Topology-Driven Learning for Images: Applications and Acceleration

A Dissertation Presented

by

Fan Wang

to

The Graduate School

in Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

August 2024

Abstract of the Dissertation

Topology-Driven Learning for Images: Applications and Acceleration

by

Fan Wang

Doctor of Philosophy

in

Computer Science

Stony Brook University

2024

The integration of topological methods into deep learning, particularly through persistent homology, has seen significant growth recently. Persistent homology, a key concept in Topological Data Analysis (TDA), offers insight into the topological characteristics of data by tracing their evolution across scales. This thesis explores the application of persistent homology to deep learning challenges, specifically focusing on ensuring the topological integrity of generated images. We enhance the robustness of a generative adversarial network by introducing a penalty based on the disparity in topological properties between the generated images and the training distribution. This approach ensures that the generated images resemble the training data not only visually but also in their topological characteristics.

As the second application, We extract persistent homology from breast DEC-MRI volumes as an approximation of fibroglandular tissues and use it to explicitly direct the attention of a 3D network

to a smaller set of voxels with high biological relevance. This targeted approach allows the network to focus on areas that are most indicative of underlying pathologies.

Despite the practical benefits, challenges remain in melding persistent homology with deep learning due to computational demands. Addressing this, the research focuses on massively parallel GPU algorithms to expedite persistent homology calculations, aiming to bridge this gap in deep learning frameworks. The shift from CPU to GPU computing is driven by the diminishing returns of Moore's Law, prompting a transition to accelerated computing with GPUs leading the performance enhancement.

This research delves into persistent homology computation for image data—essential in fields ranging from medical imaging to physical simulations, and crucial in deep learning as convolutional network outputs. This thesis introduces a streaming GPU algorithm for computing persistent homology in 2D and 3D digital images, addressing a notable bottleneck in deep learning training times. By constructing a cubical complex and utilizing discrete Morse theory for gradient vector field development, the proposed algorithm applies innovative massively parallel algorithms for topological sorting and path parity to formulate Morse boundaries. It assembles partial boundary relations from discrete data segments into a unified global boundary matrix for reduction, employing specialized measures to manage Morse matchings at chunk and GPU borders. This ensures the fidelity of persistent homology within a streaming context. The resulting algorithm exhibits exceptional speed improvements in preprocessing and persistent homology computations, surpassing existing state-of-the-art methods.

This work aims to contribute substantially improved algorithms to the TDA community, significantly reducing computation times and fostering further applications of topological loss in deep learning, thereby removing computational barriers and advancing the field.

Contents

List of Figures

ix

List of Tables

Acknowledgements

The completion of this dissertation was made possible by the combined support and guidance of mentors, the camaraderie of friends, and the unwavering encouragement from my family.

At the forefront, I must express my deep gratitude to my advisor, Prof. Chao Chen. His guidance has been nothing short of a beacon throughout my academic endeavors. His fervor for research, extensive knowledge, and meticulous care for detail have not only influenced my work but also my approach to challenges. He has provided an unwavering support system and has been a pillar of encouragement. His mentorship is an enduring source of inspiration that I will carry with me in the years ahead.

I also wish to thank my committee members, Prof. Dimitris Samaras, Prof. Rezaul Chowdhury, and Prof. Haibin Ling, for their insightful feedback and contributions that were instrumental to my research. I would like to extend my special thanks to Prof. Hubert Wagner for his considerable assistance throughout my research. His support has been immensely valuable to my work.

My journey has been greatly enriched by my peers and colleagues, whose discussions and perspectives have profoundly shaped my professional path. Special thanks are extended to a group of individuals who have been integral to my support system: Prof. Prateek Prasanna, Prof. Fuxin Li, Weimin Lyu, and Jiachen Yao. Additionally, I owe a word of thanks to my friends, Jiaju Miao and Lu Sun, for their companionship and unwavering support throughout this journey.

Lastly, to my parents, your boundless love and solid support have been the cornerstone of my journey. The success I've achieved is a reflection of your belief in me, and for that, I am eternally grateful.

Vita

Fan Wang hails from Liaoning Province, China. He completed his B.S. degree at Lanzhou University in 2012 and earned his M.S. degree from the National University of Singapore in 2013. He joined Advanced Digital Sciences Center, Singapore for 3 years. In August 2016, he embarked on his PhD journey at Stony Brook University's Department of Computer Science. His research explores the confluence of parallel programming, GPU programming, medical imaging, computer vision, 3D vision, and machine learning. He is dedicated to developing swift and dependable tools to benefit the community of topological data analysis.

