Disentangled Compositional Diffusion for Controllable Scientific Data Generation

Nandan Madhuj
Cornell University
nm736@cornell.edu

Meet Hemant Parikh
Cornell University
mhp66@cornell.edu

Anirban Samaddar
Argonne National Laboratory
asamaddar@anl.gov

Yixuan Sun
Argonne National Laboratory
yixuan.sun@anl.gov

Sandeep Madireddy
Argonne National Laboratory
smadireddy@anl.gov

Jian-Xun Wang
Cornell University
jw2837@cornell.edu

Abstract

Diffusion and flow-based generative models are capable of generating high quality samples, but their internal representation is difficult to manipulate for controllable and interpretable generation, particularly for scientific applications. We employ decomposed diffusion models, which encode data into a latent representation, enabling interpretability by disentangling individual components and controllability by selecting components with desired features to generate new samples. We demonstrate applications on different datasets, including interpretability and controlled generation on temperature fields and vortex-induced flow fields, synthesizing human aortic geometries from very few samples, and 2D turbulent flow fields.

1 Introduction

Diffusion and flow-based generative models have shown remarkable performance for high-dimensional perceptual data (Ho et al., 2020; Song et al., 2020). Similar high fidelity generation has also been shown for many scientific tasks (Price et al., 2025; Schuette et al., 2025). Therefore, diffusion and flow-based generative models hold the promise of accelerating scientific discovery through accurate surrogate models and data augmentation (Chen et al., 2024). However, standard unconditional generation merely traces a path from a Gaussian source to the broad, implicitly learned marginal distribution. In the context of scientific discovery, this is insufficient; we often require steerable generation that strictly adheres to physical requirements or preferences, thus effectively sampling from a tilted distribution conditioned on specific geometric components or structural features.

Current methods for achieving such structural alignment face significant trade-offs. Explicit conditioning (Dasgupta et al., 2025; Yang et al., 2025), where separate diffusion models are trained for different conditions, is the most straightforward way to achieve controlled generation. But such an approach requires condition-paired datasets and training multiple models. Another way of achieving control is through inference-time guidance (Chung et al., 2025; Guo et al., 2024; Bansal et al., 2023). Guidance utilizes the gradient of conditional functions such as a regressor to modify the score function used for sampling to achieve control, which is commonly used in inverse design. In a similar spirit, compositional approaches (Du et al., 2021; Liu et al., 2022) combine the scores of different sub-models to perform generation in a product-of-experts fashion such that the generated samples satisfy all sub-models simultaneously. While flexible, both guidance and composition typically require additional models (regressors or sub-models) on top of the base diffusion model, creating computational bottlenecks that limit the application for generally scarce and expensive-to-obtain scientific data.

To leverage the flexibility of compositional generation without the overhead of auxiliary models, we propose a framework for intrinsic structural alignment based on decomposed diffusion (Su et al., 2024) for scientific data generation. Unlike standard guidance which aligns models post-hoc, our method introduces a specially designed loss term during training to enforce the disentanglement of latent factors into interpretable components. This facilitates zero-shot steerability at inference

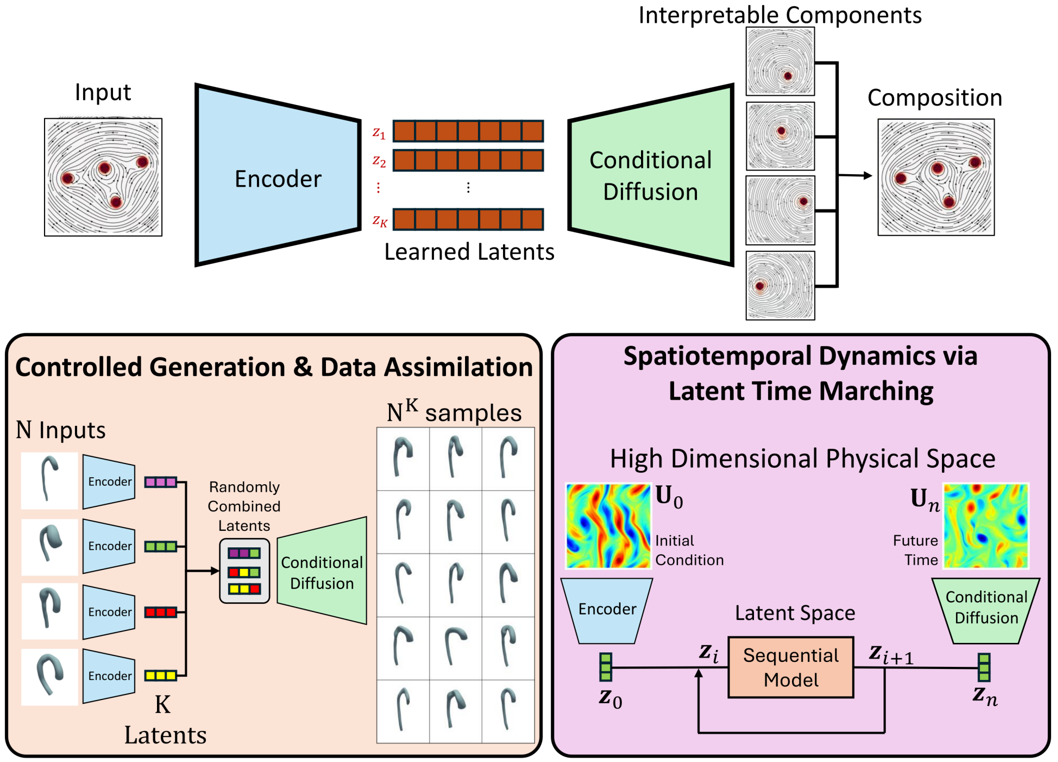

Figure 1: Decomposed Diffusion consists of an encoder with a generative decoder whose latents can be mixed to generate physically coherent scientific fields. (Top) Decomposition of vortex induced flow fields: The model isolates individual latents capturing effect of specific vortices on the flow field. (Bottom Left) Decomposed Diffusion on Human Aorta: application on anatomical data, where individual latents may be recombined to synthesize novel, realistic aorta samples from a very small training dataset. (Bottom Right) Latent Spatiotemporal Dynamics: High dimensional 2D turbulence can be represented in lower dimensional latent space for reduced-order surrogate modeling.

time: we can control generation by selectively composing desired latent components responsible for specific physical factors. This strategy is particularly advantageous for small datasets, enabling meaningful generation beyond observed training samples. We demonstrate this alignment-by-decomposition approach on two-dimensional temperature fields with multiple heat sources, flow fields with vortices, turbulent flow, and coronary artery contours.

Our contributions include, adapting decomposed diffusion models to scientific data generation where compositional decomposition aligns with physically interpretable features. We also introduce a latent orthogonality regularizer which promotes disentanglement between components in more complex systems. Additionally, we demonstrate realistic generation by latent recombination even in extreme low-data regimes where generative models usually succumb to memorization. Finally, we discuss the possibility of leveraging the latent representation for reduced-order surrogate modeling.

2 Related Works

Generative models, such as diffusion models (Ho et al., 2020; Song et al., 2020) and flow matching (Lipman et al., 2022), have enabled highly-fidelity sample generation for high-dimensional natural images. These models learn a score function (or a vector field), parameterized by a neural network, to transport random noise to the training data manifold. During inference, the learned score network is used in various ordinary (Lipman et al., 2022; Tong et al., 2023; Karras et al., 2022), and stochastic (Song et al., 2020; Karras et al., 2022) differential equation solvers to generate random samples from the data manifold. Standard flow-based generative models lack control in their generation process. Several recent works have focused on improving control for the flow-based generative models.

Guided generation Several works (Dhariwal & Nichol (2021); Zheng et al. (2023)) in computer vision have sought to improve control by conditioning the image generation process on auxiliary information, such as text. During training, these models condition the score neural network on the representation learned from the conditioning variable. This enables the model to generate diverse samples based on unseen prompts during inference. However, training text-conditioned generative models requires labeling the training images with textual descriptions, which can be expensive. Built on this idea Chung et al. (2025); Guo et al. (2024); Bansal et al. (2023) propose training-free guidance approaches that, during inference, aim at steering a pretrained unconditional diffusion model using user-specified objective functions. Recent works in flow matching (Guo & Schwing (2025); Samaddar et al. (2025)) have extended the idea of conditional generative modeling to condition the generation process on latent variables learned directly from training images. These methods have shown improved training efficiency and control through generating samples conditioned on the features extracted from the training data. However, these approaches do not disentangle features in the latent space and have not been extensively evaluated on scientific datasets.

