Published in Image Processing On Line on 2025–03–00.
Submitted on 2024–09–26, accepted on 2025–01–29.
ISSN 2105–1232 © 2025 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2025.580

Latent Diffusion Approaches for Conditional Generation of Aerial Imagery: A Study

Roger Marí, Rafael Redondo

Eurecat, Centre Tecnològic de Catalunya, Multimedia Technologies, Barcelona, Spain
{roger.mari, rafael.redondo}@eurecat.org

Communicated by Pablo Musé *Demo edited by* Roger Marí

Abstract

Generative artificial intelligence is increasingly being applied in diverse areas such as architecture design, music composition, or character animation. Among the generative methods, diffusion models are today the state of the art in the synthesis of high quality images with inherent diversity and realism. This paper aims to evaluate the fidelity and realism of the synthesis achieved by different architectural variations of a latent diffusion model, which is used to generate aerial images conditioned to semantic maps. As shown in the results, the diffusion model tends to correctly capture the overall semantic structure and generates realistic textures, often with a lack of fine-grained detail. Among the conditioning variations, cross-attention layers were crucial to outline the semantic segments more accurately and exploit conditional data more effectively.

Source Code

The source code and documentation for this algorithm are available from the web page of this article[1]. Usage instructions are included in the `README.md` file of the archive.
This is an MLBriefs article. The source code has not been reviewed!

Keywords: generative artificial intelligence; diffusion models; latent diffusion; remote sensing.

1 Introduction

Deep generative models are a family of deep neural networks able to synthesize new samples by learning complex probability distributions from real data. Capable of synthesizing data of almost any nature, generative models are being applied for a wide variety of tasks (e.g., in creative industries), but also to augment databases for downstream tasks and mitigate the lack of data in some areas, such as medical imaging.

[1] https://doi.org/10.5201/ipol.2025.580

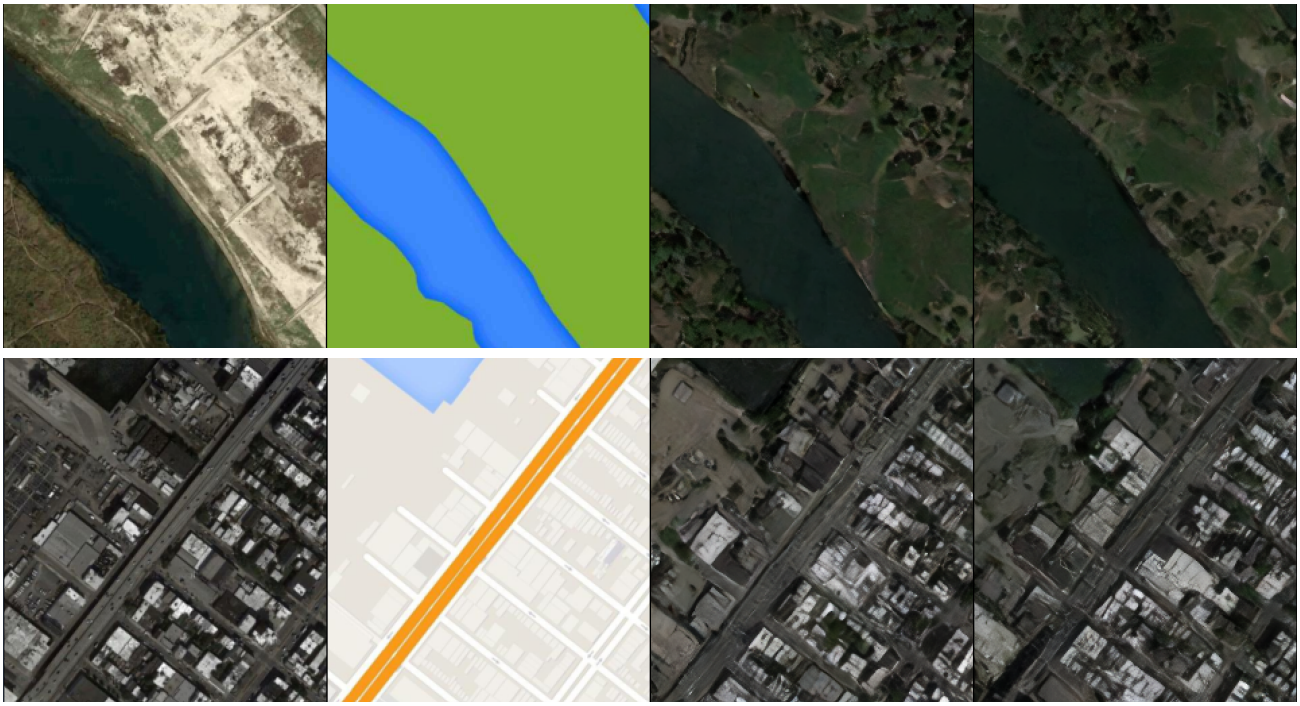

Figure 1: Left to right: real aerial image, conditional map input to the diffusion model and 2 different synthetic output samples.

