a linear noise schedule ($\beta_0 = 0.0001, \beta_T = 0.02$), $T = 400$ diffusion steps, a standard U-Net architecture [16] for the noise predictor, and the simplified MSE noise prediction training objective [9]. Our diffusion models are implemented using the open-source Diffusers toolbox [15] and trained for 1000 epochs (around 3 days on a single NVIDIA A80 GPU).

2.1 Accuracy of the classifiers on noisy data

In figure 1, we compare the classification accuracy of the robust and the non-robust classifiers over the noisy validation data set (by applying equation 2) and see that the non-robust classifier accuracy (red) drops significantly with increasing noise levels added through increasing diffusion steps, up to the point of random guessing at less than 25% of the total diffusion steps T . This analysis of the classification performance is a simple way to understand classifier robustness over different noise levels. However, the classification performance analysis works over the validation set perturbed by different amount of random noise, disregarding previous time steps (equation 2). In the diffusion forward process, the dependency on the previous sample is critical for the model definition and the sampling procedure (Markov property).

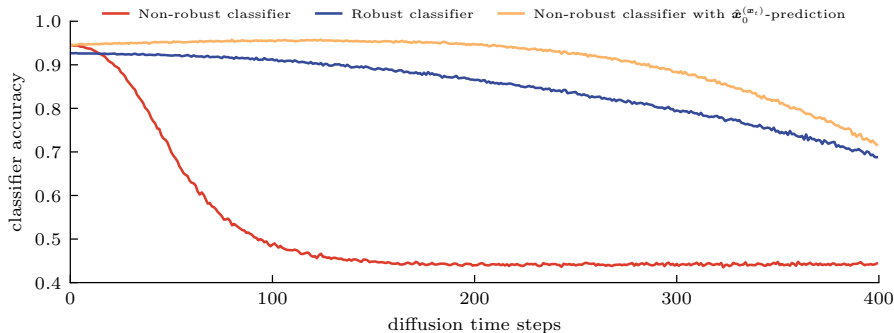


Fig. 1: Classification accuracy comparison of the robust, non-robust, and non-robust with $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction (eq. 5) classifiers on the CelebA binary attribute female. The metric is reported as the average over the validation data set.

2.2 Sensitivity of the classifier logits

To further investigate the implications of low classification accuracy in the presence of noisy data points, we propose to analyze the sensitivity of the classifier’s output scores (logits) to small changes in input features over time. This approach allows us to measure how the quantity of noise in the data points affects the decision boundary and robustness of the classifier. Specifically, starting from the same image \mathbf{x}_0 , we do not sample two adjacent noisy versions \mathbf{x}_t and \mathbf{x}_{t-1} independently (equation 2), but instead use the same noise to produce both noisy images. This results in small changes by construction, where the same features are perturbed in \mathbf{x}_t and \mathbf{x}_{t-1} but at different scales based on the β schedule of the DDPM forward process. This is in line with the diffusion forward process definition in section 1, in which \mathbf{x}_t is a more noisy version of \mathbf{x}_{t-1} . We consider a classification function $f : \mathcal{X} \rightarrow \mathcal{Y}^D$ which maps an RGB input image to a D -dimensional vector of class logits and define the metric S_t as:

$$S_t(\mathbf{x}_t, \mathbf{x}_{t-1}) = \frac{\|f(\mathbf{x}_t) - f(\mathbf{x}_{t-1})\|_2}{\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2} . \quad (3)$$

For two noisy data points \mathbf{x}_t and \mathbf{x}_{t-1} on the same diffusion trajectory (starting from the same \mathbf{x}_0), small differences between these points should correspond to small differences in logits for a robust classifier. Note that the metric S_t is similar to the discrete approximation of derivatives. We compare the score S_t (equation 3) over the entire diffusion forward process for our classifiers in figure 2 to analyze the noise sensitivity of classifier logits over time. The results confirm that the non-robust classifier is indeed much more sensitive to small input changes than the robust classifier. This means that the non-robust classifier function is not smooth and reacts with different output logits for small input perturbations, hinting to possibly undesired behavior for the guidance of the diffusion reverse process based on classifier gradients.

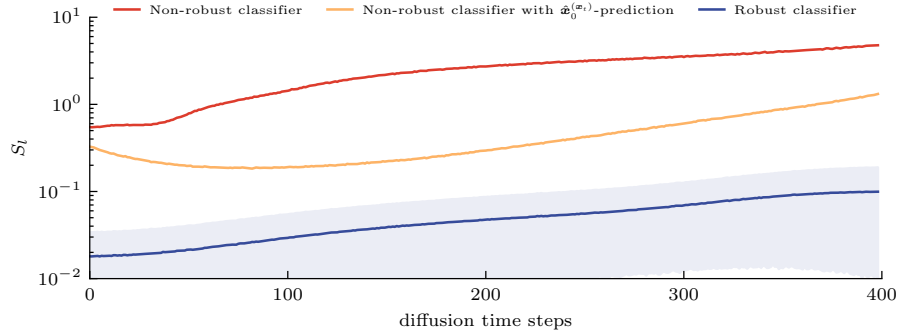


Fig. 2: Logit sensitivity S_l (log scale) as defined in eq. 3 over time t for the robust, non-robust, and non-robust with $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction (eq. 5) classifiers on CelebA. The metric is reported as the average (and std) over the validation data set.