Compositional generative models A recent body of works (Du & Mordatch, 2019; Du et al., 2020; Liu et al., 2021; 2022; Du et al., 2023; Liu et al., 2023; Wang et al., 2025; Bradley et al., 2025; Thornton et al., 2025) has proposed alternate ways to reuse pretrained energy based and conditional diffusion models, focusing on the task of compositional generation. Using pretrained text-to-image generative models, these methods compose different concepts (using textual descriptions) in a single generated image. However, as highlighted in (Bradley et al., 2025) (Fig. 7), compositional generation works well for disentangled concepts. To alleviate this challenge, (Su et al., 2024) propose a method for decomposing an image into concepts and composing them for reconstruction. Given an input image, this approach learns a conditional diffusion model, conditioned on a set of latent feature vectors, using an encoder neural network to encode different features. This approach has been able to decompose factors in natural images (like background, texture etc.) and enabled controlled generation by recombining different factors from the training data. However, this method does not explicitly enforce the disentanglement of the learned latent features in their loss function, which can lead to components learning overlapping features especially when latent dimensionality is high.

In this study, we extensively evaluate the decompose-and-compose diffusion models (Su et al., 2024) on several scientific datasets. We introduce a contrastive penalty term in the loss function to enforce orthogonality of the latent factors extracted by the encoder. This simple modification enables learning disentangled concepts encoded in the latents from the training data. Without any auxiliary information, these concepts often show consistency with the underlying physical process and enable diverse sample generation.

3 METHODOLOGY

Our work builds on the decomposed diffusion framework of Su et al. (2024). They used compositional decomposition for concept discovery in image datasets. The present work, however, deals with scientific datasets where disentanglement and orthogonality are attractive properties (Schmid, 2010; Berkooz et al., 1993). Due to this, our training loss specifically involves an orthogonality term between latent vectors. We found this helpful in disentangling different components for more complicated datasets. (See A.2.3)

The model is trained to encode high dimensional physical fields into K separate latent vectors each capturing some representative feature of the input. A conditional diffusion model (See A.1, A.2), conditioned on one or more latent vectors, generates samples in the physical field and acts as a decoder. The training of decomposed diffusion models is similar to standard diffusion models (See Alg. 1). The standard denoising step, however, now consists of contributions from all latent components to generate the denoised sample.

During inference, the diffusion model maybe conditioned on individual latent vectors alone, generating components which represent particular features of the full field. If instead the diffusion model

Algorithm 1: Training Algorithm

```
Input: data distribution $p_D$, Encoder $\text{Enc}_\phi$, denoising model $\epsilon_\theta$, number of components $K$
while not converged do
    ▶ Sample data distribution
    $x_i \sim p_D$
    ▶ Extract K latent variables from $x_i$
    $\boldsymbol{z}_1, \dots, \boldsymbol{z}_K \leftarrow \text{Enc}_\phi(x_i)$
    ▶ Compute denoising direction;
    $\epsilon \sim \mathcal{N}(0, 1), \quad t \sim \text{Unif}(\{1, \dots, T\})$;
    $x_i^t = \sqrt{1 - \beta_t}\, x_i + \sqrt{\beta_t}\, \epsilon$;
    $\epsilon_{\text{pred}} \leftarrow \frac{1}{K} \sum_k \epsilon_\theta(x_i^t, t, \boldsymbol{z}_k)$
    ▶ Compute total loss $\mathcal{L}_{\text{MSE}}$ with diffusion and latent orthogonality terms
    $\mathcal{L}_{MSE} = \|\epsilon_{\text{pred}} - \epsilon\|^2 + \lambda_{orthog} \sum_{ij} (\boldsymbol{z}_i \cdot \boldsymbol{z}_j - \delta_{ij})^2$
    ▶ Optimize objective $\mathcal{L}_{\text{MSE}}$ w.r.t. $\zeta = \{\phi, \theta\}$
    $\Delta\zeta \leftarrow \nabla_\zeta \mathcal{L}_{MSE}$
end while
```

Algorithm 2: Inference Algorithm (full field)

```
Input: Diffusion steps T, denoising model $\epsilon_\theta$, latent vectors $\boldsymbol{z}_1, \dots, \boldsymbol{z}_K$, step size $\gamma$
$x^T \sim \mathcal{N}(0, 1)$
for $t = T, T - 1, \dots, 1$ do
    ▶ Sample Gaussian noise
    $\xi \sim \mathcal{N}(0, 1)$
    ▶ Compute denoising direction With all latents
    $\epsilon_{\text{pred}} \leftarrow \frac{1}{K} \sum_k \epsilon_\theta(x^t, t, \boldsymbol{z}_k)$ ;
    ▶ Noisy Gradient Descent
    $x^{t-1} \leftarrow \frac{1}{\sqrt{1-\beta_t}} \left( x^t - \gamma \epsilon_{\text{pred}} + \sqrt{\beta_t}\, \xi \right)$ ;
end for
```

is conditioned on all latent vectors, it reconstructs the full field with properties represented by all the latent vectors (See Alg. 2).

4 Results and Discussion

In this section, we consider a various datasets and explore different uses of this framework for scientific datasets. Interesting applications such as controlled generation, generation even with severely limited data, and the use of latent representation for downstream scientific tasks such as reduced-order surrogate modeling are discussed.

4.1 Decomposing Scalar Temperature Fields

We first apply diffusion-based decomposition on the solutions to the steady heat equation with four randomly placed, constant-temperature heat sources. A central advantage of decomposed diffusion for representing scientific datasets is its ability to identify and isolate individual components that contribute to the full physical field. Such ability can be useful for scientific datasets (and scientific community in general) since relatively simple physical laws/interactions give rise to complicated physical phenomena.

Figure 2 shows results from a decomposed diffusion model trained on this dataset. The training data consists of 5000 temperature fields obtained by Gauss-Seidel method. A choice of $K = 4$ is natural in this case. The hyperparameter K is dataset dependent and requires careful tuning according to the expected number of components. In this case, it is straightforward to see that each of the latent vectors correctly represent a particular heat source.

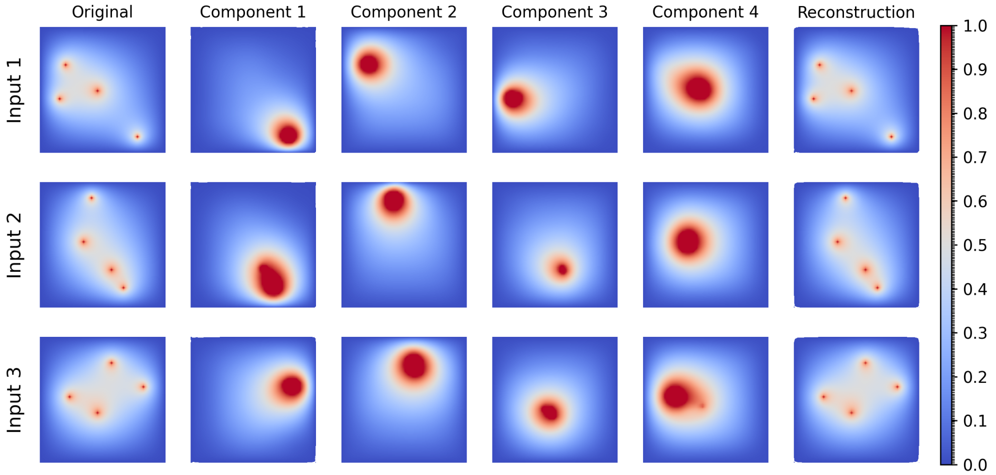

Figure 2: Decomposed Diffusion on the solutions to the steady heat equation with four heat sources. Results shown for three different inputs. The model learns a latent encoding with 4 latent vectors. Conditional diffusion on each latent vector generates the corresponding component representing individual heat source location. Conditioning diffusion on all 4 latents reconstructs the input.

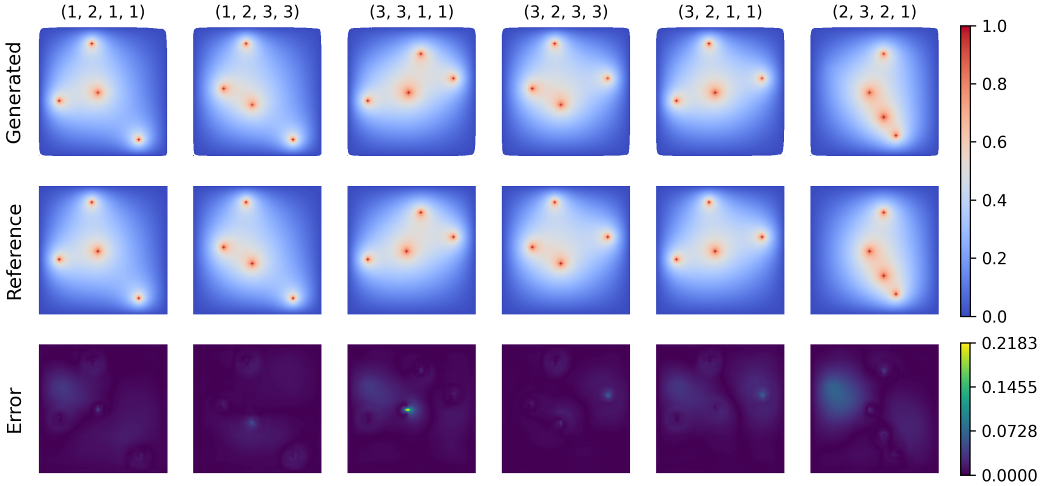

Figure 3: Controlled generation via latent permutation. The first row shows solutions generated by combining different heat sources from latents of Fig. 2 inputs. For example, (2,3,2,1) means Latents 1 and 3 were chosen from input 2, latent 2 from input 3 and latent 4 from input 1. The second row shows the reference numerical solution obtained by iterative Gauss–Seidel Relaxation for that particular source configuration. Notably, the decomposed diffusion predictions match closely with the reference solution and are obtained without numerical solution.