Diffusion models represent today the state of the art in high-quality image synthesis. The diversity and realism achieved by diffusion models surpasses earlier generative models, such as variational autoencoders (VAEs) [11, 23, 16] or generative adversarial networks (GANs) [3, 12, 8, 9]. The diffusion fundamentals were first introduced in 2015 by Sohl-Dickstein et al. [18], and were extended for high-quality image synthesis in 2020 by Ho et al. [5]. Diffusion models became globally famous in 2022, after OpenAI released DALL-E 2 [14] for high-resolution text-to-image generation.[2]

The success of diffusion models is undisputed, but in this rapidly evolving field there is little consensus on how users should adapt these models to their own datasets. In this paper we explore a common scenario: given a collection of about a thousand images, the objective is to generate new samples conditioned to some additional information. Specifically, given a schematic map —as in Google Maps[3] or OpenStreetMap[4]— the goal is to generate realistic aerial images following the semantics of the map, as shown in Figure 1. The high cost of remote sensing images often limits the availability of large datasets, making it a field where diffusion models can play a significant role in synthesizing useful datasets.

2 Related Work

This section covers the fundamentals of diffusion models for image synthesis, followed by a focus on advanced techniques aimed at enhancing efficiency and performance, particularly for high-resolution image generation.

[2]OpenAI previously released in 2021 a first version of DALL-E [15] based on transformers instead of diffusion.
[3]https://www.google.es/maps
[4]https://www.openstreetmap.org

2.1 Diffusion Fundamentals and DDPMs

Diffusion-based image generation was first addressed by Denoising Diffusion Probabilistic Models (DDPMs) [5]. Denoising diffusion models consist of two processes: a *forward diffusion process* that gradually adds noise to the input, and a *reverse denoising process* that generates new samples by reversing the noise addition or denoising. As shown in Figure 2, both processes represent Markov chains, where each state depends on the previous state only.

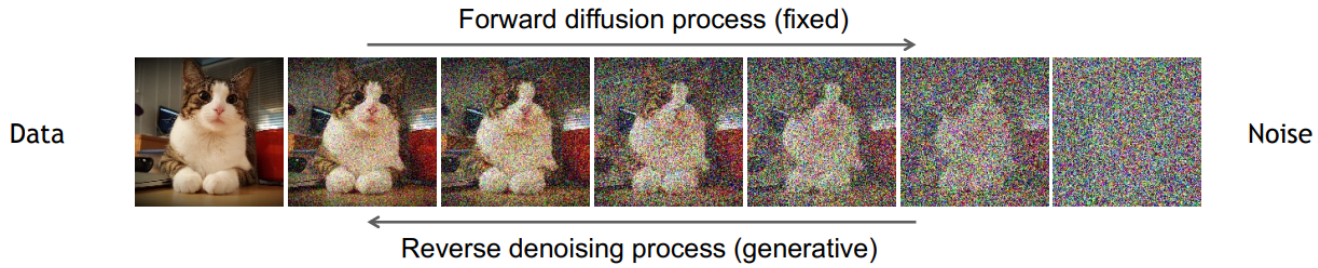

Figure 2: Forward diffusion process and reverse denoising process.Reproduced from [21].

Forward diffusion process. The addition of Gaussian noise to an image $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ can be expressed as $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \sigma^2)$, with mean $\boldsymbol{\mu}$ and variance σ^2. In the forward diffusion process, Gaussian noise is added iteratively and often reparameterized as

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right), \tag{1}$$

where $\mathbf{x}_t$ is the data at time step t, $\mathbf{x}_{t-1}$ is the previous state, and β_t is a hyperparameter controlling the noise variance at each step. New samples can be drawn from $\mathcal{N}$ as $\mathbf{x} = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is noise from a standard normal distribution, i.e., $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Equation (1) shows how to add noise to each state one step at a time. However, given the initial sample $\mathbf{x}_0$, it is possible to directly generate the noisy sample $\mathbf{x}_t$ corresponding to any step t as

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right), \tag{2}$$

where, by reparameterization, $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Equation (2) is also known as the diffusion kernel.

The set of noise variances used at each diffusion step, i.e., $\{\beta_1, \beta_2, \ldots, \beta_t, \ldots, \beta_T\}$, are scheduled so that $\bar{\alpha} \to 0$ and therefore $q(\mathbf{x}_T \mid \mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ in (2). Thus, the data distribution follows a standard normal distribution at the end of the diffusion process at time step T.

Reverse denoising process. In this process a particular function of the form $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ must be learned, which removes noise iteratively starting from the last sample of the forward process $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ until converging to the source sample $\mathbf{x}_0$. Interestingly, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ can be approximated as a normal distribution for small noise variances β_t used in the diffusion steps (1). Thus, each denoising step can be expressed as

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I}\right), \tag{3}$$

also known as the parametric denoising distribution. Note that θ represents the learnable parameters of a neural network and σ is a hyperparameter controlling the variance at each step t.

At each denoising step, the network yields an estimation of the mean less-noisy image of the previous iteration, denoted as $\boldsymbol{\mu}_\theta$. In practice, instead of directly predicting $\boldsymbol{\mu}_\theta$, it is advantageous to express $\boldsymbol{\mu}_\theta$ in terms of a noise residual $\boldsymbol{\epsilon}_\theta$, which is easier to predict for the network [5].