Chapter 1

Introduction and Overview

Persistent homology, a central concept in topological data analysis (TDA), provides a powerful framework for analyzing and understanding the shape of data across multiple scales. By tracking how topological features like holes and voids appear and disappear as one filters through the data, it offers a robust method for capturing the intrinsic geometric and topological properties of complex datasets. Applications of persistent homology span a wide range of fields, from identifying new materials in chemistry to understanding the structure of the universe in cosmology, and from enhancing machine learning algorithms to improving medical diagnoses through better analysis of biomedical images. Its versatility and ability to provide meaningful insights into the underlying structure of data have led to a surge in popularity across various disciplines, heralding a new era of interdisciplinary research where mathematics meets real-world challenges.

This thesis first explores the integration of persistent homology within the framework of deep learning, showcasing its potential to significantly improve the interpretability and effectiveness of deep learning models. Through a series of applications, this work demonstrates how persistent homology can provide deep insights into data structure and dynamics, opening new avenues for research and innovation in fields where understanding the shape and features of high-dimensional data is crucial.

Furthermore, this thesis is dedicated to tackling the challenge of efficiently computing persistent homology, a pivotal tool in topological data analysis that reveals the underlying topological features of complex data sets across different scales. It delves into the development and refinement of computational methods that enhance the speed and accuracy of persistent homology calculations, making it more accessible for broader applications.

1.1 Applications of Persistent Homology

The adoption of persistent homology in the realm of data analysis has garnered significant interest due to its profound ability to elucidate the underlying topological and geometric structures of complex datasets. As a branch of topological data analysis (TDA), persistent homology offers a nuanced perspective on data by capturing the birth, persistence, and death of features across multiple scales, providing insights that are imperceptible through traditional data analysis techniques. Its applications span a broad spectrum, ranging from the study of brain connectivity networks in neuroscience to the characterization of material properties in physics. Recently, there has been a notable trend toward integrating persistent homology with deep learning, aiming to leverage the strengths of both fields. This integration seeks to enhance the interpretability and performance of deep learning models by incorporating topological and geometric information, opening new avenues for research and application in areas where understanding the shape and structure of data is critical.

In this context, my work introduces two significant contributions: TopoGAN in Chapter 3 and TopoTxR in Chapter 4. TopoGAN is an innovative approach that integrates topological loss into the training of generative adversarial networks (GANs), ensuring the topological correctness of generated images. By employing topological loss, TopoGAN effectively guides the generative model to produce images that not only are visually convincing but also adhere to the desired topological properties, bridging the gap between visual realism and topological accuracy. Meanwhile, TopoTxR stands at the intersection of medical imaging and topological data analysis, offering a novel biomarker for predicting pathological complete response (pCR) to breast cancer treatment. Utilizing persistent homology as a means to approximate the fibroglandular tissue structures in breast dynamic contrast-enhanced magnetic resonance imaging (DCE-MRI), TopoTxR represents a groundbreaking step towards harnessing the power of TDA for significant clinical applications, potentially improving the accuracy of treatment response predictions and personalizing therapeutic strategies.

1.2 GPU Computation of Persistent Homology

The computation of persistent homology remains a resource-intensive and time-consuming process, which significantly restricts its broader application across various fields. Despite its proven utility in uncovering the intricate

topological structures of data, the high computational cost and the substantial amount of time required to analyze large datasets act as barriers. This limitation not only hampers the adoption of persistent homology in disciplines that could benefit from its insights but also challenges researchers striving to integrate it into real-time or large-scale analysis workflows. Overcoming these obstacles is crucial for unlocking the full potential of persistent homology in widespread applications, from scientific research to industry solutions.

In this thesis, we embark on enhancing the computational efficiency of topological data analysis by introducing several innovative solutions. Initially, we present a GPU-based algorithm specifically designed for calculating Euler Characteristics in Chapter 6, setting the stage for more efficient topological analyses. Subsequently, we delve into a hardware design aimed at accelerating boundary matrix reduction, which is arguably the most time-consuming aspect of persistent homology computation in Chapter 5. This approach addresses a critical bottleneck, paving the way for faster processing times. We intend to propose a comprehensive GPU algorithm tailored for persistent homology computation itself. This algorithm is developed to leverage the parallel processing power of GPUs, significantly reducing the computational load and time required for analyzing complex datasets. Together, these contributions represent a substantial advancement in the field, offering tools and methodologies that could broaden the application and impact of persistent homology in various research areas.

Chapter 2

Related Work

In this chapter, we delve into an extensive review of the related works found in the literature. We explore various studies and developments that have contributed to the field, highlighting the key methodologies, findings, and theoretical frameworks that have shaped current practices.

2.1 Topological Data Analysis and Persistent Homology

Topological Data Analysis (TDA) employs topological techniques, notably persistent homology, to analyze datasets, drawing on foundational and advanced works in the field [1–7]. This approach has found diverse applications across various domains [3, 8–20], showcasing the versatility and breadth of TDA.