The four components, generated by conditioning the diffusion model on one latent at a time, indicate what the corresponding latents represent. If for example, we wanted to generate a new sample with a heat source place close to the bottom right of the domain, we may choose latent 1 from Input 1 of Figure 2. Thus, we can choose particular latent vectors from inputs of Figure 2 to control the location of heat sources in the new samples. Some of the generated samples are shown in Figure 3 (and Figure 12). The generated samples with the desired location of heat sources show little difference from the reference numerical solution. This demonstrates controlled generation through interpretable latent representation.

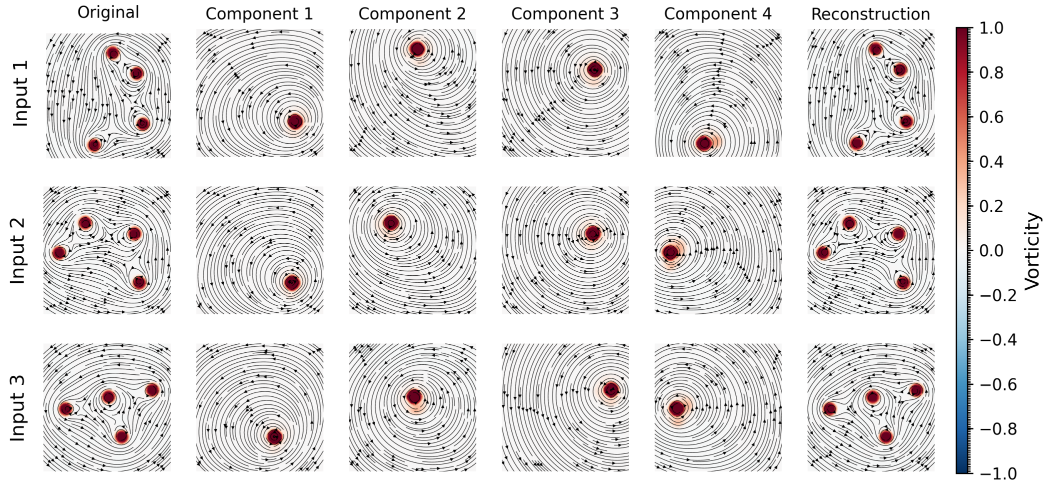

Figure 4: Decomposition of flow fields generated by four identical rankine vortices. Each row corresponds to a specific input shown in first column while last column shows the reconstruction. The columns 2-5 show how different vortices can be captured by individual latents.

4.2 Decomposing Fluid Flows induced by Vortices

Solution to the heat equation are scalar temperature fields. We now consider vector flow fields induced by rotating Rankine vortices. In this case, 5000 samples were used for training a model with latent dimensionality $d = 48$ and number of latents $K = 4$.

Figure 4 shows the decomposed diffusion on flow fields induced by these vortices. An interpretable representation is learned in which the four latent vectors capture global effects of individual vortices as rotational velocity fields. As in the temperature field case, the different latents vectors may be combined to generate new flow fields not present in the training data. (See A.2.4, Figure 13). We have also done experiments on flow fields with dissimilar vortices (Figure 10). In this case, we found that enforcing latent orthogonality was useful in disentangling the individual vortices. Learning physically disentangled latents is important to ensure efficient data representation and minimum redundancy. The additional term in the loss function promotes latent vectors to be orthogonal to one another and reduces the possibility of redundant information shared between multiple components. Orthogonality promotes more independence between the learned components (See A.2.3)

4.3 Synthesizing realistic Aortic Geometries via Latent Permutation

The previous two cases sought to demonstrate interpretability with abundant training data. However, scientific data is usually not abundant. In general, scientific data is either non-trivial to collect (e.g., patient specific human aorta dataset considered presently) or computationally expensive to simulate (such as turbulence data considered in the following section). In this section, we consider the problem of generating realistic and anatomically valid aortic geometries from a very small dataset. In this case, only 21 examples of arteries are available (See A.3). With severe data limitation, data hungry generative models fail to generalize. Onset of memorization has been shown to occur at training times proportional to dataset size (Bonnaire et al., 2025; Favero et al., 2025). For very small dataset, this happens very soon and methods such as early stopping produce out-of-distribution samples which may violate real world constraints required for scientific data.

We observe that decomposed diffusion can produce realistic samples even with extremely limited data. Instead of memorizing the training data, the model memorizes the latent representation. However, the memorized latent representation allows flexibility and the latents may be permuted to generate new, high quality samples which are realistic (Figure 5). These geometries can be useful for downstream biomedical tasks such as AI-assisted cardiovascular diagnosis, data-driven surrogate modeling, and patient-specific treatment planning.

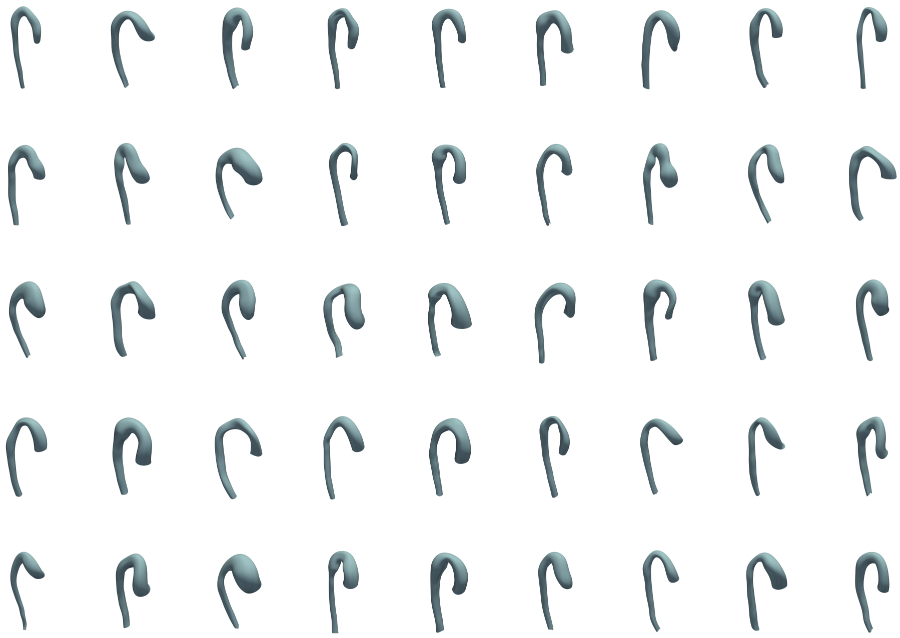

Figure 5: Generation via latent permutation: Only $N = 20$ training samples were used for training, the model with $K = 3$ latent vectors. New latent representations were formed by selecting latent vectors at random from the latent representation of training data. A total of N^K latent encoding are possible each generating new samples which are realistic and anatomically valid.

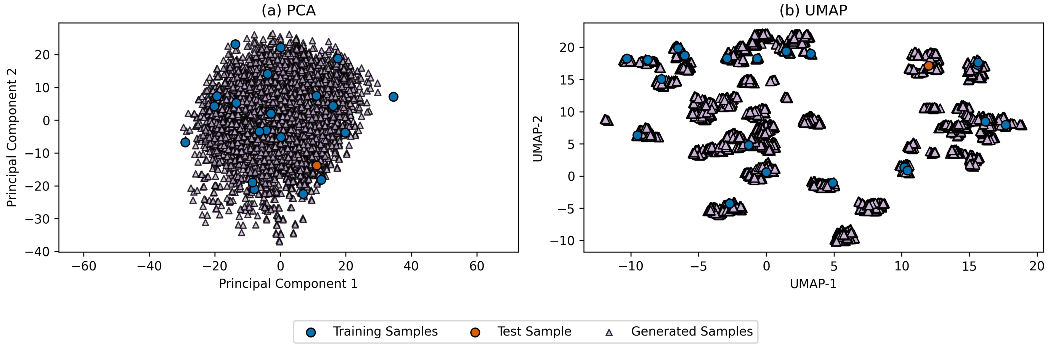

Figure 6: Synthetic Human Aorta geometries: (Left) Generated samples projected onto the Principal Component basis of Training Samples showing good generalization despite memorizing the individual latent vectors from only 20 training samples. (Right) Training Samples are projected onto a UMAP basis learned over generated samples.