2.3 Stability of the classifier gradients

Going a step further beyond logits, we can directly compute gradients just as they would be used in the sampling process of the diffusion model to confirm that unstable logits over time t indeed affect the gradients necessary in classifier guidance. We run the same experiment, but compare the sensitivity of gradients over time t instead of logits:

$$S_g(\mathbf{x}_t, \mathbf{x}_{t-1}) = \frac{\|\nabla_{\mathbf{x}_t} f(\mathbf{x}_t) - \nabla_{\mathbf{x}_{t-1}} f(\mathbf{x}_{t-1})\|_2}{\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2} . \quad (4)$$

An alternative interpretation of equation 4 is in terms of geometry. Equation 4 quantifies to what degree the guidance vectors point in similar directions for adjacent time steps t and $t - 1$. We note that S_g is connected to the discrete approximation of second-order derivatives, that is the curvature of the classification function over time. In practice, a low S_g score would correspond to gradual introduction of features during conditional diffusion sampling instead of sudden feature changes. Based on this intuition, we can quantify through the metric S_g how informative classifier gradients are for conditional sampling.

In figure 3, we show the metric S_g over time for the same experimental setup as previously, confirming that the non-robust classifier with unstable logit outputs (as demonstrated in figure 2), indeed does not have informative gradients and is therefore not suitable for conditional guidance. On the contrary, the robust classifier (blue line in figure 3) shows low gradient sensitivity as measured by S_g , enabling the use of the robust classifier for classifier guidance.

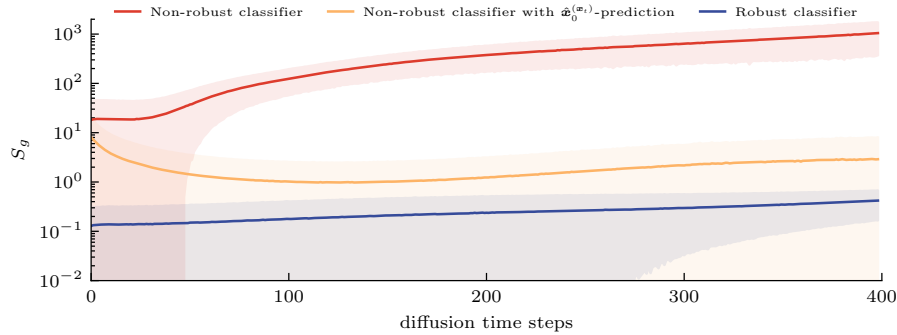


Fig. 3: Gradient sensitivity S_g (log scale) as defined in eq. 4 over time t for the robust, non-robust, and non-robust with $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction (eq. 5) classifiers on CelebA. The metric is reported as the average (and std) over the validation set.

2.4 Informative classifier gradients through $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction

To summarize, we have shown that classifying noisy data points with a classifier not trained over the same noise results in a loss of accuracy and high sensitivity of the classifier outputs to such noise. We have also shown that this results in unstable and therefore non-informative gradients, which are not suitable for the use in classifier guidance. One approach to resolve this issue is to apply the classifier not on the noisy diffusion data \mathbf{x}_t but on an approximation of the fully denoised image \mathbf{x}_0 [2,1,20]. We can estimate the $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction via:

$$\hat{\mathbf{x}}_0^{(\mathbf{x}_t)} = \frac{\mathbf{x}_t}{\sqrt{\bar{\alpha}_t}} - \frac{\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) . \quad (5)$$

The $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction seemingly resolves the issue of non-robust classifiers for classifier guidance, supported by the high classification accuracy over noisy data when applying the $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction before the classification (orange line in figure 1). In addition to classification accuracy, however, we also consider the gradient sensitivity for $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction:

$$\hat{S}_g(\mathbf{x}_t, \mathbf{x}_{t-1}) = \frac{\|\nabla_{\mathbf{x}_t} f(\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}) - \nabla_{\mathbf{x}_{t-1}} f(\hat{\mathbf{x}}_0^{(\mathbf{x}_{t-1})})\|_2}{\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2} . \quad (6)$$

We show in figure 3, that the classifier with $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction (orange line) substantially reduces gradient sensitivity (and hence improves gradient stability), but does not yet achieve the level of a robust classifier. Hence, we believe that further improvements, beyond the $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction, are required. We note that $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction dramatically increases memory cost because gradients need to be propagated not only through the classifier but also through the diffusion model at each denoising step.

