Universal Guided Diffusion

Vladislav Pyzh
vladislav.pyzh@polytechnique.edu

Igor Gogarev
igor.gogarev@polytechnique.edu

Patrick Tourniaire
patrick.tourniaire@polytechnique.edu

Abstract

In this paper, we investigate the impact of the universal guidance method proposed in [1] as a diffusion guidance method on adhering to additional constraints in the reverse diffusion process. Specifically, we measure universal guidance's ability to respect the additional constraints with respect to the baseline that uses textual scene descriptions. We clearly observe that universal guidance outperforms the baseline when injected with mask constraints. We observe some challenges when using bounding boxes as constraints such as adversarial attacks. Finally, we analyzed the parameter sensitivity of the method on the MNIST dataset highlighting the importance of hyperparameter analysis for the specific applications of this method.

1. Introduction

Image diffusion is a state-of-the-art (SOTA) technique for visual synthesis [4, 11, 7]. Diffusion models generate synthesized images based on constraints injected to the model. Typically, this has been text latents describing the scene of a desired image. However, this can be limiting for complex scenes and therefore a large research area in diffusion models has been dedicated to injecting different modalities. ControlNet by Zhang Lvmin, et al. [13] is one popular example, enabling segmentation maps, binary edges and other visual constraints to guide the diffusion process. However, this is a classifier free method, requiring training of the diffuser with these constraints.

In earlier works by Dhariwal Prafulla, et al. [4] classifier guidance was used by injecting CLIP into the backward process to directly alter the diffusion path. However, one limitation of this approach was the fact that the guidance was applied directly in the noisy domain. For CLIP which was not trained on this domain it was hard to justify how well it could perform in a noisy domain. Furthermore, other guidance methods at the time before Universal Guidance [12, 2] relied on simple guidance functions rather than a comprehensive study of universal guidance functions for multiple downstream applications.

Therefore, in this paper we aim to evaluate Universal Guidance as a method for injecting different guidance constraints in the backward diffusion process for downstream tasks. Therefore, our aim is to reproduce the Universal Guidance method, and experiment with our own dataset for quantitative evaluation using class labeled segmentation masks during diffusion. Additionally, to gain a perspective on the versatility of the method we experiment with a second guidance function, based on labeled bounding boxes for compositional control. Lastly, we will perform a more extensive look into the hyperparameter sensitivity of the method using the MNIST dataset to reduce the computational load.

Research Question 1. *Does universal guidance with segmentation masks truly respect the composition better than pure textual constraints?*

Research Question 2. *How closely do bounding box guidance follow ground truth positions during guidance?*

Research Question 3. *What hyper-parameter sensitivities exists in Universal Guidance on a simple dataset like MNIST?*

To better frame our study on Universal Guidance, we first present the foundational theory of image diffusion and Universal Guidance as presented by Arpit Bansal, et al. [1].

2. Background

The two core processes in image diffusion are the forward- and backward process, as presented in Definition (1) and Definition (2) respectively. Furthermore, we note that current SOTA diffusion models operate on a latent representation of the image to boost inference time [7] . We follow this by denoting the latent representation of some image $\mathbf{x}$ by $\mathbf{z}$.

Definition 1 (Forward Process)**.** $\mathbf{z}_0$ *denotes some image, which is progressively noised through a Markov chain of*

Gaussian distributions from $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_T$. *Each transition* $t \rightarrow (t+1)$ *in the Markov chain produces noise* $q(\mathbf{z}_t|\mathbf{z}_{t-1})$ *defined by.*

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) := \mathcal{N}\left(\mathbf{z}_t; \sqrt{1-\boldsymbol{\beta}_t}\mathbf{z}_{t-1}, \mathbb{1}\boldsymbol{\beta}_t\right) \tag{1}$$

Where $\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T$ *defines the variance schedule, dictating how much noise is added until* $\mathbf{x}_T$ *is pure noise in the form* $\mathcal{N}(\mu, \mathbb{1})$, *where* μ *is centered at 0. The computation of some* $\mathbf{z}_t$ *can be simplified to an immediate computation as shown by Jonathan Ho, et al. [5].*

$$\mathbf{z}_t = \sqrt{\alpha_t}\mathbf{z}_0 + (\sqrt{1-\alpha_t})\epsilon \tag{2}$$

Where $\epsilon \sim \mathcal{N}(\mu, \mathbb{1})$, *and* α_t *is computed from the beta schedule as per Jonathan Ho, et al. [5].*

As noise is added according to a Markov chain of Gaussian noise in Definition (1), we can estimate the noise added for some transition $t \rightarrow (t-1)$ using the theory in Definition (2) and an architecture like denoising UNets [10, 5].