Figure 6 shows projection of the generated samples onto the first two principal components of the training data showing good generalization. In the same figure, training samples are projected onto a UMAP of generated samples showing that the anatomically valid training data lies close to clusters of the generated samples on the projected manifold. A quantitative comparison against standard diffusion models is discussed in Appendix A.4

4.4 Two-dimensional Turbulence

In this section we utilize Kolmogorov flow dataset of Shu et al. (2023). We preprocess the vorticity to get velocity snapshots for training a decomposed diffusion model. Latent dimensionality $d = 128$

and number of components $K = 4$ was chosen. The results are shown in Figure 7. For more complicated cases, interpretation becomes difficult. However, component 1 seems to capture bigger structures. In contrast, component 4 shows smaller scale structures. Components 2 and 3 show intermediate length scales. Additionally, the components also show directionality. For example, component 1 has horizontal structures, component 2 and 4 have diagonally oriented structures while component 3 seems to have preference for structures near domain boundaries.

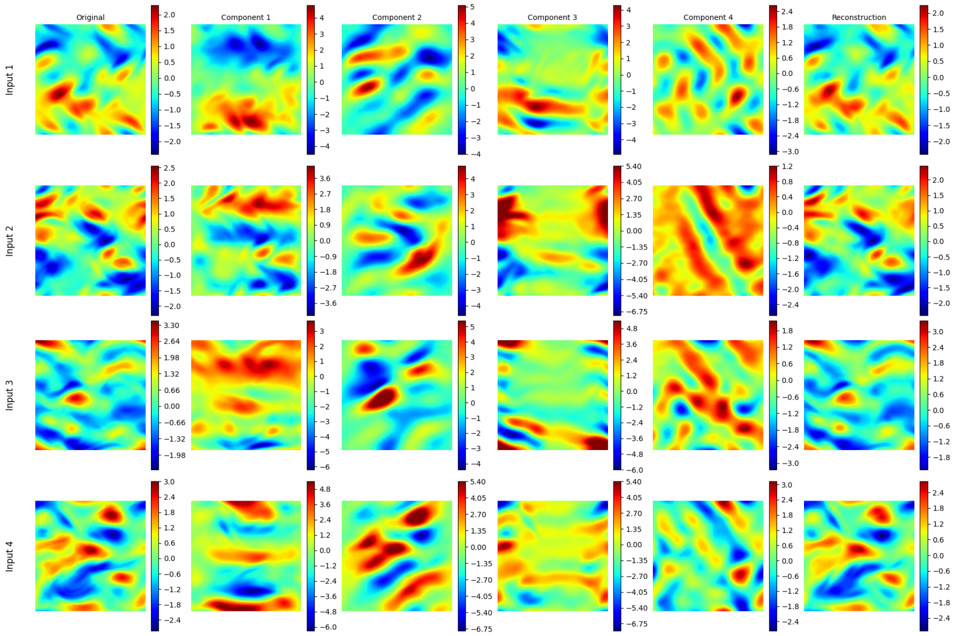

Figure 7: Decomposed Diffusion on two-dimensional turbulence dataset. x-velocity is plotted for four different inputs, their components and reconstruction. A clear interpretation is difficult. However, spatial and directional preference is evident in the components. Moreover, some components capture large scale structures while other capture smaller or intermediate scale structures.

A rigorous hyperparametric study over number of components K and latent dimensionality d needs to be done. Unlike the simpler cases, we do not know the underlying features in this case. Some separation of length scales is expected. However, a value of $K = 4$ is arbitrary and may have led to information redundancy. Nevertheless, new samples may be generated by combining the different latent components. Some of them are shown in Figure 8.

5 Conclusion and Future Work

This work demonstrates decomposed diffusion models in scientific datasets for controllable and interpretable generation. Initial results show promising use cases across different scientific domains where data scarcity and interpretability hold central importance. As such, these results are preliminary and various potential limitations need to be addressed. Latent interpretability is the key attraction of decomposed diffusion models. However, this cannot always be guaranteed and depends on model hyperparameters (such as number of latents K, latent dimensionality d) and the dimensionality and complexity of the data itself. We demonstrate interpretability for simpler cases of heat source generated temperature fields and vortex flows and only partial interpretability for two-dimensional turbulence. However, even in these cases, hyperparameter tuning and in some cases, an additional orthogonality term was necessary to achieve disentangled components. In addition to interpretability, another promising direction for decomposed diffusion for scientific tasks is its ability to generate new valid samples by latent recombination. We observe this behavior even under extreme data scarcity as demonstrated in the case of coronary artery synthesis case. This behavior seems to be rooted in latent component memorization rather than training data memorization as is the case with standard diffusion models. Improvements over standard diffusion models in low data

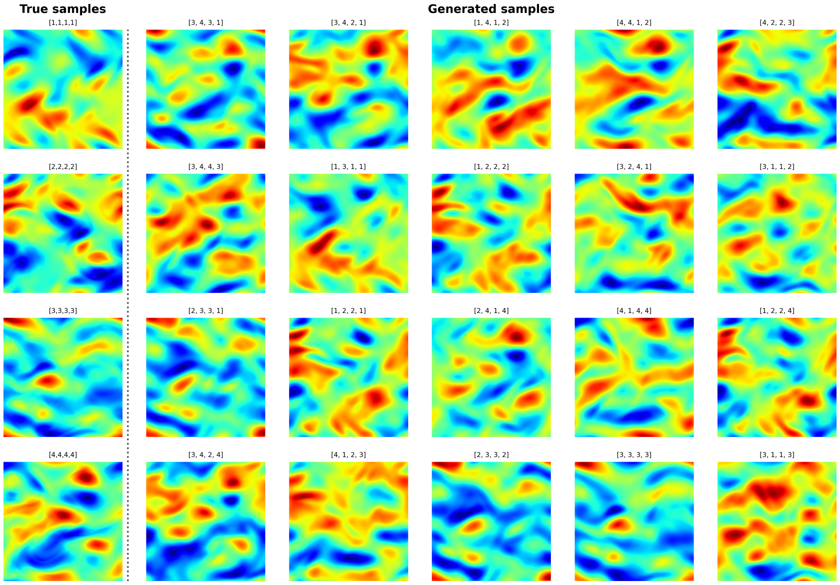

Figure 8: Turbulence snapshots generated by latent permutation. Latent corresponding to inputs of Figure 7 are used to generate new samples. The origin of the latent vector is shown above each sample. For example, (1,1,2,3) indicates that the third latent was taken from Input 2, the fourth latent was taken from Input 3 while first and second latents are first and second latents of Input 1.

regimes needs further exploration for quantifying the benefits. We are also interested in utilizing the structure of the latent representation for downstream tasks such as reduced order modeling. Fig. 9 shows the latent representation of training data for the Rankine vortex-induced flow fields. While decomposed diffusion models encode spatial features, we find that latent trajectories evolve continuously in time. Since the temporal evolution in latent space is continuous, a sequential model maybe trained to auto-regressively predict future latents from initial conditions. This can be an interesting direction for data-driven surrogate modeling.

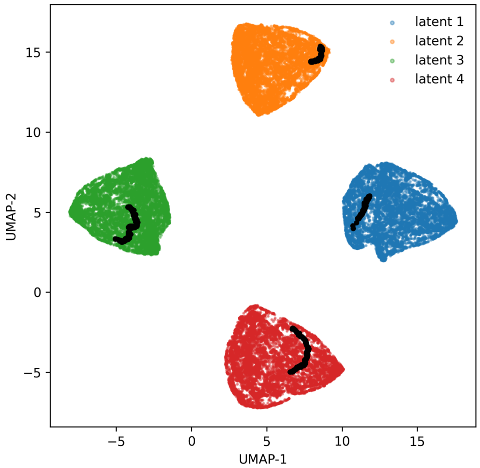

Figure 9: UMAP of latent vectors for Rankine Vortex Flows. UMAP basis learned on latent vectors corresponding to all training data. Four clusters show clear separation between the four latent vectors. Moreover, if any particular sample is allowed to evolve in time (vortices allowed to move around due to induced velocities), the corresponding latent vectors also evolve continuously. This may be leveraged for reduced-order modeling. i.e., learning temporal dynamics in latent space by a lower-order model and subsequently using diffusion model for high-fidelity future snapshots.

References

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*, pp. 843–852, 2023.

G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25:539–575, 1993.

Tony Bonnaire, Raphael Urfin, Giulio Biroli, and Marc Mézard. Why diffusion models don't memorize: The role of implicit dynamical regularization in training. *arXiv preprint arXiv:2505.17638*, 2025. URL https://arxiv.org/abs/2505.17638.

Arwen Bradley, Preetum Nakkiran, David Berthelot, James Thornton, and Joshua M Susskind. Mechanisms of projective composition of diffusion models. *arXiv preprint arXiv:2502.04549*, 2025.