2.5 Stable classifier gradients through moving averages

We begin our improvements with the insight that the classifier guidance process in equation 1 effectively acts as a moving average because the mean of the reverse sampling process in every step is the sum of the mean of the previous step and the classifier guidance vector. However, the guidance vectors are computed independently in each step t , meaning that their directions can drastically change between time steps as discussed in section 2.3 and shown in figures 2 and 3. Accordingly, it stands to reason to adjust classifier guidance to explicitly perform a moving average over the guidance vectors, thus enhancing the gradient stability. We explore two stabilization techniques inspired by the two most common stochastic optimization algorithms, SGD with momentum [17] and ADAM [13]. For a given guidance gradient \mathbf{g} , momentum strength β and $\epsilon > 0$, we define:

$$\nu_t^{\text{ema}}(\mathbf{g}, \beta) = \beta \nu_{t-1}^{\text{ema}} + (1 - \beta) \mathbf{g} , \quad (7)$$

$$\nu_t^{\text{adam}}(\mathbf{g}) = \frac{\nu_t^{\text{ema}}(\mathbf{g}, \beta = 0.9)}{\sqrt{\nu_t^{\text{ema}}(\mathbf{g}^2, \beta = 0.999) + \epsilon}} . \quad (8)$$

We do not include any de-biasing terms into equations 7 and 8 to compensate for extremely noisy samples with barely any signal in the initial denoising steps ($\mathbf{x}_T, \mathbf{x}_{T-1}, \dots$), which results in unreliable gradients. We therefore omit these de-biasing terms deliberately to bias the guidance terms toward zero. In the reverse process, this will avoid adding unreliable conditioning information early in the sampling steps, which could potentially break the diffusion sampling process due to unlikely starting points.

We again experimentally validate on the forward process how these stabilization techniques change the gradient stability over time. For this, we apply both techniques (equations 7 and 8) directly on the gradients in our gradient stability metric S_g (equation 6). We show the gradient stability over time t in figure 4, contrasting the robust classifier to the non-robust classifier with $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction and with the stabilization techniques. For ADAM, the gradient stability deteriorates over increasing time t due to the rescaling of the gradients by the running estimate of the second moment (see equation 8), amplifying differences between time steps t and $t - 1$ based on the variance of the gradient (denominator of equation 8). For exponential moving averaging, the differences between neighboring diffusion time steps become naturally smaller, with a larger window size ($\beta = 0.99$) contributing to even more stability over time. Interestingly, the EMA stabilization with the large window size reaches the gradient stability of the robust classifier, especially during the first half of the forward process ($t < 200$).

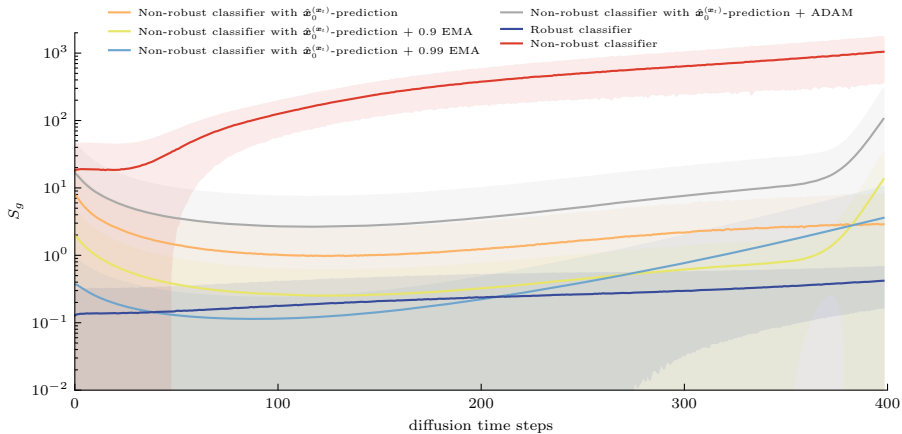


Fig. 4: Gradient sensitivity S_g (log scale) over time t for the robust, non-robust, and non-robust with $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction (eq. 5) classifiers, as well as multiple stabilization techniques (eq. 7,8). The metric is reported as the average (and std) over the CelebA validation data set.

Our analysis demonstrates that gradient stability, measured by pairwise differences over time, is connected to the classifier accuracy. Additionally, we show that $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction enhances gradient quality from this perspective. Furthermore, by explicitly enforcing stable feature changes over the diffusion time steps t through exponential moving averaging of classifier gradients, we can bridge the gap between the non-robust classifier and the robust classifier in terms of gradient stability. These observations have so far been on the diffusion forward process to observe the gradient behavior in isolation without interference of the diffusion reverse (sampling) process. In section 3, we will translate these findings to the diffusion reverse process.

3 Diffusion reverse process

In the diffusion reverse process, we apply the techniques from section 2 to diffusion sampling. In algorithm 1, we provide details about the implementation of our guided sampling setup. The only difference to the standard DDPM classifier guidance are lines 3 and 4, which is where we apply our stabilization techniques. We introduce the data sets in section 3.1, define the metrics used to evaluate the generated samples in section 3.2, and then use algorithm 1 in section 3.3 to produce conditional samples for our non-robust classifiers.