Definition 2 (Backward Process)**.** *The backward process uses the learned parameters* $\boldsymbol{\theta}$ *of a denoiser architecture to estimate the noise added during a forward step. Thus, estimating the noise added in* $(t-1) \rightarrow t$ *enabling it to be removed from the image* $\mathbf{z}_t$. *Giving the noise estimation* $p_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}|\mathbf{z}_t)$.

$$p_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}|\mathbf{z}_t) := \mathcal{N}\left(\mathbf{z}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_t, t), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)\right) \tag{3}$$

For the model to learn to estimate the added noise, the model is typically trained such that the estimated noise $\epsilon_{\boldsymbol{\theta}} \approx \epsilon$. *Then using this estimated noise we can reverse the process using inference techniques such as DDIM [11] or DDPM [5] to obtain a clean sythesized image.*

$$\hat{\mathbf{z}}_0 = \frac{\mathbf{z}_t - (\sqrt{1-\alpha_t})\epsilon_\theta(\mathbf{z}_t, t)}{\sqrt{\alpha_t}} \tag{4}$$

Naturally during inference, the path taken to generate $\mathbf{z}_0$ is based on the parameters θ given some optional constraint $\mathbf{z}_c$ which can be injected into the diffuser or guide the backward process directly. Universal Guided Diffusion by Arpit Bansal, et al. [1] uses the latter approach, which enables any form of guidance function to be injected in the backward process. This idea is highlighted in Definition (3) and Definition (4).

Definition 3 (Forward Universal Guidance)**.** *Given a guidance function* f *and a loss* l, *universal guidance directs the estimate of* $\hat{\epsilon}_\theta(\mathbf{z}_t, t)$ *using the loss gradient wrt. the guidance function and the synthetic image* $\hat{\mathbf{z}}_0$. *Which is computed by.*

$$\hat{\epsilon}_\theta(\mathbf{z}_t, t) = \epsilon_\theta(\mathbf{z}_t, t) + s(t)\nabla_{\mathbf{z}_t} l(\mathbf{c}, f(\hat{\mathbf{z}}_0)) \tag{5}$$

Naturally, this assumes that f *is a differentiable function with a target labeling of* $\mathbf{c}$.

The authors note that forward universal guidance can over priorities the realness of the image and thus lose some of the effect of the guidance. Therefore, a supplementary method called backward universal guidance was proposed to address this issue, defined in Definition (4).

Definition 4 (Backward Universal Guidance)**.** *Instead of computing* $\nabla_{\mathbf{z}_t} l(\mathbf{c}, f(\hat{\mathbf{z}}_0))$, *they compute a guided change* $\Delta\mathbf{z}_0$.

$$\Delta\mathbf{z}_0 = \underset{\Delta}{\operatorname{argmin}} \left[l(\mathbf{c}, f(\hat{\mathbf{z}}_0 + \Delta))\right] \tag{6}$$

Which is computed using m-step gradient descent. Where $\Delta = 0$ *is the starting point. Here* $\Delta\mathbf{z}_0$ *is the best respecting change in the clean data space and thus they translate this into the noisy space.*

$$\mathbf{z}_t = \sqrt{\alpha_t}(\hat{\mathbf{z}}_0 + \Delta\mathbf{z}_0) + (\sqrt{1-\alpha_t})\tilde{\epsilon} \tag{7}$$

Where $\tilde{\epsilon}$ *is an augmentation of the original denoising prediction.*

$$\tilde{\epsilon} = \epsilon_\theta(\mathbf{z}_t, t) - \left(\sqrt{\alpha_t/(1-\alpha_t)}\right)\Delta\mathbf{z}_0 \tag{8}$$

Naturally, the advantage of this approach is that we can inject any differentiable loss function into the process to utilize additional constraints as the guidance functions operate on the clean data space. Additionally, the method does not require any additional training procedure, which can be desirable in downstream tasks which lack access to high quality datasets.

