Reproducibility Statement. We provide experimental details regarding the datasets, baselines, evaluation metrics, and implementation in Sec. 4. Additional details of the dataset are provided in Appendix C. In Appendix D, we provide additional details about the baselines, the implementation of our method, and the computation resources used. Code available at `https://github.com/Saumya-Gupta-26/TopoDiffusionNet`

Acknowledgments. I would like to express my gratitude to Eisha Gupta for her valuable feedback on the writing and presentation of this paper. I also sincerely appreciate the anonymous reviewers for their time and effort in providing constructive critiques during the review and rebuttal process. This research was partially supported by the National Science Foundation (NSF) grants CCF-2144901, IIS 2212046 and SCH 2123920, the National Institute of Health (NIH) grants R01NS143143 and R01CA297843, and the Stony Brook Trustees Faculty Award.

References

Shahira Abousamra, Minh Hoai, Dimitris Samaras, and Chao Chen. Localization in the crowd with topological constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 872–881, 2021.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18208–18218, 2022.

Samik Banerjee, Lucas Magee, Dingkang Wang, Xu Li, Bing-Xing Huo, Jaikishan Jayakumar, Katherine Matho, Meng-Kuan Lin, Keerthi Ram, Mohanasankar Sivaprakasam, et al. Semantic segmentation of microscopic neuroanatomical data by combining topological priors with encoder–decoder deep networks. *Nature machine intelligence*, 2(10):585–594, 2020.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023.

Dina Bashkirova, José Lezama, Kihyuk Sohn, Kate Saenko, and Irfan Essa. Masksketch: Unpaired structure-guided masked image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1879–1889, 2023.

Rickard Brüel-Gabrielsson, Bradley J Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J Guibas, and Gunnar Carlsson. A topology layer for machine learning. *arXiv preprint arXiv:1905.12200*, 2019.

Nick Byrne, James R Clough, Israel Valverde, Giovanni Montana, and Andrew P King. A persistent homology-based topological loss for cnn-based multiclass segmentation of cmr. *IEEE transactions on medical imaging*, 42(1):3–14, 2022.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1209–1218, 2018.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

Chao Chen, Xiuyan Ni, Qinxun Bai, and Yusu Wang. A topological regularizer for classifiers via persistent homology. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2573–2582. PMLR, 2019.

Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5343–5353, 2024.

James R Clough, Ilkay Oksuz, Nicholas Byrne, Julia A Schnabel, and Andrew P King. Explicit topological priors for deep-learning based image segmentation using persistent homology. In *International Conference on Information Processing in Medical Imaging*, pp. 16–28. Springer, 2019.

James R Clough, Nicholas Byrne, Ilkay Oksuz, Veronika A Zimmer, Julia A Schnabel, and Andrew P King. A topological loss function for deep-learning based image segmentation using persistent homology. *TPAMI*, 2020.

Tamal K Dey, Jiayuan Wang, and Yusu Wang. Road network reconstruction from satellite images with machine learning supported by topological methods. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 520–523, 2019.

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

Shiyi Du, Qicheng Lao, Qingbo Kang, Yiyue Li, Zekun Jiang, Yanfeng Zhao, and Kang Li. Distilling knowledge from topological representations for pathological complete response prediction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 56–65. Springer, 2022.

Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28:511–533, 2002.

Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Soc., 2010.

Azade Farshad, Yousef Yeganeh, Yu Chi, Chengzhi Shen, Böjrn Ommer, and Nassir Navab. Scenegenie: Scene graph guided diffusion models for image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 88–98, 2023.

J Funke, S Saalfeld, DD Bock, SC Turaga, and E Perlman. Miccai challenge on circuit reconstruction from electron microscopy images, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Saumya Gupta, Xiaoling Hu, James Kaan, Michael Jin, Mutshipay Mpoy, Katherine Chung, Gagandeep Singh, Mary Saltz, Tahsin Kurc, Joel Saltz, et al. Learning topological interactions for multi-class medical image segmentation. In *ECCV*, 2022.

Saumya Gupta, Yikai Zhang, Xiaoling Hu, Prateek Prasanna, and Chao Chen. Topology-aware uncertainty for image segmentation. *Advances in Neural Information Processing Systems*, 36, 2024.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. *arXiv preprint arXiv:2303.09556*, 2023.

Hongliang He, Jun Wang, Pengxu Wei, Fan Xu, Xiangyang Ji, Chang Liu, and Jie Chen. Toposeg: Topology-aware nuclear instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21307–21316, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. *Advances in neural information processing systems*, 30, 2017.

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In *NeurIPS*, 2019.

Xiaoling Hu, Yusu Wang, Li Fuxin, Dimitris Samaras, and Chao Chen. Topology-aware segmentation using discrete morse theory. In *ICLR*, 2021.

Xiaoling Hu, Dimitris Samaras, and Chao Chen. Learning probabilistic topological representations using discrete morse theory. In *ICLR*, 2023.

Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Shizuo Kaji, Takeki Sudo, and Kazushi Ahara. Cubical ripser: Software for computing persistent homology of image and volume data. *arXiv preprint arXiv:2005.12692*, 2020.

Valentin Khrulkov and Ivan Oseledets. Geometry score: A method for comparing generative adversarial networks. In *International conference on machine learning*, pp. 2621–2629. PMLR, 2018.

Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2426–2435, 2022.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22511–22521, 2023.

Maneesh Agrawala Lvmin Zhang. Controlnet-seg model. ControlNet-Seg model card, `https://huggingface.co/lllyasviel/sd-controlnet-seg`. 2023.

Inc Midjourney. Midjourney. `https://www.midjourney.com/`. 2024.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

James R Munkres. *Elements of algebraic topology*. CRC press, 2018.

Xiuyan Ni, Novi Quadrianto, Yusu Wang, and Chao Chen. Composing tree graphical models with persistent homology features for clustering mixed-type data. In *International Conference on Machine Learning*, pp. 2622–2631. PMLR, 2017.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

OpenAI. Dall·e 3. `https://openai.com/product/dall-e-3`, a. 2023.

OpenAI. Lsun bedroom model. `https://openaipublic.blob.core.windows.net/diffusion/march-2021/lsun_uncond_100M_2400K_bs64.pt`, b. 2021.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3170–3180, 2023.

Yaopeng Peng, Hongxiao Wang, Milan Sonka, and Danny Z Chen. Phg-net: Persistent homology guided medical image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7583–7592, 2024.

Quynh Phung, Songwei Ge, and Jia-Bin Huang. Grounded text-to-image synthesis with attention refocusing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7932–7942, 2024.

Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, et al. Unicontrol: A unified diffusion model for controllable visual generation in the wild. *arXiv preprint arXiv:2305.11147*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.

Runway. Stability ai. Stable diffusion v1.5 model card, `https://huggingface.co/runwayml/stable-diffusion-v1-5`. 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *URL https://arxiv.org/abs/2205.11487*, 4, 2022a.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022b.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2025.

Suprosanna Shit, Johannes C Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien PW Pluim, Ulrich Bauer, and Bjoern H Menze. cldice-a novel topology-preserving loss function for tubular structure segmentation. In *CVPR*, 2021.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Nico Stucki, Johannes C Paetzold, Suprosanna Shit, Bjoern Menze, and Ulrich Bauer. Topologically faithful image segmentation via induced matching of persistence barcodes. In *ICML*, 2023.

Student. The probable error of a mean. *Biometrika*, 6(1):1–25, 1908.

Robert Endre Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM (JACM)*, 22(2):215–225, 1975.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Fan Wang, Huidong Liu, Dimitris Samaras, and Chao Chen. Topogan: A topology-aware generative adversarial network. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pp. 118–136. Springer, 2020.

Fan Wang, Saarthak Kapse, Steven Liu, Prateek Prasanna, and Chao Chen. Topotxr: a topological biomarker for predicting treatment response in breast cancer. In *International Conference on Information Processing in Medical Imaging*, pp. 386–397. Springer, 2021.

Haotian Wang, Min Xian, and Aleksandar Vakanski. Ta-net: Topology-aware network for gland segmentation. In *WACV*, 2022a.

Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022b.

Meilong Xu, Xiaoling Hu, Saumya Gupta, Shahira Abousamra, and Chao Chen. Semi-supervised segmentation of histopathology images with noise-aware topological consistency. In *European Conference on Computer Vision*, pp. 271–289. Springer, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.

Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

The appendix is organized as follows.

Appendix A provides the illustration of super-level sets for 1-dim topology.

Appendix B provides the illustration of $\mathcal{L}_{\text{preserve}}$ and $\mathcal{L}_{\text{denoise}}$ for 1-dim topology.

Appendix C provides additional details of the datasets.

Appendix D provides additional baseline and implementation details.

Appendix E contains 0-dim constraint-wise results on the COCO dataset.

Appendix F presents experiments on 1-dim topology where there are non-boundary (standalone) holes.

Appendix G provides qualitative results of Stable Diffusion (Rombach et al., 2022; Runway), DALL·E 3 (OpenAI, a), and AR (Phung et al., 2024) for 1-dim topological constraints.

Appendix H includes an ablation study on the 'Encoding Network' for the topological constraint c.

Appendix I provides a discussion on the limitations and future work of our method.

