🌀 Spatial Reasoners for Continuous Variables in Any Domain

Bart Pogodzinski [1] **Christopher Wewer** [1] **Bernt Schiele** [1] **Jan Eric Lenssen** [1]

Abstract

We present 🌀 **Spatial Reasoners**, a software framework to perform spatial reasoning over continuous variables with generative denoising models. Denoising generative models have become the de-facto standard for image generation, due to their effectiveness in sampling from complex, high-dimensional distributions. Recently, they have started being explored in the context of reasoning over multiple continuous variables. Providing infrastructure for generative reasoning with such models requires a high effort, due to a wide range of different denoising formulations, samplers, and inference strategies. Our presented framework aims to facilitate research in this area, providing easy-to-use interfaces to control variable mapping from arbitrary data domains, generative model paradigms, and inference strategies. Spatial Reasoners are openly available online[2].

1. Introduction

Denoising generative models, such as DDPM [6], DDIM [16], Flow Matching [9], or Rectified Flow [10] have achieved unmatched levels of generation quality, and the research work in this field only continues to accelerate. Typically, these models learn to approximate a conditional data distribution $p(x \mid c)$ and learn to sample from it, where x represents a variable like images and c can be text or other conditioning signals.

In the recent year, the trend evolved further and interest grew in diffusion models that allow sampling over multiple variables, where each has its own noise level [3; 15; 20]. This scheme allows a wide range of sampling techniques, such as auto-regressive generation (with planned order), generation with infinite horizon, and overlapping generation, essentially turning denoising models into an engine for general probabilistic inference. Spatial Reasoning Models (SRMs) [20] formalized this framework into a general variant of such models that, given some partitioning of the data format into variables $\{x_1, ..., x_n\}$, e.g. image patches, video frames, skeleton joint positions, language tokens, etc., allows sequential conditional inference across these variables by decomposition using the chain-rule of probability:

$$p(x_1, .., x_n) = \prod_{i=1}^{n} p(x_{\pi(i)} | \{x_{\pi(j)}\}_{j=i+1}^{n}) \qquad (1)$$

As shown, optimizing the specific inference strategy, such as order and amount of sequentialization, can significantly reduce hallucinations in the generations [20].

Currently there are many different diffusion formulations, noise schedules, samplers, and inference variants. Thorough analysis and adaption to other data domains requires large-scale ablations and significant implementation effort. It is common that in the research field rapid development is prioritized over good separation of concerns, modularity and readability. This allows quickly testing ideas, but makes it harder to build on top of them. We believe an intuitive, modular and expandable framework and project template would therefore immensely help to further develop the paradigm of reasoning with denoising generative models.

In this work, we present 🌀 **Spatial Reasoners**, a software framework for performing spatial reasoning over sets of continuous random variables via multi-noise-level denoising generative models. We hope it will facilitate solving generative tasks in a wide range of new domains that go beyond image representations. Spatial Reasoners expose the following degrees of freedom in an easy-to-use interface:

- The choice of the input domain by providing a generic mapper interface that transforms arbitrary data domains into sets of variables to reason over.
- Explicit control over training and inference schedules, e.g., order and amount of sequentialization, individual noise levels, and the denoising formulation.
- A range of denoiser architectures, such as UNet [14], DiT [12], LightningDiT[22], U-ViT [7; 17], MAR [8], xAR [13], and AEs [4; 14; 22] for latent modeling, to be used depending on modality and task at hand.

[1]Max Planck Institute for Informatics, Saarland Informatics Campus, Germany. Correspondence to: Bart Pogodzinski <bpogodzi@mpi-inf.mpg.de>, Christopher Wewer <cwewer@mpi-inf.mpg.de>, Bernt Schiele <schiele@mpi-inf.mpg.de>, Jan Eric Lenssen <jlenssen@mpi-inf.mpg.de>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