3. Experiments

To evaluate universal guided diffusion's ability to adhere to mask constraints in the reverse diffusion process, we compared its performance with text guided diffusion using Stable Diffusion 1.4 [9] for both. For this baseline we used textual prompts to describe the scene. Then for Universal Guidance we used the same SD v1.4 model but with forward/backward guidance injected into the reverse diffusion process, where the guidance function loss was the cross entropy between semantic segmentation masks. These enabled us to gain core insights into the universal guidance approach and therefore answer Research Question (1).

We computed the forward pass for the entire dataset, whereas for the backward pass we performed computations only for one image as a proof of concept. This limitation is due to the high computational demands of the backward pass: we need to solve an optimization problem at every

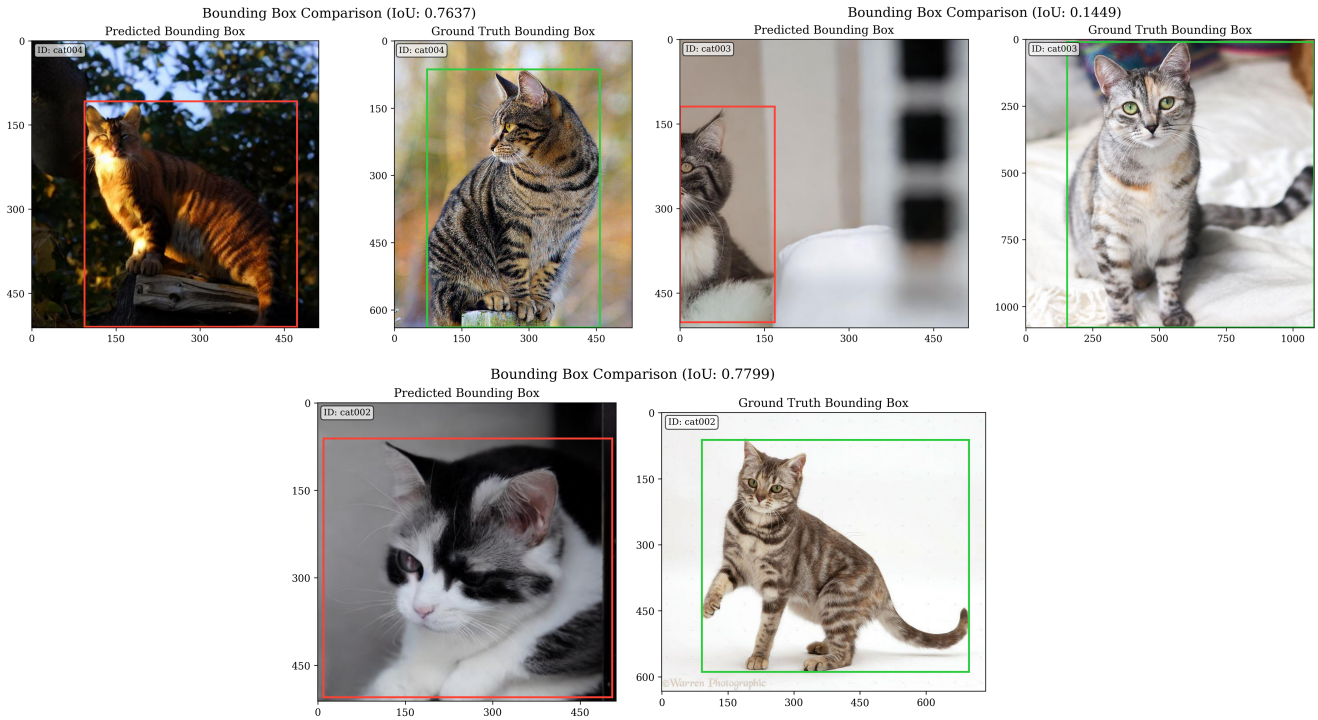


Figure 1: Qualitative results for the use of bounding boxes as a guidance.

diffusion step, resulting in around 40 calls to the guiding segmentator. The backward pass for a single image required approximately 5 hours of computation using RTX 4090 GPU. It is worth mentioning that in the original paper, the backward pass is not used for mask guidance.