Appendix J provides further details about the persistent homology computation.

Appendix K provides FID scores for the experiments.

Appendix L presents experiments on simultaneously using 0-dim and 1-dim topological constraints as conditions.

Appendix M shows experiments using Conditional Variational AutoEncoders (cVAE), justifying the use of Diffusion Models in our work.

Appendix N provides results of using masks containing basic shapes (circles and squares) to denote the cardinality. We compare it against our TopoDiffusionNet's masks which preserve the object shape. We present results on both SD1.5 and SDXL-Turbo ControlNet backbones.

A Persistent Homology

From Sec. 3.2, the equivalent of Fig. 4 for 1-dimensional topology is shown in Fig. 9.

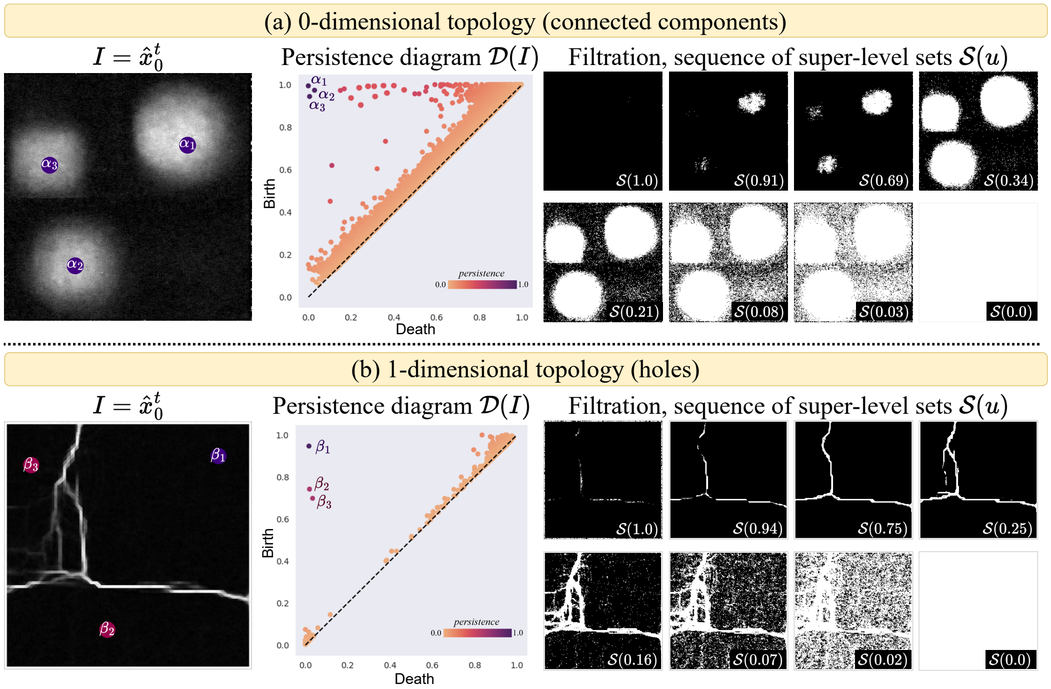


Figure 9: Illustration of persistent homology and persistence diagrams of both types of topological structures, 0-dim connected components and 1-dim holes. (a) Despite the noise, we can visually see three prominent structures $\alpha_1, \alpha_2, \alpha_3$ in I. In the topological space, $\alpha_1, \alpha_2, \alpha_3$ thus appear in the top-left corner of the persistence diagram $\mathcal{D}(I)$, persisting through most of the filtration $\mathcal{S}$. Similarly in (b), $\beta_1, \beta_2, \beta_3$ denote the prominent holes. All the remaining connected components and holes are noisy, persisting over a short threshold in $\mathcal{S}$, thus appearing closer to the diagonal in $\mathcal{D}(I)$. Persistence diagrams are useful to distinguish between salient and noisy structures in an image.

B Methodology

From Sec. 3.3, the equivalent of Fig. 5 for 1-dimensional topology is shown in Fig. 10.

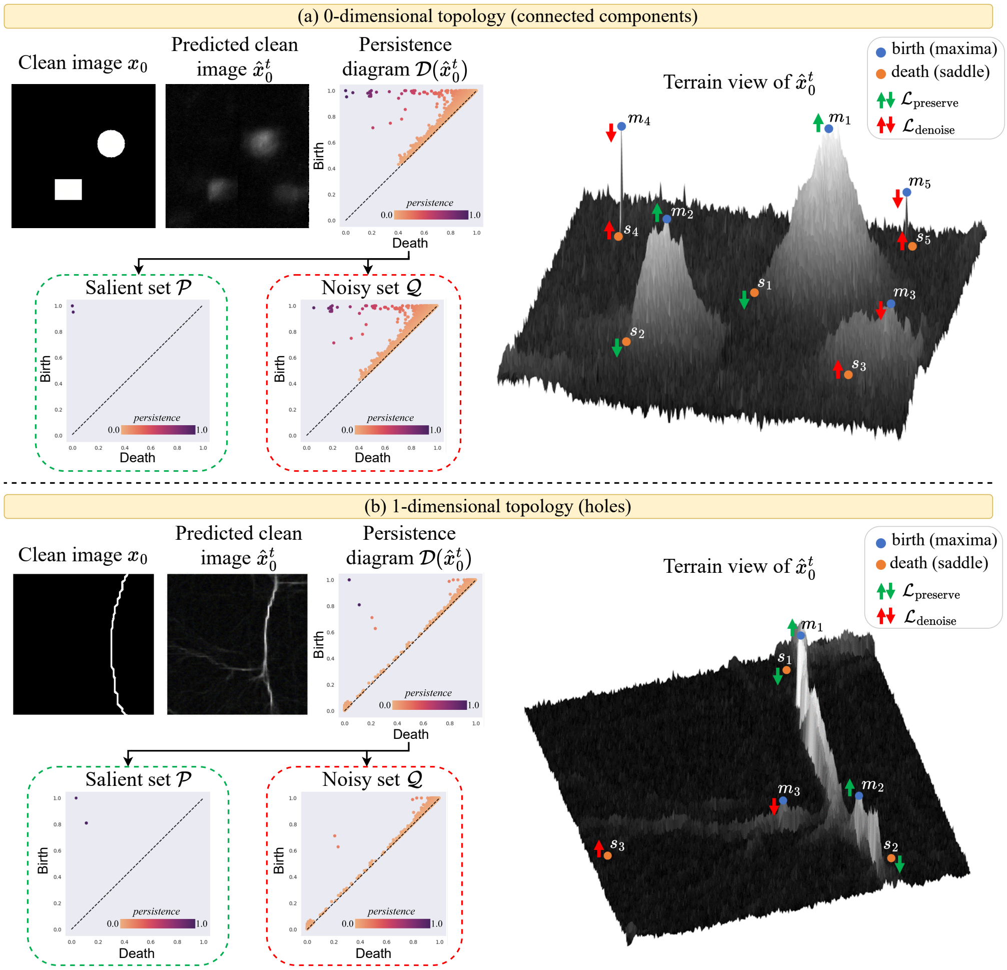

(a) 0-dimensional topology (connected components)
Clean image x_0
Predicted clean image $\hat{x}_0^t$
Persistence diagram $\mathcal{D}(\hat{x}_0^t)$
Birth
Death
persistence
0.0
1.0
Salient set $\mathcal{P}$
Noisy set $\mathcal{Q}$
Terrain view of $\hat{x}_0^t$
birth (maxima)
death (saddle)
$\mathcal{L}_{\text{preserve}}$
$\mathcal{L}_{\text{denoise}}$
m_1
m_2
m_3
m_4
m_5
s_1
s_2
s_3
s_4
s_5
(b) 1-dimensional topology (holes)
Clean image x_0
Predicted clean image $\hat{x}_0^t$
Persistence diagram $\mathcal{D}(\hat{x}_0^t)$
Birth
Death
persistence
0.0
1.0
Salient set $\mathcal{P}$
Noisy set $\mathcal{Q}$
Terrain view of $\hat{x}_0^t$
birth (maxima)
death (saddle)
$\mathcal{L}_{\text{preserve}}$
$\mathcal{L}_{\text{denoise}}$
m_1
s_1
m_2
m_3
s_2
s_3

Figure 10: Illustration of $\mathcal{L}_{\text{preserve}}$ and $\mathcal{L}_{\text{denoise}}$ for 0-dim connected components and 1-dim holes, with $c = 2$ as seen in x_0. (a) After computing $\mathcal{D}(\hat{x}_0^t)$, we partition it into sets $\mathcal{P}$ (the top c structures) and $\mathcal{Q}$ (remaining ones). For each dot $p \in \mathcal{D}(\hat{x}_0^t)$, the birth and death values respectively correspond to local maxima m_p and saddles s_p in $\hat{x}_0^t$. In the terrain view of $\hat{x}_0^t$, structures (m_1, s_1) and (m_2, s_2) belong to $\mathcal{P}$; hence optimizing $\mathcal{L}_{\text{preserve}}$ increases their saliency by increasing $\hat{x}_0^t(m_1), \hat{x}_0^t(m_2)$ and decreasing $\hat{x}_0^t(s_1), \hat{x}_0^t(s_2)$. All the remaining n structures $(m_3, s_3), (m_4, s_4), \cdots, (m_n, s_n)$ belong to $\mathcal{Q}$. Optimizing $\mathcal{L}_{\text{denoise}}$ suppresses these noisy structures by decreasing $\hat{x}_0^t(m_3), \hat{x}_0^t(m_4), \cdots, \hat{x}_0^t(m_n)$ and increasing $\hat{x}_0^t(s_3), \hat{x}_0^t(s_4), \cdots, \hat{x}_0^t(s_n)$. (b) mirrors this process for holes, where $\mathcal{L}_{\text{preserve}}$ enhances the saliency of the two holes (m_1, s_1) and (m_2, s_2), and $\mathcal{L}_{\text{denoise}}$ suppresses the appearance of all the remaining holes like (m_3, s_3).