Algorithm 1 Guided DDPM Sampling

```

1:  $\mathbf{x}_T \sim \mathcal{N}(0, I)$ , classifier guidance scale  $s$ , unconditionally trained DDPM  $\mu_\theta(\mathbf{x}_t)$ ,
   DDPM forward process variance  $\Sigma_t(\mathbf{x}_t)$ , guidance stabilization function  $\nu$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{g} = \nabla_{\mathbf{x}_t} \log p_{\text{cl}}(y | \hat{\mathbf{x}}_0^{(\mathbf{x}_t)})$  if  $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction (eq.5), else  $\nabla_{\mathbf{x}_t} \log p_{\text{cl}}(y | \mathbf{x}_t)$ 
4:    $\mathbf{g} = \nu(\mathbf{g})$  if guidance-stabilization ▷ See eq. 7 and eq. 8
5:    $\mathbf{x}_{t-1} = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t), \Sigma_t(\mathbf{x}_t))$  ▷ See diffusion reverse transition in sec. 1
6:    $\mathbf{x}'_{t-1} = \mathbf{x}_{t-1} + s \Sigma_t(\mathbf{x}_t) \nabla_{\mathbf{x}_t} \mathbf{g}$  ▷ See eq. 1
7: end for
8: return  $\mathbf{x}'_0$ 

```

3.1 Data sets

For the data sets used in conditional sampling, we chose CelebA [14] as a standard image generation benchmark data set, use the SportBalls data set [19] as a custom data set specifically created for conditional generations, and use Celeba-HQ [12] as the real-world high resolution data set with an off-the-shelve diffusion model. We train a standard MobileNetV3 [11] classifier on all data sets. We train the non-robust classifier without data augmentation and the robust classifier with the training data corrupted by the same noise occurring in the diffusion forward process.

For the CelebA data set (64x64), we train the classifiers on the binary attribute *female* (58.3% of the total images). This class was chosen based on easily distinguishable features of the classes and the relatively clear classification boundary. The classifiers are trained on over more than 160k training images and evaluated over 20k validation images.

For a synthetic, more controllable, conditional sampling setup we use the SportBalls data set (64x64). The custom data set is created by randomly selecting one out of three sport balls (multi-class classification) and placing them at random coordinates on white background with random rotation and scaling. The data set is carefully created to have similar objects (i.e., scaling, shape, size, rotation and placement) but with clear semantic differences (i.e., colors and pattern). This data set is specifically constructed for conditional sampling due to clear class boundaries with balanced classes for the classifier and the white background for unambiguous generations without artifacts in the images. The goal for the conditional DDPM sampling is to generate *baseballs*. The classifiers are trained on 80k training images and evaluated on 20k validation images.

For the real-world use-case on CelebA-HQ-256 (256x256), we train the same simple classifier just as for the other data sets, but use the pre-trained DDPM model from [9]. We use the DDPM model without modification in our stabilized guided sampling setup to showcase how our contribution translates to third-party models and higher dimensional data. The class to generate is *female* (64.1% of the total images). The non-robust classifier is trained on 28k training images and evaluated on 2k validation images.

3.2 Metrics

To evaluate the resulting samples on the CelebA and SportBalls data sets, we compute all following metrics over 50176 conditionally generated samples for the different stabilization setups (3-7 hours on a single NVIDIA A80 graphics card). For CelebA-HQ, we compute the metrics over 1024 samples (4 hours on a single NVIDIA A80 graphics card) due to computational constraints.

To quantify if the guiding classifier successfully introduced class-conditional features, we apply the classifier on the final generated samples and compute the accuracy for the target class. Different stabilization setups and guidance scales will lead to higher accuracy at the expense of image quality and diversity.

A common metric to quantify the visual quality of generated images is the Fréchet inception distance (FID) [8], which compares statistics of extracted features from a pre-trained network between the training data and generated images. For this comparison, we randomly draw the same amount of generated samples from the training data as we generate. A low FID score indicates visual similarity of the generated samples to the training data. This metric also serves as a measure for diversity, as generated samples with only class-specific features are generally less close to the training set with a diverse set of features.

To complement the accuracy and the unconditional FID metric for visual quality, we compute a class-specific FID score which only operates on the data of the target class (cFID). In practice, this means we compare the statistics of the conditionally generated samples not to that of the entire data set but only to training images of the target class. A low cFID score ensures that the generated samples are visually close to the ground-truth images of the target class, ignoring potential features of other classes.

3.3 DDPM sampling with stabilized non-robust classifiers

We start our improved sampling experiments on the CelebA data set and conclude the section with experiments on the SportBalls and the CelebA-HQ data sets. The key hyperparameter in classifier guidance is the scale s (see algorithm 1 and equation 1), known to trade-off class conditioning and sample diversity [4]. We explore the robust classifier and the non-robust classifier with stabilization techniques, and present the accuracy over different guidance scales in figure 5, the unconditional FID in figure 6 and the conditional FID in figure 7.