[2]spatialreasoners.github.io

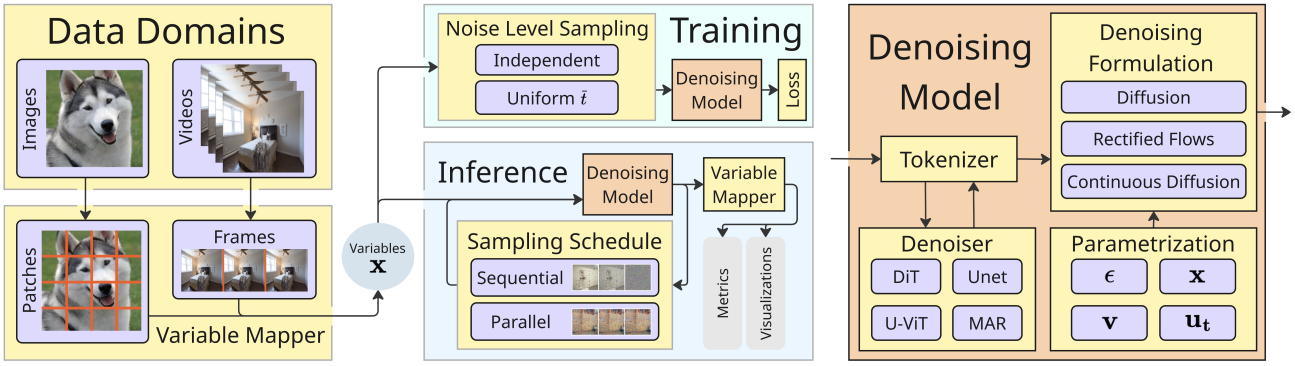


Figure 1: **Overview of Spatial Reasoners**. The variable mapper transforms different input modalities, such as images, videos or others, into variables, which are then used for training and inference. Yellow building blocks are exchangeable and can be extended with custom functionality. On the right, we zoom into individual aspects of the denoising model.

2. Related Work

Applying denoising generative models with individual noise levels per variable has been a recent trend in several works. Rolling Diffusion [15] and Diffusion Forcing [3] have proposed video diffusion models that are trained to denoise different noise levels per frame, allowing to generate long sequences in a rolling window. History-guided diffusion [17] recently expanded on the video generation with the Diffusion Forcing scheme, by exploring the impact of classifier-free guidance with respect to clean frames. MAR [8] and xAR [13] present auto-regressive diffusion on images that denoise a single or k variables at the same time. Spatial Reasoning Models (SRMs) [20] present a general framework for such strategies on sets of variables. Another line of work involves denoising models applied across multiple modalities. UniDiffuser [2] performs diffusion jointly on text and images, by independently sampling the noise level for each modality during training. The model can be used to conditionally generate images from given text, vise versa, or joint generation. Spatial Reasoners unifies all of these paradigms into a single framework, allowing to mix and explore individual choices, such as inference schedules, architectures, and denoising models.

Other packages. Related software frameworks for denoising generative models have been very successful in recent years. Examples include HuggingFace Diffusers [18], a framework for diffusion models for image generation, which supports most of the research in this domain. Another widely used toolkit is the `denoising-diffusion-pytorch` repository [19]. None of the existing frameworks explicitly supports generation and reasoning across multiple variables.

3. Spatial Reasoning with Spatial Reasoners

In this section, we first introduce the core reasoning framework of Spatial Reasoners in Sec. 3.1, before giving an overview of different toolkit building blocks in Sec. 3.2. Then, we detail exposed degrees of freedom in Sec. 3.3 and explain the easy-to-implement interfaces that allow to fast adaption of the framework to new domains in Sec. 3.4.

3.1. Core Framework

The Spatial Reasoners toolkit is built upon the framework of Spatial Reasoning Models (SRMs) [20]. Given a set of variables $\{x_1, ..., x_n\}$, SRMs define *reasoning* as an iterative denoising process over the set of variables:

$$\hat{x}_1^{t_1}, ..., \hat{x}_n^{t_n} \sim q(x_1^{t_1}, ..., x_n^{t_n} \mid x_1^{t'_1}, ..., x_n^{t'_n}), \tag{2}$$

where t_i encode individual noise levels for each variable. Depending on the task at hand, variables can represent different types of data, e.g. image patches, whole images of a sequence, or other entities, and can contain positional encodings to locate them in an arbitrary space. For a denoising process of d steps, a matrix $\mathbf{T} \in \mathbb{R}^{n \times d}$, containing the noise levels t_i for all n variables and all d steps, fully specifies the reasoning process during inference. One step of the process can be carried out by predicting scores or flows, according to typical generative denoising formulations, such as DDPM [6], DDIM [16], or Rectified Flows [10].