C Dataset Details

We provide the number of images per topological constraint c used for training on each dataset in Tab. 5. For COCO (Caesar et al., 2018), since we also consider the animal class, each animal is distributed unequally across the different constraint values. For example, there were more images for 'birds' having $c = 10$ than compared to, say, 'elephant.' All the 10 animal classes are present in the dataset; they are bear, bird, cat, cow, dog, elephant, giraffe, horse, sheep, and zebra.

When curating the dataset for CREMI (Funke et al., 2016) and Google Maps (Isola et al., 2017), we manually added a (white) border to all the images. By definition of 1-dim topology, a hole is completely surrounded by a boundary. Hence, we needed to add a border to obtain the correct number of holes/regions.

Table 5: Dataset composition

Dataset	**Topological Constraint (Betti Number)**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Shapes	2K	2K	2K	2K	2K	2K	2K	2K	2K	2K
COCO (Animals)	520	517	503	297	176	85	64	38	30	27
Google Maps	549	669	1099	1220	1343	1806	602	1470	1054	662
CREMI	2160	1992	3726	3505	1644	580	187	207	170	112

D Implementation Details

All ADM-T and TDN experiments were conducted on 1 NVIDIA RTX A6000 GPU, with a batch size of 16 and a learning rate of 2×10^{-5}. As mentioned in Sec. 3.3, our diffusion model is parameterized to predict in noise space, and we use Eq. (2) to get an estimate of the noiseless image. Although diffusion models can be parameterized to predict the noiseless state directly, we find from existing works (Hang et al., 2023; Wang et al., 2022b) that their performance is poorer compared to predicting the noise. Hence we stick to the configuration of predicting the noise. This also allows us to load pretrained weights from OpenAI (b) for our experiments instead of training from scratch.

For training ADM-T and TDN, we use the PyTorch codebase from Dhariwal & Nichol (2021)[4] and use the LSUN Bedrooms pretrained model checkpoint (OpenAI, b) to fine-tune from. To compute the birth death pairs of each topological structure, we use the Cubical Ripser (Kaji et al., 2020) library.

In Fig. 1, Fig. 6, Fig. 12, and Fig. 13, the Stable Diffusion (Rombach et al., 2022) results are generated using the Diffusers[5] library with pretrained checkpoint from Runway. For DALL·E 3, we generate images using the OpenAI API[6]. For Attention Refocusing (AR) (Phung et al., 2024), we use their publicly available codebase[7] alongwith GPT-4 (Achiam et al., 2023) from the OpenAI API to generate the layout maps. For rendering images from masks via ControlNet (Zhang et al., 2023), we use the Diffusers library with pretrained checkpoint from Lvmin Zhang (SD1.5 backbone). For Fig. 7, however, we fine-tune ControlNet on the CREMI dataset (Funke et al., 2016) so as to generate appropriate results for the corresponding text prompt.

[4] https://github.com/openai/guided-diffusion

[5] https://huggingface.co/docs/diffusers/en/index

[6] https://platform.openai.com/docs/api-reference/introduction

[7] https://github.com/Attention-Refocusing/attention-refocusing

E Constraint-wise Results

In Fig. 11, we plot the accuracy of the different methods on the COCO dataset. For smaller object counts, that is, $c <= 5$, the performance of each method is better than $c > 5$. At higher object counts, although the accuracy of each method reduces, TDN still significantly outperforms the baselines.

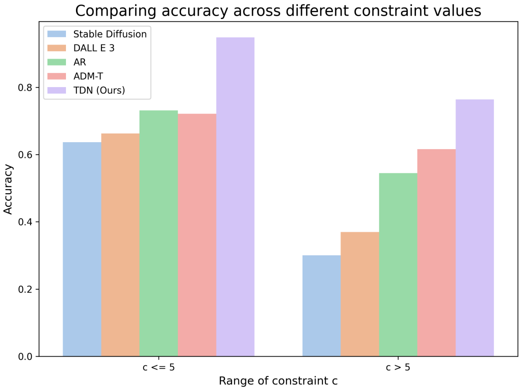


Figure 11: Accuracy results on low and high object counts (COCO dataset).

F Additional Experiments on Holes

In the main paper, we show experiments on 1-dim topology using the CREMI (Funke et al., 2016), and Google Maps (Isola et al., 2017) datasets. In these datasets, it makes sense that the holes are with respect to the image frame/boundary and span the whole image. However, TDN can handle 1-dim holes in general, not just those with respect to the boundary. To demonstrate this, we conduct experiments on standalone holes (not relative to the boundary), and present the results in Tab. 6 and Fig. 12. We generate a synthetic dataset of circular rings (similar to the Shapes dataset) to train TDN, and use the prompt ‘donuts’ in ControlNet to render the image. While generating c donuts could also be achieved using the 0-dim topological constraint, this experiment highlights the 1-dim generalizability of TDN.

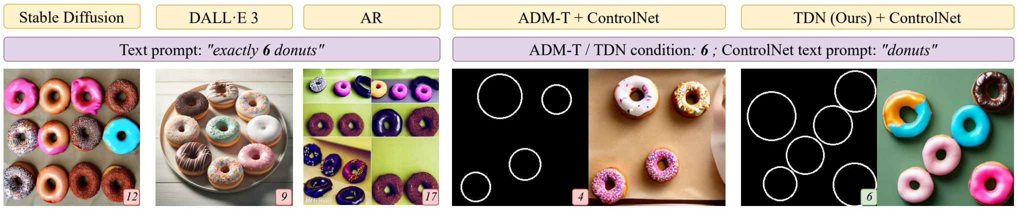

Stable Diffusion
DALL·E 3
AR
ADM-T + ControlNet
TDN (Ours) + ControlNet
Text prompt: "exactly 6 donuts"
ADM-T / TDN condition: 6 ; ControlNet text prompt: "donuts"
12
9
17
4
6

Figure 12: Results on standalone holes (holes not relative to the image frame/boundary). Number of donuts within each mask/image is noted in its bottom-right inset.

Table 6: Standalone holes

Method	Accuracy ↑	F1 ↑
ADM-T	0.79 ± 0.12	0.81 ± 0.11
TDN (Ours)	**0.95 ± 0.03**	**0.96 ± 0.02**

G Additional 1-dim Qualitative Results

In Fig. 13, we show qualitative results of pretrained Stable Diffusion (Rombach et al., 2022), DALL·E 3 (OpenAI, a) and Attention Refocusing (AR) (Phung et al., 2024) for the same 1-dim constraints shown in Fig. 7 of the main paper. As 1-dim topology is hard to describe in words, we tried a lot of variations for the text prompts, and show the results from the best ones in the figure.

As the pretrained Stable Diffusion, DALL·E 3, and AR are not trained on Electron Microscopy images of cell neurons, their inaccurate results are understandable. For generating roads, however, these methods do generate visually appropriate images but struggle to maintain the correct number of holes/regions. AR additionally tends to generate images that appear to be divided into separate, unconnected sections. This is due to the use of layout maps in the reverse process. All these methods are limited to generating images with 0-dim topology (i.e., distinct objects), and do not extend to 1-dim topology. This shortcoming motivates our work on TopoDiffusionNet.