We can observe that for a high enough guidance scale, all classifier setups except the non-robust classifier without stabilization techniques produce consistent class-conditional samples according to the classifier (figure 5). The non-robust classifier guidance fails without stabilization by offsetting the unconditional diffusion mean by so much that the diffusion reverse process can not recover, ultimately not producing any samples. We can see that the ADAM stabilization requires a much lower scaling than the other stabilization techniques as the rescaling of the gradients by the variance (equation 8) amplifies the guidance scale.

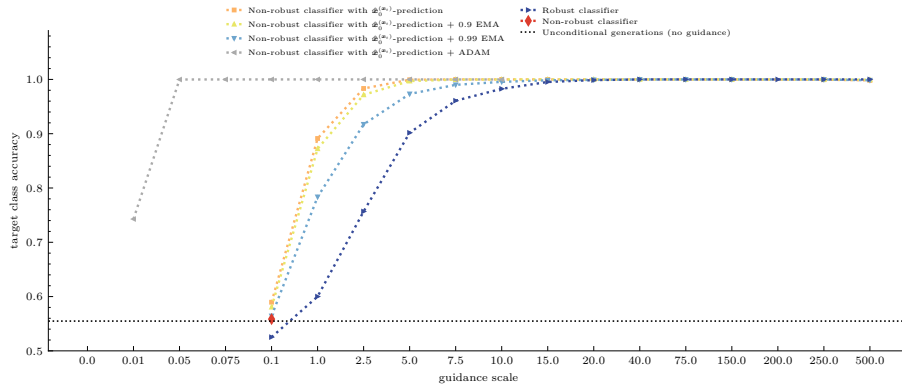


Fig. 5: Accuracy comparison for conditional sampling on CelebA with various stabilization setups (eq. 7,8). The accuracy is presented as the average over 50176 generated samples.

From the image quality as measured by the FID in figure 6, we can notice the FID of the robust classifier increases with a higher guidance scale. This is expected behavior, as the quality of the images measured by the closeness to the entire data set should decrease if the diffusion model is constrained to generate features of one class only and therefore loses diversity. The non-robust classifier with $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction and ADAM stabilization exhibits a more rapid increase in FID as the guidance scale increases, compared to the robust classifier. The non-robust classifiers with $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction and with $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction + EMA stabilization all exhibit similar behavior with an increasing FID just as the target class accuracy increases. This levels off to a stable FID value even for very high guidance scales until too much guidance strength (>500 in this case) eventually increases the FID again. This indicates an optimal range just before this increase, where substantial guidance strength can be applied without compromising sample quality or overwhelming the diffusion process.

For the class-conditional FID in figure 7, we observe a decrease in cFID score with more guidance strength for the robust classifier up to a turning point (here $s > 40$) when the cFID score increases again. This means for higher guidance strength the overall sample quality (FID) decreases due to lower diversity as compared to the complete mixed-class training set, but the class-conditional sample quality (cFID) increases. However, if the guidance strength is too high, the cFID reaches a turning point where the conditioning overpowers the diffusion process, generating samples not coherent with the underlying data distribution. A similar behavior is shown by all guidance setups, highlighting that the choice of guidance strength trades-off sample quality and class conditioning. Our proposed guidance setup, using $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction and EMA stabilization with $\beta = 0.99$, achieves the best cFID score (13.9) while maintaining good overall image quality (FID of 29.37). This guidance setup outperforms even the unmodified robust classifier,

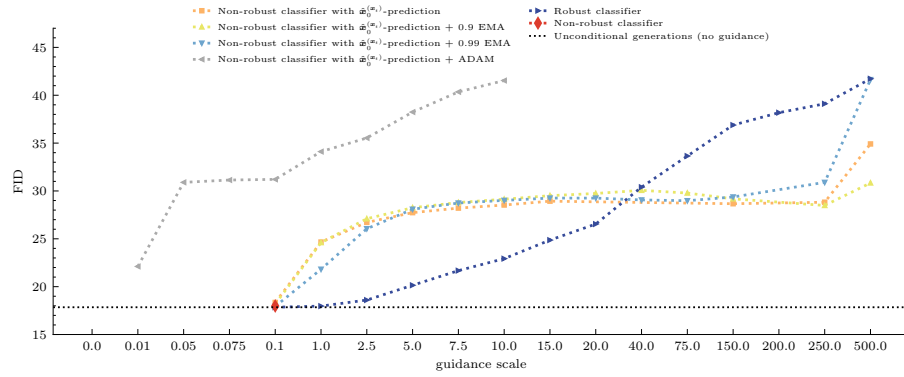






















Fig. 6: Unconditional FID comparison for conditional sampling on CelebA with various stabilization setups (eq. 7,8). The unconditional FID is calculated over 50176 generated samples.

demonstrating the potential of non-robust classifiers for conditional sampling when appropriately stabilized. This confirms our findings from the diffusion forward process analysis in section 2. By introducing the $\hat{\mathbf{x}}_0^{(x_t)}$ -prediction into the classifier guidance, the classifier gradients of the non-robust classifier are more meaningful. Through exponential moving averaging of the classifier gradients, we can enforce stable feature changes over the guided reverse diffusion process. The combination of the $\hat{\mathbf{x}}_0^{(x_t)}$ -prediction and the exponential moving average of the gradients leads to successful classifier guidance even for the non-robust classifier. We show generations for our best guidance setup as well as without guidance in table 1. More images are provided in the supplementary material.

