TOPODIFFUSIONNET: A TOPOLOGY-AWARE DIFFUSION MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models excel at creating visually impressive images but often struggle to generate images with a specified *topology*. The Betti number, which represents the number of structures in an image, is a fundamental measure in topology. Yet, diffusion models fail to satisfy even this basic constraint. This limitation restricts their utility in applications requiring exact control, like robotics and environmental modeling. To address this, we propose TopoDiffusionNet (TDN), a novel approach that enforces diffusion models to maintain the desired topology. We leverage tools from topological data analysis, particularly persistent homology, to extract the topological structures within an image. We then design a topology-based objective function to guide the denoising process, preserving intended structures while suppressing noisy ones. Our experiments across four datasets demonstrate significant improvements in topological accuracy. TDN is the first to integrate topology with diffusion models, opening new avenues of research in this area.

1 INTRODUCTION

Over the past few years, diffusion models have become prominent for image generation tasks (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Song et al., 2020a;b; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021). Naturally, text-to-image (T2I) diffusion models are a popular choice for creating high-quality images based on textual prompts (Saharia et al., 2022a; Rombach et al., 2022; Avrahami et al., 2022; Ruiz et al., 2023; Nichol et al., 2021; Kim et al., 2022; Ramesh et al., 2021; Midjourney; OpenAI, a). Despite their ability to generate visually impressive images, T2I models are still far from the desired intelligence level. In particular, they often struggle to interpret textual prompts that involve basic reasoning and logic. This includes preserving global and semantic constraints, such as consistent number of objects as well as structural patterns (e.g., enclosed regions or loops). Improving this capability would be a step forward in the control and precision of diffusion models, moving beyond qualitative attributes such as style and texture.

Topology, in a general sense, defines how different parts of an image interact with each other, dictating their overall layout within an image. Preserving topology is essential for generating images that not only look realistic but also adhere to correct semantics. The simplest measure in topology is the Betti number, which is equivalent to the number of connected components (0-dimension) and holes/loops (1-dimension). In natural images, 0-dimensional topology corresponds to the number of distinct objects, while 1-dimensional topology refers to the number of enclosed regions. Yet, current diffusion models fail to preserve even these basic topological properties. This is particularly evident in applications such as urban planning, robotics, and environmental modeling, where it is crucial to generate scenes with a specific topology or number of entities (like animals or road intersections). Fig. 1(a-b) shows instances where popular T2I diffusion models fail to generate images with the specified topology, such as specific numbers of animals, or holes/regions in road layouts.

Recognizing the limited spatial reasoning of T2I models, existing methods use *spatial maps* (such as object masks, edge maps, etc) to control the generated images (Zhang et al., 2023; Bar-Tal et al., 2023; Bashkirova et al., 2023; Huang et al., 2023; Mou et al., 2023; Li et al., 2023). These methods have indeed shown promise, offering a more guided approach to image generation that aligns closely with the provided controls. Nonetheless, generating the spatial map itself is a bottleneck: an automatic way to generate spatial maps with a desired topology is an unaddressed problem.

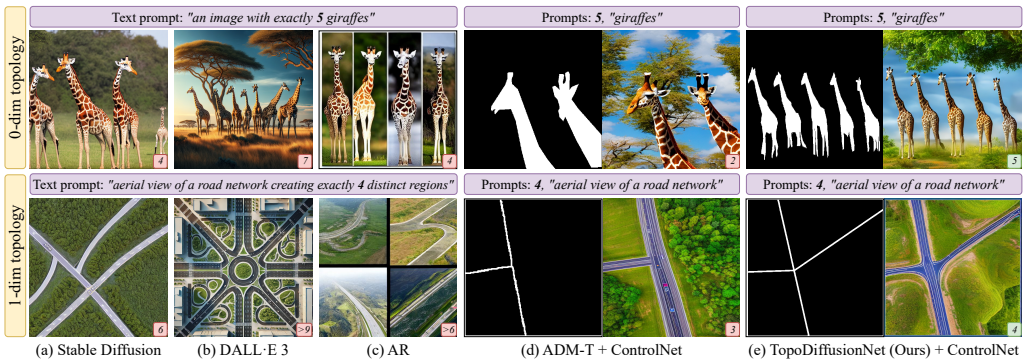

Figure 1: **Comparison of existing diffusion models in preserving topological constraints.** Top row: 0-dim topological constraint to generate exactly five giraffes. Bottom row: 1-dim topological constraint to generate a road layout with exactly four distinct regions. Text-to-image methods like (a) Stable Diffusion (Rombach et al., 2022) and (b) DALL·E 3 (OpenAI, a) struggle to respect both 0-dim and 1-dim constraints. (c) Attention Refocusing (AR) (Phung et al., 2024) requires bounding boxes for each object but struggles with higher object counts and often creates partitioned images. (d)-(e) shows a two-step process: mask generation followed by ControlNet (Zhang et al., 2023) rendering. (d) ADM-T generates masks by fine-tuning ADM (Dhariwal & Nichol, 2021) with the topological constraint as a condition, but this alone is insufficient. (e) Our TopoDiffusionNet, trained with a topology-based objective function, generates masks with the precise number of objects or regions, which when fed to ControlNet generates the desired image of five giraffes (top row) and four regions (bottom row). Giraffe/region counts are noted in the bottom-right inset of each image.

In this work, we address the challenge of generating topologically faithful images using diffusion models. Our focus is on generating images with a specific *topology*, defined by properties such as the number of connected components (0-dimension) or loops/holes (1-dimension). These topological structures, quantified by Betti numbers, serve as our topological constraints. Spatial maps such as masks have shown success in guiding the semantics of the generated image. Leveraging this, we propose using diffusion models to automate generating masks that satisfy the desired topological constraint. A straightforward approach is to condition the diffusion model on the constraint information and fine-tune it on enough samples. However, as shown in Fig. 1(d), we find that conditioning on the constraint alone falls short of effectively preserving the topology of the generated masks.

We propose TopoDiffusionNet (TDN), a novel approach that incorporates topology to guide the mask generation process. To ensure the final mask satisfies the topological constraint, a topology-aware objective function is necessary for steering the denoising process so that each timestep is one step closer to preserving the desired topological constraint. However, designing such a function is not straightforward. The intermediate timesteps are very noisy – especially at larger timesteps – so extracting meaningful information from them is challenging. We thus need tools that are robust to noise. This leads us to *persistent homology* (Edelsbrunner et al., 2002; Edelsbrunner & Harer, 2010), a mathematical theory that can, amidst the noise, extract the topological structures within an image. Using persistent homology, we can partition an image in terms of topology: separating out the significant structures from the noisy ones. We can thus design a dedicated objective function to preserve the significant structures and suppress the rest. The function guides the denoising process to progress in such a way so as to ultimately preserve the topology at the final timestep, as we see in Fig. 1(e). In summary, our contributions are as follows:

- To the best of our knowledge, we are the first to integrate topology with diffusion models to address topologically faithful image generation in both 0-dimension and 1-dimension. Specifically, we generate spatial maps (masks) to tackle the challenge of generating an image with a specific number of structures.
- We present TopoDiffusionNet (TDN), which utilizes a topology-based objective function to improve diffusion models' ability to follow simple topological constraints. It serves as a denoising loss, guiding the diffusion denoising process in a topology-aware manner.

- We evaluate TDN on four datasets to demonstrate its versatility and effectiveness. TDN exhibits large improvements in maintaining the topological integrity of the generated image.

The success of our method suggests a surprising harmony between diffusion models and topology. Diffusion models are trained to denoise but are rather hard to control for preserving global semantics. Meanwhile, topological methods such as persistent homology provide a principled solution to extract global structural information from a noisy input, and can thus successfully guide the diffusion model. We hope the coupling of diffusion models and topology, as well as the techniques developed in this paper, will shed light on more sophisticated control of these generative models in the near future.