The integration of TDA with deep learning has seen innovative attempts to infuse topological insights into neural network models, leveraging the differentiable nature of persistent homology. This integration hinges on the concept that persistence diagrams and barcodes, which document topological transformations, are differentiable relative to input data. The pioneering work by [8] introduced a topological loss function for image segmentation that aligns persistence diagrams in a supervised fashion. Echoing this approach, [21] utilized persistence barcodes to impose topological constraints on target objects, yielding improvements in structural precision. Such topological loss functions have since been extended to other imaging tasks [22, 23] and broader learning challenges [6, 24–26], illustrating the method's adaptability.

Unlike previous strategies that apply topological priors in supervised learning scenarios, primarily for segmentation, our research adopts these priors in an unsupervised context. We explore the use of topological constraints within a reverse engineering framework to streamline the identification of triggers, en-

suring that the deduced triggers exhibit minimal connected components. This novel approach aims to harness topological properties to refine and guide the learning process, potentially opening new pathways for unsupervised learning applications.

2.2 Deep Learning with Topology

Recent research in deep learning has broadened its impact across various fields, showcasing significant innovations in technology and applications. Lin et al. proposed an end-to-end immediate implant placement AI tool through sagittal root inclination measurements without intermediate steps and extra labeling on images [27]. Zhan et al. transformed multiple transmitter localization into a series of computer vision challenges, introducing a deep learning-based solution [28, 29]. Similarly, Wu et al. developed a system for refining clinical note embeddings through representation learning [30] and enhanced embeddings using a triplet network and metric learning [31]. In another advancement, a novel topology-guided attention mechanism has been utilized to focus CNNs on biologically relevant regions [32]. Yao et al. addressed the issue of noisy segmentation ground truths by proposing a spatial correction method for annotations [33]. Furthermore, Lyu et al. explored attention anomalies in Trojaned BERT models [34], introduced Trojan Attention Loss to improve attack efficacy [35], and extended their studies into backdoor attacks [36–38], enriching the landscape of security in AI models. Zheng et al. study the Trojan attach problem by introducing the concept of Trojan Twin Model [39]. Lyu et al. apply transformer for mortality prediction [40]. Dong et al. use LSTM-HeterRGNN model for overdose risk prediction [41]. Li et al. calibrate uncertainty for semi-supervised crowd counting [42] and estimate confidence using unlabelled data [43, 44].

Miao and colleagues have made significant contributions to the understanding of behavioral patterns in DeepFake videos as outlined in their 2022 study [45]. They have also pioneered the patch distribution prediction approach for tracking eye movement, as detailed in their subsequent 2023 work [46]. Complementing these insights, Ren et al. have utilized convolutional neural networks for the analysis of dental panoramic radiographs, demonstrating the versatility of CNNs in dental imaging studies [47, 48]. Beyond the realm of dentistry, Ren and team have introduced a self-supervised learning framework for the removal of motion artifacts in optical coherence tomography angiography [49], which has implications for enhanced 3D cerebral blood flow imaging in conscious rodents [50]. Their research extends to the development of a weakly supervised approach for identifying cell activation, further illustrating

the potential of machine learning in biological imaging [51].

In response to the computational demands of neural networks, Wang et al. have spearheaded multiple strategies for optimization. They have introduced an efficient approach to neural architecture search, showcasing its efficacy in [52]. Building on their momentum, they further advanced the field with a novel neural network pruning method detailed in [53], aiming to streamline network structures without compromising performance. Their research continued with the development of a balanced training paradigm for generative adversarial networks (GANs), which they elucidated in [54]. This balanced approach addresses the often challenging training process of GANs, fostering stability and improving output quality. Additionally, they presented an innovative network initialization technique in [55], which optimizes the starting conditions for neural network training, contributing to a faster and potentially more convergent learning process.

GAN for images with fine structures. Generative Adversarial Nets (GANs) [56] have gained significant popularity for their ability to model complex data distributions effectively. A transformer based GAN has been used for time-series data generation [57, 58]. A GAN comprises two main components: a discriminator and a generator. The generator's goal is to create synthetic data that can fool the discriminator, which in turn tries to distinguish between the real and the synthetic data. Training of GANs continues until a Nash equilibrium is reached where the generator produces data indistinguishable from actual data, characterized by the discriminator loss which reflects the Jensen-Shannon divergence between the distributions of real and synthetic data.

One known issue in GAN training arises when the support between the real and synthetic data distributions does not overlap, leading to zero gradients for the discriminator and thus stalling the generator's learning. The introduction of Wasserstein GAN (WGAN) [59] mitigates this problem by utilizing the Wasserstein distance to more robustly measure the distance between real and synthetic distributions. However, implementing WGAN involves maintaining a 1-Lipschitz function for the discriminator, which is challenging. Techniques to enhance this aspect of WGANs have been developed [60–62], including various gradient penalty methods proposed to stabilize GAN training and encourage convergence to a local equilibrium [63–65].