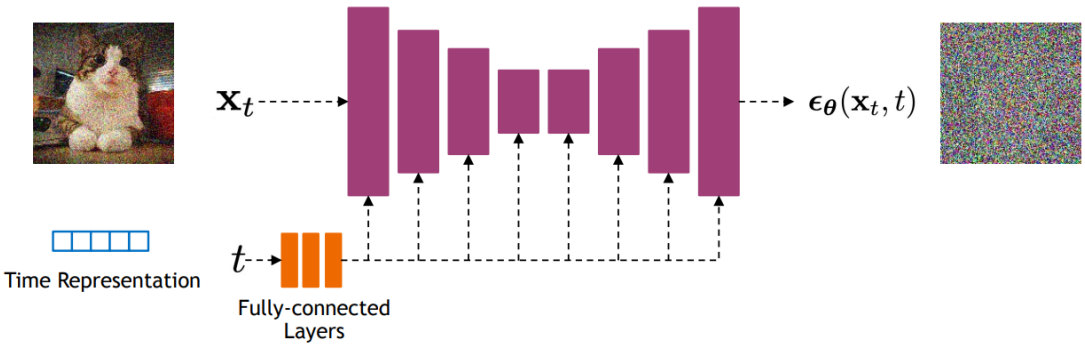

Figure 3: Diffusion model basic architecture diagram. Reproduced from [21].

Loss function. DDPMs are trained using the variational upper bound condition originally used in VAEs. As shown in [18, 5] this condition can be decomposed in different terms, where the driving force penalizes the difference between the prediction $\boldsymbol{\epsilon}_\theta$ and the actual noise $\boldsymbol{\epsilon}$ as follows

$$L_{t-1} = \mathbb{E}_{t\sim\mathcal{U}(0,T),\mathbf{x}_0\sim q(\mathbf{x}_0),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\lambda_t \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}_{\mathbf{x}_t}, t)\|^2] + C, \tag{4}$$

where C is a constant and the time-dependent coefficient λ_t depends on β_t and σ_t, although in practice $\lambda_t = 1$ is effective enough [5].

Network and hyperparameters (noise schedule). The loss function (4) implies that DDPMs are fundamentally noise prediction networks. This offers great flexibility to select a particular neural architecture, with the input and output being the same size as the only restriction. In the literature, U-Net architectures with ResNet blocks and self-attention layers are commonly used.

The time step t is embedded by a sinusoidal positional encoding or random Fourier features. It is accessible by all network layers by simple addition or other methods, as illustrated in Figure 3.

The noise schedule, i.e., hyperparamters β_t and σ_t^2, controls the variance of the forward diffusion and reverse denoising processes. A linear scheduler $\sigma_t^2 = \beta_t$ is commonly used, although other alternatives have been proposed [10, 13].

2.2 Scaling Up Diffusion Models

Score based generative models. Score-based models can be seen as an extension of DDPMs where the forward diffusion process and the reverse denoising process are expressed in terms of stochastic differential equations (SDEs), in which the time variable is continuous [20]. The key innovation in score-based models is that the model learns the gradient of the log probability density, which is easier to learn than the data distribution itself, as in DDPMs. Score-based frameworks have several advantages, e.g., more flexible and efficient sampling or exact likelihood computation.

Accelerated sampling. Accelerated sampling techniques in diffusion models aim to reduce the computational cost and time required to generate high-quality samples, which is a key challenge due to the iterative nature of the diffusion process. In standard DDPMs, sampling often involves hundreds or thousands of denoising steps. Accelerated methods, such as DDIM (Denoising Diffusion Implicit Models) [19], focus on reducing the number of steps while maintaining sample quality. These techniques leverage approximations or modified reverse processes to improve efficiency and make diffusion models more practical for real-time applications.

High-resolution synthesis. Latent diffusion pipelines are one of the most popular diffusion models for high-resolution image synthesis [17, 22]. Instead of working in the pixel domain, which would

require an extremely large number of iterations, they operate in a latent space, which is a more compact representation. Pre-trained autoencoders are often used to cover the conversion between pixel and latent space.

Classifier guidance is another popular strategy to improve the performance of diffusion models for high-resolution synthesis. In this case, a classifier is trained in parallel to the diffusion model and the gradients are mixed during sampling [2]. Another approach is to use implicit classifier guidance, which avoids the need for additional classifiers by randomly discarding the class condition during training [7]. Lastly, cascaded approaches involve using a sequence of diffusion models to improve the output resolution [6]. The initial model in the sequence is unconditional and produces low-resolution images, while the subsequent models enhance these images through super-resolution, conditioning on the outputs of the preceding models.

Conditional generation. Conditional diffusion models are a powerful approach to guide the image generation process towards specific features. Both the forward and reverse processes are influenced by input information to generate images that match the characteristics of this input. These models are frequently conditioned on data such as text prompts, class labels, or auxiliary images. The input information is encoded, e.g., with CLIP embeddings [14] or image encoders [17], and combined with noisy samples at each step. Additionally, complementary strategies such as the incorporation of conditional data into intermediate layers or using cross-attention mechanisms can further enhance the use of conditional data effectively.

3 Methodology

This work aims to experiment with different variations of conditional diffusion models in the generation of high-resolution aerial images by means of a small dataset made of around a thousand samples. The conditional data consists of a schematic map containing the contours of buildings and roads, as well as areas of water and vegetation.

3.1 Model Architecture

A latent diffusion model is used, consisting of two networks: a VQ-VAE and a U-Net, illustrated in Figure 4.

VQ-VAE. A variational autoencoder that performs the conversion between pixel space and latent space with a $\times 4$ spatial compression factor. Vector quantization (VQ) discretizes the latent samples into a codebook, which represents data in a more compact way, easing learning and optimization [23]. Running the diffusion process in the latent domain instead of the pixel domain reduces the training steps and achieves a high-resolution synthesis more efficiently [17].