Minshuo Chen, Song Mei, Jianqing Fan, and Mengdi Wang. Opportunities and challenges of diffusion models for generative AI. *Natl. Sci. Rev.*, 11(12):nwae348, 3 December 2024. doi: 10.1093/nsr/nwae348.

Hyungjin Chung, Jeongsol Kim, and Jong Chul Ye. Diffusion models for inverse problems. *arXiv preprint arXiv:2508.01975*, 2025.

Agnimitra Dasgupta, Harisankar Ramaswamy, Javier Murgoitio-Esandi, Ken Y Foo, Runze Li, Qifa Zhou, Brendan F Kennedy, and Assad A Oberai. Conditional score-based diffusion models for solving inverse elasticity problems. *Comput. Methods Appl. Mech. Eng.*, 433(117425):117425, 1 January 2025. doi: 10.1016/j.cma.2024.117425.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Pan Du, Mingqi Xu, Xiaozhi Zhu, and Jian-Xun Wang. Hug-vas: A hierarchical nurbs-based generative model for aortic geometry synthesis and controllable editing. *arXiv preprint arXiv:2507.11474*, 2025. doi: 10.48550/arXiv.2507.11474.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in neural information processing systems*, 32, 2019.

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation and inference with energy based models. *arXiv preprint arXiv:2004.06030*, 2020.

Yilun Du, Shuang Li, Yash Sharma, Josh Tenenbaum, and Igor Mordatch. Unsupervised learning of compositional energy concepts. *Advances in Neural Information Processing Systems*, 34:15608–15620, 2021.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pp. 8489–8510. PMLR, 2023.

Alessandro Favero, Antonio Sclocchi, and Matthieu Wyart. Bigger isn't always memorizing: Early stopping overparameterized diffusion models. 2025.

Pengsheng Guo and Alexander G Schwing. Variational rectified flow matching. *arXiv preprint arXiv:2502.09616*, 2025.

Yingqing Guo, Hui Yuan, Yukang Yang, Minshuo Chen, and Mengdi Wang. Gradient guidance for diffusion models: An optimization perspective. *Advances in Neural Information Processing Systems*, 37:90736–90770, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Nan Liu, Shuang Li, Yilun Du, Josh Tenenbaum, and Antonio Torralba. Learning to compose visual relations. *Advances in Neural Information Processing Systems*, 34:23166–23178, 2021.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European conference on computer vision*, pp. 423–439. Springer, 2022.

Nan Liu, Yilun Du, Shuang Li, Joshua B Tenenbaum, and Antonio Torralba. Unsupervised compositional concepts discovery with text-to-image generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2085–2095, 2023.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044): 84–90, January 2025. doi: 10.1038/s41586-024-08252-9.

Anirban Samaddar, Yixuan Sun, Viktor Nilsson, and Sandeep Madireddy. Efficient flow matching using latent variables. *arXiv preprint arXiv:2505.04486*, 2025.

Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010. doi: 10.1017/S0022112010001217.

Greg Schuette, Zhuohan Lao, and Bin Zhang. ChromoGen: Diffusion model predicts single-cell chromatin conformations. *Sci. Adv.*, 11(5):eadr8265, 31 January 2025. doi: 10.1126/sciadv.adr8265.

Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, pp. 111972, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv [cs.LG]*, 26 November 2020.

Jocelin Su, Nan Liu, Yanbo Wang, Joshua B. Tenenbaum, and Yilun Du. Compositional image decomposition with diffusion models. In *Proceedings of the 41st International Conference on Machine Learning (ICML) / PMLR*, 2024.

James Thornton, Louis Béthune, Ruixiang Zhang, Arwen Bradley, Preetum Nakkiran, and Shuangfei Zhai. Composition and control with distilled energy diffusion models and sequential monte carlo. *arXiv preprint arXiv:2502.12786*, 2025.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Yanbo Wang, Justin Dauwels, and Yilun Du. Compositional scene understanding through inverse generative modeling. *arXiv preprint arXiv:2505.21780*, 2025.

Yuwei Yang, Shukai Gu, Bo Liu, Xiaoqing Gong, Ruiqiang Lu, Jiayue Qiu, Xiaojun Yao, and Huanxiang Liu. DiffMC-gen: A dual denoising diffusion model for multi-conditional molecular generation. *Adv. Sci. (Weinh.)*, 12(22):e2417726, 1 June 2025. doi: 10.1002/advs.202417726.

Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky TQ Chen. Guided flows for generative modeling and decision making. *arXiv preprint arXiv:2311.13443*, 2023.

A Appendix

A.1 Background on Denoising Diffusion Models

Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020) constitute a class of generative frameworks that approximate a data distribution $p(\mathbf{x}_0)$ over $\mathbf{x}_0 \in \mathbb{R}^d$ by reversing a gradual stochastic diffusion process. The framework is defined by two dual Markov chains: a fixed forward process q that progressively injects noise into the data, and a parameterized reverse process p_θ trained to reconstruct the data structure. Formally, given a sample from the data distribution $\mathbf{x}_0 \sim p(\mathbf{x}_0)$, the forward diffusion process is a Markov chain $q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$ that gradually adds Gaussian noise according to a variance schedule $\beta_1, \ldots, \beta_T$. The transition kernel at step t is defined as:

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

Defining $\alpha_t := 1 - \beta_t$ and the cumulative product $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, the Gaussian properties of the forward process allow us to marginalize over intermediate steps. This yields a closed-form expression for $q(\mathbf{x}_t|\mathbf{x}_0)$ at an arbitrary timestep t, enabling efficient sampling via the reparameterization trick:

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

By design, as $t \to T$, the marginal distribution $q(\mathbf{x}_T)$ converges to a standard isotropic Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The generative process is defined as the learned reverse Markov chain $p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, initialized from $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The transition kernel p_θ is parameterized to approximate the intractable true posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. For sufficiently small β_t, these reverse transitions are well-approximated by a Gaussian distribution:

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

To optimize the model parameters θ, we parameterize the mean $\boldsymbol{\mu}_\theta$ to estimate the noise component $\boldsymbol{\epsilon}$ inherent in $\mathbf{x}_t$. The function approximator $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$, typically a neural network, is trained via a simplified variational lower bound on the negative log-likelihood:

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\left[|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, t)|^2\right], \tag{4}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and t is sampled uniformly from $\{1, \ldots, T\}$. Following training, synthesis is performed via ancestral sampling. Starting from pure noise $\mathbf{x}_T$, the data is iteratively reconstructed using the learned noise prediction:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t\mathbf{z}, \tag{5}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ represents stochastic noise injection for $t > 1$, and σ_t is a variance term derived from the schedule β_t.

To enable steerable synthesis, the DDPM framework can be extended to model the conditional distribution $p(\mathbf{x}_0|\mathbf{c})$, where $\mathbf{c}$ represents an auxiliary context vector (e.g., class labels, text embeddings, or scalar constraints). In this setting, the reverse diffusion process is modified to approximate the conditional posterior $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})$. This is achieved by augmenting the neural network to accept the conditioning signal, denoted as $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c})$. Consequently, the training objective is adapted to minimize the conditional noise prediction error. Assuming paired data $(\mathbf{x}_0, \mathbf{c})$, the optimization problem becomes:

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t}\left[|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, t, \mathbf{c})|^2\right], \tag{6}$$

During inference, the sampling update rule remains analogous to the unconditional case, with the noise estimate replaced by its conditional counterpart $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}, t)$, effectively biasing the trajectory of the reverse Markov chain towards the region of the data manifold specified by $\mathbf{c}$.

A.2 Decomposed Diffusion Models

Standard diffusion models learn a single denoising process over the full data representation. In contrast, decomposed diffusion models (Su et al., 2024) explicitly factor the generative process into multiple components or concepts. The central goal is to exploit this factored representation for clear interpretability often missed in standard diffusion models.

A.2.1 Latent Encoding

Given a data sample $x_i^0 \sim p_D$ with dimensionality D, an encoder E_ϕ maps data to K separate latent vectors.

$$E_\phi(x_i^0) = (\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_K) \quad \boldsymbol{z}_j \in \mathbb{R}_k^d, \quad d \ll D. \tag{7}$$

A.2.2 Generative Decoder

A conditional diffusion model generates samples via gradually denoising pure Gaussian noise. However, the reverse posterior now contains explicit conditioning on the latent vectors $\boldsymbol{z}_j$. Instead of predicting a single noise, the model predicts K noises which should together compose the total added corruption noise added during forward diffusion. The training loss is accordingly updated to optimize the encoder and the diffusion models.

$$\theta, \phi = \arg\min \ \mathbb{E}_{x_i,t,\epsilon}\left[\left\|\epsilon - \frac{1}{K}\sum_{k=1}^{K}\epsilon_\theta(x_t^i, t, \boldsymbol{z}_k)\right\|^2\right], \quad \{\boldsymbol{z}_j\}_{j=1}^K = E_\phi(x_t^i). \tag{8}$$

To generate the component representing a particular latent vector, the denoising is carried out conditioned only on that latent vector. Thus, using z_k, we can iteratively denoise a Gaussian sample using the noise predictor $\epsilon_\theta(\cdots|\boldsymbol{z}_k)$ to generate Component k. If instead the composition is desired, then the denoising step is carried out using the average of the K contributions $\frac{1}{K}\sum_k \epsilon_\theta(\cdots|\boldsymbol{z}_k)$.