Despite these advancements, while GANs excel at generating visually realistic images, they often do not guarantee the correctness of finer topological details in the generated images. This issue points to the need for incorporating topology-aware regularization in GANs. Addressing topological inaccuracies is vital as it enhances the structural integrity and realism of generated images. Techniques such as topology regularization have been proposed to address this

issue by reducing topological noise in the images generated by GANs [66].

Furthermore, advanced strategies like spectral normalization [61] and the use of large batch sizes along with truncation tricks [67] have shown to improve GAN training, achieving state-of-the-art results on benchmarks like ImageNet. Local patch-based approaches such as PatchGAN [68] focus on capturing high-frequency details by applying GANs to small sections of the image, proving effective in various applications including Pix2pix [68], CycleGAN [69], and SinGAN [70]. These methods emphasize the importance of preserving detailed structures in generated images, thus enhancing the overall quality and utility of GAN-generated visuals.

In the realm of Generative Adversarial Networks (GANs) that focus on geometric aspects, several models integrate geometry to enhance image generation. The geometricGAN [71] utilizes the concept of large margins, akin to Support Vector Machines (SVMs) [72], to refine both discriminator and generator learning processes. The Localized GAN (LGAN) [73] employs local coordinates to articulate the data manifold's local geometry. Specifically designed for facial imagery, the Geometry-Aware GAN (GAGAN) [74] incorporates face shape priors, while the Geometry-Consistent GAN (GcGAN) [75] ensures semantic integrity of images through a geometry-consistency constraint. These transformations typically involve image flipping and rotation.

Regarding high-order structural data in adversarial networks for semantic segmentation, several existing techniques [76–78] utilize adversarial losses within the semantic segmentation framework to harness high-order structural data. Nonetheless, these approaches largely overlook the preservation of topology.

Topological information for image analysis. Various methodologies have been suggested for using persistent homology directly as a tool for feature extraction. These topological features are vectorized [14] and subsequently integrated into kernel machines [79–81] or employed within deep neural network architectures [82]. In fully supervised image segmentation tasks, topological considerations are often used as a directive constraint or loss to enhance the quality of segmentation [8, 83, 84]. Mosinska et al. [85] adopt an implicit approach to modeling topology via feature maps from pretrained VGG networks [86], though this technique falls short in adapting to unseen geometric structures. Additional studies have focused on specific applications like retinal vessels [87] and lung airways [88], primarily concentrating on connectivity (0-dimensional topology) and showing limitations in extending to higher-dimensional topologies. In machine learning, topological data is increasingly being used to explore the topology of data manifolds [13, 24, 26, 89] and to enhance graph learning through sophisticated structural information [17, 90].

In the context of evaluating generative models, Khrulkov and Oseledets [91] have advanced a method that employs the topology of the data manifold to assess the disparity between synthetic and real data distributions. This provides a qualitative metric for generative models' effectiveness. Despite its innovation, this metric primarily addresses the standard image feature space and falls short in accurately determining whether the generator has effectively captured the real image topology.

Furthering the exploration of topology in generative models, Brüel-Gabrielsson et al. [92] implemented a specialized loss function to uphold connectivity constraints within generated images. However, the application of such predefined topological constraints, like connectedness, may not truly aid the generator in learning the authentic topological distribution inherent in real data. This is because these constraints are artificially imposed rather than learned from the data itself.

Addressing these limitations, TopoGAN emerges as the pioneering generative model that autonomously learns topological features directly from real images. This advancement marks a significant step forward in allowing generative models to more deeply understand and replicate the complex topological properties observed in natural data sets.

2.3 Topological Data Analysis for Breast DCE-MRI

Quantitative imaging characteristics, in tandem with machine learning techniques, have been pivotal for predicting pathological complete response (pCR), as evidenced by various studies [93, 94]. The field of radiomics, which deals with the quantitative analysis of tumor features such as texture and shape, has been promising in gauging treatment outcomes. Specifically, these features are adept at depicting both the primary tumors and, more recently, the surrounding tissue zones [95, 96]. However, the utility of these methods is often curtailed by their prescriptive nature, a shortfall in universal applicability, dependency on precise lesion demarcation, and a gap in elucidating phenotypic variances beyond the immediate tumor vicinity.

Convolutional Neural Networks (CNNs) have been deployed for analyzing breast Dynamic Contrast-Enhanced Magnetic Resonance Imaging (DCE-MRI) to forecast pCR [97–100]. Given the less-than-ideal performance of models based solely on images, there has been a move towards integrating image-based CNN models with clinical data not captured in images, thereby enhancing predictive accuracy [101]. Efforts have been made to decompose

stain in multiplex immunohistochemistry images with unsupervised learning [102]. Yao et al. applies topological analysis of mouse brain vasculature via 3D light-sheet microscopy images [103].