U-Net. A convolutional neural network performing the forward and reverse processes in latent space. The U-Net is commonly used in diffusion models for its efficiency, its capacity to capture long-range dependencies, and its ability to maintain the resolution between input and output [1, 17]. Instead of predicting a less noisy image at each step, the U-Net actually learns to predict the amount —and shape— of noise present in the feedforward sample. Thus, in the reverse process, the predicted noise is iteratively subtracted from the sample as many times as performed in the diffusion process.

3.2 Conditioning Configurations

In image-conditioned diffusion models, the conditional image is commonly concatenated with the U-Net input, previously encoded or downsampled to fit the dimensionality of the latent space. Addi-

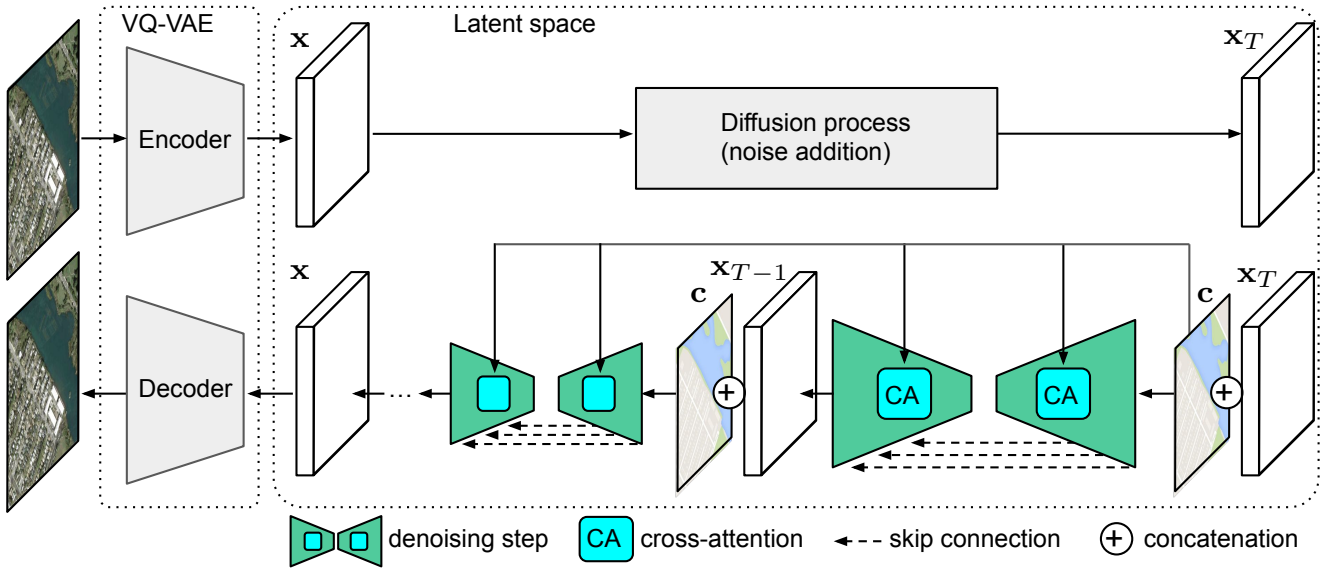

Figure 4: Diagram of the conditional diffusion pipeline. The input satellite image is converted by the VQ-VAE encoder into a latent space of lower dimensionality. The U-Net (green blocks) is trained to predict the noise in the latent samples (denoising step). The conditional image $\mathbf{c}$, is attached to the noisy latent vector $\mathbf{x}_t$ at each denoising step t. In addition, the condition $\mathbf{c}$ can be injected into cross-attention layers (CA) to further exploit conditional data.

tionally, the conditional image can also be injected into Cross-Attention (CA) layers [2, 17] at different resolution levels of the U-Net. Cross-attention layers focus on relevant and long-distance parts of images and the associated activations, leveraging semantic information more effectively. This conditioning mechanism encourages the model to focus on conditional data to generate synthetic samples consistent with it.

Being the main objective of this work, different conditioning configurations of the latent diffusion pipeline will be evaluated. In particular:

1. No conditioning: no areal semantic maps are fed into the model.
2. Downsampling condition: the model's input is concatenated with a bilinear downsampled version of the semantic map.
3. Encoding condition: the model's input is concatenated with a VQ-VAE-encoded semantic map.
4. Downsampling condition + CA: the semantic map is additionally flattened and injected into the cross-attention layers after being downsampled.
5. Encoding condition + CA: the semantic map is additionally flattened and injected into the cross-attention layers after being encoded.

3.3 Model Optimization

The diffusion model is optimized to minimize the square of the L^2-norm between the actual $\boldsymbol{\epsilon}$ and predicted noise $\boldsymbol{\epsilon}_\theta$, formulated as $\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{m}, t)\|_2^2$. Thus, at time t the prediction of the diffusion process relies on a noisy latent $\mathbf{x}_t$ and the conditional map $\mathbf{m}$. The larger the time step t is, the noisier the input sample $\mathbf{x}_t$. A total of 1,000 time steps are used for both training and inference. A pre-trained VQ-VAE [17] was used from the open-source Diffusers library.[5] The weights of the U-Net are optimized during training, while the VQ-VAE weights are frozen.