2 Related Work

Diffusion models. Diffusion models, first introduced by Sohl-Dickstein et al. (2015), are now prevalent in image generation (Ho et al., 2020; Song et al., 2020a;b; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021), evolving from unconditional models, to conditional models using class labels (Ho & Salimans, 2022; Dhariwal & Nichol, 2021), and later to text-to-image models (Saharia et al., 2022a; Rombach et al., 2022; Avrahami et al., 2022; Ruiz et al., 2023; Nichol et al., 2021; Kim et al., 2022; Ramesh et al., 2021; Midjourney; OpenAI, a). However, textual prompts have limitations in controlling spatial composition, like layouts and poses. Recognizing this, several works propose to use spatial maps (such as masks, edge maps, etc) as condition to guide the image generation process (Zhang et al., 2023; Qin et al., 2023; Zhao et al., 2024; Bar-Tal et al., 2023; Bashkirova et al., 2023; Huang et al., 2023; Mou et al., 2023). These approaches aim to overcome the limitations of text-based conditioning by providing more explicit spatial guidance.

Numeric control in diffusion models. Betti numbers quantify topological structures, such as the number of connected components (0-dimension) or loops/holes (1-dimension), which is conceptually related to the task of counting. Recent works have explored approaches to enhance the counting performance in diffusion models. Paiss et al. (2023) enhances CLIP's (Radford et al., 2021) text embeddings for counting-aware text-to-image (T2I) generation via Imagen (Saharia et al., 2022b). While this results in some improvement, the performance is still limited, supporting our motivation that T2I models often struggle with textual prompts involving semantic reasoning. Layout-based methods (Chen et al., 2024; Phung et al., 2024; Farshad et al., 2023) use layout maps, that is, maps containing bounding boxes of each object/entity, to guide the reverse diffusion process. While these methods show promise, they are not scalable as their complexity increases with the number of objects. Furthermore, focusing on these boxes often results in images that appear partitioned, as shown in Fig. 1(c) top row, where the backgrounds of each giraffe differ. Finally, all of the methods mentioned above are limited to 0-dimensional topological structures (i.e., discrete objects) and do not extend to higher-dimensional topological constraints.

Deep learning with topology. Methods from algebraic topology, under the name of *topological data analysis* (TDA) (Carlsson, 2009), have found use in various machine learning problems owing to their versatility (handling data such as images, time-series, graphs, etc.) and robustness to noise. The most widely-used tool from TDA, *persistent homology* (PH) (Edelsbrunner et al., 2002; Edelsbrunner & Harer, 2010), has been applied to several image classification (Peng et al., 2024; Wang et al., 2021; Du et al., 2022; Hofer et al., 2017; Chen et al., 2019) and segmentation tasks (Abousamra et al., 2021; Clough et al., 2019; Hu et al., 2019; Clough et al., 2020; Stucki et al., 2023; Byrne et al., 2022; He et al., 2023) as it can track topological changes at multiple intensity values. Other TDA theories like discrete Morse theory (Dey et al., 2019; Hu et al., 2021; 2023; Gupta et al., 2024; Banerjee et al., 2020), topological interactions (Gupta et al., 2022), and center-line transforms (Shit et al., 2021; Wang et al., 2022a), have also enhanced performance in these areas.

In the realm of generative models, TDA has been used with generative adversarial networks (GANs) (Goodfellow et al., 2014) to evaluate performance through topology comparisons (Khrulkov & Oseledets, 2018), and enhance image quality via topological priors (Brüel-Gabrielsson et al., 2019) and PH-based loss functions (Wang et al., 2020). These methods have mainly focused on quality enhancement without directly controlling the topological characteristics of the images. In the case of diffusion models, incorporating TDA has not yet been explored. Our work, TopoDiffusionNet, represents a unique effort in this direction, employing PH within diffusion models to control the topology of the generated images. This is a significant shift from existing applications of TDA in generative models, moving beyond quality improvement to precise topological control.

3 METHODOLOGY

Given a topological constraint c, our goal is to generate a *mask* containing c number of *structures*. A structure can either correspond to an object or a hole/region. We illustrate these structures in Fig. 2. In (a), the four objects are in white. In (b), the four holes correspond to the four black regions/partitions the white lines create with the border. In the formal language of algebraic topology (Munkres, 2018), c is the Betti number, that is, it is the rank of the homology group, in which objects (connected components) correspond to the 0-dimensional (0-dim) homology classes and holes/loops/regions correspond to the 1-dimensional (1-dim) homology classes.

(a) Four objects (b) Four holes

Figure 2: Illustration of topological structures.

To enforce the topological constraint c, the diffusion model needs to be conditioned on it. However, c alone is not sufficient to control the topology of the final generated image[1]. Therefore, we propose a topology-based objective function $\mathcal{L}_{\text{top}}$ to guide the reverse denoising diffusion process at each timestep during training. $\mathcal{L}_{\text{top}}$ uses persistent homology (Edelsbrunner et al., 2002; Edelsbrunner & Harer, 2010) to distinguish between the desired and the spurious structures, aiming to enhance the former and reduce the latter to ensure the topology matches c closely. Fig. 3 provides an overview.

The rest of this section is organized as follows. We briefly discuss diffusion models in Sec. 3.1, followed by a quick background on persistent homology in Sec. 3.2. We tie these concepts together to finally introduce our method TopoDiffusionNet (TDN) in Sec. 3.3.

3.1 DIFFUSION MODELS

Diffusion models (Ho et al., 2020) are able to sample images from the training data distribution $p(x_0)$ by iteratively denoising random Gaussian noise in T timesteps. The framework consists of a forward and a reverse process.

In the forward process, at every timestep $t \in T$, Gaussian noise is added to the clean image $x_0 \sim p(x_0)$ until the image becomes an isotropic Gaussian. The forward noising process is denoted by $q(x_t \mid x_0) \coloneqq \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\,\boldsymbol{I}\right)$, which can be rewritten as,

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{(1-\bar{\alpha}_t)}\epsilon \tag{1}$$

$\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is a noise variable, $\bar{\alpha}_t$ is the noise scale at timestep t, and $\mathcal{N}$ is the normal distribution.

The reverse process aims to learn the posterior distribution $q\left(x_{t-1} \mid x_t, x_0\right)$, using which we can recover x_{t-1} given x_t. This is typically done by training a denoising neural network (U-Net (Ronneberger et al., 2015)) with network parameters θ. The denoising model $\epsilon_\theta(x_t, t)$ takes the noisy input x_t at timestep t and predicts the noise ϵ added in Eq. (1) of the forward process. The model is trained using the simplified objective function $\mathcal{L}_{\text{simple}}$ (Ho et al., 2020): $\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon_\theta(x_t,t) - \epsilon\|_2^2\right]$. In our case, we additionally provide the topological constraint c as a condition to control the topology of the generated image. Thus, the denoising model becomes $\epsilon_\theta(x_t, c, t)$, where c is injected into the denoising neural network.

During training, although the denoising model predicts the noise $\epsilon_\theta(x_t, c, t)$ at timestep t, we can deterministically (without iterative sampling) recover the predicted noiseless image $\hat{x}_0^t$ (an estimate of the true x_0) from Eq. (1) as,

$$\hat{x}_0^t = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t, c, t)\right) \tag{2}$$

This alternate form of the prediction will enable us to compute $\mathcal{L}_{\text{top}}$ as we will see in Sec. 3.3.

3.2 PERSISTENT HOMOLOGY

Persistent homology (PH) (Edelsbrunner et al., 2002; Edelsbrunner & Harer, 2010), owing to its differentiable nature, is an attractive candidate for integrating topological information into the training of deep learning methods. In the case of image data, it can detect the changes in topological structures (connected components and holes) across a varying threshold (also called the filtration value). More importantly, persistent homology is robust to noise, that is, it can extract these structures even in noisy scenarios. Structures that exist for a wide range of thresholds are *salient*, while the remaining structures are deemed as *noise* in the image.

[1] In this section, we use 'image' to mean the mask.