The literature is replete with instances of CNN applications in mammogram-based cancer diagnosis, underscoring their utility in this domain [104–106]. However, the inherent limitation of mammograms, being 2D projections of 3D structures, introduces challenges in interpreting the data they provide. While 2D mammography struggles with data interpretation, 3D mammography, though offering depth, does not adequately inform on the dynamics of tissue enhancement, both of the tumor and its surrounding areas. Comparative analyses have demonstrated MRI's superiority in detailing breast cancer's scope over mammography, which, alongside ultrasound, often misestimates tumor dimensions following neoadjuvant chemotherapy in about 8% of instances, as highlighted by retrospective evaluations [107–109]. The aim of conducting post-chemotherapy examinations is to assess residual disease, yet traditional methods like mammography, ultrasound, and physical assessments only identify pCR in 13-25% of cases, as observed in specific studies [110–112]. Despite their limitations, MRI and ultrasound are preferable to mammography for detecting remaining tumors, with MRI particularly outperforming in detecting multifocal or multicentric diseases [109].

The utilization of topological data, especially through the lens of persistent homology [113], presents a sophisticated method for encoding the intricate topological nuances present within images. This form of data representation, encapsulated through persistence diagrams or barcodes, has been applied across a spectrum of image analysis endeavors, including the examination of cardiac imagery [84], analysis of brain networks [114], and the segmentation of neuron images [8]. In a novel stride, the integration of this topological perspective with deep learning architectures has been explored to introduce topological rigor into image segmentation tasks, showcasing significant advancements [8, 21, 115, 116]. The exploration into translating persistence diagrams into learnable frameworks spans several methodologies, ranging from vectorization strategies [14], to the utilization of kernel methods [79–81], and the incorporation into deep learning paradigms [82]. Moreover, the adoption of graph-based formulations of topology, efficiently synergized with Graph Neural Networks (GNN), has broadened applications to include pathology image categorization [117] and the distinction of retinal blood vessels [118]. An intriguing, albeit less robust form of topological data, the Betti curve, has been harnessed for enhancing learning with breast imagery [119]. Yet, the potential of directly engaging with the geometric manifestations of topological constructs, such as cycles and voids, remains largely untapped. These constructs hold the key

to unraveling the geometric intricacies of breast tissue structures, like fibroglandular tissues, offering a novel pathway for CNN analysis by mapping these topological insights back onto the original volumetric data.

This discourse introduces a pioneering deep learning strategy that capitalizes on the geometric attributes of topological constructs, utilizing them as an explicit mechanism for directed attention within our model. By identifying and focusing on loops and bubbles that outline the detected topological features, we forge a deep learning architecture that concentrates on these significant topological elements. This approach diverges from conventional reliance on hand-crafted features, demographic information, or learned representations. Instead, it leverages the geometric fidelity of tissue structures revealed through persistent homology, translating these insights into a format amenable for CNN processing.

The extraction of explicit tissue structures presents a formidable challenge, notably due to the frequent absence of definitive ground truths for training segmentation models. Persistent homology offers a resolution to this dilemma within various biomedical imaging contexts, enabling the unsupervised extraction of structurally informative features [8, 21, 84, 120–122]. Despite its potential, many existing methodologies have largely confined their use of persistence diagrams to a role as direct features. The geometric wealth contained within the topological structures identified by these algorithms remains underutilized, suggesting a fertile ground for innovation.

Self-attention, a concept ushered in by the transformer architecture [123], revolutionizes the dynamic capture of relationships within data sequences through its mechanism. In the realm of visual tasks, spatial attention crafts an attention map by mining the spatial relationships within a feature map, while channel attention distills information across different channels into a singular attention vector [124]. The introduction of mask-guided attention into spatial attention modules sharpens the focus of the model, directing it towards regions of interest with heightened precision [125]. By clearly delineating these areas, the model is empowered to selectively engage with relevant sections, sidelining potential noise or less pertinent areas as dictated by the mask. This targeted approach not only streamlines the training process but also enhances overall model efficacy.

2.4 Hardware Acceleration of Persistent Homology Computation

In this section, we delve into the existing literature surrounding the hardware acceleration of persistent homology computation. Our focus spans across a range of hardware solutions specifically tailored to enhance the computation and access patterns inherent to various aspects of persistent homology analysis. This encompasses both custom-designed hardware optimized for the unique computational demands and access patterns of persistent homology processes, as well as more generalized solutions utilizing Graphics Processing Units (GPUs). The exploration of these hardware accelerations aims to identify strategies that significantly reduce computational time and improve efficiency, thereby facilitating the broader application and deeper investigation of persistent homology in complex data analysis. Through this review, we aim to highlight key developments and identify potential areas for innovation in accelerating persistent homology computations, bridging the gap between theoretical mathematical concepts and their practical application in data analysis.