Table 1: Metrics and first 10 samples for unconditional diffusion sampling (left) and conditional diffusion sampling (right) with the non-robust classifier, $\hat{\mathbf{x}}_0^{(x_t)}$ -prediction (eq. 5), 0.99-EMA stabilization (eq. 7) and guidance scale of 150.0 on CelebA. Metrics calculated over a batch of 50176 samples. More images are shown in the supplementary material.

Unconditional					Conditional (best)				
									
									
acc.: 55.67%, FID: 17.68 , cFID: 30.96					acc.: 99.99%, FID: 29.37 , cFID: 13.90				

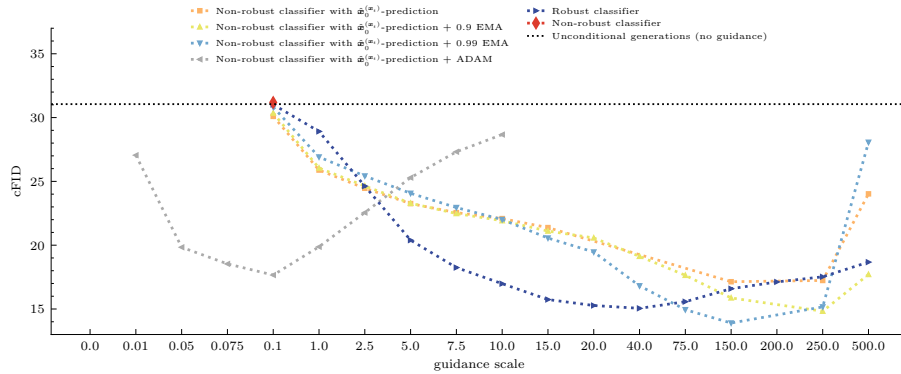


Fig. 7: Target class FID comparison for conditional sampling on CelebA with various stabilization setups (eq. 7,8). The class-conditional FID is calculated over 50176 generated samples.

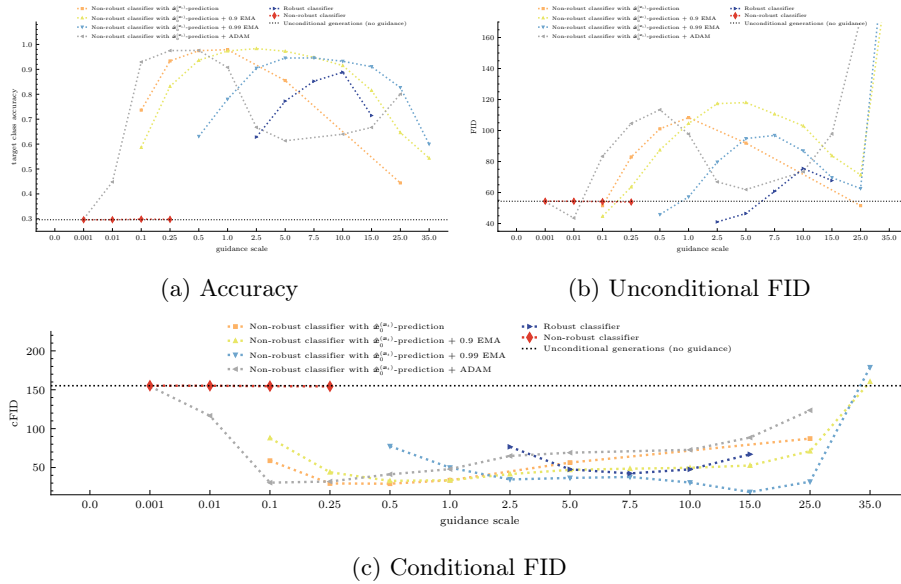






















Fig. 8: Accuracy, FID and cFID metrics for conditional sampling on SportBalls with stabilization setups (eq. 7,8) calculated over 50176 generated samples.

We repeat the previous experiment on the more controlled multi-class SportBalls data set. We present the accuracy in figure 8a, the unconditional FID in figure 8b and the conditional FID in figure 8c for the different guidance setups and scaling strength. Sample images are shown in figure 2. The robust classifier





















generates class-conditional samples, trading off the overall image quality with the amount of conditioning added to the diffusion reverse process (figure 8c). All guidance setups improve the guidance mechanism for the non-robust classifier, with the 0.99-EMA stabilization reaching the lowest cFID score of 18.5 while maintaining good overall image quality with a FID of 69.6. The guidance by the non-robust classifier fails similarly as on the CelebA data without any stabilization techniques. The special setup of the data set with clear class boundaries and unambiguous class features is visible in the results, where the cFID decreases drastically when classifier guidance is successfully applied (table 2).