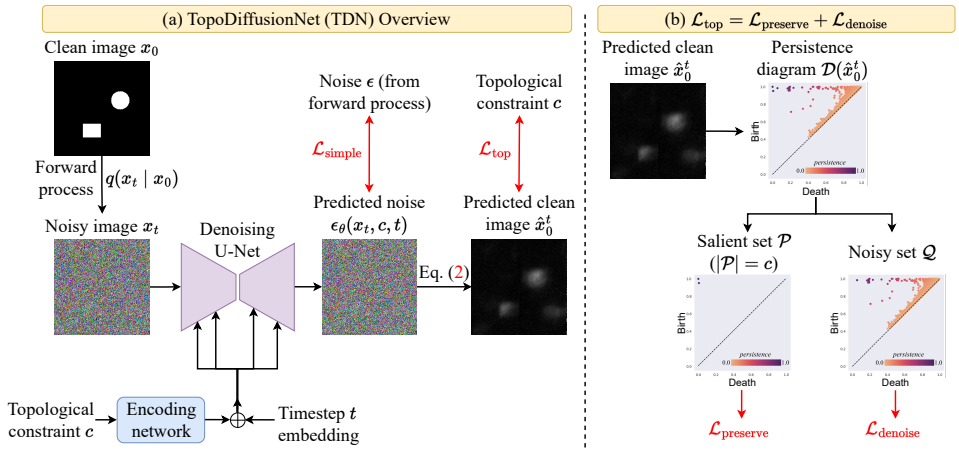

Figure 3: (a) TDN overview: We condition the diffusion model on the topological constraint c (here $c = 2$). During training, we first add noise ϵ to the input x_0 using the forward process (Eq. (1)) to obtain x_t, where t is sampled uniformly. The U-Net is trained as part of the reverse process, predicting the added noise $\epsilon_\theta(x_t, c, t)$, with which we obtain the noiseless image $\hat{x}_0^t$ (Eq. (2)). Alongside the standard loss $\mathcal{L}_{\text{simple}}$, we propose $\mathcal{L}_{\text{top}}$ to enforce the topological integrity of $\hat{x}_0^t$. (b) To compute $\mathcal{L}_{\text{top}}$, the persistence diagram $\mathcal{D}(\hat{x}_0^t)$ captures all the topological structures in $\hat{x}_0^t$, partitioning them into salient/desired structures $\mathcal{P}$ and noisy ones $\mathcal{Q}$. Terms $\mathcal{L}_{\text{preserve}}$ and $\mathcal{L}_{\text{denoise}}$ respectively amplify $\mathcal{P}$ and suppress $\mathcal{Q}$, guiding the training to eventually satisfy c.

In our setting, during training, we can apply persistent homology to every intermediate image $\hat{x}_0^t$ (from Eq. (2)) predicted by the diffusion model at timestep t. For ease of reference, we denote $\hat{x}_0^t$ by I, having size $h \times w$. In practice, I has continuous probability values in a normalized range, say, $[0, 1]$[2]. We now consider *super-level sets* of I, i.e., the set of pixels (i, j) for which I_{ij} is above some threshold value u. Let $\mathcal{S}$ denote the super-level set, then, $\mathcal{S}(u) \coloneqq \{(i, j) \in [1, h] \times [1, w] \mid I_{ij} \geq u\}$. This is nothing but thresholding, and we call the resulting binary image the super-level set at u, $S(u)$. Decreasing u continuously generates a sequence of sets, i.e. a filtration, which grows as the threshold parameter u is brought down: $\varnothing \subseteq \mathcal{S}(1) \subseteq \mathcal{S}(u_1) \subseteq \mathcal{S}(u_2) \subseteq \cdots \subseteq \mathcal{S}(0) = [1, h] \times [1, w]$. We demonstrate this filtration in Fig. 4.

When u is high, only a few pixels can exceed the threshold, and hence the size of $\mathcal{S}(1)$ is small (an almost black image). As u decreases, new pixels join the set, and topological structures in $\mathcal{S}$ are created and destroyed. Eventually, at $u = 0$, the entire image is in the super-level set. In this manner, persistent homology can track the evolution of all the topological structures.

The output of the persistent homology algorithm includes the *birth* and *death* threshold values for each topological structure. We can keep track of the birth (creation) b and death (destruction) d thresholds of all the topological structures and put the tuples (b, d) in a diagram – the *persistence diagram* $\mathcal{D}$ – where the y-axis represents birth and the x-axis death.[3] The persistence diagram is thus a graphical representation of topological structures throughout the filtration process, consisting of multiple dots in a 2-dimensional plane (see Fig. 4). These dots are called persistent dots, where each dot corresponds to one topological structure. The *persistence*, or the lifetime of a structure, is given by the difference between its death and birth times. Structures that persist over a wide range of thresholds are considered significant or salient, indicating stable and prominent structures within the image, while short-lived structures are the noise. The diagonal $b = d$ represents structures of zero persistence and dots far from the diagonal represent salient structures with high persistence.

With this information, we can identify noisy structures due to their low persistence and proximity to the diagonal, allowing us to filter them out and retain the salient topological structures. The number of salient structures we retain is precisely the number of structures we desire in the final image. As we show in the next subsection, we compute the persistence diagram of the predicted image $\hat{x}_0^t$ to optimize it from a topological perspective.

[2]The range is typically $[-1, 1]$ in implementation.

[3]If using a sub-level instead of super-level set filtration, the diagram would have the x-axis as birth, and the y-axis as death.

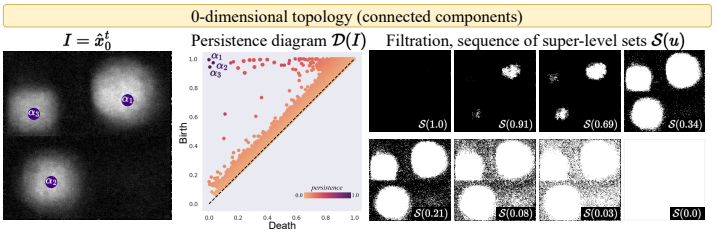

Figure 4: Illustration of persistent homology and persistence diagrams of 0-dim topological structures (connected components). Despite the noise, we can visually see three prominent structures $\alpha_1, \alpha_2, \alpha_3$ in I. In the topological space, $\alpha_1, \alpha_2, \alpha_3$ thus appear in the top-left corner of the persistence diagram $\mathcal{D}(I)$, persisting through most of the filtration $\mathcal{S}$. All the remaining connected components 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. The equivalent illustration for 1-dim topological structures (holes) is in Appendix A.

3.3 Proposed TopoDiffusionNet (TDN)

Conditioning. We condition the diffusion model on c to enable it to generate a mask containing exactly c number of structures. We follow Nichol & Dhariwal (2021) to inject the condition information. We first obtain an embedding of c from a trainable network composed of a few linear layers. Next, we inject the embedding through the same pathway as the timestep embedding of t. Consequently, both embeddings are passed to residual blocks throughout the denoising model.

Objective function $\mathcal{L}_{\text{top}}$. Conditioning alone is not sufficient to control the topology of the generated image. To address this, we introduce $\mathcal{L}_{\text{top}}$, a topology-based objective function to force the predicted image at every timestep to preserve c as closely as possible. Since the diffusion model is parameterized to predict in the noise space, directly analyzing topology from this noise is not meaningful. We need to map the prediction from the noise space back to the image space in order to infer the topology. From Eq. (2), we obtain $\hat{x}_0^t$ – an estimate of the noiseless image x_0 at timestep t – from $\epsilon_\theta(x_t, c, t)$. The estimate $\hat{x}_0^t$ is noisy, especially when t is large, making it ideal to use the theory of persistent homology to separate out its salient structures from the noisy ones.

Given the prediction $\hat{x}_0^t$, we compute its persistence diagram $\mathcal{D}(\hat{x}_0^t)$ containing either 0-dim or 1-dim information based on the desired topological structure (object or regions). Recall that we desire c topological structures in the predicted image. For a persistent dot $p \in \mathcal{D}$, with birth b_p and death d_p, its persistence value, $|b_p - d_p|$, measures its significance, according to the theory. We rank all dots in $\mathcal{D}$ by their persistence values in descending order. The top c dots are the structures we aim to preserve, reflecting our desired topology, whereas the rest denote noisy structures to be suppressed/denoised. Thus, we decompose the diagram $\mathcal{D}$ into two disjoint sets, $\mathcal{D} = \mathcal{P} \dot{\bigcup} \mathcal{Q}$, where $\mathcal{P}$ contains the c largest persistence dots ($|\mathcal{P}| = c$), while $\mathcal{Q}$ contains all the remaining dots.