2.4.1 GPU Computation of Euler Characteristic Curves

Due to their straightforwardness, both the Euler characteristic curve (ECC) and the Euler characteristic (EC) are widely utilized across various disciplines, particularly in imaging-related domains. An insightful overview of this subject can be found in the comprehensive review authored by Worsley [126], from which we draw inspiration for the applications discussed below. Subsequently, we delve into the existing computational literature pertinent to this area.
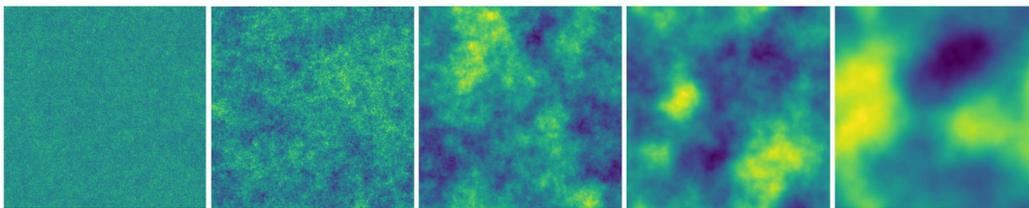
Figure 2.1: Visualizations of Gaussian random fields generated with different levels of smoothness.

Diverse Applications. Concepts akin to the Euler characteristic (EC) emerged within the realm of astrophysics as early as the 1970s, where the Euler characteristic curve (ECC) was referred to as the *genus curve*. Formalization of these concepts occurred in 1986 with the work of Gott et al. [127], investigating the

sponge-like topology evident in large-scale cosmic structures. Subsequently, the ECC gained prominence in the analysis of imaging data related to the cosmic microwave background (CMB) radiation [128], with roots traced back to earlier research on the topology of Gaussian random fields (GRFs) by Adler and Hasofer [129], utilized for modeling the CMB. Notably, Fig. 2.1 showcases visual representations of GRFs.

Within the field of bone morphometry, ideas surrounding the EC gained traction, culminating in formal mathematical treatment in 1993 [130]. Here, the EC played a pivotal role in characterizing trabecular bone structures, particularly in computing the first Betti number, referred to as the *connectivity* within this context.

Moreover, the EC serves as a fundamental tool in morphological image processing [131], where it is commonly employed to delineate the shape of thresholded (binary) images under the guise of the *Euler number*. Subsequently, it found application in computing the ECC across all thresholds of grayscale images, known as the *stable Euler number* [132]. Notably, the zero-crossing of the ECC plays a crucial role in selecting segmentation thresholds, as illustrated in Fig. 6.2, where it distinctly highlights riverbeds in terrain datasets.

Emergence of ECC in Topological Data Analysis (TDA). While persistent homology remains the dominant approach in contemporary topological data analysis (TDA), several recent studies have highlighted the utility of the Euler characteristic (EC) and Euler characteristic curve (ECC) in this domain. Bobrowski and Skraba [133] demonstrate the remarkable effectiveness of ECC in analyzing percolation thresholds within random cubical filtrations and other stochastic models. Crawford et al. [134] propose a novel statistic based on EC, proving its efficacy in predicting clinical outcomes for brain cancer patients using brain imaging data.

Furthermore, Amezquita et al. [135] explore the application of EC-based image transforms in analyzing barley shapes. Lin et al. [136] investigate the utility of various handcrafted features, including ECC, for analyzing magnetic resonance images of the liver. Smith and Zavala [137] advocate for ECC as a versatile data descriptor applicable across diverse data types such as graphs, point clouds, and images, with a specific focus on its effectiveness in characterizing liquid crystal system images.

The computation of Euler characteristic transforms, as explored by Betthauser et al. [138] and Jiang et al. [139], presents an intriguing avenue. Notably, these computations often necessitate manipulation of filtration values on a per-cell basis, contrasting with the standard case where filtration values are assigned per voxel, as addressed by our software.

Algorithmic Developments. The computational techniques employed diverge from the simplistic framework outlined earlier. Instead, they often leverage the intimate connection between the Euler characteristic (EC) and differential geometry. Notably, an efficient algorithm for EC computation in 3D voxel data was introduced by Gott et al. [127], notable for its reliance on precomputed tables of voxel neighborhoods. In contrast, our approach is rooted in the mathematical and algorithmic framework of cubical homology, a direction pioneered by Kaczynski, Mischaikow, and Mrozek in the 1990s, with further elaboration in their seminal work [140].

Originating in computational dynamics, this framework evolved into a broader context, leading to the development of the first efficient, general-dimension algorithm for EC computation of binary images by Ziou and Allili [141] in 2001. This approach treats a binary image as a cubical complex, compactly encoding its information. Despite its simplicity and efficiency, one drawback lies in the memory overhead associated with storing the cubical complex.

The advent of efficient algorithms for Euler characteristic curve (ECC) computation further propelled advancements. Snidaro and Foresti [132] introduced the first efficient algorithm for 2D image ECC computation in 2003, followed by Richardson and Werman's 2014 work [142] for 3D images, and Heiss and Wagner's 2017 contribution [143] for arbitrary-dimensional images. This methodology extends Ziou and Allili's concept to cubical filtrations, accommodating grayscale data. Notably, it introduces parallel computation and streaming of images into memory in manageable chunks, enabling the handling of images of arbitrary size. Our GPU implementation builds upon this approach.