3.1. Dataset

To verify how well universal guided diffusion adhered to the given mask constraints, we used a public animal image dataset[1]. This dataset contained around 300 images of dogs, cats and foxes, where each class had an approximately equal amount of images. Due to the lack of computational resources, we limited our research only to the "cat" class. Naturally, the goal of the reverse diffusion model was to replicate these images based on the position of the animal in the image. With universal guided diffusion we needed two things to accomplish this; a textual prompt describing the scene and a mask of the desired animal position/pose.

Therefore, we extracted segmentation masks of the animals using Grounded-SAM [8]. Grounded-SAM is a variant of SAM connected to Grounded-DINO to extract bounding boxes given a text prompt which is then used as input to SAM to extract high quality segmentation masks. This process looked like the following; (1) we fed the label of the animal (cat/dog/fox) to Grounded-SAM to extract segmentation masks, (2) described the image scene using LLaVa model [6] to obtain a global image description which was used as the textual prompt in reverse diffusion.

3.2. Additional Experiments

Along side our main research objective using segmentation masks, we investigated Universal Guidance's performance on bounding boxes and performed and a hyper parameter analysis using the MNIST dataset. We chose to investigate with bounding boxes due to the fact that we could easily use the same bounding boxes which is extracted from grounding DINO as a part of grounding SAM. Then for the guidance function, we used a light weight detector from the PyTorch library[2]. In the same fashion as for segmentation masks, due to computational constraints we only performed experiments on the cat images of our dataset. Using the servers provided by Polytechnique, both the mask and bounding box based methods spent $\tilde{2}$ hours per image and we found that the storage capacity was too low for us to be able to load model weights and the dataset. Therefore, we used two instances of RTX4090 GPUs from VastAI paid by ourselves which reduced the computation time to $\tilde{1}5$ min per image. Given that there was 100 images per label, it was infeasible for us to pay more for the compute to also

[1] https://www.kaggle.com/datasets/snmahsa/animal-image-dataset-cats-dogs-and-foxes

[2] https://pytorch.org/vision/master/_modules/torchvision/models/detection/faster_rcnn.html#FasterRCNN_ResNet50_FPN_V2_Weights

perform the experiments on dogs and foxes. Finally, this enabled us to answer Research Question (2).

3.3. Hyperparameters Analysis on MNIST dataset

As a part of our project, we conducted a research on the main hyperparameters of the Universal Guidance algorithm. It was done on the MNIST dataset, because we were limited in the resources available for the multiple large model inferences. For the diffusion model we used UNet2DModel. We trained it for 3 epochs with the configuration of 1000 denoising steps, using Adam optimizer with learning rate 1e-4. For the classifier (used as a Guidance function) we used a tiny 2 layer convolutional neural network with ReLU activations and MaxPool layers. The classifier was trained for 5 epochs with Adam optimizer and learning rate equal to 1e-3. It achieved the desired high performance on the MNIST classification task so that we were sure it is guiding correctly.

We tested multiple hyperparameters. We investigated the influence of the number of optimization steps in the universal backward pass on the resulting quality of the images. Furthermore, we studied the relationship between the number of self-recurrence steps and the resulting image quality.

Lastly, we compared the images using different schedulers used in the forward pass. We tried to see how different reducing dynamics for the scheduler steps will influence the final result. By tuning the coefficients of the functions, we ensured that the sum of all points in each function is close to each other, which means that overall during the image generation we used the same value of guidance, but with different dynamics. We tested 4 different schedulers.

$$s(t) = \frac{1}{k \cdot x + b} + a \tag{9}$$

$$s(t) = s \cdot \log_{10}(x + f) + g \tag{10}$$

$$s(t) = -x + u \tag{11}$$

$$s(t) = const \tag{12}$$

This schedulers (Fig. 2) provided options with a slower or faster decrease in the beginning of the denoising process, as well as edge cases like linear schedulers or constant value.