To constrain $\hat{x}_0^t$ to have c structures in the *image space*, in the *topological space* we need to maximize the persistence or saliency of the dots $p \in \mathcal{P}$, and suppress all the noisy dots $p \in \mathcal{Q}$. To achieve this, we introduce two loss terms:

$$\mathcal{L}_{\text{preserve}} = -\sum_{p \in \mathcal{P}} |b_p - d_p|^2 \quad \text{and} \quad \mathcal{L}_{\text{denoise}} = \sum_{p \in \mathcal{Q}} |b_p - d_p|^2 \tag{3}$$

$$\mathcal{L}_{\text{top}} = \mathcal{L}_{\text{preserve}} + \mathcal{L}_{\text{denoise}} \tag{4}$$

Minimizing $\mathcal{L}_{\text{top}}$ is equivalent to maximizing the saliency of the top c structures (via $\mathcal{L}_{\text{preserve}}$) and suppressing the saliency of the rest (via $\mathcal{L}_{\text{denoise}}$). In the ideal case, $\mathcal{L}_{\text{top}} = -c$, as each of the top c structures will have a persistence of 1, while all the noisy structures will have zero persistence. The image $\hat{x}_0^t$ will then have exactly c topological structures as desired. In the absence of $\mathcal{L}_{\text{top}}$, if the denoising process were to originally result in $> c$ structures, now $\mathcal{L}_{\text{denoise}}$ will suppress all the extra/noisy structures, preventing them from appearing in the final clean image. One concern is whether there could be less than c structures in total. In practice, at large timesteps t, $\hat{x}_0^t$ always has several thousand noisy topological structures. Thus, if the denoising process were to originally proceed with $< c$ structures, $\mathcal{L}_{\text{preserve}}$ will now increase the persistence of a less salient dot to ensure the final image has exactly c structures.

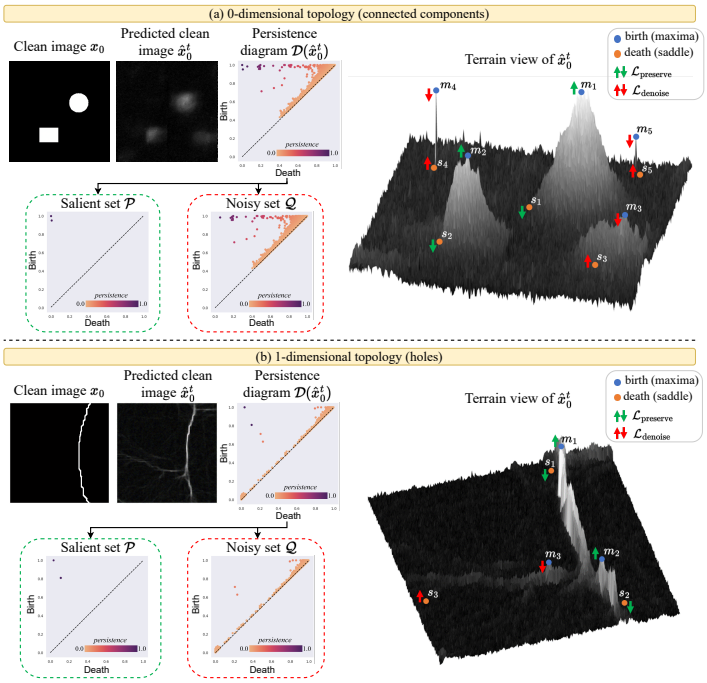

Figure 5: Illustration of $\mathcal{L}_{\text{preserve}}$ and $\mathcal{L}_{\text{denoise}}$ for 0-dim connected components and 1-dim holes, with $c = 2$ as seen in x_0. (a) After computing $\mathcal{D}(\hat{x}_0^t)$, we partition it into sets $\mathcal{P}$ (the top c structures) and $\mathcal{Q}$ (remaining 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).

Implementation and differentiability. For every topological structure, the birth and death values b and d correspond to function values of a maximum-saddle pair; b and d are function values of a local maximum m and a saddle point s, respectively. These pairs can be determined by the almost linear union-find algorithm (Edelsbrunner & Harer, 2010; Ni et al., 2017) which locates and pairs all local maxima and saddle points to reflect the birth and death of topological structures. For every persistent dot $p \in \mathcal{D}$, let m_p and s_p respectively denote the 2D coordinates of the corresponding local maximum and saddle point in the prediction $\hat{x}_0^t$. Then, Eq. (3) and Eq. (4) can be rewritten as,

$$\mathcal{L}_{\text{top}} = -\sum_{p \in \mathcal{P}} |\hat{x}_0^t(m_p) - \hat{x}_0^t(s_p)|^2 + \sum_{p \in \mathcal{Q}} |\hat{x}_0^t(m_p) - \hat{x}_0^t(s_p)|^2 \tag{5}$$

We illustrate this in Fig. 5. During training, for every topological structure to preserve, i.e., $p \in \mathcal{P}$, the function $\mathcal{L}_{\text{preserve}}$ increases the intensity value $\hat{x}_0^t(m_p)$ at the local maximum m_p and decreases $\hat{x}_0^t(s_p)$ at the saddle point s_p. This strengthens the saliency of the desired structures. At the same time, to prevent exceeding c structures in the final image, $\mathcal{L}_{\text{denoise}}$ suppresses structures $p \in \mathcal{Q}$ by reducing the intensity value $\hat{x}_0^t(m_p)$ at the local maximum m_p while increasing $\hat{x}_0^t(s_p)$ at the saddle point s_p. $\mathcal{L}_{\text{denoise}}$ forces $\hat{x}_0^t(m_p)$ to be equal to $\hat{x}_0^t(s_p)$, leading to a homogenous region. This effectively eliminates the noisy structure, as it was neither born nor died, rendering it non-existent.

With this, we see that $\mathcal{L}_{\text{top}}$ is differentiable, as Eq. (5) is written as polynomials of the prediction $\hat{x}_0^t$ at certain pixels. From Eq. (5) we can compute the gradient of $\mathcal{L}_{\text{top}}$ with respect to $\hat{x}_0^t$, and via chain rule, we can ultimately compute the gradient with respect to the denoising model's parameters θ. The training optimization adjusts θ to ensure that the topological space, i.e. the persistence diagram $\mathcal{D}(\hat{x}_0^t)$, has exactly c persistent dots, in turn resulting in c structures in the image space $\hat{x}_0^t$.

End-to-end training. The overall training objective $\mathcal{L}_{\text{total}}$ of TDN is formulated as: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{simple}} + \lambda \mathcal{L}_{\text{top}}$ where λ is the loss weight. The standard denoising loss $\mathcal{L}_{\text{simple}}$ produces visually good results, whereas $\mathcal{L}_{\text{top}}$ helps respect the topological constraint c.

4 EXPERIMENTS

Datasets. We train ADM-T and TDN on four datasets: **Shapes**, **COCO** (Caesar et al., 2018), **CREMI** (Funke et al., 2016), and **Google Maps** (Isola et al., 2017). The Shapes dataset is a synthetic dataset created by us that contains objects such as circles, triangles, and/or rectangles. For CREMI, COCO, and Google Maps, we train the diffusion models on their segmentation masks. For COCO, we select masks which contain at least one instance of the super category 'animal'. For COCO and Shapes, we use 0-dim, the number of connected components, as the topological constraint. For COCO, we also use the animal class as a condition to generate masks of specific animals. CREMI is an Electron Microscopy dataset, and Google Maps contains aerial photos from New York City. For CREMI and Google Maps, we use 1-dim, the number of holes, as the topological constraint. Each of the datasets contains masks consisting of up to ten structures. More details are in Appendix B.

Baselines. Stable Diffusion (Rombach et al., 2022) and DALL·E 3 (OpenAI, a) are popular T2I diffusion models. Attention Refocusing (AR) (Phung et al., 2024) uses layout maps (bounding boxes for each object) generated by GPT-4 (Achiam et al., 2023) to guide the reverse process.