A.2.3 Latent Orthogonality

The goal of predicting K noise components is to possibly exploit the factored representation for interpretability. However, learning disentangled factors is not always guaranteed. For example, if the latent dimensionality d is large, the individual components can become identical to the original sample.

.

We found that enforcing orthogonality between latent components seems to help disentangling the factors. This is shown in Figure 10 where 4 dissimilar Rankine vortices are not disentangled by standard decomposed diffusion. To this end, we added an additional term in the loss function $\mathcal{L}_\mathcal{T} = \mathcal{L}_{simple} + \mathcal{L}_{orthogonality}$, where the diffusion part of the loss ($\mathcal{L}_{simple}$) comes from Eqn. 8 while the orthogonality term is given by,

$$\mathcal{L}_{orthogonality} = \sum_{ij}\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_j - \delta_{ij}\right)^2 \tag{9}$$

Where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. The optimization becomes,

$$\theta, \phi = \arg\min \mathcal{L}_\mathcal{T} \tag{10}$$

Figure 10 shows the results when a standard decomposed diffusion model is trained on 8000 samples of Rankine vortex induced velocities. The model fails to separate out the different vortices. However, a decomposed diffusion model with identical hyperparameters but with an added term in the loss function for latent orthogonality is able to separate out different vortices showing more independence between the different components.

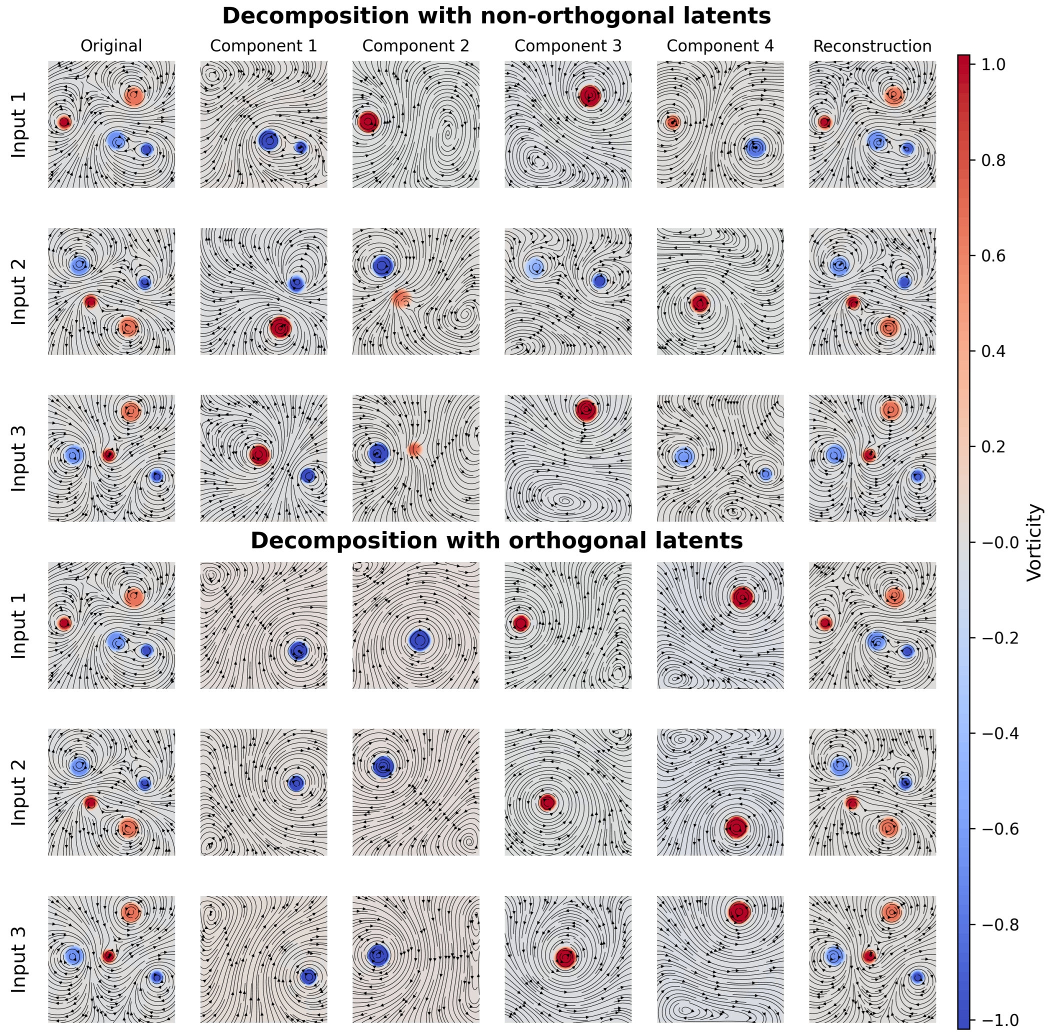

Figure 10: Decomposition of flow fields generated by four dissimilar Rankine vortices. Three different inputs are considered for two different Models. The first three rows show decomposition through standard decomposed-diffusion model. As is evident, the model is unable to decouple different vortices. The last three rows show decomposition through a decomposed-diffusion model with added constraint for latent orthogonality. Notably, a soft constraint for latent orthogonality disentangles the vortices and makes the training more efficient in learning interpretable components.

A.2.4 Permutation for Generation

Once a latent representation has been learned, new samples may be generated in a controlled manner by choosing latent vectors representing desired features in the generated sample. For example, the heat sources shown in Figure 2 may be permuted to generate new samples. Starting from N inputs with K latent vectors each, any generated sample requires forming a new latent representation consisting of K vectors. For this new representation, every component may be chosen from the N available options and therefore a total of N^K valid samples may be generated.

This is shown in Figure 12 for the heat sources and Figure 13 for the vortex flows where the latent vectors from the inputs of Figure 2 and Figure 4 are permuted to generate new samples. Even for very small datasets, we find the memorizing the latents instead of the training samples lets us generate new samples. This is demonstrated for the Human artery dataset.

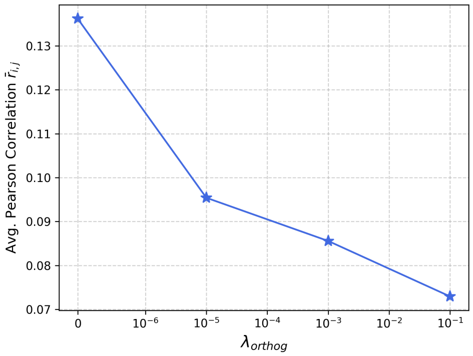

Figure 11: Impact of orthogonality penalty (λ_{orthog}) on component independence: The semantic disentanglement between the latent conditioned components is quantified using the Pearson correlation coefficient for 100 different inputs to models trained with different orthogonality penalty weights. At $\lambda = 0$, higher inter-component correlation indicates redundant feature representation. Increasing the penalty decreases the correlation while also separating the physical features as seen in Figure 10.

A.3 Human Aortic Geometries

The human aortic geometries considered in the present work are shown in Figure 14. 21 samples of patient specific aortic geometries are available. A non-uniform rational B-splines (NURBS)-based parametrization introduced by Du et al. (2025) was used to work with these geometries which are characterized by their centerline and control points on the surface.

A decomposed diffusion model is then trained to learn a latent representation of the control points with $K = 3$ latent vectors. Naturally, with a small number of samples, there is memorization of the latent representation. The last original sample, unseen during training, is not represented accurately by the model. This shows memorization of the latents for the 20 training samples. However, latent permutation lets us generate new, anatomically valid geometries despite memorization (See Figure 15). Usually, generative models would memorize the training data but decomposed diffusion models memorize the latent representation which provides additional flexibility through latent permutation.

A.4 Generation in low-data regime

High quality generation through generative models is difficult in low-data settings. Memorization sets in quickly and the model generates samples increasingly similar to training data. Decomposed diffusion also suffers from memorization. Under low-data regime, the encoder fails to learn interpretable components. However, provided the latent dimensionality is chosen to be sufficiently small, the memorized latents do represent some underlying feature of the training data. Every sample seen during training has a latent representation which the model memorizes. As a result, the model correctly encodes and reconstructs the training samples. An unseen sample (shown in red in Figure 16) is not correctly encoded by the network and the reconstruction fails. This shows memorization in latent space. However, latent permutation, which is essentially a non-linear composition, still yields realistic, diverse samples despite memorization. Since the encoder is not generalizable, the generation is neither controllable nor interpretable. However, unlike standard diffusion models, the decomposed components may be combined to yield new samples. Figure 16 shows the contrast between identically parameterized and trained decomposed diffusion and standard diffusion models. Standard diffusion baseline, trained with identical parameters and for an identical number of training iterations, fails to generate diverse samples.