Table 2: Metrics and first 10 samples for unconditional diffusion sampling (left) and conditional diffusion sampling (right) with the non-robust classifier, $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction (eq. 5), 0.99-EMA stabilization (eq. 7) and guidance scale of 15.0 on the SportBalls data set. Metrics calculated over a batch of 50176 samples. More images are shown in the supplementary material.

Unconditional					Conditional (best)				
									
									
<i>acc.: 29.63%, FID: 54.37 , cFID: 155.19</i>					<i>acc.: 91.05%, FID: 69.64 , cFID: 18.52</i>				

For our real-world CelebA-HQ data set, we test our best guidance setup on an off-the-shelf DDPM. We evaluate the metrics on 1024 samples and show the first 10 generated samples as well as the corresponding metrics in figure 3. The stabilized classifier guidance for the non-robust classifier with $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction and 0.99-EMA successfully generates class-conditional samples by achieving $> 99\%$ target class accuracy, reducing the cFID by 7.52 points and trading-off conditioning with overall sample quality (FID increased by 5.1 points). Visually, male faces are slightly altered towards what the classifier believes are female features.

Table 3: Metrics and samples for unconditional diffusion sampling (left) and conditional diffusion sampling (right) with $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction (eq. 5), 0.99-EMA stabilization (eq. 7) and guidance scale of 10.0 on CelebA-HQ. Metrics calculated over a batch of 1024 samples. The top row shows the first 5 generations, the bottom row shows 5 hand-picked seeds for which the unconditional model produces the class male. More images are shown in the supplementary material.

Unconditional					Conditional				
									
									
<i>acc.:</i> 77.1%, <i>FID:</i> 45.57 , <i>cFID:</i> 49.70					<i>acc.:</i> 99.02%, <i>FID:</i> 50.67 , <i>cFID:</i> 42.18				

4 Related Work

Classifier guidance as proposed in [4] requires a classifier, which was trained on the same noise as introduced in the diffusion forward process. A one-step estimate of the denoised image from the diffusion model was proposed to apply classifier guidance to noise-unaware classifiers [2], which we refer to as $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction in our paper. In combination with the $\hat{\mathbf{x}}_0^{(\mathbf{x}_t)}$ -prediction, a robust classifier restricting the non-robust classifier gradients can be used in conjunction to enable guidance on arbitrary classifiers [1]. However, to the best of our knowledge, no paper has so far specifically addressed the challenge of non-robust classifiers for classifier guidance without training a specialized classifier or diffusion model. For classifier guidance with robust classifiers, multiple improvements have been suggested, for example [5,6].

Classifier-free guidance [10], as the predecessor of classifier guidance, subsumes the auxiliary classifier into a Bayesian implicit classifier in the form of a conditional diffusion model. Through training a conditional diffusion model, the unconditional and conditional denoising steps can be traded-off to achieve conditioning during sampling. We mention this parallel line of work for completeness, but note that classifier-free guidance always requires training a conditional diffusion model, which therefore does not allow adding arbitrary conditioning information in the diffusion reverse process without retraining.

5 Conclusion

In this study, we have extended classifier guidance techniques to non-robust classifiers within denoising diffusion probabilistic models (DDPMs). By addressing the inherent limitations of requiring specifically trained robust classifiers for classifier guidance, we built on top of previously proposed one-step denoised image predictions to stabilize guidance gradients during the sampling process. Our findings demonstrate that incorporating stabilization techniques, particularly exponential moving averages, enhances gradient stability, bridging the performance gap between non-robust and robust classifiers. The experimental results on the CelebA data set indicate that our approach not only improves classification accuracy but also maintains sample diversity and visual quality in generated images. Future work will focus on refining these methods and exploring their applicability to other generative models and diffusion samplers. Especially other techniques from stochastic optimization and dynamic guidance schedules will be explored.

Limitations Classifier-guidance is sensitive to hyperparameter choices, especially the guidance scaling. We explored many hyperparameter choices in this study but did not specifically optimize for state-of-the-art FID scores. We only explored two stabilization techniques based on SGD with momentum and ADAM as the two most commonly used methods from stochastic optimization. This shows stabilization techniques are promising candidates to improve classifier guidance, other techniques not explored in this study may however improve gradient stability even further. We also use the FID metric as-is with the feature extractor pre-trained on ImageNet [3]. This results in higher FID values for CelebA and very high FID values for SportBalls, since the features in the data sets are different to features extracted on ImageNet. In our analysis, we used one representative classifier architecture (MobileNetV3). Other architectures may require different hyperparameter choices. The same applies for the diffusion process, where we only used the standard DDPM setup. Translating our findings to other diffusion reverse samplers is subject to future work.

Acknowledgments. This research is supported by the Center for Artificial Intelligence (CAIRO) at the Technical University of Applied Sciences Würzburg-Schweinfurt (THWS), Würzburg, Germany and the Bavarian Hightech Agenda.