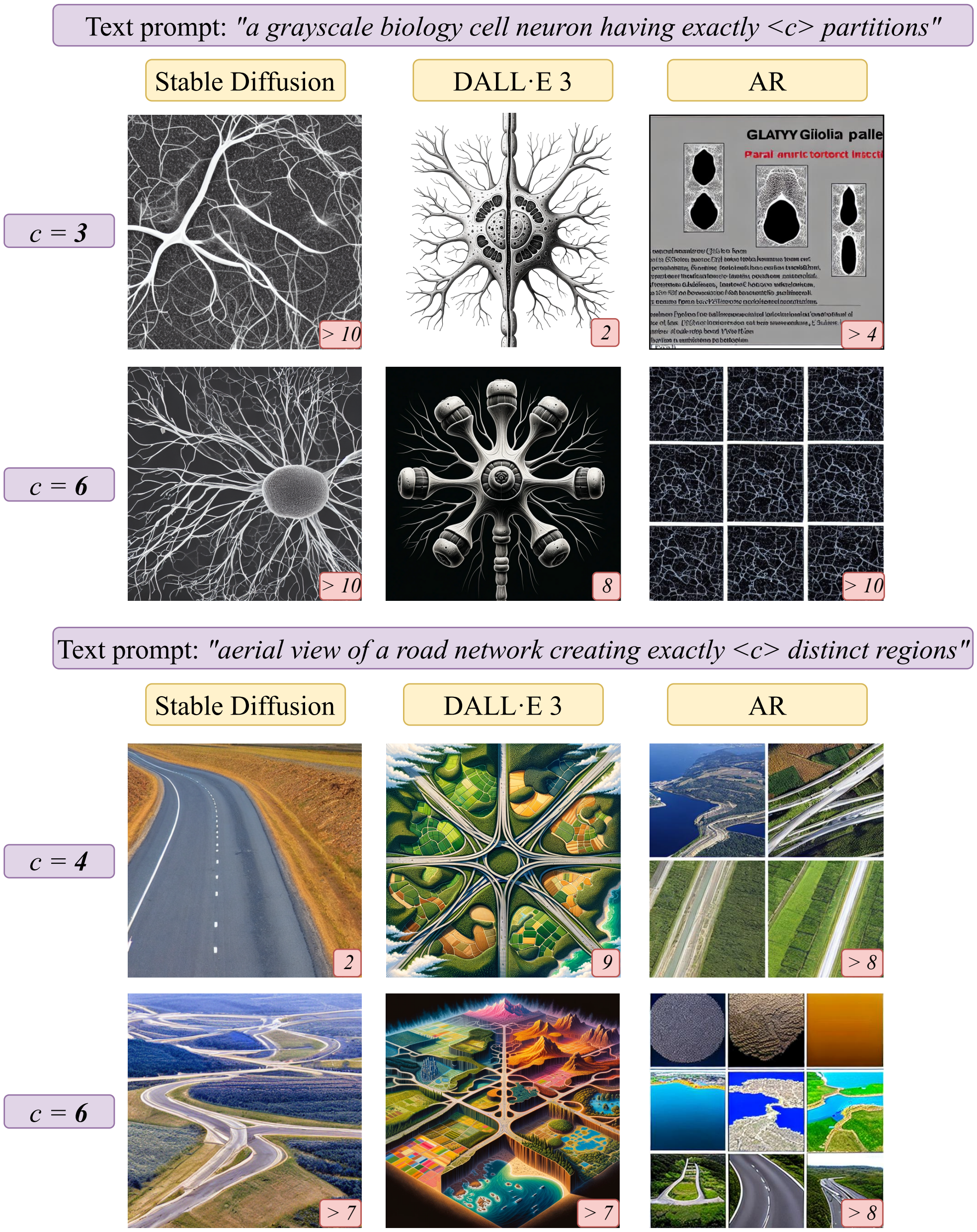

Text prompt: "a grayscale biology cell neuron having exactly <c> partitions"
Stable Diffusion
DALL·E 3
AR
c = 3
> 10
2
> 4
c = 6
> 10
8
> 10
Text prompt: "aerial view of a road network creating exactly <c> distinct regions"
Stable Diffusion
DALL·E 3
AR
c = 4
2
9
> 8
c = 6
> 7
> 7
> 8

Figure 13: Qualitative results for 1-dim topological constraint. Stable Diffusion, DALL·E 3 and AR take text prompts as input (purple box). Rows 1-2: Results equivalent to CREMI. Rows 3-4: Results equivalent to Google Maps. Number of holes within each image is noted in its bottom-right inset.

H Additional Ablation Study

We conduct an additional ablation study apart from the ones presented in the main paper.

Ablation study of the encoding network. In TDN, we use the topological constraint c as a condition. We do this by first obtaining an embedding of c via an Encoding Network (Fig. 3), and then passing it to all the residual blocks in the denoising model.

In the results reported in the main paper, the Encoding Network is composed of a few linear layers. We use 'LL' to denote this configuration.

Another way to configure the Encoding Network is to use the Transformer sinusoidal position embedding (Vaswani et al., 2017). Let 'PE' denote this configuration. It uses the same code as the network generating the sinusoidal timestep embedding.

We show results comparing LL and PE in Tab. 7 when using our proposed objective function $\mathcal{L}_{\text{top}}$. We find that both configurations have comparable performance, with LL slightly outperforming PE.

Table 7: Ablation study on Encoding Network for TDN

Dataset	Encoding	Accuracy ↑	Precision ↑	F1 ↑
Shapes	LL	**0.9478 ± 0.0420**	**0.9499 ± 0.0492**	**0.9488 ± 0.0370**
	PE	0.9011 ± 0.0730	0.9132 ± 0.0683	0.9033 ± 0.0385
COCO (Animals)	LL	**0.8557 ± 0.0805**	**0.8670 ± 0.0636**	**0.8613 ± 0.0970**
	PE	0.8395 ± 0.0952	0.8436 ± 0.1152	0.8411 ± 0.1014

I Discussion

Limitations. Presently, TDN requires at least a few samples of each constraint in the training data and is not guaranteed to extrapolate to unseen constraints. This limitation stems from the broader challenges associated with state-of-the-art diffusion models, which require a large amount of training data. Exploring the numeric relationships between the different constraints, instead of treating them independently, has the potential to generalize to unseen constraints.

Future work. Our proposed TDN currently uses persistent homology to control the number of objects (in 0-dim) and the number of holes (in 1-dim). However, persistent homology can theoretically be extended to higher dimensions, as persistence diagrams can capture topological features in arbitrary dimensions. For future work, we are looking into graph network generations, as well as 3D applications, where we can control not just connected components (0-dim) and holes (1-dim), but also voids (2-dim). 3D point clouds and volumetric medical imaging data are important applications where maintaining topology is crucial in generating realistic synthetic data.

J Details on Persistent Homology Computation

We expand on Sec. 3.2 regarding the persistent homology and persistence diagram (PD) computation. As mentioned in the implementation details (Appendix D), we use the Cubical Ripser library Kaji et al. (2020) for the homology computation. Their manuscript has details about the optimized algorithms they implement for the computation. Here, we provide a summarized version of the details.

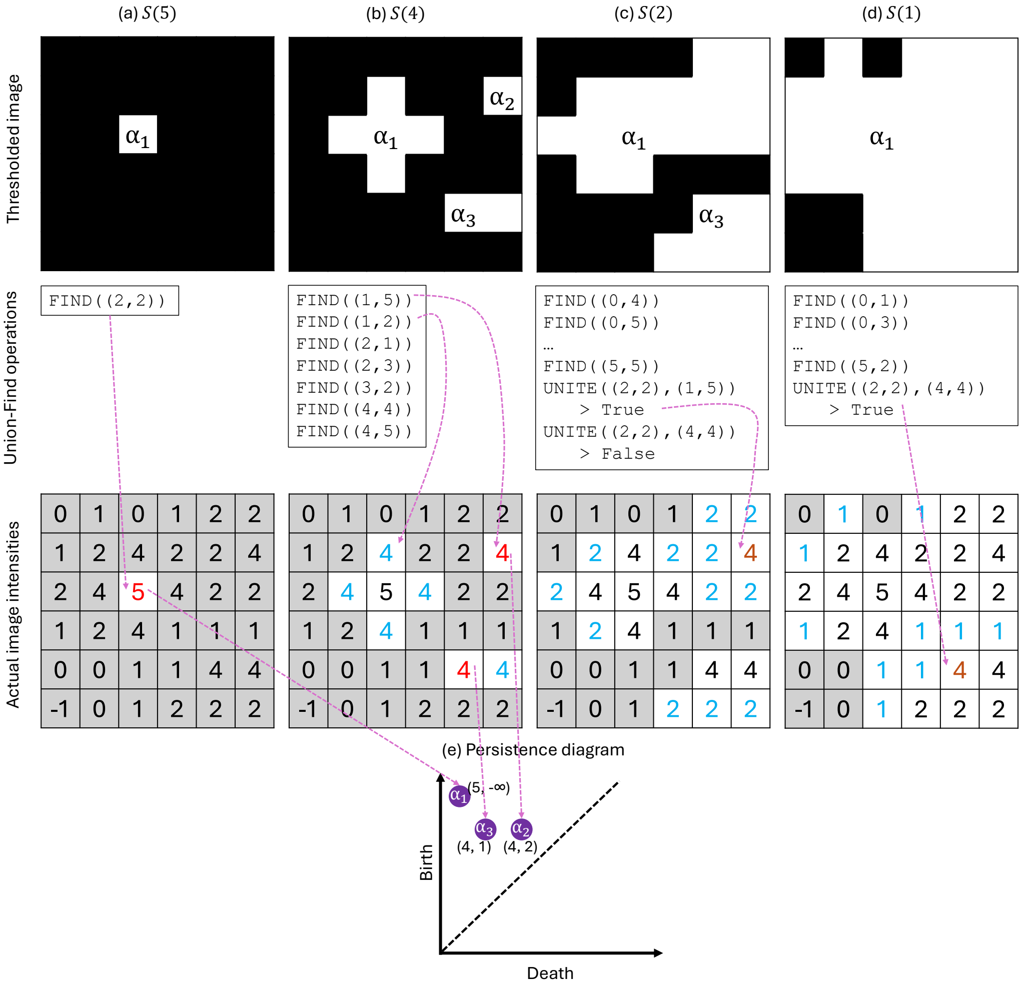
Consider a 2D image $I \in \mathbb{R}^2$. We consider the 0-dim case in which we track the birth and death of connected components (CC). As we construct a super-level filtration $\mathcal{S}$, we start thresholding an image I starting from the maximum value to the minimum value. For computational purposes, we do not need to consider all threshold values $u \in \mathbb{R}$, rather, since the image has size $H \times W$, the maximum number of unique values in the image is $H * W$. Hence we need to consider atmost $H * W$ values of threshold u to determine the birth and death times of all the CC in the image.