3.2. Overview

Fig. 2 shows an overview of the full toolkit, where exchangeable and customizable building blocks are shown in yellow. The `VariableMapper` can transform data from different domains into the `Variable` format, which is then unified for the rest of the pipeline. Training and inference routines work on top of this format. During training, noise is added to ground truth variables according to the chosen noise level sampling algorithm, before fed into the denoising model, which is trained to denoise them. During inference, variables can be (partially) initialized with random noise and denoised according to a defined schedule.

3.3. Individual Degrees of Freedom

The Spatial Reasoners framework exposes a wide range of degrees of freedom to facilitate exploration and further research. We discuss them individually in the following.

Denoising Paradigms. We support original diffusion with discrete steps, implementing DDPM [6] and DDIM [16], diffusion with continuous steps, mixing diffusion and flow matching, cosine (variance preserving) flows [1; 11] and Rectified Flows [10].

Parameterizations. Spatial Reasoners supports a wide range of parameterizations, including ϵ prediction (noise), x_0 prediction (clean data), u_t prediction (direction in flow models), v prediction (direction vector in diffusion).

Training $\mathbf{t}$-sampling. We support different $\mathbf{t}$-samplers for training with sets of variables, such as the independent uniform sampling strategy [3], or Uniform-$\bar{t}$ sampling [20]. It is straightforward to implement additional sampling strategies. All $\mathbf{t}$-samplers for variable sets can be additionally combined with tailored scalar noise level samplers like the logit-normal distribution for Rectified Flows [5].

Architectures. The framework makes it easy to exchange the neural architecture, which predicts the noise. Currently, we support DiT [12], LightningDiT [22], UNet [14], MAR [8], xAR [13], and U-ViT-Pose [17]. We support loading the checkpoints from the original works.

Inference Schedules. We support all the inference schedules from existing works, such as sequentialized sampling, with variable blend between autoregressive and parallel generation (overlap), and with predicted, manually-defined or random order [20], as well as next k variable prediction [13].

Dependency Graph Injection. We provide functionality to inject domain-specific knowledge by providing dependency structure between variables in form of graphs. Those can be exploited in choosing the order of inference.

Uncertainty Prediction. All models can easily parameterized to also predict uncertainty, allowing for uncertainty-based ordering of generation [20].

Learned Variance. By implementing a unified interface for different denoising paradigms, we support improvements for diffusion modes like the learning of the variance in the generative process [11] also in combination with flow formulations like Rectified Flows [10].

Latent Denoising. Spatial Reasoners supports reasoning and generation in latent spaces, by including typical image autoencoders like SD-VAE [14], VAVAE [22], and DC-AE [4].

Modular Losses. We support different losses including standard MSE for noise prediction, VLB [11] for learning the variance of the reverse process, and cosine similarity for additional supervision of the velocity direction with flow models [21]. It is easy to add additional losses, e.g., other losses for uncertainty predictions besides NLL [20].

3.4. Adapting to New Domains

A main goal of Spatial Reasoners is to make it easy to adopt SRMs to new data domains. We achieve this by providing two interface classes that need to be customized to support a new modality: the `VariableMapper` and the `Tokenizer`.

In the `VariableMapper`, the user needs to define how a data example should be partitioned into variables, atomic elements that maintain the same noise level. For latent space diffusion, an autoencoder can be defined here that pre-processes the data before partitioning.

The `Tokenizer` allows to transform the variables-format data to the arbitrary input format of the trained denoiser. Architectures such as DiT are domain agnostic and just require the definition of the token positions for positional encodings, e.g., sinusoidal, RoPE, etc., depending on the generation task.

In addition to the two mentioned interfaces, the user can implement custom visualization and metrics in `Evaluation` classes. Thanks to the underlying variable format, the rest of the framework remains domain agnostic.

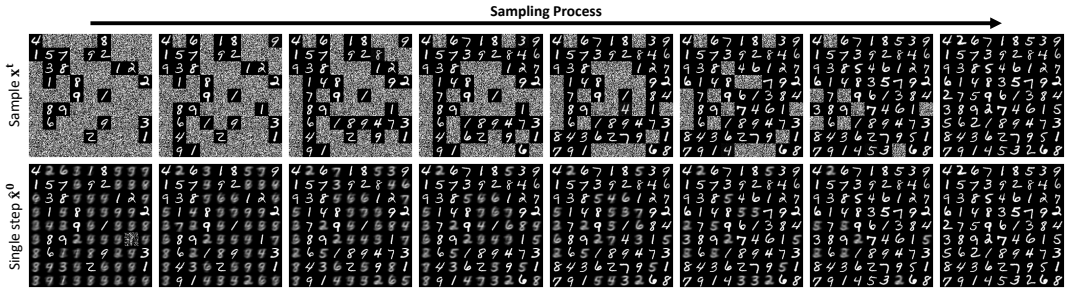
4. Application Examples

In this section, we provide a few application examples to showcase the generality of Spatial Reasoners. We show examples for reasoning over image-based MNIST Sudoku [20], auto-regressive image generation [13], and auto-regressive, overlapping video generation [3; 17].