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

References

1. Augustin, M., Boreiko, V., Croce, F., Hein, M.: Diffusion Visual Counterfactual Explanations. *NeurIPS* (2022)
2. Avrahami, O., Lischinski, D., Fried, O.: Blended Diffusion for Text-driven Editing of Natural Images. *CVPR* (2022)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A Large-scale Hierarchical Image Database. *CVPR* (2009)
4. Dhariwal, P., Nichol, A.: Diffusion Models Beat Gans on Image Synthesis. *NeurIPS* (2021)
5. Dinh, A.D., Liu, D., Xu, C.: Pixelasparam: A Gradient View on Diffusion Sampling with Guidance. *ICML* (2023)
6. Dinh, A.D., Liu, D., Xu, C.: Rethinking Conditional Diffusion Sampling with Progressive Guidance. *NeurIPS* (2024)
7. Gruver, N., Stanton, S., Frey, N., Rudner, T.G., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., Wilson, A.G.: Protein Design with Guided Discrete Diffusion. *NeurIPS* (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. *NeurIPS* (2017)
9. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models. *NeurIPS* (2020)
10. Ho, J., Salimans, T.: Classifier-free Diffusion Guidance. *NeurIPS Workshop on Deep Generative Models and Downstream Applications* (2021)
11. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for Mobilenetv3. *ICCV* (2019)
12. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive Growing of Gans for Improved Quality, Stability, and Variation. *ICLR* (2018)
13. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. *ICLR* (2015)
14. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep Learning Face Attributes in the Wild. *ICCV* (2015)
15. von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Wolf, T.: Diffusers: State-of-the-art Diffusion Models. <https://github.com/huggingface/diffusers> (2022)
16. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional Networks for Biomedical Image Segmentation. *MICCAI* (2015)
17. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Representations by Back-propagating Errors. *Nature* (1986)
18. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep Unsupervised Learning using Nonequilibrium Thermodynamics. *ICML* (2015)
19. Vaeth, P., Fruehwald, A.M., Paassen, B., Gregorova, M.: Generative Example-based Explanations: Bridging the Gap Between Generative Modeling and Explainability. *arXiv:2410.20890* (2024)
20. Vaeth, P., Fruehwald, A.M., Paassen, B., Gregorova, M.: Gradcheck: Analyzing Classifier Guidance Gradients for Conditional Diffusion Sampling. *arXiv:2406.17399* (2024)
21. Weiss, T., Mayo Yanes, E., Chakraborty, S., Cosmo, L., Bronstein, A.M., Gershoni-Poranne, R.: Guided Diffusion for Inverse Molecular Design. *Nature Computational Science* (2023)

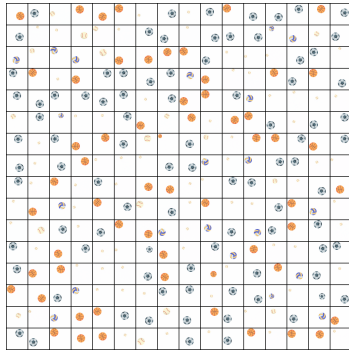
A Supplementary material (compact version)



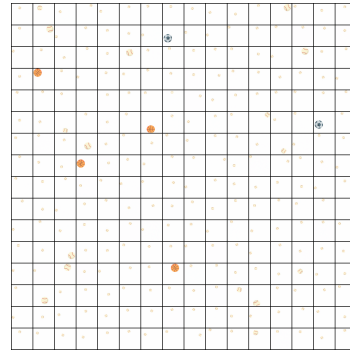
(a) Unconditional CelebA samples



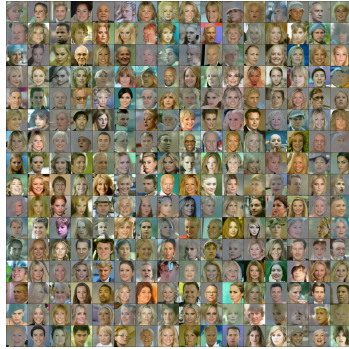
(b) Conditional CelebA samples, s=150



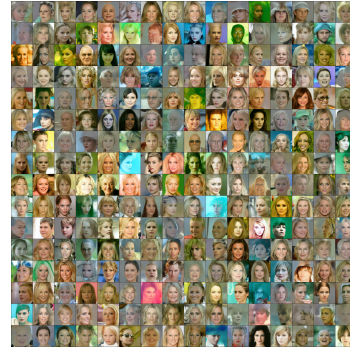
(c) Unconditional SportBalls samples



(d) Conditional SportBalls samples, s=15



(e) Unconditional CelebA-HQ samples



(f) Conditional CelebA-HQ samples, s=10

Fig. 9: More generations for the different data sets. All conditional samples are generated with our guidance setup of $\hat{x}_0^{(x^t)}$ -prediction, 0.99-EMA stabilization and the data set specific guidance scale s . We show **the first 256** generations.