2.4.2 Hardware Acceleration of Boundary Matrix Reduction

In persistent homology computation, a critical step involves reducing a boundary matrix. This matrix's columns and rows correspond to simplices of various dimensions (vertices, edges, triangles, and tetrahedra) within a domain's discretization, illustrated in Fig. 2.3. A binary boundary matrix ∂ marks with a 1 when simplex σ_u, represented by row u, is part of the boundary of σ_v, represented by column v. The goal of boundary matrix reduction is to transform this matrix into a simplified form using column additions in a binary field. This process is often computationally intensive due to the massive size of boundary matrices, even for relatively small input images. For instance, 28×28 pixel images from the MNIST dataset yield 1D and 2D boundary matrices of sizes 784×1512 and 1512×729, respectively. As boundary matrix reduction
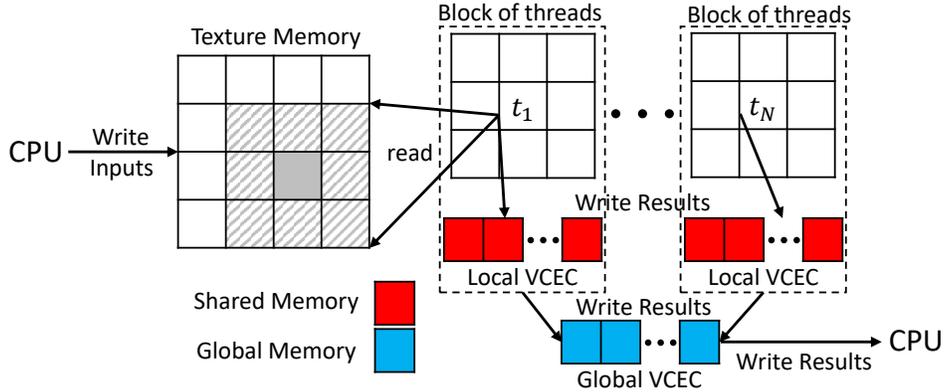
Figure 2.2: The image is initially transferred from RAM to the GPU's global memory. Each block of threads undertakes the processing of a specific section of the image. Within each block, every thread accesses a voxel and its eight neighboring voxels, all denoted in a lighter blue hue. Upon completion of block-level processing, the local result obtained by the block is amalgamated with the global result. Finally, the comprehensive result for the entire image is conveyed back to RAM.

becomes a significant bottleneck in persistent homology analysis, hindering broader application, there's a pressing need for its acceleration. Hardware accelerators have shown potential in speeding up such computations significantly, surpassing traditional CPU and GPU capabilities. Our approach, focusing on hardware-specific optimizations, aims to enhance this process by up to 20,000 times, marking a pioneering effort in hardware-accelerated persistent homology analysis.

2.4.3 GPU Computation of Persistent Homology

Research on applying topological methods to grid data, specifically through persistent homology and Morse complexes, dates back several decades. Initially, Verri et al. [144] explored algebraic topology in the context of grid data by analyzing topological features of size functions in 1993. The field expanded with the work of Kaczynski et al. [145], who applied the Conley index to study dynamics in nonlinear systems, with further applications emerging over time [146]. Wang et al. propose an efficient GPU algorithm for Euler Characteristic Curve computation [147] and a hardware accelerator for boundary matrix reduction [148].