[5] https://github.com/huggingface/diffusers

For feasibility reasons, images are resized to 256×256 pixels, so latent samples are downscaled to spatial size of 64×64. The batch size is fixed to 8 —or 4 when CA layers are employed to alleviate memory cost— and the number of training epochs is fixed to 400.

3.4 Online Demo

This publication includes an online demo to test the proposed method on a set of predefined samples or on the user's own data. The method can be tested in inference, with pre-trained weights, and generates multiple synthetic samples for each selected map. The demo also allows to modify the number of time steps T in the generative process to regulate the amount of detail added to the synthesized sample, as illustrated in Figure 5.

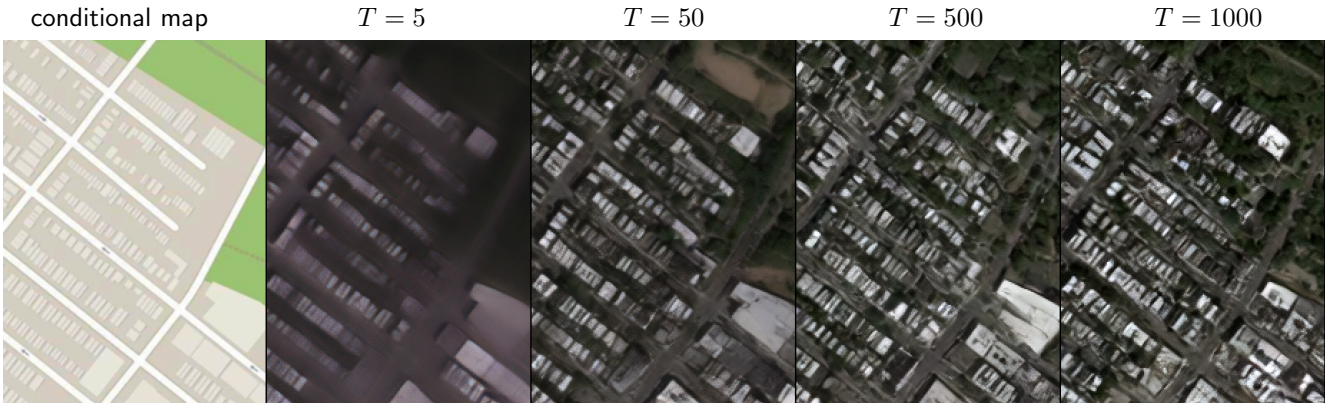

Figure 5: Effect of the number of time steps T on the synthesis quality. A small number of denoising steps at inference mostly produces low-frequency signals, while gradually increasing T induces more detailed and contrasted images.

4 Experiments

This section presents the experimental data and assesses the performance of the latent diffusion pipeline based on the conditioning approaches used to integrate information from the semantic map.

We use the publicly available *Maps* dataset [8]. It consists of 1,096 training samples and 1,098 validation samples taken from Google Maps. Each sample provides a base map of 600×600 pixels and its corresponding aerial RGB image of the same size. Some examples are shown in Figure 6.

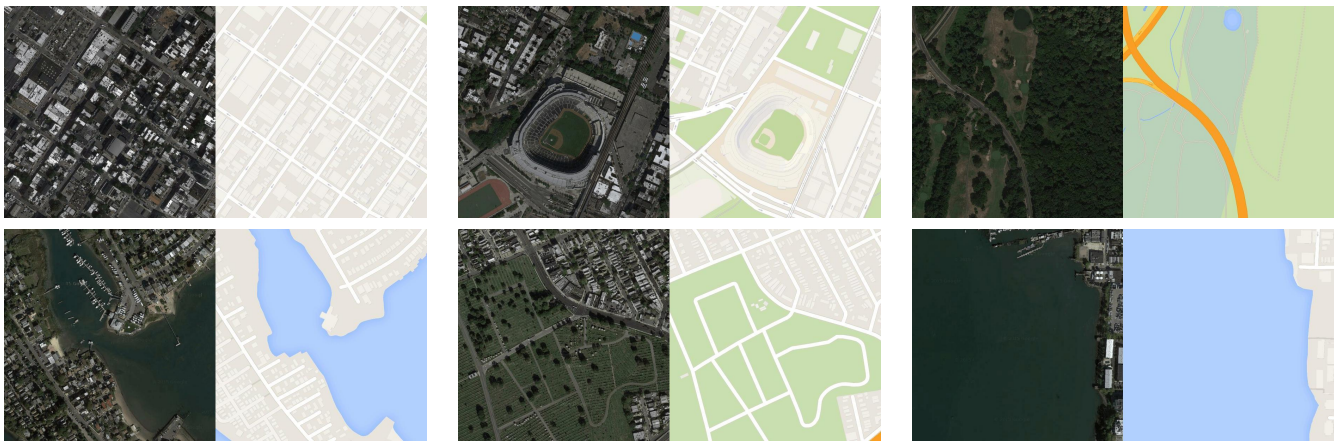

Figure 6: Some samples from the *Maps* dataset (train split).

4.1 Quantitative Assessment

The Fréchet Inception Distance (FID) [4] is used to quantify the quality of synthetic images compared to real samples. Lower FID score is generally associated with better image quality and diversity. The FID is calculated against images of the training set and the validation set. Note that, to generate the synthetic aerial images, only maps from the validation dataset are used as input condition.