Implementation details. Our work extends the ADM (Dhariwal & Nichol, 2021) diffusion model. We use 'ADM-T' to denote the modification of using a topological constraint as a condition. We obtain an embedding of the constraint using an encoding network. Following the approach in Nichol & Dhariwal (2021), we then feed this embedding to all the residual blocks in the network by adding it to the timestep embedding. For COCO, we similarly inject animal class embedding to further control the generated mask. For every dataset, we use 256×256 as the image resolution. Our diffusion models use a cosine noise scheduler (Nichol & Dhariwal, 2021), with $T = 1000$ timesteps for training. During inference, however, we use only 50 steps of DDIM (Song et al., 2020a) sampling. For the ADM-T baseline, we load a pretrained checkpoint (OpenAI, b) and then fine-tune on our datasets using the constraint information as condition. For TDN, we follow the same approach but additionally use $\mathcal{L}_{\text{top}}$ in the training. More details are listed in Appendix C.

Evaluation metrics. To evaluate whether the generated image satisfies the input constraint, we use metrics such as Accuracy, Precision, and F1. We report the mean and standard deviation of the results across different constraint values. To measure the performance, for 0-dim, we generate 50 samples per constraint $c \in [1, 10]$ per animal/shape category (resulting in 5K images for COCO). We similarly generate 50 images per constraint $c \in [1, 10]$ for the 1-dim datasets. We perform the unpaired t-test (Student, 1908) (95% confidence interval) to determine the statistical significance of the improvement. In all the tables, performances that are statistically significantly better are in **bold**.

4.1 RESULTS

Qualitative and quantitative results for 0-dim. In Fig. 6 and Tab. 1, we present qualitative and quantitative comparisons, respectively, of pretrained Stable Diffusion, DALL·E 3, AR, ADM-T, and our proposed TDN. In Appendix D, we provide constraint-wise results. As ADM-T and TDN produce masks, we employ pretrained ControlNet (Zhang et al., 2023; Lvmin Zhang) to create textured images from these masks. In practice, any of the methods (Qin et al., 2023; Zhao et al., 2024; Bar-Tal et al., 2023; Bashkirova et al., 2023; Huang et al., 2023; Mou et al., 2023) could also be used for this purpose, but we chose ControlNet for its simplicity. From Fig. 6, we see that T2I models Stable Diffusion and DALL·E 3 are unable to respect the explicit topological constraint of generating c objects. Due to limited spatial and semantic reasoning, both methods have the lowest performance in Tab. 1. AR uses layouts generated by GPT-4, which improves over T2I models, as seen in Tab. 1. In general, however, GPT-4 does not have much spatial awareness and when the count is high, the bounding boxes tend to either be too small or highly overlap with each other. This leads to incorrect counts in the generated image as seen in Fig. 6. Additionally, when the object count is high, AR often produces fragmented results, with objects isolated within assigned areas, leading to a divided and visually disjointed image. ADM-T, despite being conditioned on the constraint c, also falls short of satisfying the constraint. This indicates that the constraint alone is not powerful enough to influence the global reasoning of the model. In contrast, TDN preserves the constraint better than ADM-T, as evident from the quantitative results in Tab. 1. TDN demonstrates significant improvements across all metrics. $\mathcal{L}_{\text{preserve}}$ aims to retain at least c objects, while $\mathcal{L}_{\text{denoise}}$ aims to maintain atmost c objects. Thus optimizing both simultaneously via training with $\mathcal{L}_{\text{top}}$ helps the diffusion model to preserve c objects in the generated mask.

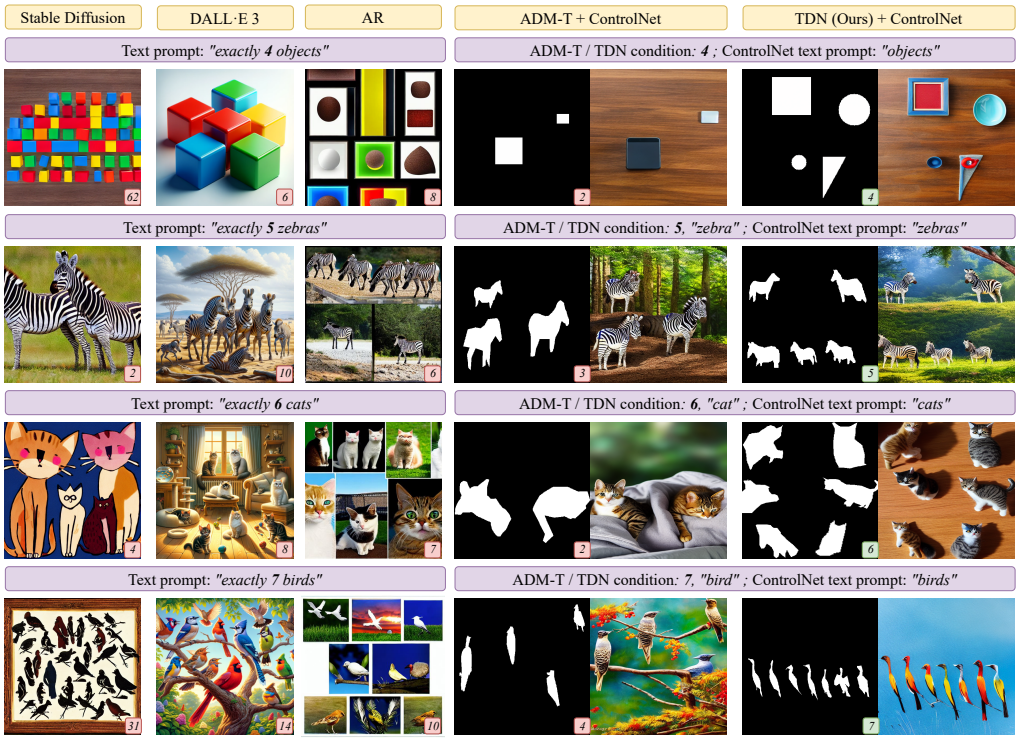

Figure 6: Qualitative results for 0-dim topological constraint. Row 1: Shapes dataset; Rows 2-4: COCO dataset. ADM-T and TDN take the constraint as the condition (purple box), and also the animal class for COCO. Stable Diffusion, DALL·E 3, and AR take the equivalent text prompt as input. Object/animal counts are noted in the bottom-right inset of each image/mask.

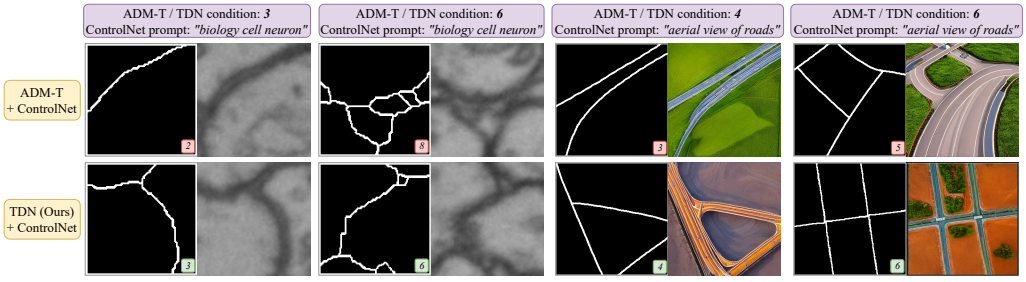

Figure 7: Qualitative results for the 1-dim topological constraint. ADM-T and TDN take the constraint (in the purple box) as the condition. Columns 1-2: CREMI. Columns 3-4: Google Maps. Number of holes within each mask is noted in its bottom-right inset.

Qualitative and quantitative results for 1-dim. In Fig. 7, we provide the qualitative comparison for ADM-T and TDN. We exclude results from Stable Diffusion, DALL·E 3, and AR as these methods are limited to generating distinct objects (0-dim topology) and currently cannot handle generating distinct holes/regions (1-dim topology). As holes are a complex topological constraint, they are challenging to describe in words, and hence the performance of such methods is limited (see Fig. 1 and Appendix E). Similar to the 0-dim case, ADM-T struggles with the topological constraint. The quantitative results in Tab. 1 highlight that preserving 1-dim topology, which involves boundaries spanning across the mask, is more challenging than 0-dim topology. This complexity necessitates powerful global reasoning capabilities from the diffusion model, an area where ADM-T shows limited performance. However, with $\mathcal{L}_{\text{top}}$, TDN achieves substantial improvements across all metrics, with $\mathcal{L}_{\text{preserve}}$ and $\mathcal{L}_{\text{denoise}}$ both working to retain exactly c holes in the generated mask.