To quantify the diversity and memorization, we calculate distances of generated samples to their nearest neighbor in the training data (See Figure 17). For the standard diffusion model, the mean and variance of these distances is much smaller compared to decomposed diffusion samples showing memorization and lack of generalization. After an equal amount of training, the diffusion models fails to generalize. If trained further, the mean and variance of these distances decreases further.

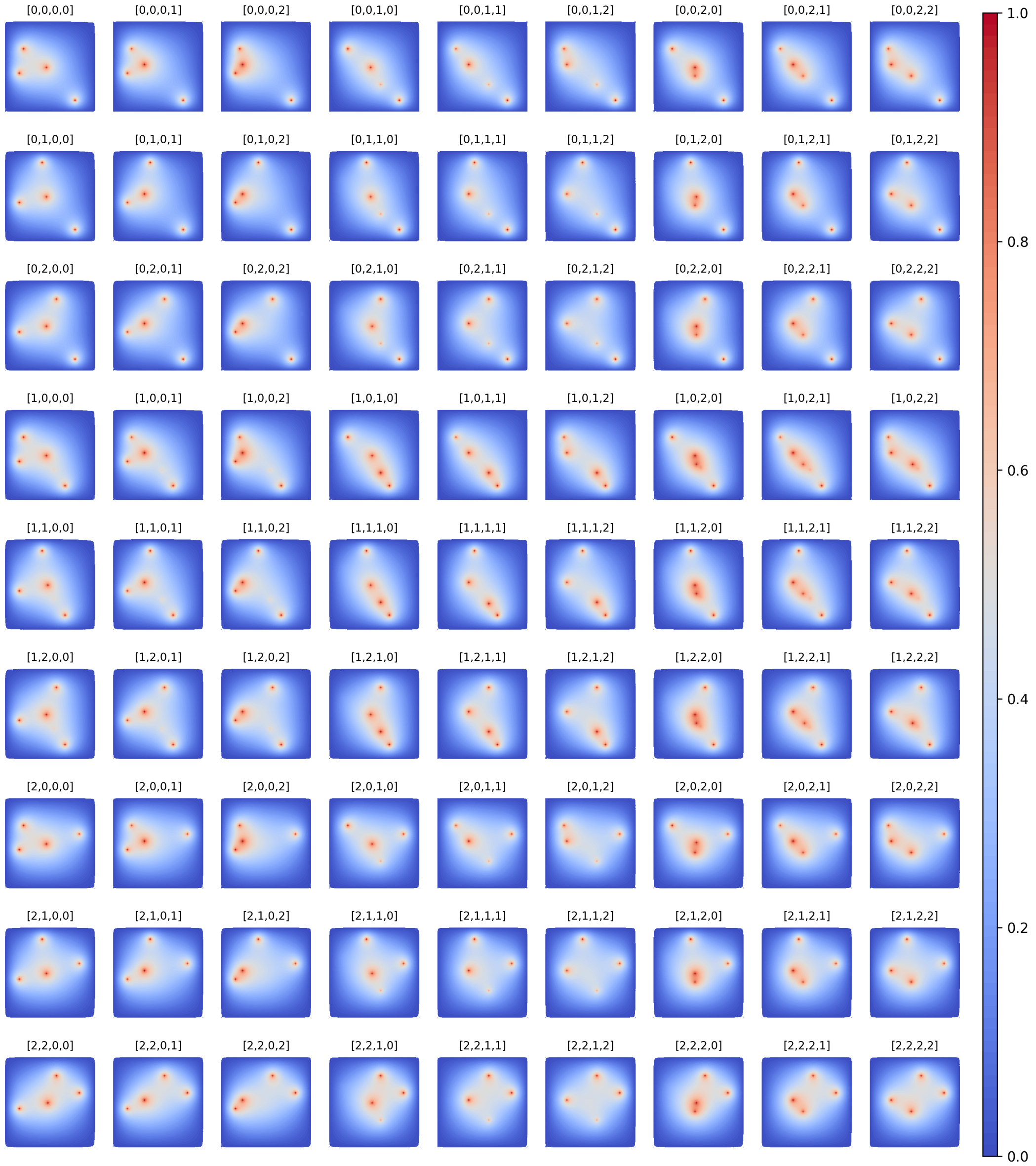

Figure 12: Latent permutation for generating new samples. Retaining control over the location of heat sources. $N^K = 3^4$ samples are generated from the inputs of Figure 2.

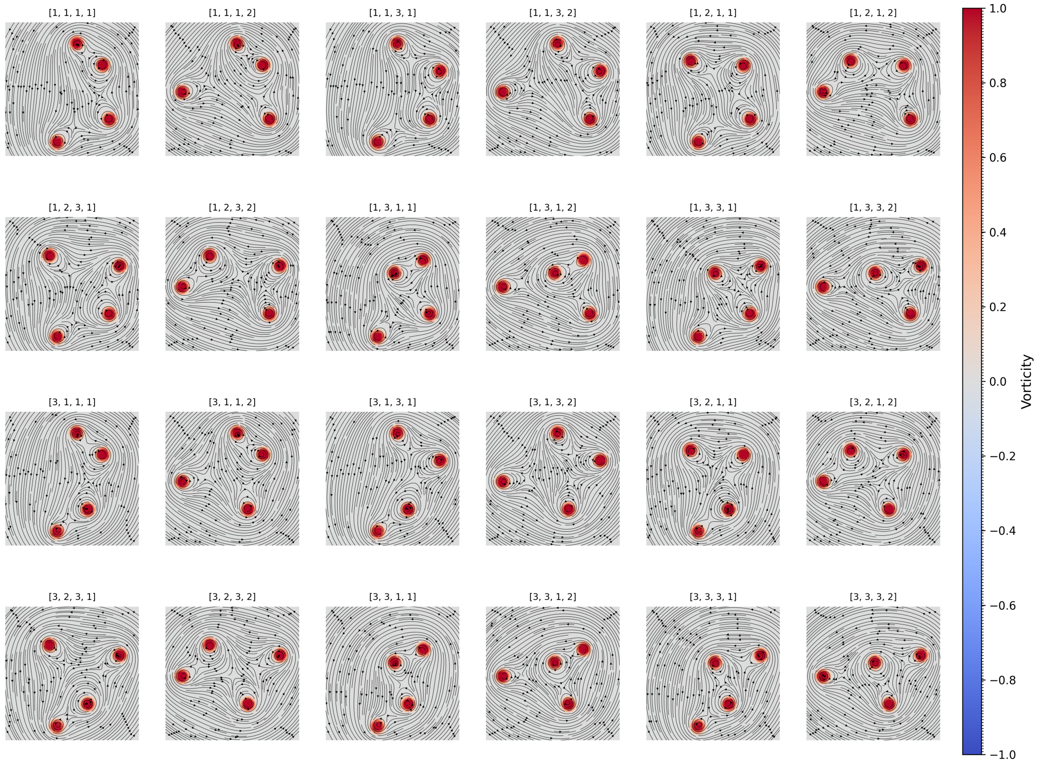

Figure 13: Latent permutation for generating new samples for the Rankine vortex induced flows considered in Figure 4.

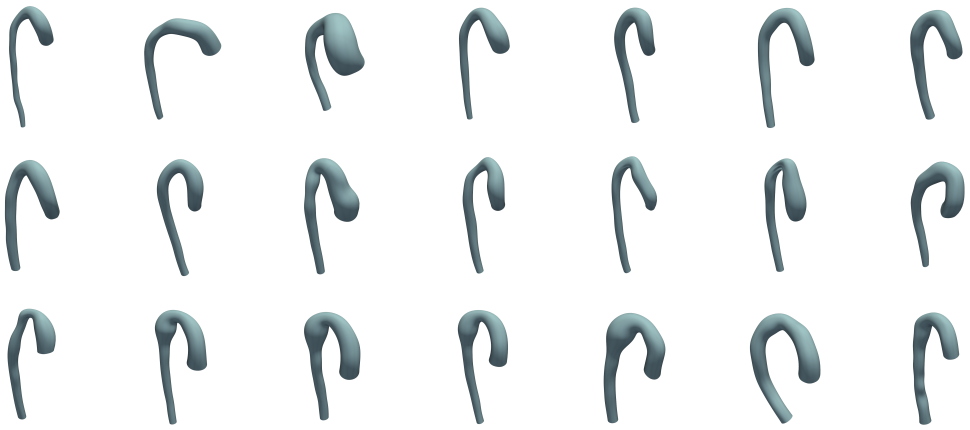

Figure 14: Human Aorta Dataset: Original 21 patient-specific geometries.