Table 1 shows the FID scores obtained across the different conditioning configurations and reveals several observations of interest. In the first place, the differences in terms of FID are remarkable between training and validation real images (first row). This could be explained by the fact that the Inception network, responsible for extracting latent features on which the FID is computed, was not trained with aerial images. In the second place, FID is significantly lower on the validation set than the training set in the conditional configurations (2)-(5). Considering that synthetic samples were generated exclusively by using semantic maps from the validation set, such difference indicates that the conditional information is being considered effectively by the diffusion model and encourages similarity of the global image structure. In the third place, conditional configurations using the encoder VQ-VAE-encoded conditional image outperform their counterparts using downsampling. And lastly, the most substantial improvement on FID was achieved by the use of cross-attention, which additionally inject conditional information throughout all the U-Net layers.

Conditioning configuration		FID↓ vs. Train-Set	FID↓ vs. Val-Set	Average
Validation set	(reference)	59.4069	-	-
Unconditional synthesis	(1)	108.4007	129.1611	118.7809
Downsampling condition	(2)	126.3592	103.7375	115.0483
Encoding condition	(3)	127.3213	101.3111	114.3162
Downsampling condition + CA	(4)	116.1103	92.6007	104.3555
Encoding condition + CA	(5)	**104.4925**	**80.9066**	**92.6995**

Table 1: Comparison of the synthesis quality, based on FID, across different configurations in conditioning the latent diffusion model. Scores computed on 2,048 synthetic samples.

		FID↓ vs. Val-Set on Top N		
Conditioning configuration		Urban	Vegetation	Water
Downsampling condition	(2)	161.7278	158.6988	174.5742
Encoding condition	(3)	161.1109	154.9866	169.1432
Downsampling condition + CA	(4)	154.9340	148.7444	161.7717
Encoding condition + CA	(5)	**137.7760**	**139.5490**	**144.6613**

Table 2: Comparison of synthesis quality, based on FID, corresponding to the N synthetic images where the input conditional map has the majority of pixels for each class (urban surface, vegetation and water), with $N = 200$.

To further analyze the performance of the conditioning configurations, FID on different target landscapes is reported on Table 2. For this purpose, the conditional maps were simplified into 3 possible classes: *urban* (buildings, roads), *vegetation* (parks) and *water* (sea, rivers, lakes). For each class, the 200 images with the majority pixels in a given category were used to compute the FID. As in Table 1, the best performance corresponds to the configuration using the encoded conditional map injected into the cross-attention layers. It is important to note that this configuration achieves the best FID in all three categories. Therefore, it is not a case of a single predominant landscape being very well represented. The FID scores for urban areas and vegetation are lower in comparison to those for water, which turns out to be the least common class in the validation set.

Figure 7 shows some samples from the *Maps* dataset and their segmentation masks with the urban, vegetation and water categories. It also shows the histogram of these categories across the training and validation splits.

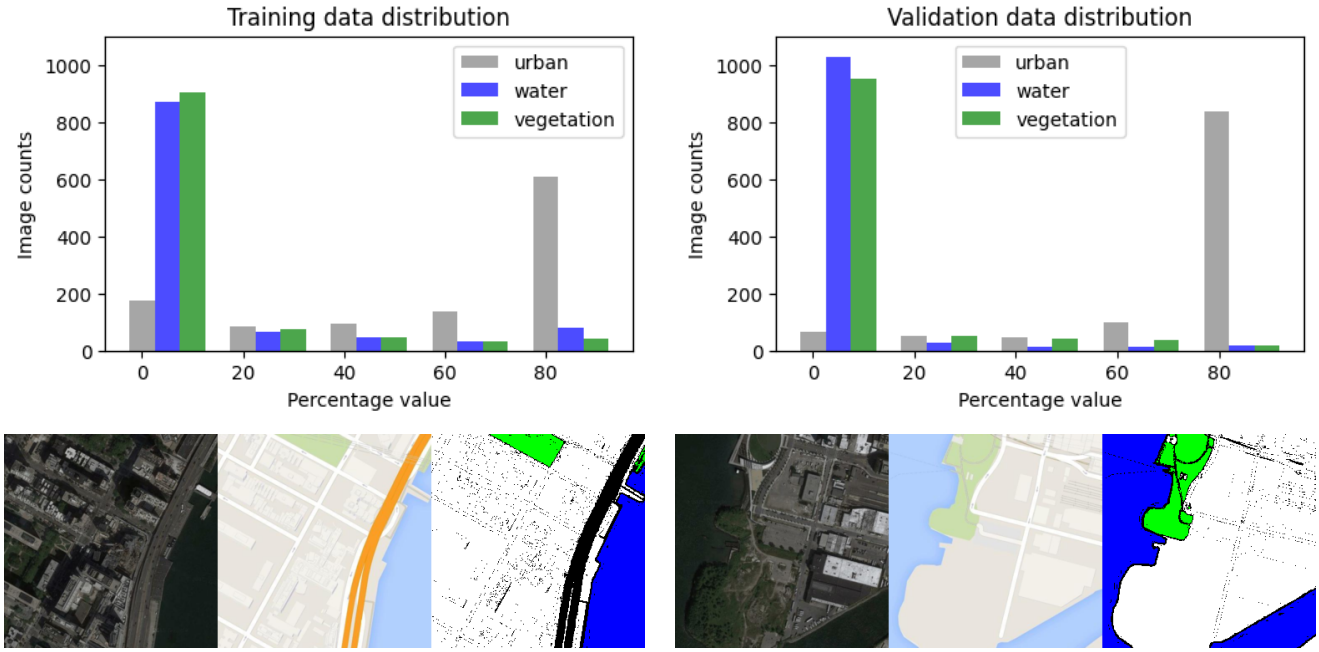

Figure 7: (Top row) Landscape distribution in the *Maps* dataset with images having a particular percentage of their pixels from *urban*, *vegetation* or *water* categories. Most images contain large urban areas, above 80%, and water or vegetation are substantially less present, below 20%. (Bottom row) Two examples of aerial images and their respective original maps (middle) and simplified maps (right), in which black pixels stand for none of the three main categories —mostly highways or other specific structures.