First, preprocessing is done to store all the intensity values in I along with their (x, y) location in decreasing order in a sorted data structure. This allows for faster identification of the next threshold value as well as the pixels that get newly included in the super-level set. Second, we utilize the standard Union-Find data structure Tarjan (1975), which maintains a collection of disjoint sets. This data structure supports the operation `FIND((x,y))`, which finds the highest-value representative of the CC containing the pixel (x, y). It also supports the operation `UNITE((x,y), (v,w))`, which unites the CCs represented by root pixels (x, y) and (v, w) and, assuming $I(x, y) > I(v, w)$, making (x, y) the representative of the merged components.

At each threshold value u, we use the sorted data structure to identify which pixels are included in the set $\mathcal{S}(u)$. `FIND` is called on all these newly added pixels to determine if they belong to any existing CC; if not, they are considered roots of a new CC, with a new entry to the list of disjoint sets in the union-find data structure. Furthermore, this results in the creation of a dot $(u, -\infty)$ in the persistence diagram whose birth time is u, and a default death time of $-\infty$. Then, `UNITE` is called for every pair of roots in the disjoint set list, to determine if any of the sets are in fact merged. Consider if sets with roots (x, y) and (v, w) are merged, with $I(x, y) > I(v, w)$. In that case, CC with root (v, w) has now 'died', and so the persistence dot whose birth time was $I(v, w)$ in the diagram, will now have an updated death time of u. Hence, $(I(v, w), u)$ will replace the old dot $(I(v, w), -\infty)$ in the persistence diagram.

We show a walkthrough of this in Fig. 14. We start with an empty union-find data structure. In (a), we start with $u = 5$, as 5 is the maximum value in the image. We call `FIND` and update the union-find data structure with a new entry. Additionally, the persistence diagram (PD) in (e) has an entry for $(5, -\infty)$. This CC is denoted as α_1. Next, in (b) $u = 4$, all the blue and red pixels are newly added to this filtration. `FIND` is called on each of them. The blue pixels indicate that they were found to be a part of an existing CC. The red pixels indicate that no parent CC was found, and hence are considered as creating new CCs. The PD includes two new dots $\alpha_2(4, -\infty)$ and $\alpha_3(4, -\infty)$. Next, as intensity value 3 is not present in I, we do not need to compute $\mathcal{S}(3)$. The sorted data structure directly leads us to the next u value which is $u = 2$ in (c). `FIND` is called on all the blue pixels. `UNITE` is called on all pairs of root CC nodes, resulting `True` for the fact that components α_1 and α_2 have now merged. Since the birth time of α_1 is larger, the component α_2 'dies' and becomes a part of α_1. Thus, the persistent dot is updated from $\alpha_2(4, -\infty)$ to $\alpha_2(4, 2)$ as it was born at $u = 4$ has died at $u = 2$. Similarly in (d), at $u = 1$, `FIND` is called on all the blue pixels. Then `UNITE` $(\alpha_1,\ \alpha_3)$ is called, showing that they are actually merged. Thus, the persistent dot is updated from $\alpha_3(4, -\infty)$ to $\alpha_3(4, 1)$ as α_3 was born at $u = 4$ and has now died at $u = 1$. Thus, the final persistent dots we end up with are $\alpha_1(5, -\infty)$, $\alpha_2(4, 2)$, $\alpha_3(4, 1)$ as shown in (e).

For 1-dimensional and higher features (e.g., loops and voids), while we do not go into details here, the Cubical Ripser library constructs a sparse coboundary matrix that encodes relationships between pixels. This matrix is reduced to upper-triangular form using column operations, optimizing the identification of the birth and death of cycles. Each non-zero pivot in the reduced matrix corresponds to a topological feature. This computational framework ensures the efficient generation of persistence diagrams, leveraging union-find for 0-dim and matrix operations for higher dimensions. More details are available in the Cubical Ripser Kaji et al. (2020) paper.



```
FIND((2,2))
```

```
FIND((1,5))
FIND((1,2))
FIND((2,1))
FIND((2,3))
FIND((3,2))
FIND((4,4))
FIND((4,5))
```

```
FIND((0,4))
FIND((0,5))
...
FIND((5,5))
UNITE((2,2),(1,5))
   > True
UNITE((2,2),(4,4))
   > False
```

```
FIND((0,1))
FIND((0,3))
...
FIND((5,2))
UNITE((2,2),(4,4))
   > True
```

0	1	0	1	2	2
1	2	4	2	2	4
2	4	5	4	2	2
1	2	4	1	1	1
0	0	1	1	4	4
-1	0	1	2	2	2

0	1	0	1	2	2
1	2	4	2	2	4
2	4	5	4	2	2
1	2	4	1	1	1
0	0	1	1	4	4
-1	0	1	2	2	2

0	1	0	1	2	2
1	2	4	2	2	4
2	4	5	4	2	2
1	2	4	1	1	1
0	0	1	1	4	4
-1	0	1	2	2	2

0	1	0	1	2	2
1	2	4	2	2	4
2	4	5	4	2	2
1	2	4	1	1	1
0	0	1	1	4	4
-1	0	1	2	2	2

Figure 14: Figure inspired from Kaji et al. (2020). The figure shows a sample 2D image, thresholded at different values. (a)-(d) shows the sequence of super-level sets $\mathcal{S}(u)$, which is nothing but thresholding the image at the value u. It shows how different connected components (CC) are created or destroyed as u changes. White-background cells indicate pixels included in the set $\mathcal{S}(u)$. Red indicates root pixels causing the birth of a CC. Blue indicates pixels that are newly added to the set $\mathcal{S}(u)$ but are absorbed in existing CCs. Brown indicates root pixels whose CC has died, having merged with another CC whose root had a higher birth time. (e) shows the corresponding 0-dim persistence diagram.

K Quality of Generated Masks

We provide FID scores across all the datasets in Tab. 8. Since our method TDN focuses on generating masks, we report the FID of the mask [FID (Mask)] and the FID of the images [FID (Image)] generated by ControlNet Zhang et al. (2023) when using these masks as condition.

Our analysis shows that the quality of the masks generated by TDN closely resembles the true masks, that is, the ground truth (GT) masks annotated by humans, as evidenced by the consistently low FID (Mask) scores across all datasets. Additionally, to evaluate the impact on final image quality, we compare the FID (Image) metric between two scenarios: ControlNet using GT masks (ControlNet + GT) versus ControlNet using TDN-generated masks. The results demonstrate comparable FID scores, indicating that using TDN-generated masks as conditioning does not degrade the image quality. In other words, TDN-generated masks are as good as GT masks for generating real images.

Results from Tab. 1 and Tab. 8 highlight that TDN achieves improved topological control while maintaining the overall visual quality of the generated images.

Table 8: **Comparison of FID (Mask) and FID (Image) against baselines across all datasets.** For Shapes and Google Maps datasets, we cannot report FID (Image) as there is no ground truth image dataset to compare to. For COCO and CREMI, we report FID (Image) against the respective dataset images. ControlNet + GT indicates that ControlNet is using the GT masks as the condition. ControlNet + FT indicates that it has been fine-tuned on the CREMI dataset to generate similarly textured images. **Bold** denotes the best results, while *italics* denotes the second best. Note that ControlNet + GT has been included as the upper bound performance; realistically, it cannot be used during inference as the GT masks are not available

Dataset	Method	FID (Mask) ↓	FID (Image) ↓
Shapes	ADM-T	0.092	-
	TDN (Ours)	**0.068**	-
COCO (Animals)	Stable Diffusion	-	29.41
	DALL·E 3	-	**17.49**
	AR	-	35.05
	ADM-T	0.267	21.72
	TDN (Ours)	**0.222**	*21.28*
	ControlNet + GT	-	20.94
Google Maps	ADM-T	0.198	-
	TDN (Ours)	**0.156**	-
CREMI	Stable Diffusion	-	48.18
	DALL·E 3	-	54.72
	AR	-	69.86
	ADM-T	0.518	3.322
	TDN (Ours)	**0.467**	**3.286**
	ControlNet + GT + FT	-	3.126

L Using 0-dim and 1-dim Simultaneous Topological Constraints

In our work, we focused on using either 0-dim (number of objects) or 1-dim (number of holes) constraints individually, depending on the property of the dataset. In this section, we conduct experiments on synthetic data using both 0-dim and 1-dim constraints simultaneously. The constraint is of the form (a, b) where a denotes 0-dim while b denotes 1-dim. Both are first separately encoded and then added together to be fed as a condition. Sample masks generated by TDN along with ControlNet-rendered images are shown in Fig. 15. We report quantitative results in Tab. 9.

