Eurographics Symposium on Geometry Processing 2025
M. Attene and S. Sellán
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 5

FRIDU: Functional Map Refinement with Guided Image Diffusion - Supplemental

Avigail Cohen Rimon, Mirela Ben-Chen and Or Litany

Technion - Israel Institute of Technology

1. Implementation Details and Network Parameters

To enable training a diffusion model on a very limited dataset, we adopt the Patch-Diffusion paradigm from [WJZ*23] and build upon the code they provide. Specifically, we use their implementation based on the UNet-based diffusion model EDM-DDPM++ [KAAL22]. Following [WJZ*23], we employ the EDM sampling strategy with 50 deterministic reverse steps during inference.

In addition to default settings, we use the following hyperparameters for the Patch-Diffusion model: num_blocks $= 2$, model_channels $= 16$, channel_mult $= [2,4]$, and channel_mult_emb $= 0.1$. We train using 3 patch resolutions, where the functional maps have dimension 128×128 in all experiments—except in Section 4.2, where we match the resolution used in DiffZO which is 130×130. The parameter duration $= 2$ is used across all experiments, except in Section 4.2, where we set duration $= 10$ for the model trained on FAUST, and duration $= 15$ for the model trained on FAUST+SCAPE.

We use a batch size of 32 for all experiments, except for training on FAUST+SCAPE, where we use a batch size of 64. We run all experiments on a single NVIDIA A40 GPU, except for the FAUST+SCAPE training, which we run on NVIDIA L40S GPU.

For guidance, we implement the algorithm described in [BCS*23] with adaptation to the EDM.

2. Initial Functional Map Computation

We use the `pyFM` Python package [Mag21] to generate initial functional maps for the experiments in Section 4.1, which also includes the computation of WKS descriptors. For the WKS we use the parameters *n_descr* $= 100$, *subsample_step* $= 5$ based on 150 eigenvectors. For SHOT descriptors, we compute them in advance using the code from [HLR*19], with the parameters num_evecs $= 150$, num_bins $= 10$, and radius $= 15$.

3. DiffZO Comparison

For comparison with the DiffZo method [MO24], we extract the weights of their pretrained Feature Extractor model (DiffusionNet [SACO22]) for models trained on FAUST and FAUST+SCAPE. To compute the initial functional map used in our model, we first compute the pointwise map Π_{init} from the extracted features and then derive the corresponding functional map using the corresponding Laplace-Beltrami eigenbases.

4. Visualizations

For visualizing functional map matrices, we use the colormap defined in [LJO19], and for rendering in Blender, we utilize BlenderToolbox [Liu18].

References

[BCS*23] BANSAL A., CHU H.-M., SCHWARZSCHILD A., SENGUPTA S., GOLDBLUM M., GEIPING J., GOLDSTEIN T.: Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 843–852. 1

[HLR*19] HALIMI O., LITANY O., RODOLA E., BRONSTEIN A. M., KIMMEL R.: Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 4370–4379. 1

[KAAL22] KARRAS T., AITTALA M., AILA T., LAINE S.: Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems 35* (2022), 26565–26577. 1

[Liu18] LIU H.-T. D.: Blender Toolbox, 12 2018. URL: https://github.com/HTDerekLiu/BlenderToolbox. 1

[LJO19] LIU H.-T. D., JACOBSON A., OVSJANIKOV M.: Spectral coarsening of geometric operators. *arXiv preprint arXiv:1905.05161* (2019). 1

[Mag21] MAGNET R.: pyfm: Functional maps in python. https://github.com/RobinMagnet/pyFM, 2021. Accessed: 2024-04-10. 1

[MO24] MAGNET R., OVSJANIKOV M.: Memory-scalable and simplified functional map learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 4041–4050. 1

[SACO22] SHARP N., ATTAIKI S., CRANE K., OVSJANIKOV M.: Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG) 41*, 3 (2022), 1–16. 1

[WJZ*23] WANG Z., JIANG Y., ZHENG H., WANG P., HE P., WANG Z., CHEN W., ZHOU M., ET AL.: Patch diffusion: Faster and more data-efficient training of diffusion models. *Advances in neural information processing systems 36* (2023), 72137–72154. 1