4.2 Qualitative Assessment

Figure 8 shows some results of the conditional synthesis along with the input conditional map and the corresponding real image. Note that the conditional maps are taken from the validation set, so their corresponding real images were not seen during training. Visual inspection of synthetic images helps to better understand the strengths and weaknesses of the diffusion model and the conditioning approaches listed in Table 1.

In general, the model captures the structure of the input conditional map correctly. Vegetation segments correspond to vegetation zones in the synthetic aerial images. Likewise for urban and water areas.

However, the quality of fine-grained textures is very variable. Roofs on large buildings, as shown in Figure 8 (b), lack homogeneity, probably because they are rare in the dataset. Instead, small buildings look more realistic, as shown in Figure 8 (a) and (d). It can also be observed that the model using cross-attention (third column) generates sharper results. This is especially noticeable along the contours of the input map, as shown in the roads in Figure 8 (c). Overall, the cross-attention layers successfully contribute to better capture contextual information and contours.

Another interesting fact is that the model seems to be prone to generate unnecessary artifacts. For example, navigation lines in large water areas lead to artifacts that replicate their shape, see Figure 8 (e). This is not surprising because there are few water images of this type in the dataset, and the model may mistake navigation lines for roads.

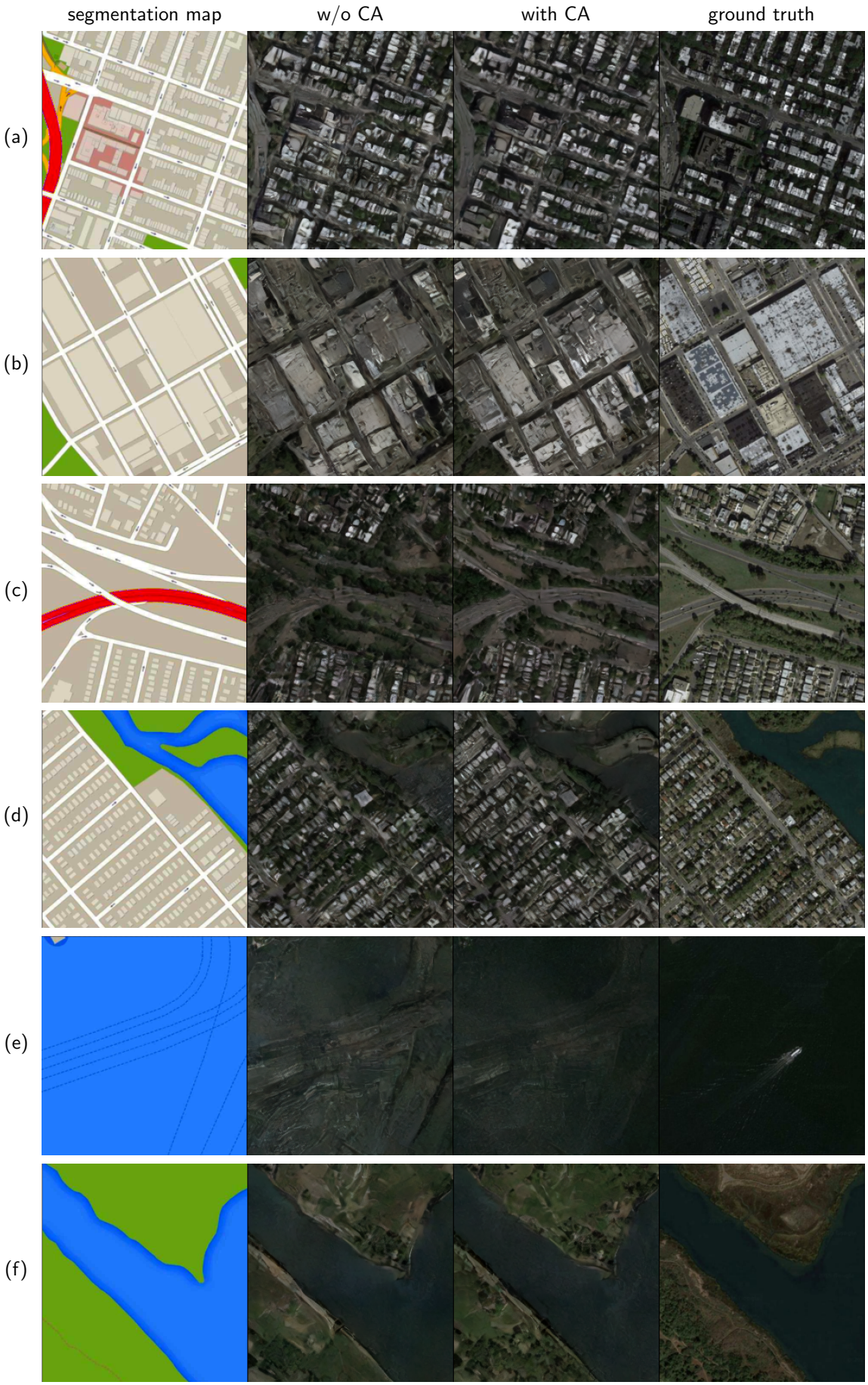

Figure 8: Comparison of synthetic examples with and without cross-attention, both using the encoded conditional map.