We find that TDN is capable of handling these joint constraints. While the performance is slightly lower compared to using each constraint individually, TDN still significantly outperforms ADM-T across all metrics. This shows that persistent homology (PH) can enhance the performance of the base diffusion model, even under multi-constraint scenarios. This finding opens up interesting possibilities for future work in handling even richer combinations of topological constraints.

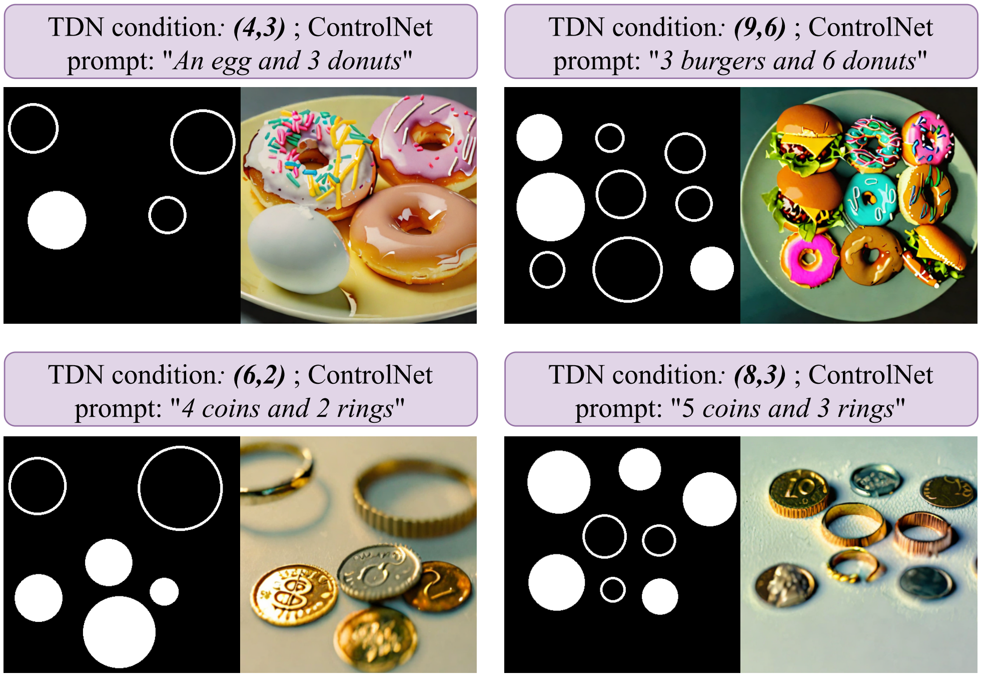

TDN condition: ***(4,3)*** ; ControlNet prompt: "*An egg and 3 donuts*"
TDN condition: ***(9,6)*** ; ControlNet prompt: "*3 burgers and 6 donuts*"
TDN condition: ***(6,2)*** ; ControlNet prompt: "*4 coins and 2 rings*"
TDN condition: ***(8,3)*** ; ControlNet prompt: "5 *coins and 3 rings*"

Figure 15: Masks generated by TDN using both 0-dim and 1-dim topological constraints simultaneously. Condition (a, b) denotes a for 0-dim (#objects) and b for 1-dim (#holes). Images are generated via ControlNet by using mask as a condition.

Table 9: Quantitative performance of using both 0-dim and 1-dim topological constraints simultaneously. The FID (Mask) is reported once for each method. Best results are highlighted in **bold**

TopoDim	**Method**	**Accuracy** ↑	**Precision** ↑	**F1** ↑	**FID (Mask)** ↓
0-dim	ADM-T	0.7383 ± 0.1305	0.7997 ± 0.1268	0.7677 ± 0.1229	0.1279
	TDN (Ours)	**0.9183 ± 0.0731**	**0.9338 ± 0.0993**	**0.9261 ± 0.0906**	**0.0982**
1-dim	ADM-T	0.7616 ± 0.0905	0.7892 ± 0.1129	0.7752 ± 0.1082	-
	TDN (Ours)	**0.9233 ± 0.0705**	**0.9492 ± 0.0913**	**0.9360 ± 0.0720**	-

L.1 Generating Specific Topology like Double Annulus

As mentioned above, we represent the joint constraints as pairs (a, b), where a specifies the number of connected components (0-dim) and b specifies the number of holes (1-dim). A double annulus consists of concentric circles, with $(a, b) = (2, 2)$. Our loss, based on homology, cannot distinguish whether two loops are positioned concentric or not, however, it can generate structures that are homotopy equivalent to the double annulus. As shown in Fig. 16, by sampling for the $(2, 2)$ condition, TDN can generate nested circles (visually the closest constraint to a double annulus) and other homotopy equivalent structures. This experiment demonstrates that our method can handle complex topological specifications through the combination of constraints, though geometric relationships (like equidistance and concentricity) remain an interesting direction for future work.

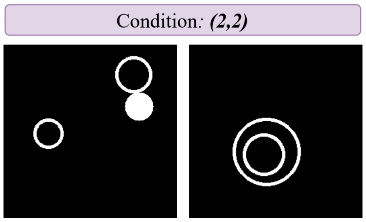


Figure 16: Sample images generated by TDN when using both 0-dim and 1-dim topological constraints simultaneously. With constraints $(a, b) = (2, 2)$, we can generate nested circles which are the closest to achieving a double annulus (two concentric circles) using our method.

M Comparison against CVAE

In this section, we justify the motivation behind using diffusion models instead of other generative models like Conditional Variational Autoencoders (CVAE) Sohn et al. (2015). We conduct experiments using CVAE as the generative model for the same task of generating masks. Although we trained on multiple image resolutions, CVAE performed poorly on larger image sizes. Hence we report results on 64×64 images, using a latent embedding dimension of 128. We show results in Tab. 10 and Fig. 17 for both standard CVAE and a version incorporating our proposed loss $\mathcal{L}_{\text{top}}$.

Our experimental results show significant limitations of the CVAE approach. From Fig. 17, we see that the quality of the masks generated by CVAE, with and without $\mathcal{L}_{\text{top}}$, is severely degraded. From Tab. 10, it has a significantly higher FID compared to our TDN masks, indicating a lower resemblance to the true masks. Additionally, CVAE also struggles to preserve topological constraints, as evidenced by the low performance on evaluation metrics like Accuracy and F1. While CVAE $+\mathcal{L}_{\text{top}}$ slightly improves performance, it remains far weaker than diffusion models. Specifically, in the 0-dim, CVAE generates fragmented objects and fails to preserve their overall shapes. In 1-dim, it fails to generate connected structures. Although mask images seem easy because they are binary images, they are in fact a challenge to generate. This is because of the difference in the number of objects, their varied shapes and sizes, and unconstrained spatial locations (eg: not fixed to the center of the image). Similarly, the diverse orientation and connection patterns make the 1-dim datasets challenging. Such complexities make it difficult for the CVAE to generalize to these datasets.

Using diffusion models is critical to the success of TDN. First, the base diffusion model ADM-T effectively captures spatial arrangements and object shapes (in 0-dim), and connectivity patterns in 1-dim. However, it struggles with the number of objects/holes. This is where our persistent homology loss brings big improvements. Preserving the number of structures requires heavy global reasoning and detail. In contrast, CVAE's bottleneck layer compresses information, losing significant information in the image space. This limits their ability to preserve object shape and enforce strict topological constraints. The significant gain of our loss in diffusion models is owing to the gradual denoising steps, as well as intact image resolution. These experiments highlight the necessity of diffusion models, making them critical to the success of our method in topological control.

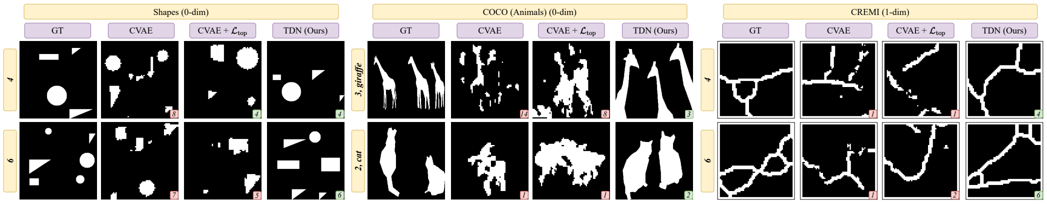


Figure 17: Qualitative comparison of using CVAE and diffusion models as the generative model. Ground truth (GT) denotes the true masks used to train the models

Table 10: Quantitative comparison of using CVAE and diffusion models as the generative model. We include ADM-T and TDN results from Tab. 1 for convenience. Best results are in **bold**