4. Results

In this section, we present a collection of experiments to directly answer our research questions. First we present a setup similar to the original Universal Guidance paper using masks, then we present results using labeled bounding boxes as a guide, and finally we will present a hyperparameter analysis of Universal Guidance using a low computation setup with the MNIST dataset [3].

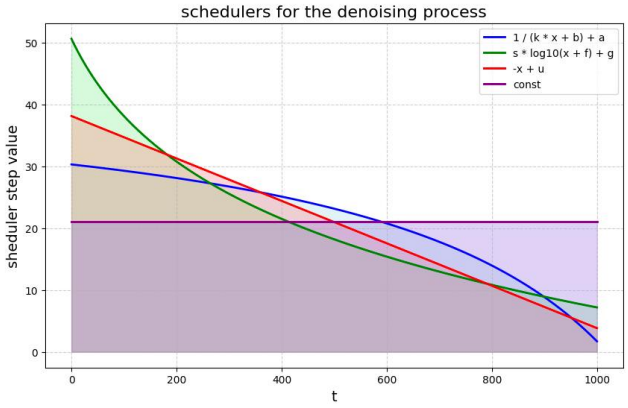
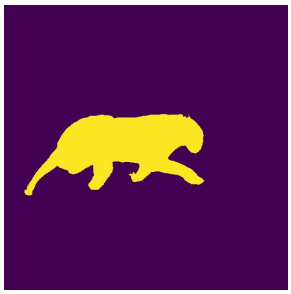


Figure 2: Multiple schedulers tested during MNIST experiments.

4.1. Mask Guidance

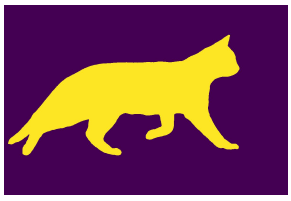
This experiment shows that the proposed guidance method significantly improves the IoU score compared to the baseline. Qualitative results for the forward pass are provided in Table 1. However, we see that the IoU of 0.53 does not fully correspond to cases where images strictly follow the guidance. After a qualitative analysis of images and mask generation, we observe that diffusion rarely fails to capture the positions of the cat's body (though it does happen), but quite often misses minor details such as tails. We provide an example of relatively successful guidance in Figure 3.

(a) Generated image



(b) Generation mask

(c) Ground-truth image



(d) Ground-truth mask

Figure 3: Comparison of non-GT vs. GT images and masks.

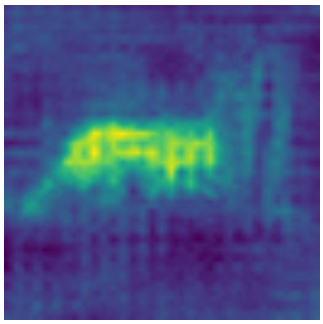


Figure 4: Guided mask

IoU comparison	
Baseline	0.29
Guided	0.53

Table 1: Quantitative results for mask segmentation guidance on cats

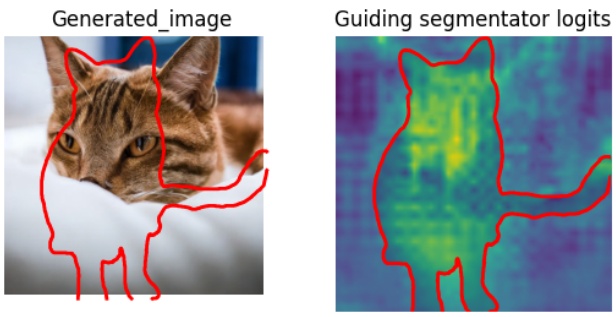


Figure 5: Backward pass results

As a significant limitation of the proposed method, we note that, due to the use of a VAE decoder, we cannot directly affect the generated image via the computed gradient; instead, we must do it through the decoder. We believe this limitation causes the poor rendering of thin details and leads to the specific structure of the mask predicted during sampling, as seen in Figure 4.

We also provide a result for the backward pass in Figure 5. Here, we observe another important limitation of both proposed methods. Although the generated image fails to capture the desired mask (contoured with a red line), the segmentator logits show relatively good results within the desired mask. We believe that during guidance, the weaker segmentator might not be robust enough and could be vulnerable to adversarial-like effects, resulting in good segmentation outcomes but a generated image that significantly diverges from the target.