Additional comparison. Paiss et al. (2023) enhances CLIP's text embeddings for counting and uses them to show counting-aware T2I generation via Imagen. We report the Accuracy of our TDN results in their setting in Tab. 2, showing significant performance improvement. This supports our motivation that T2I models often struggle with textual prompts involving semantic reasoning.

Table 1: Quantitative comparison on preserving the topological constraint c

Dataset	Method	Accuracy ↑	Precision ↑	F1 ↑
Shapes	Stable Diffusion (Runway)	0.6381 ± 0.2559	0.6660 ± 0.1759	0.6537 ± 0.2198
	DALL·E 3 (OpenAI, a)	0.6857 ± 0.2561	0.7059 ± 0.3198	0.6956 ± 0.2649
	AR (Phung et al., 2024)	0.7384 ± 0.2178	0.7596 ± 0.2039	0.7474 ± 0.2165
	ADM-T	0.7500 ± 0.1889	0.7809 ± 0.1582	0.7651 ± 0.1210
	TDN (Ours)	**0.9478 ± 0.0420**	**0.9499 ± 0.0492**	**0.9488 ± 0.0370**
COCO (Animals)	Stable Diffusion (Runway)	0.4686 ± 0.2361	0.5154 ± 0.1747	0.4909 ± 0.1976
	DALL·E 3 (OpenAI, a)	0.5162 ± 0.3674	0.5491 ± 0.2724	0.5194 ± 0.3256
	AR (Phung et al., 2024)	0.6379 ± 0.2062	0.7360 ± 0.1658	0.6611 ± 0.1851
	ADM-T	0.6685 ± 0.1485	0.6917 ± 0.1079	0.6799 ± 0.1931
	TDN (Ours)	**0.8557 ± 0.0805**	**0.8670 ± 0.0636**	**0.8613 ± 0.0970**
Google Maps	ADM-T	0.5494 ± 0.1386	0.5642 ± 0.1861	0.5567 ± 0.1185
	TDN (Ours)	**0.8318 ± 0.1159**	**0.8471 ± 0.1797**	**0.8394 ± 0.1969**
CREMI	ADM-T	0.5357 ± 0.1879	0.4777 ± 0.1797	0.4881 ± 0.1571
	TDN (Ours)	**0.7785 ± 0.1901**	**0.8142 ± 0.1925**	**0.7959 ± 0.1659**

Table 2: Additional baseline

Method	Accuracy ↑
Paiss et al. (2023)	0.5018
TDN (Ours)	**0.7969**

Table 3: Ablation on loss terms

$\mathcal{L}_{\text{preserve}}$	$\mathcal{L}_{\text{denoise}}$	Accuracy ↑
✗	✗	0.7500 ± 0.1889
✓	✗	0.8926 ± 0.1821
✗	✓	0.9186 ± 0.1129
✓	✓	**0.9478 ± 0.0420**

Table 4: Ablation on λ

Loss weight λ	Accuracy ↑
0	0.7500 ± 0.1889
1e-3	0.9066 ± 0.0612
1e-5	**0.9478 ± 0.0420**
1e-7	0.9176 ± 0.0407
Min-SNR (5)	0.9286 ± 0.0320

Effect across timesteps. While we use 50 steps of DDIM (Song et al., 2020a) sampling for inference, the training was conducted with $T = 1000$ timesteps using DDPM (Ho et al., 2020). To understand the effect of $\mathcal{L}_{\text{top}}$ throughout training, we analyze intermediate results from the DDPM inference procedure. In Fig. 8, we visualize $\hat{x}_0^t$ across different timesteps on the Shapes dataset, highlighting the impact of $\mathcal{L}_{\text{top}}$ on denoising efficiency. Notably, TDN achieves a closer approximation to the true x_0 by timestep 750, nearly 200 timesteps before ADM-T. Furthermore, we plot the average Fréchet Inception Distance (FID) (Heusel et al., 2017) for 1000 samples per timestep. The plot shows that $\mathcal{L}_{\text{top}}$ accelerates denoising and enhances the quality of $\hat{x}_0^t$ at earlier stages, working as intended. This demonstrates that $\mathcal{L}_{\text{top}}$ improves both speed and accuracy of the denoising process.

4.2 Ablation Studies

To demonstrate the efficacy of TDN, we conduct ablation studies on the loss components and the effect of hyperparameter value changes. Appendix G includes an ablation study on the 'Encoding Network' for the topological constraint c. All analyses are on 0-dim Shapes dataset.

Ablation study on $\mathcal{L}_{\text{preserve}}$ and $\mathcal{L}_{\text{denoise}}$. In Tab. 3, we show the contribution of $\mathcal{L}_{\text{denoise}}$ and $\mathcal{L}_{\text{preserve}}$ in meeting the topological constraint. While both terms individually improve performance, $\mathcal{L}_{\text{denoise}}$ has a more significant effect. This is because $\hat{x}_0^t$, especially at larger timesteps, has several spurious structures. $\mathcal{L}_{\text{denoise}}$ works by suppressing the birth of these structures, leaving only the desired number of structures in the mask. Meanwhile, $\mathcal{L}_{\text{preserve}}$ is crucial when the model generates fewer structures than expected. Naturally, combining both terms yields the optimal performance.

Ablation study on loss weight λ. In Tab. 4, we show experiments with different weights for $\mathcal{L}_{\text{top}}$, including the Min-SNR ($\gamma = 5$) (Hang et al., 2023) strategy where the loss weight is a function of timestep t. When $\lambda = 1e-5$, TDN achieves the best performance. Nonetheless, a reasonable range of λ always results in improvement, demonstrating the efficacy and robustness of $\mathcal{L}_{\text{top}}$.

5 Conclusion

We propose TopoDiffusionNet, the first method to integrate topology with diffusion models. Our approach generates images that preserve topology by producing masks with a specified number of structures (Betti number). Empirical results show significant improvement in preserving this topological constraint, demonstrating that our method guides the denoising process in a topology-aware manner. This paves the way for further research on topological control in image generation.

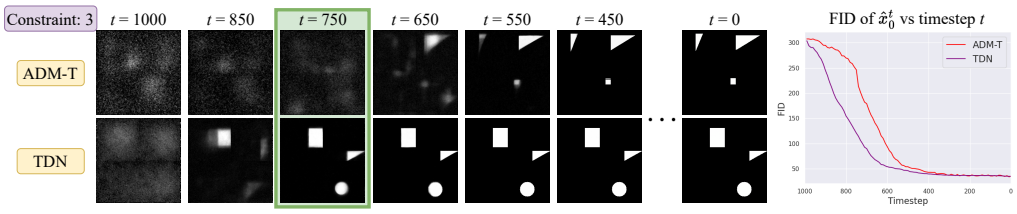

Figure 8: TDN has better FID of $\hat{x}_0^t$ at larger timesteps compared to ADM-T.

Reproducibility Statement. We provide experimental details regarding the datasets, baselines, evaluation metrics, and implementation in Sec. 4. Additional details on the dataset are provided in Appendix B. In Appendix C, we provide additional details about the baselines, the implementation of our method, and the computation resources used.

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.

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.

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 additional details of the datasets.

Appendix C provides additional baseline and implementation details.

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

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

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

Appendix H provides a discussion on the limitations of our method.