Dataset (TopoDim)	Method	Accuracy ↑	Precision ↑	F1 ↑	FID (Mask) ↓
Shapes (0-dim)	CVAE Sohn et al. (2015)	0.3133 ± 0.1321	0.3235 ± 0.1984	0.3078 ± 0.1443	2.231
	CVAE $+\mathcal{L}_{\text{top}}$	0.3816 ± 0.0919	0.3444 ± 0.1276	0.3220 ± 0.1365	1.981
	ADM-T	0.7500 ± 0.1889	0.7809 ± 0.1582	0.7651 ± 0.1210	0.092
	TDN (Ours)	**0.9478 ± 0.0420**	**0.9499 ± 0.0492**	**0.9488 ± 0.0370**	**0.068**
COCO (0-dim)	CVAE Sohn et al. (2015)	0.2442 ± 0.0410	0.2696 ± 0.0285	0.2562 ± 0.0318	4.208
	CVAE $+\mathcal{L}_{\text{top}}$	0.3094 ± 0.0967	0.3342 ± 0.0729	0.3213 ± 0.1165	4.083
	ADM-T	0.6685 ± 0.1485	0.6917 ± 0.1079	0.6799 ± 0.1931	0.267
	TDN (Ours)	**0.8557 ± 0.0805**	**0.8670 ± 0.0636**	**0.8613 ± 0.0970**	**0.222**
CREMI (1-dim)	CVAE Sohn et al. (2015)	0.2785 ± 0.4296	0.2267 ± 0.1156	0.2499 ± 0.1391	5.971
	CVAE $+\mathcal{L}_{\text{top}}$	0.3267 ± 0.2834	0.3091 ± 0.1079	0.3176 ± 0.1447	5.751
	ADM-T	0.5357 ± 0.1879	0.4777 ± 0.1797	0.4881 ± 0.1571	0.518
	TDN (Ours)	**0.7785 ± 0.1901**	**0.8142 ± 0.1925**	**0.7959 ± 0.1659**	**0.467**

N Comparison against Basic Masks

To justify the importance of our approach, we conduct experiments using simple shapes (circles and squares) as conditioning masks for ControlNet Zhang et al. (2023). Quantitative results are shown in Tab. 11, and qualitative examples are provided in Fig. 18.

Our experiments reveal several limitations when using simplified shapes as conditioning masks, as they fail to provide sufficient spatial and structural guidance. This setup is comparable to AR Phung et al. (2024) (shown in Tab. 1 and Fig. 6 of the main paper), which uses rectangular bounding boxes with an additional attention step. From our observations, we detect multiple failure modes. First, from Fig. 18, we see that such conditions tend to introduce visual artifacts in the generated images. The images are often fragmented, with objects isolated within assigned areas, leading to a divided and visually disjointed image. Second, the basic shapes also fail to constrain the number of objects within them, often resulting in 0 objects or multiple objects per shape (see the 'cats' row). Finally, for large animals like zebras, ControlNet struggles to complete the image beyond the boundary of the shape—while the zebra texture is present within the shape, the overall image is not as desired. Similarly, for even larger animals like giraffes, the discrepancy between the basic shape and the natural object proportions leads to incomplete or distorted generations.

In contrast, the masks generated by TDN provide tighter spatial and structural control by offering sufficient detail to guide the network effectively. This results in correct object counts and better visual quality, as evidenced by superior FID scores and count metrics. TDN's masks strike a balance between oversimplification and unnecessary complexity, as generative models are powerful enough to fill in finer textural details.

Our results highlight that relying solely on a text prompt and simplified masks is insufficient for the model to generate visually coherent images. Instead, meaningful masks like those generated by TDN are essential for ensuring reliable control over cardinality and topology in generative models.

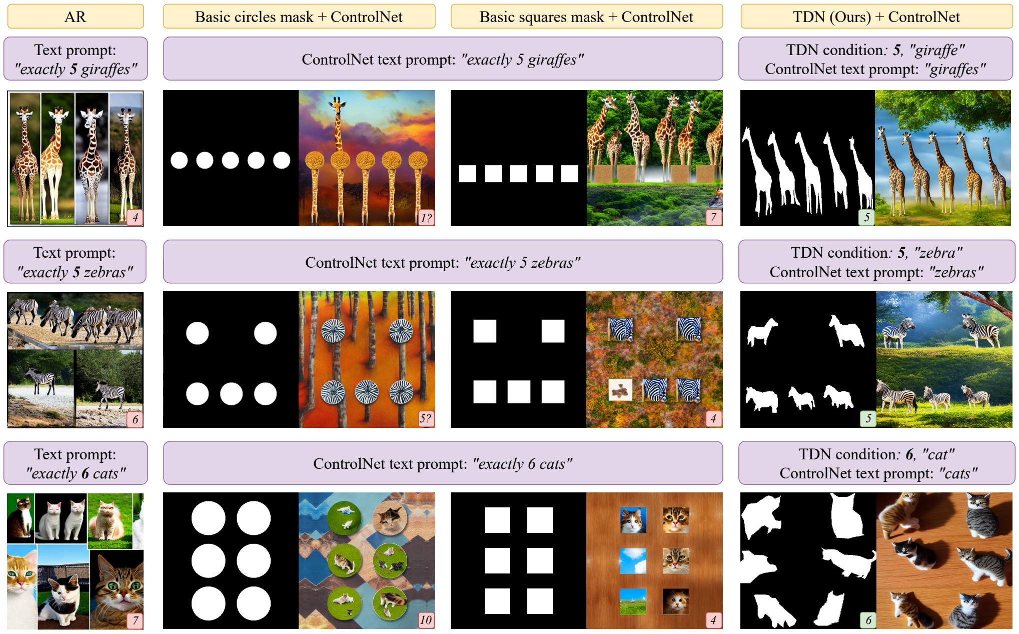

AR
Basic circles mask + ControlNet
Basic squares mask + ControlNet
TDN (Ours) + ControlNet
Text prompt: "exactly 5 giraffes"
ControlNet text prompt: "exactly 5 giraffes"
TDN condition: 5, "giraffe"
ControlNet text prompt: "giraffes"
4
1?
7
5
Text prompt: "exactly 5 zebras"
ControlNet text prompt: "exactly 5 zebras"
TDN condition: 5, "zebra"
ControlNet text prompt: "zebras"
6
5?
4
5
Text prompt: "exactly 6 cats"
ControlNet text prompt: "exactly 6 cats"
TDN condition: 6, "cat"
ControlNet text prompt: "cats"
7
10
4
6

Figure 18: Qualitative comparison of using different conditioning types, particularly masks containing basic shapes like circles/squares to guide cardinality. ControlNet uses the SD1.5 backbone (Lvmin Zhang).

Table 11: Quantitative comparison of using different conditioning types for the COCO dataset. We include AR (Phung et al., 2024) and TDN results from Tab. 1 for convenience. Best results in **bold**

Method / Mask condition type	Accuracy ↑	Precision ↑	F1 ↑	FID (Image) ↓
Basic circles mask + ControlNet	0.2972 ± 0.3592	0.3194 ± 0.3046	0.3079 ± 0.2208	49.47
Basic squares mask + ControlNet	0.3123 ± 0.2114	0.3459 ± 0.2209	0.3282 ± 0.2618	47.16
AR (Bounding box w/ attention)	0.6379 ± 0.2062	0.7360 ± 0.1658	0.6611 ± 0.1851	35.05
TDN (Ours)	**0.8557 ± 0.0805**	**0.8670 ± 0.0636**	**0.8613 ± 0.0970**	**21.28**

N.1 Using SDXL-Turbo Backbone for ControlNet

The ControlNet results generated in this paper have used the SD1.5 backbone (Lvmin Zhang). In this sub-section, we generate ControlNet results using the newer SDXL-Turbo backbone[8] (Sauer et al., 2025). The results are in Fig. 19, Fig. 20, and Fig. 21.