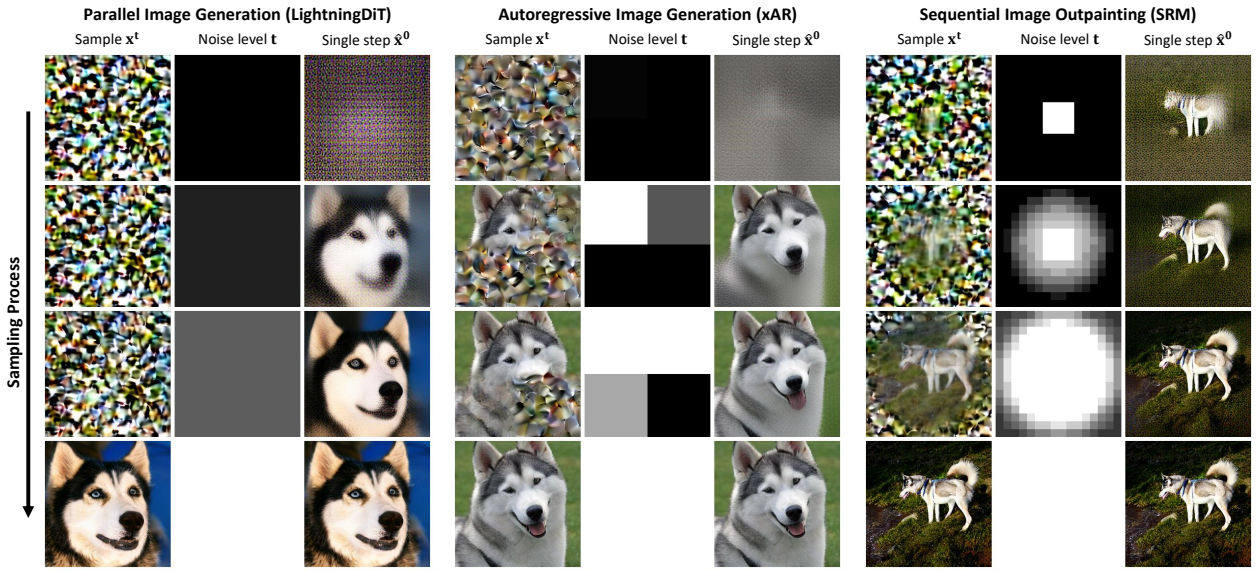
Visual Reasoning Tasks. SRMs [20] introduced multiple visual reasoning benchmarks, where variables are image patches. In Fig. 2a we show sequential solving of visual Sudoku, consisting of MNIST numbers. It is fully auto-regressive and the order is predicted based on uncertainty. The more numbers on the board, the less ambiguous the remaining ones, which is visible in the $\hat{\mathbf{x}}^{\mathbf{0}}$ prediction.

Image Generation and Editing. Fig. 2b shows multiple examples for image generation (left and middle), and outpainting (right). Spatial Reasoners supports a variety of sampling schedules, such as standard parallel generation with LightningDiT [22], next-k variable generation of xAR [13], or manually defined schedules from SRMs [20]. The SRM example (right) shows a locality-based order, painting outwards from existing variables. All shown models are latent diffusion models and generate in the latent space of a VAE.

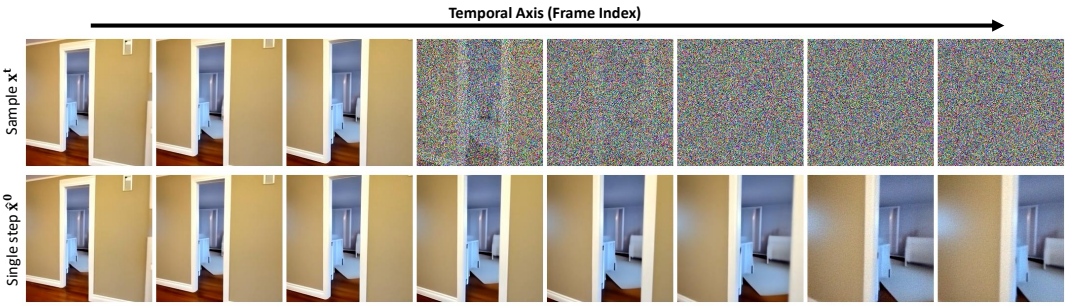
Soft-sequential Video Generation. We allow to perform soft-sequential video generation with a U-ViT model [17], where each video frame is represented as one variable.