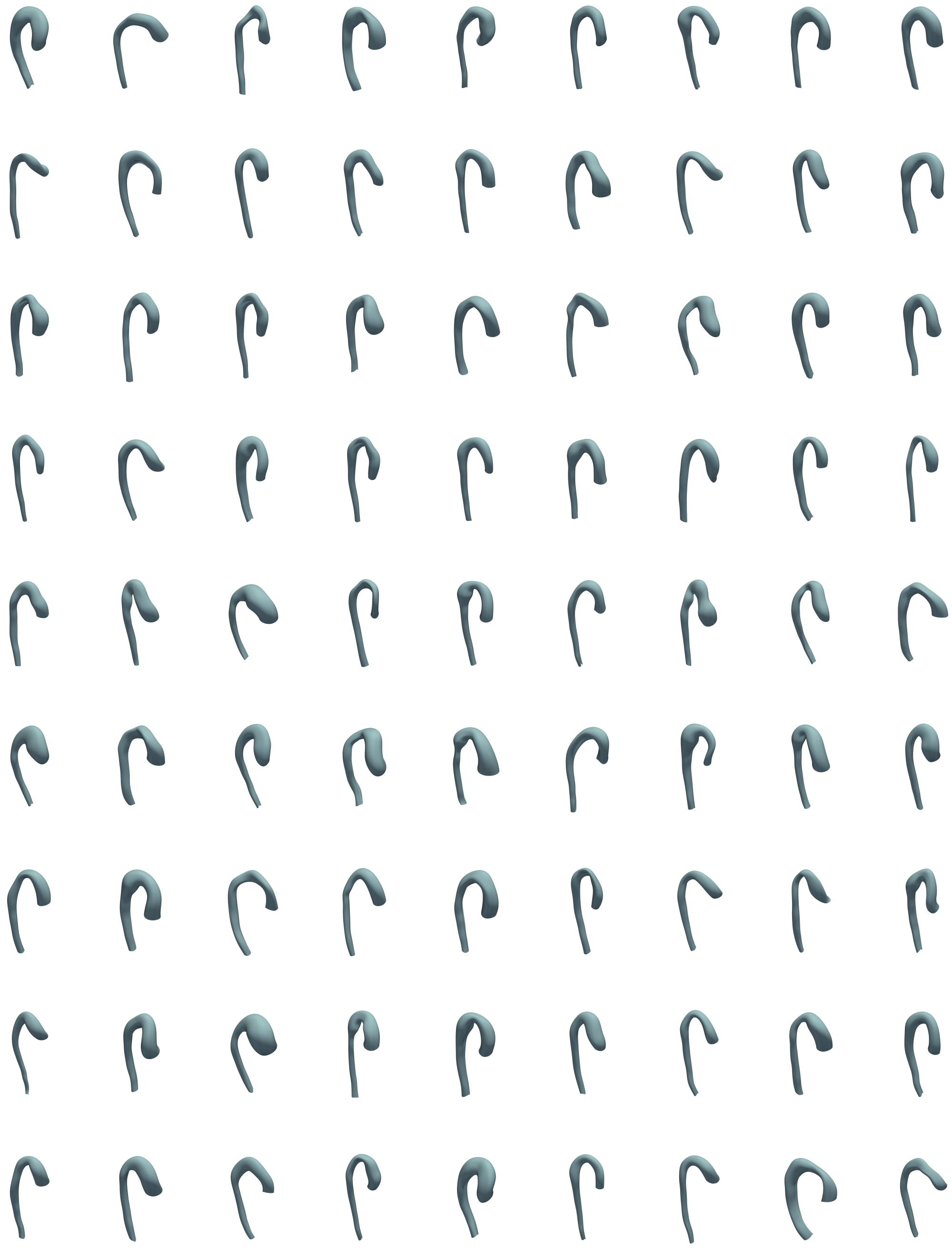

Figure 15: Generated Human Aortic Geometries: samples generated by latent permutation in very small dataset settings. Note that this figure complements Figure 5 which shows some of these samples in the main body.

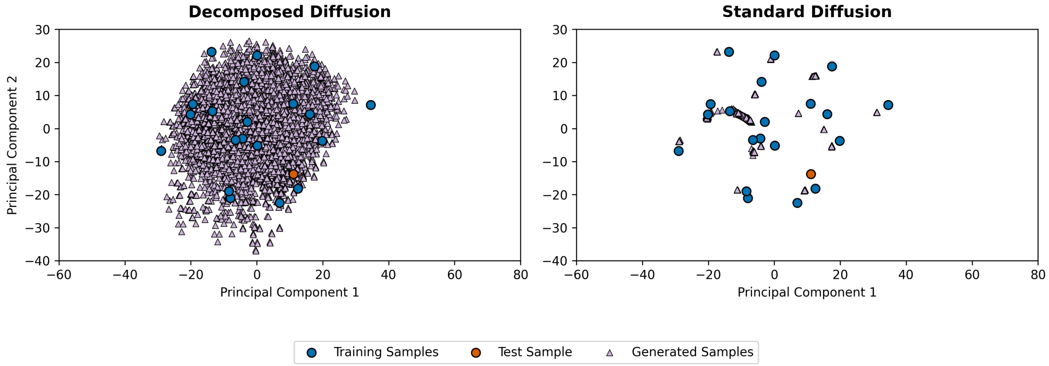

Figure 16: Generation under data scarcity: A decomposed diffusion model and a standard diffusion model are trained with the same network architecture, hyperparameters and on the same aortic geometries. The first two principal components of original dataset (21 samples) are used to project the generated samples after identical training iterations. (Left) Decomposed diffusion (with N=20, K=3) generates a maximum of $N^K = 8000$ samples that show good generalization. (Right) Samples generated via unconditional generation through the standard diffusion model. These samples show little diversity and occur in small clusters that move closer to the training data as the model trains further.

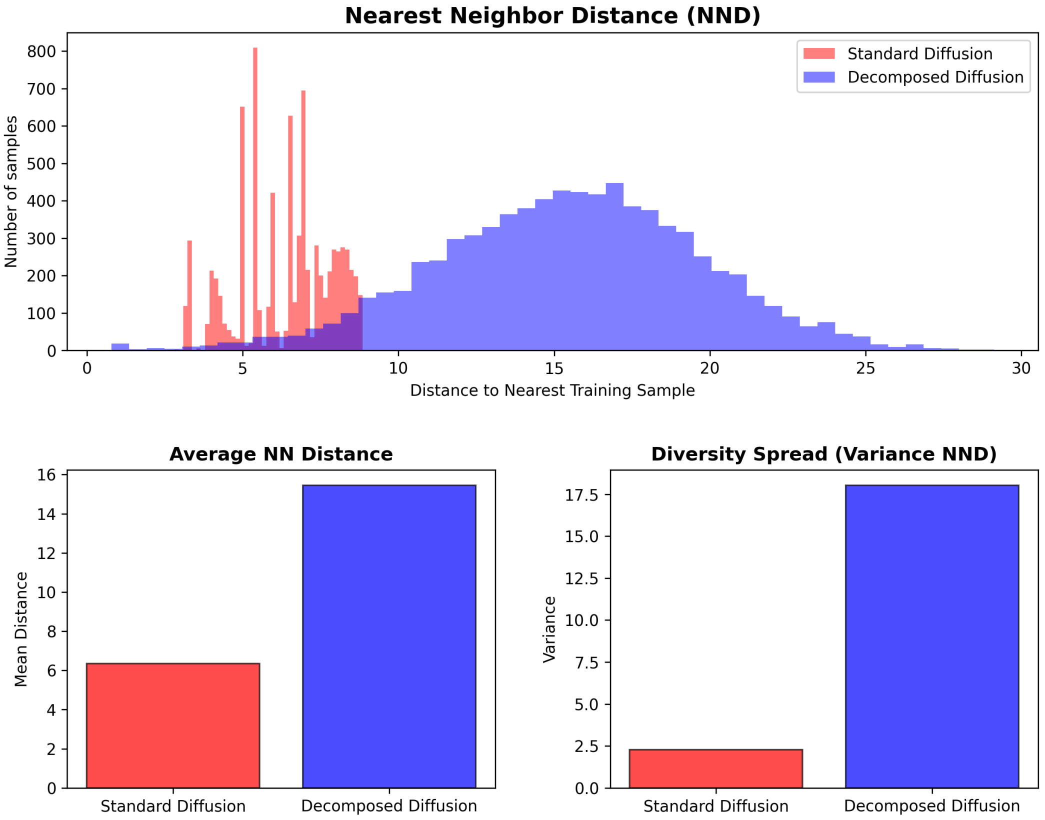

Figure 17: Distance of generated samples to nearest training sample: (Top) Human aorta geometries generated through standard diffusion and decomposed diffusion are characterized by their distances to the nearest training sample in principal component space. (Bottom Left) On average, standard diffusion produces sample much closer to training data. This deteriorates further on continued training. (Bottom Right) The diversity within the training sample, characterized by variance of NND, is much larger for decomposed diffusion samples as is also qualitatively evident from PCA projection.