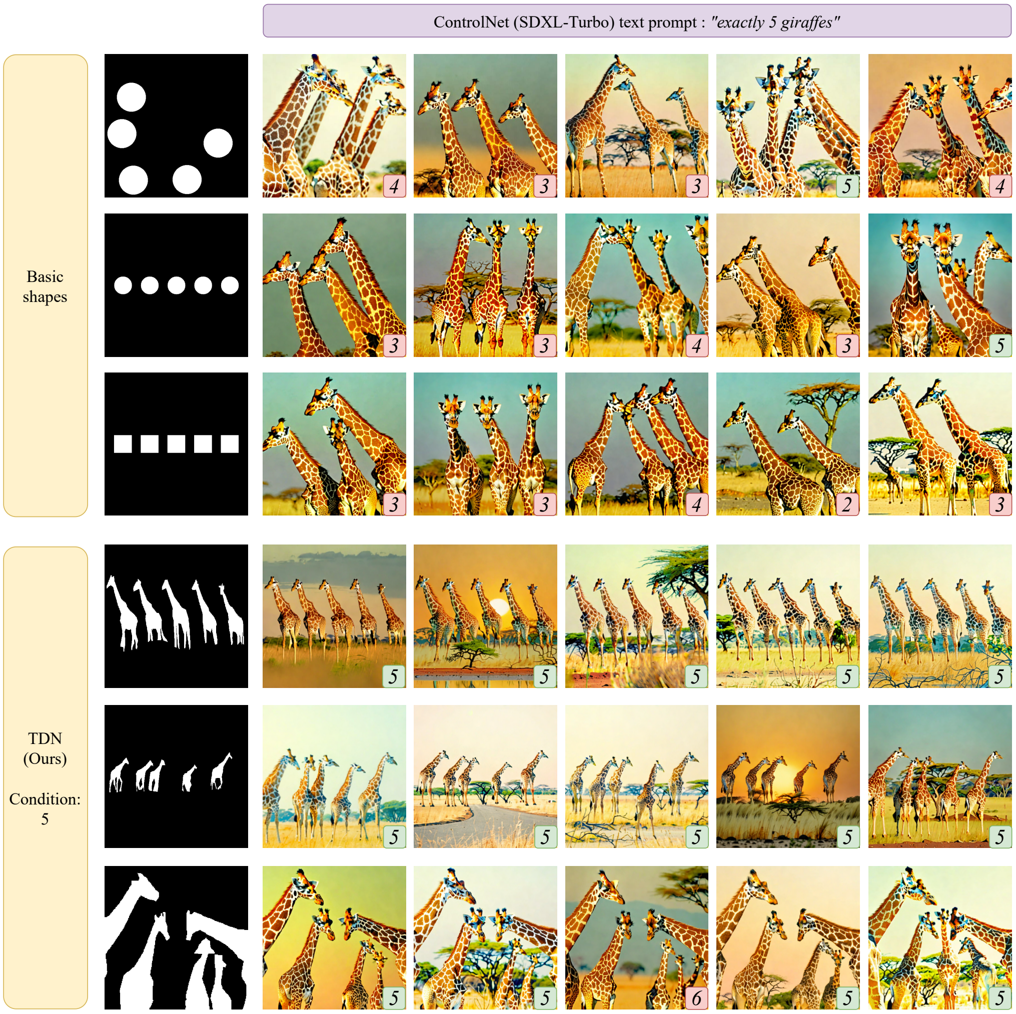


Figure 19: Generating exactly 5 giraffes. Qualitative comparison of using different conditioning types, such as masks containing basic shapes like circles/squares to guide cardinality. ControlNet uses the SDXL-Turbo backbone (Sauer et al., 2025).

[8] `https://huggingface.co/stabilityai/sdxl-turbo`

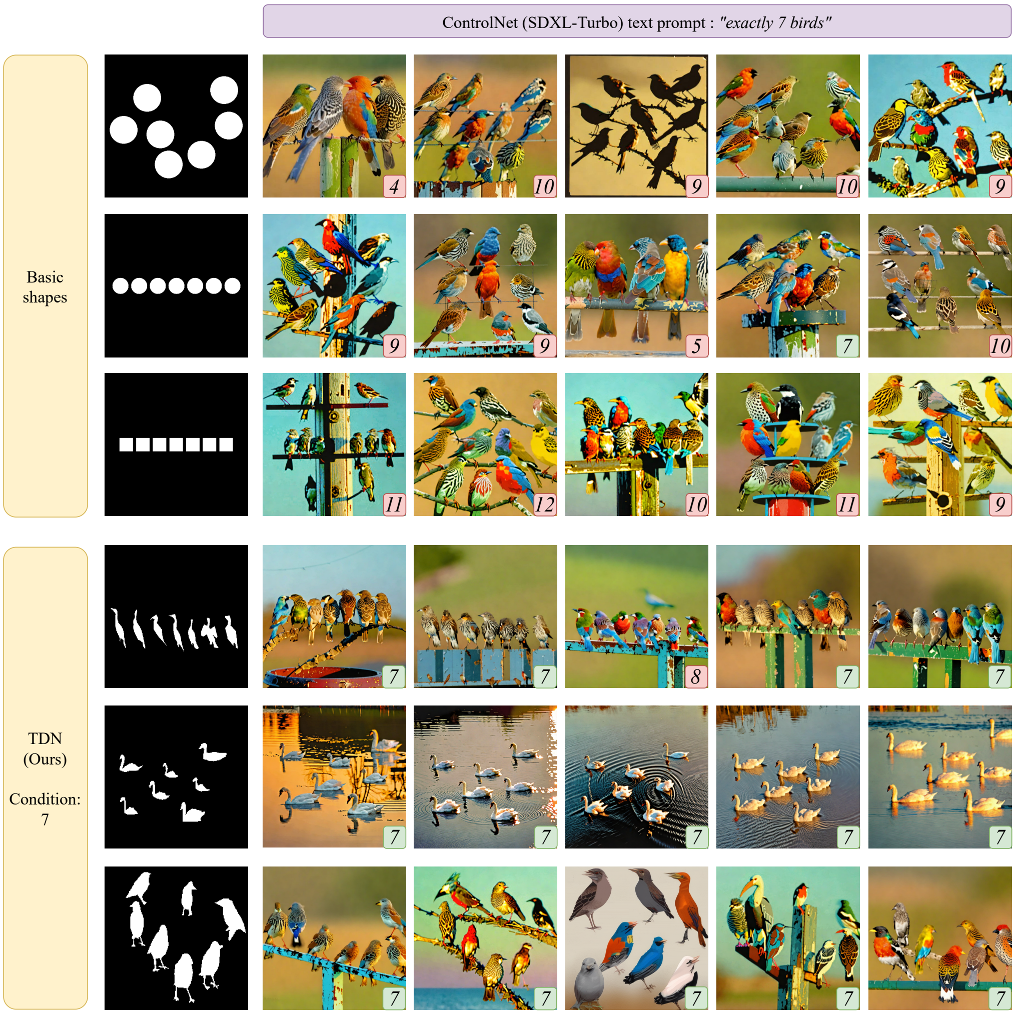


Figure 20: Generating exactly 7 birds. Qualitative comparison of using different conditioning types, such as masks containing basic shapes like circles/squares to guide cardinality. ControlNet uses the SDXL-Turbo backbone (Sauer et al., 2025).

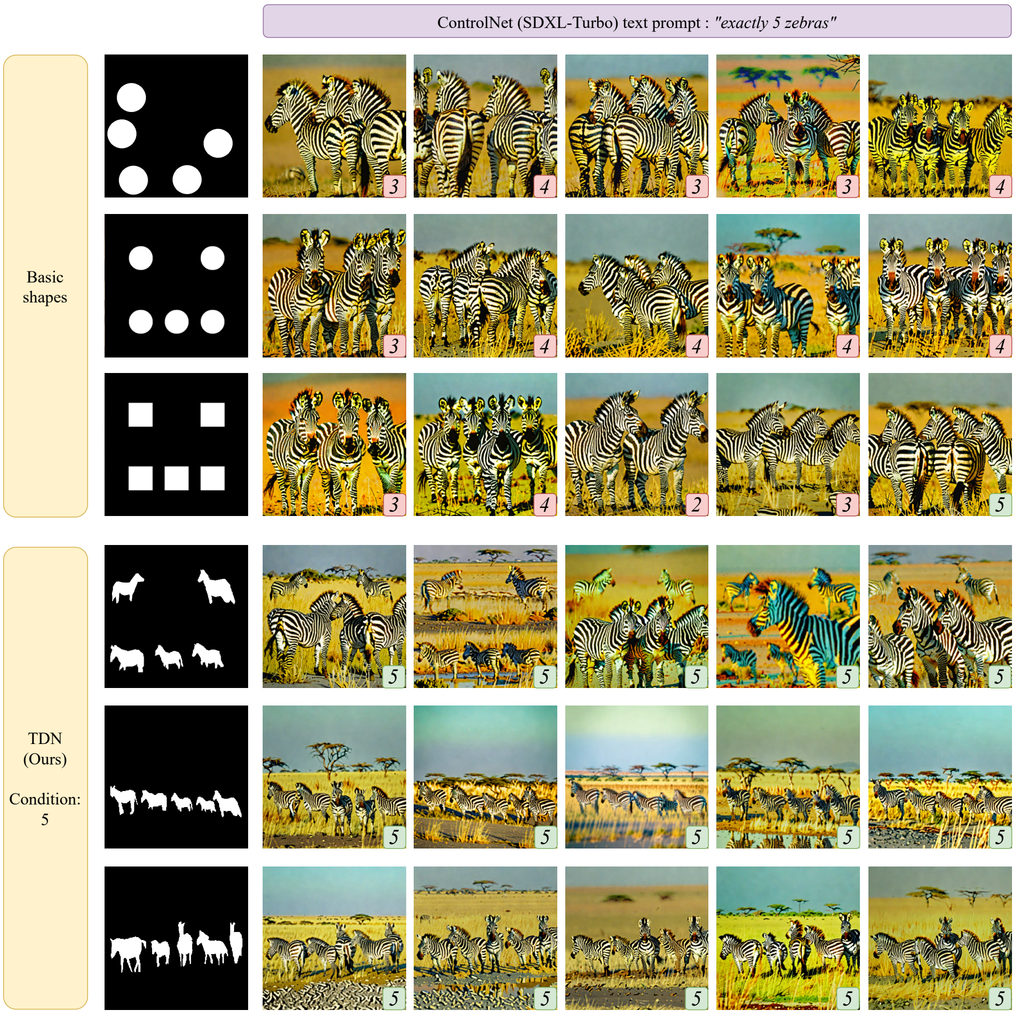


Figure 21: Generating exactly 5 zebras. Qualitative comparison of using different conditioning types, such as masks containing basic shapes like circles/squares to guide cardinality. ControlNet uses the SDXL-Turbo backbone (Sauer et al., 2025).