A Persistent Homology

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

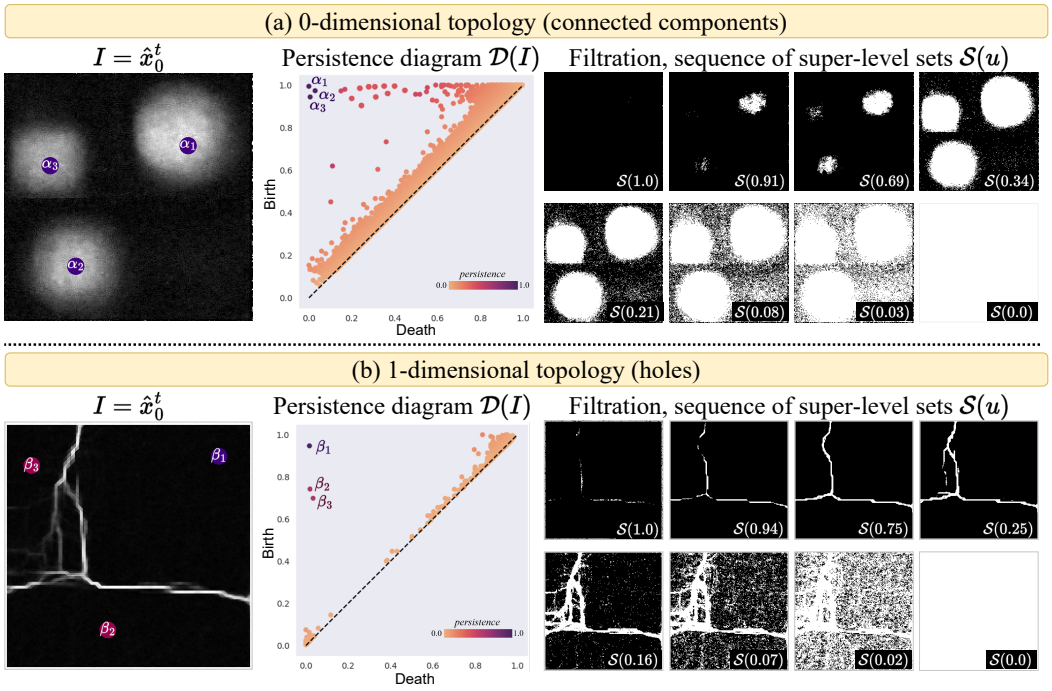

Figure 9: Illustration of persistent homology and persistence diagrams of both types of topological structures, 0-dim connected components and 1-dim holes. (a) Despite the noise, we can visually see three prominent structures $\alpha_1, \alpha_2, \alpha_3$ in I. In the topological space, $\alpha_1, \alpha_2, \alpha_3$ 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 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

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

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.

C 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. We will publicly release the code upon acceptance of the paper.

In Fig. 1, Fig. 6, Fig. 7, Fig. 11, and Fig. 12, 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. 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.

D Constraint-wise Results

In Fig. 10, 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.

E Additional 1-dim Qualitative Results

In Fig. 11, 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.

[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

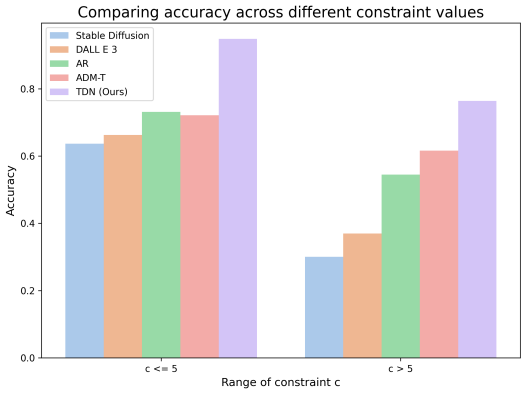

Figure 10: Accuracy results on low and high object counts (COCO dataset).

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.

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.

Table 6: Standalone holes

Method	Accuracy ↑	F1 ↑
ADM-T	0.79 ± 0.12	0.81 ± 0.11
TDN (Ours)	$\mathbf{0.95 \pm 0.03}$	$\mathbf{0.96 \pm 0.02}$

G 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.

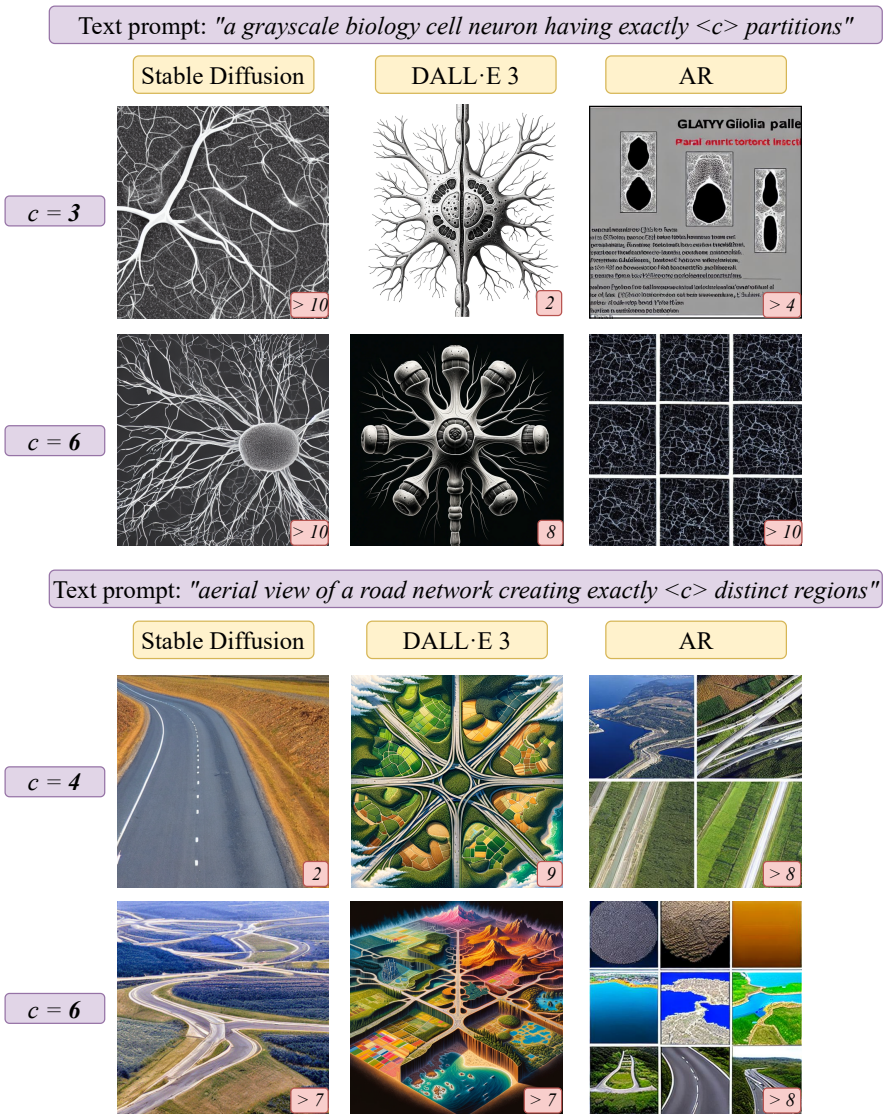

Figure 11: Qualitative results for 1-dim topological constraint. Stable Diffusion, DALL·E 3 and AR take text prompts as input (purple box). Rows 1-2: Results equivalent to CREMI. Rows 3-4: Results equivalent to Google Maps. Number of holes within each image is noted in its bottom-right inset.

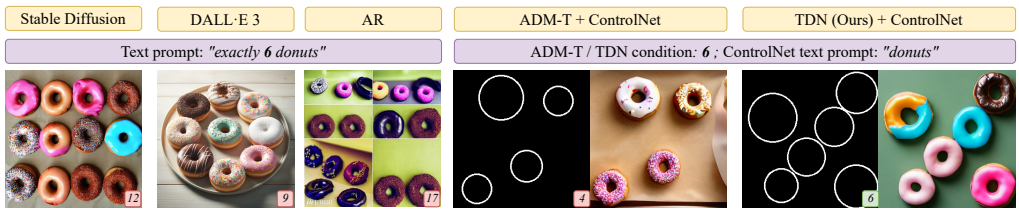

Figure 12: Results on standalone holes (holes not relative to the image frame/boundary). Number of donuts within each mask/image is noted in its bottom-right inset.

Table 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

H 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.

I 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 C), 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. 13. 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` (α_1, α_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.