5 Conclusion

This paper reviews the fundamentals of diffusion models and applies a latent diffusion pipeline to generate synthetic samples of aerial images with different conditioning configurations. The results show that the model captures the overall structure of the map correctly and generates realistic textures. However, it lacks quality in the details, especially in rare structures. Cross-attention layers are crucial to fully exploit the conditional information and improve the appearance of boundaries. Encoding the conditional map in latent space using a pre-trained encoder, instead of directly downsampling, also helps improve the quality of synthetic samples.

Future research will experiment with lower compression ratios of the latent space to better preserve spatial resolution and encourage fine-grained details in the synthetic samples.

Acknowledgements

This work was financially supported by the Catalan Government through the funding grant ACCIÓ-Eurecat (Project TRAÇA: "IAGenerativa" 2023-2025).

References

[1] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, *Diffusion Models in Vision: A Survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2023), https://doi.org/10.1109/TPAMI.2023.3261988.

[2] P. Dhariwal and A. Nichol, *Diffusion Models Beat Gans on Image Synthesis*, Advances in Neural Information Processing Systems, 34 (2021), pp. 8780–8794.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative Adversarial Nets*, Advances in Neural Information Processing Systems, 27 (2014).

[4] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*, Advances in Neural Information Processing Systems, 30 (2017).

[5] J. Ho, A. Jain, and P. Abbeel, *Denoising Diffusion Probabilistic Models*, Advances in Neural Information Processing Systems, 33 (2020), pp. 6840–6851.

[6] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, *Cascaded Diffusion Models for High Fidelity Image Generation*, Journal of Machine Learning Research, 23 (2022), pp. 1–33. http://jmlr.org/papers/v23/21-0635.html.

[7] J. Ho and T. Salimans, *Classifier-Free Diffusion Guidance*, ArXiv Preprint ArXiv:2207.12598, (2022). https://doi.org/10.48550/arXiv.2207.12598.

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-To-Image Translation with Conditional Adversarial Networks*, in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134, https://doi.org/10.1109/CVPR.2017.632.

[9] T. Karras, S. Laine, and T. Aila, *A Style-Based Generator Architecture for Generative Adversarial Networks*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4401–4410.

[10] D. Kingma, T. Salimans, B. Poole, and J. Ho, *Variational Diffusion Models*, Advances in Neural Information Processing Systems, 34 (2021), pp. 21696–21707.

[11] D. P. Kingma and M. Welling, *Auto-Encoding Variational Bayes*, ArXiv Preprint ArXiv:1312.6114, (2013). https://doi.org/10.48550/arXiv.1312.6114.

[12] M. Mirza and S. Osindero, *Conditional Generative Adversarial Nets*, ArXiv Preprint ArXiv:1411.1784, (2014). https://doi.org/10.48550/arXiv.1411.1784.

[13] A. Q. Nichol and P. Dhariwal, *Improved Denoising Diffusion Probabilistic Models*, in International Conference on Machine Learning, PMLR, 2021, pp. 8162–8171.

[14] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, *Hierarchical Text-Conditional Image Generation with CLIP Latents*, ArXiv Preprint ArXiv:2204.06125, 1 (2022), p. 3. https://doi.org/10.48550/arXiv.2204.06125.

[15] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, *Zero-Shot Text-To-Image Generation*, in International Conference on Machine Learning, PMLR, 2021, pp. 8821–8831.

[16] A. Razavi, A. Van den Oord, and O. Vinyals, *Generating Diverse High-Fidelity Images with VQ-VAE-2*, Advances in Neural Information Processing Systems, 32 (2019).

[17] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, *High-Resolution Image Synthesis with Latent Diffusion Models*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695, https://doi.org/10.1109/CVPR52688.2022.01042.

[18] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, *Deep Unsupervised Learning Using Nonequilibrium Thermodynamics*, in International Conference on Machine Learning, PMLR, 2015, pp. 2256–2265.

[19] J. Song, C. Meng, and S. Ermon, *Denoising Diffusion Implicit Models*, ArXiv Preprint ArXiv:2010.02502, (2020). https://doi.org/10.48550/arXiv.2010.02502.

[20] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, *Score-Based Generative Modeling Through Stochastic Differential Equations*, ArXiv Preprint ArXiv:2011.13456, (2020). https://doi.org/10.48550/arXiv.2011.13456.

[21] A. Vahdat, K. Kreis, and R. Gao, *Diffusion-Based Generative Modeling: Foundations and Applications*, in Tutorial IEEE Conference on Computer Vision and Pattern Recognition, 2022. https://www.youtube.com/watch?v=cS6JQpEY9cs. Accessed 28 February 2024.

[22] A. Vahdat, K. Kreis, and J. Kautz, *Score-Based Generative Modeling in Latent Space*, Advances in Neural Information Processing Systems, 34 (2021), pp. 11287–11302.

[23] A. Van Den Oord, O. Vinyals, and K. Kavukcuoglu, *Neural Discrete Representation Learning*, Advances in Neural Information Processing Systems, 30 (2017).