4.2. Classification Guidance

Bounding boxes as a guidance provides a much coarser constraint than masks, and it is mainly useful to fix the general position of objects in the image. Indeed, this is useful considering the difficulty of constraining object compositions using text prompts in diffusion models. Table (2) highlights the quantitative results of this experiment on bounding boxes, focusing only on cats.

Classification Guidance		
Label	IoU	CI 95%
Cat	0.48	$\pm$0.15

Table 2: Quantitative results for classification guidance on cats

The main takeaway from this experiment is that the choice of the object detection model is extremely important. Due to our computational constraints we had to go for a less powerful object detector model, which affected the results. Mainly we observed that even when the generated image clearly contained a cat, it would label it as something completely irrelevant such as a stop sign. To mitigate this we had to set a very low prediction threshold around 40%. Naturally, this will influence the results. We see that for the successful cases where cats where found, the IoU with the gold is not the best. With a mean IoU of 0.48 and a quite large 95% of 0.15, indicating that the model struggles to adopt the bounding boxes in the universal guidance effectively. Potentially due to the classification model performance. This is further highlighted in Figure (1), where we have some successful cases but also one case which completely fails.

4.3. Parameter Analysis on MNIST

While conducting the experiments with different hyperparameter values, we decided to visually compare the output images. To make the comparison more stable, for each experimental setup we generated 10 images. Furthermore, to be able to compare certain patterns of the digits, we decided to generate only the digit 2. We put in green box all the examples that are successful in our opinion.

4.3.1 Number of self-recurrence steps

We tested the influence of the value of the self-recurrence parameter on the quality of the resulting images. We chose to generate images with the parameter values **k** 1, 4 and 8 (Fig. 6). It is clearly seen that the increase of this parameter value improves the quality of generated images. We compared the results visually and concluded that with **k =**

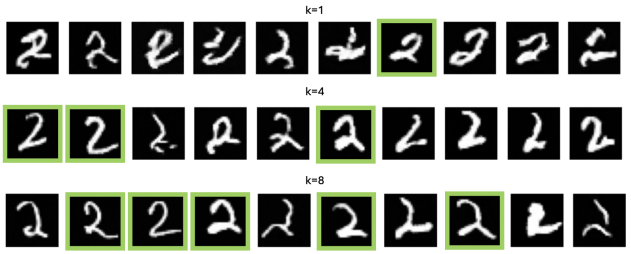


Figure 6: Generated examples with different number of self recurrence steps.

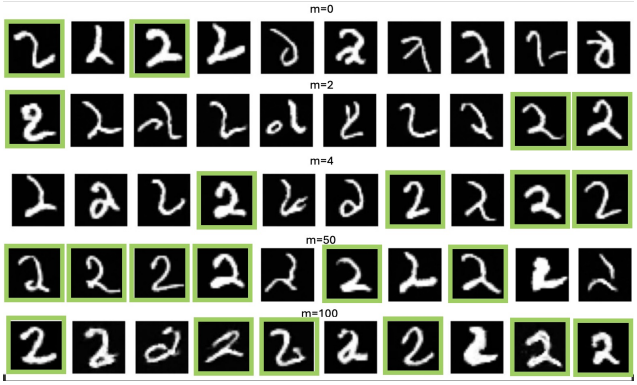


Figure 7: Generated examples with different numbers of optimizations steps.

1 algorithm generated 1 successful image, with **k = 4** - 5 successful images, with **k = 4** - 8 successful images.

4.3.2 Number of optimization steps in the Universal backward step

We tested multiple values for the number of optimizations steps **m** in the universal backward step, including 0 (i.e. only forward pass algorithm), 2, 4, 50, 100 and achieved the following results (Fig. 7). It can be seen that without backward step or with small number of optimization steps the algorithm is not providing many successful examples. However, increasing the number of steps to 50 leads to more frequent correct examples. At the same time, an increase further up to 100 steps does not greatly influence the result, which means that the optimal number of optimization steps in this set-up probably lies between 50 and 100.