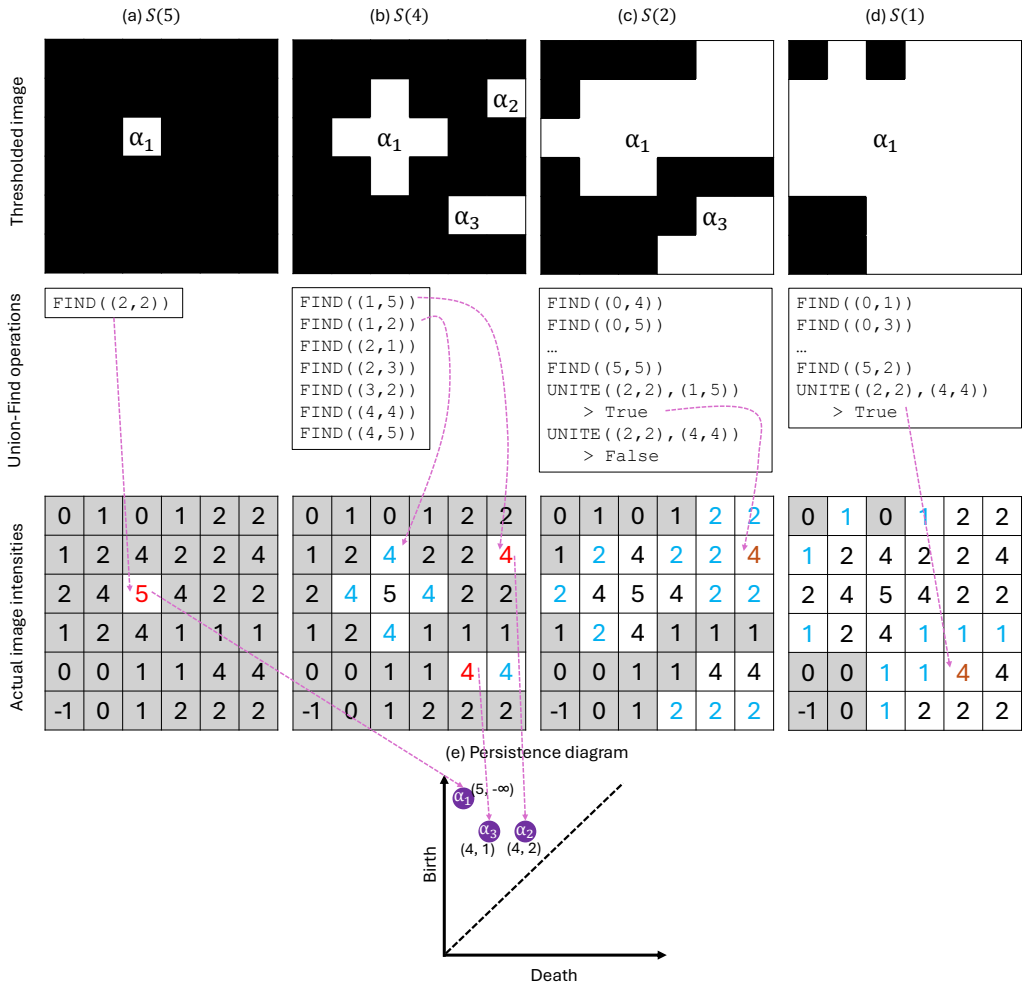

Figure 13: Figure inspired from Kaji et al. (2020). The figure shows a sample 2D image, thresholded at different values. (a)-(d) shows the sequence of super-level sets $\mathcal{S}(u)$, which is nothing but thresholding the image at the value u. It shows how different connected components (CC) are created 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.

J 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

K 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 images generated by TDN are shown in Fig. 14. 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.

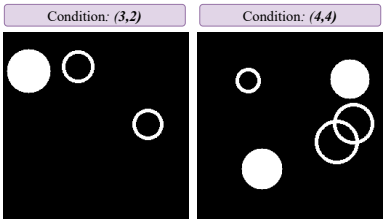

Figure 14: Images generated by TDN using both 0-dim and 1-dim topological constraints simultaneously. Condition (a, b) denotes a for 0-dim (#objects) and b for 1-dim (#holes).

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)	$\mathbf{0.9183 \pm 0.0731}$	$\mathbf{0.9338 \pm 0.0993}$	$\mathbf{0.9261 \pm 0.0906}$	**0.0982**
1-dim	ADM-T	0.7616 ± 0.0905	0.7892 ± 0.1129	0.7752 ± 0.1082	-
	TDN (Ours)	$\mathbf{0.9233 \pm 0.0705}$	$\mathbf{0.9492 \pm 0.0913}$	$\mathbf{0.9360 \pm 0.0720}$	-

K.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. 15, 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.

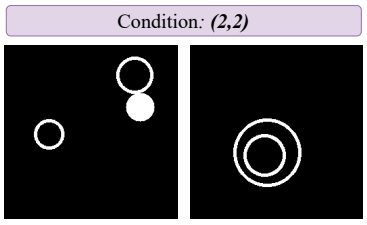

Figure 15: Sample images generated by TDN when using both 0-dim and 1-dim topological constraints simultaneously. With constraints $(a, b) = (2, 2)$, we can generate nested circles which are the closest to achieving a double annulus (two concentric circles) using our method.

L 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. 16 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. 16, 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.

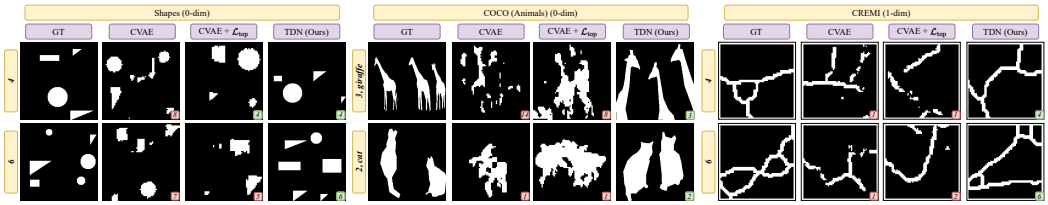

Figure 16: Qualitative comparison of using CVAE and diffusion models as the generative model. Ground truth (GT) denotes the true masks used to train the models

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**

M 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. 17.

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. 17, 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.

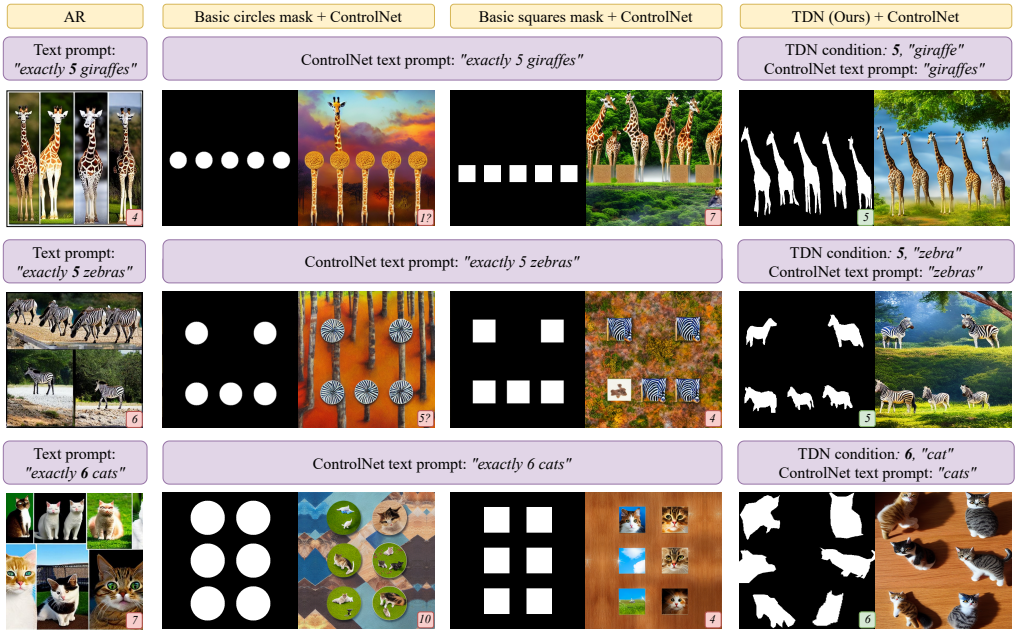

Figure 17: Qualitative comparison of using different conditioning types, particularly masks containing basic shapes like circles/squares to guide cardinality.

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 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.