(a) Visual Reasoning: Solving Visual Sudoku



(b) Latent Image Generation and Editing



(c) Soft-Sequential Video Generation in Pixel Space

Figure 2: **Application Examples**. Spatial Reasoners supports **(a)** visual reasoning benchmarks like visual Sudoku [20], **(b)** various image generation and editing strategies including parallel denoising with LightningDiT [22], autoregressive next-X prediction with xAR [13], and soft, certainty-based sequentialization with Spatial Reasoning Models [20], and **(c)** auto-regressive, long horizon video generation, including history-guidance [3; 17].

Fig. 2c illustrates a moment during inference. While the first three frames are already fully denoised, the others are partially or fully noisy. However, the information from the already denoised frames and the camera pose conditioning is sufficient conditioning for the model to provide a good single-step $\hat{\mathbf{x}}^{\mathbf{0}}$ prediction for fully noisy frames.

5. Conclusion

Denoising models have proven to be powerful tools for generative tasks, and recent developments have extended their utility to reasoning over multiple variables with distinct noise levels. By offering a clean, modular interface for defining variable mappers, training and inference schedules, together with access to a vast lineup of denoising architec-

tures, Spatial Reasoners aims to broaden the applicability of multi-noise-level generative models beyond traditional domains. We hope that Spatial Reasoners will become a useful tool for researchers who want to take a deeper dive into probabilistic reasoning with structured generative models.

Acknowledgements

This project was partially funded by the Saarland/Intel Joint Program on the Future of Graphics and Media. We thank Philipp Schröppel for insightful discussions regarding the design choices of the software architecture.

References

[1] Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions, 2023. URL https://arxiv.org/abs/2303.08797.

[2] Bao, F., Nie, S., Xue, K., Li, C., Pu, S., Wang, Y., Yue, G., Cao, Y., Su, H., and Zhu, J. One transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning (ICML)*, 2023.

[3] Chen, B., Monso, D. M., Du, Y., Simchowitz, M., Tedrake, R., and Sitzmann, V. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, 2024.

[4] Chen, J., Cai, H., Chen, J., Xie, E., Yang, S., Tang, H., Li, M., Lu, Y., and Han, S. Deep compression autoencoder for efficient high-resolution diffusion models. In *ICLR*, 2025.

[5] Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

[6] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

[7] Hoogeboom, E., Heek, J., and Salimans, T. simple diffusion: End-to-end diffusion for high resolution images. 2023.

[8] Li, T., Tian, Y., Li, H., Deng, M., and He, K. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems*, 2024.

[9] Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

[10] Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

[11] Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 2021.

[12] Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *International Conference on Computer Vision (ICCV)*, 2023.

[13] Ren, S., Yu, Q., He, J., Shen, X., Yuille, A., and Chen, L.-C. Beyond next-token: Next-x prediction for autoregressive visual generation, 2025.

[14] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[15] Ruhe, D., Heek, J., Salimans, T., and Hoogeboom, E. Rolling diffusion models. In *International Conference on Machine Learning*, 2024.

[16] Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[17] Song, K., Chen, B., Simchowitz, M., Du, Y., Tedrake, R., and Sitzmann, V. History-guided video diffusion, 2025.

[18] von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Nair, D., Paul, S., Berman, W., Xu, Y., Liu, S., and Wolf, T. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[19] Wang, P. denoising-diffusion-pytorch. https://github.com/lucidrains/denoising-diffusion-pytorch, 2020.

[20] Wewer, C., Pogodzinski, B., Schiele, B., and Lenssen, J. E. Spatial reasoning with denoising models. In *International Conference on Machine Learning (ICML)*, 2025.

[21] Yao, J., Wang, C., Liu, W., and Wang, X. Fasterdit: Towards faster diffusion transformers training without architecture modification. 2024.

[22] Yao, J., Yang, B., and Wang, X. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *CVPR*, 2025.