4.3.3 Different guidance schedulers analysis

We created multiple schedulers with different dynamics while ensuring that in total they provide the same amount of guidance. As a result (Fig. 8), we did not see much difference between using schedulers that are described above. However, we suppose that this behaviour might be due to the simplicity of the task. Thus, more complex tasks, such as animals generation, can be more sensitive to the scheduler choice and require more precise parameter tuning.

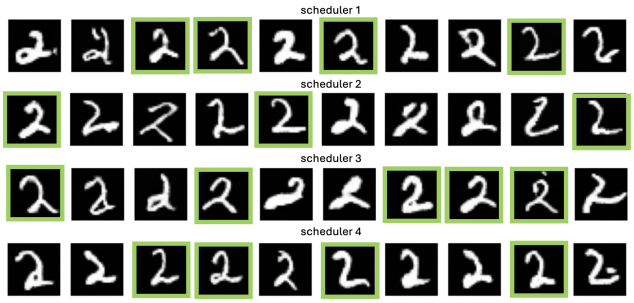


Figure 8: Generated examples with different guidance schedulers

5. Conclusion

In this work, we evaluated the proposed guidance approach for injecting information using different modalities. We observed that for segmentation masks, this approach adjusts the generation to be more coherent with the target. Specifically, the segmentation mask-guided method demonstrated a substantial improvement in mean IoU scores (0.53 vs. 0.29 for textual baselines), confirming its strength in better capturing scene compositions.

However, our experiments also highlight notable limitations of the method. Universal Guidance in the setup with Stable Diffusion struggles with finer image details, as observed in the provided examples. We believe this issue arises from guidance via latent variables and VAE decoders, causing an inability to directly adjust the image in correspondence with the guidance mask.

Additionally, to experiment with the flexibility of the universal guidance method for different constraints we also used a object detector with bounding boxes. Where we focused only on the placement of the object class in the image. Indeed, a notable limitation in this experiment was the low quality of the detector itself. For example, for small pixel perturbations we observed that cats could be labeled as traffic signs. Demonstrating the importance of a robust guidance network when it comes to universal guidance diffusion.

We also tested the algorithm on a simple setup with MNIST dataset to see the influence of hyperparameters on the final images quality. We evaluated the outputs by changing the number of self-recurrence steps, number of optimization steps and forward pass scheduler functions. We conclude that even in a simple experimental setup with MNIST bigger numbers of self-recurrence steps and optimizations steps give better results. At the same time, we have not noticed any difference when using different sched-

uler functions.

We also observed challenges beyond just the quality of generated images. In particular, we would like to emphasize the high hardware requirements of this method. Diffusion models are inherently computationally expensive, and the additional cost of computing both the denoising model and classifier, alongside optimization during the backward pass, significantly increases resource requirements. Finding suitable guidance functions and hyperparameters is thus complicated by these excessive computational demands. We believe that the work could benefit from a technique that introduces automatic evaluation of the required guidance schedule.

We hypothesize that combining the backward pass with self-recurrence may lead to unpredictable behavior, as updates to z_t depend on changes applied to z_0. A rigorous mathematical analysis of this interaction would be valuable for better understanding the behavior and stability of this approach. We want to note that the initial paper does not use a backward pass for segmentation masks and does not contain insights into how these two methods fit together.

6. Team Work Split

- Vladislav Pyzh: Developed the core parts of the forward and backward passes, tested them on a toy example, and implemented mask segmentation guidance. Generated a ground-truth dataset using Grounded SAM and benchmarked the results against it. Wrote the corresponding section and conclusion.

- Patrick Tourniaire: created a generalized script for accepting any guidance loss as a class injected into our universal guidance implementation. Performed experiments on universal guidance with bounding boxes. Written introduction, background, experimental setup and bbox guidance sections.

- Igor Gogarev: did all the experiments with the MNIST part. Worked on the schedulers functions tuning, writing code for the MNIST experimental setup and the corresponding sections in the report. Participated in the implementation of the forward universal guidance pass.

References

[1] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 843–852, June 2023.

[2] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

[3] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[4] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2024. Curran Associates Inc.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[6] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[7] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023.

[8] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[11] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.

[12] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *The Eleventh International Conference on Learning Representations*, 2023.

[13] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824, 2023.