In recent years, there has been a notable increase in the topological analysis of grid data. For example, Bendich et al. [149] devised a method to approx-
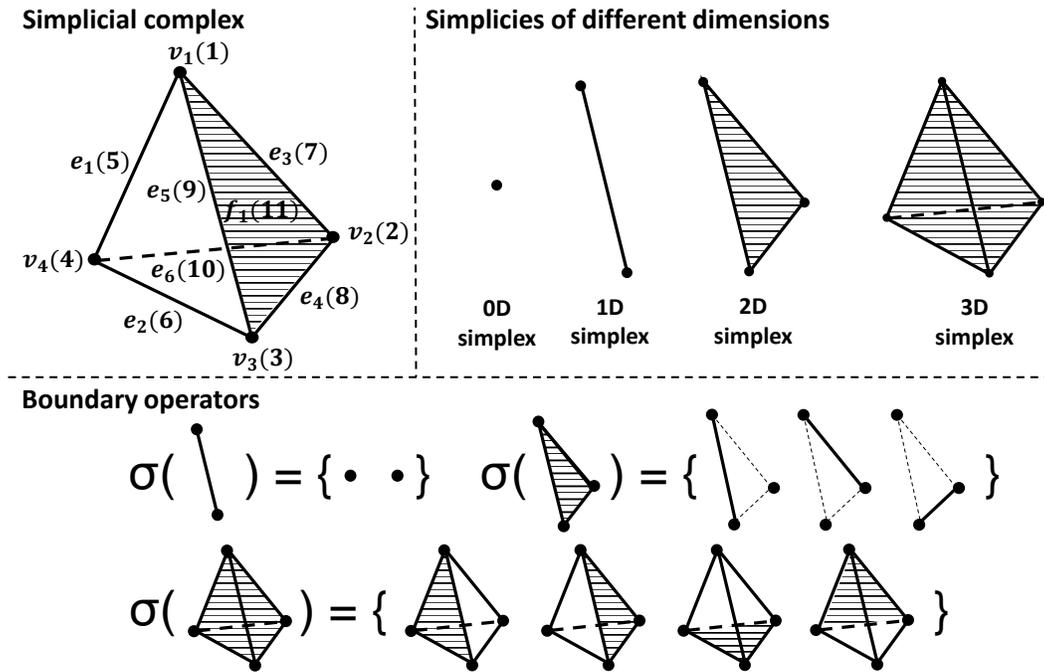
Figure 2.3: The figure showcases key elements of topological data analysis. At the top left, it displays a simplicial complex annotated with filter function values next to vertices, edges, and faces. The top right focuses on simplices of dimensions 0, 1, 2, and 3. The bottom row provides examples of boundary operators acting on 1-, 2-, and 3-dimensional simplices, illustrating the mathematical foundation behind the construction and analysis of simplicial complexes in topological studies.

imate the persistent homology of 3D grid data. Employing discrete Morse theory for efficient gradient pair assignments, Robins et al. [150] achieved a reduction in the size of the boundary matrix while maintaining the integrity of the original topology. Concurrently, Wagner et al. [151] introduced a new technique for generating boundary matrices efficiently, facilitating exact persistence calculations for grid data across various dimensions. This method has gained widespread adoption in tools such as Gudhi [152] and DIPHA [153], although its application often necessitates further simplification due to the large size of the boundary matrices involved.

Günther et al. [154] introduced a memory-efficient method for storing matchings to handle large datasets, although this approach notably increased the complexity of implementation. Expanding upon the work of Robins et al. [150], Mischaikow and Nanda [155] extended the application of these methods beyond cubical complexes, implementing the Perseus software to compute per-

sistence across various data types. Their methodology emphasized flexibility, enabling a wide range of applications, albeit at some cost to efficiency. They also pioneered a streaming preprocessing technique where individual level sets are stored, simplified, and then merged separately.

In contrast, our approach uses spatial decomposition of the volume, building on Robins et al.'s *ProcessLowerStars* method. This strategy was further developed by Wagner [156], who introduced a topology-preserving streaming method that partitions large grid data into slices for parallel processing. Our research adapts this streaming scheme to GPU architectures, enabling highly parallel execution of *ProcessLowerStars* on individual GPU threads for separate voxels, as opposed to Wagner's method which assigned each CPU thread to process an entire data slice.

Moreover, the utility of discrete Morse theory for data visualization simplifications was showcased by Gyulassy et al. [157], who processed a billion-voxel volume on standard hardware in a single day. However, it was noted that these simplified complexes are not suitable for computing persistent homology of the original data [158].

The advent of modern GPU capabilities has spurred the development of rapid, memory-efficient parallel methods for topological analysis of extensive scientific datasets. For instance, Zhang et al. [159] developed Ripser++, a GPU-accelerated algorithm that enhances the efficiency of Vietoris-Rips computations. Concurrently, a method taking advantage of GPU parallelism was introduced by [147] for computing cubical complexes and Euler characteristic curves on grid data. Subhash et al. [160] presented a GPU parallel algorithm that efficiently addresses the path counting challenge in Morse-Smale complexes by transforming graph traversals into parallel tasks. Our contribution to this area includes a novel parallel topological sorting method and a parallel algorithm for path parity calculations, effectively solving the path counting issue.

Chapter 3

Topology-Aware Generative Adversarial Network

3.1 Introduction

This thesis introduces an application of persistent homology—a topology-aware generative adversarial network. Generative adversarial networks (GANs) [56] have demonstrated significant success in producing highly realistic images. These networks involve training a generator to create images that mimic real ones, while simultaneously training a discriminator to differentiate between these synthetic and actual images. Through a minimax game, the generator evolves towards a model capable of sampling synthetic images from a distribution that aligns closely with that of genuine images.

A critical aspect of designing GANs is addressing how to effectively minimize the disparity between synthetic and real image distributions, not just in terms of visual likeness but also in semantic content. Research has highlighted [161, 162] that while traditional GANs [56, 59, 60, 163–165] typically match the first order moments of these distributions within a CNN-based feature space, more advanced techniques strive to align higher order statistics. For instance, methods have been developed to match second order statistics of image features [162, 166]. Specifically, Kossaifi et al. [74] integrate a statistical shape prior for face images into the generator. This approach is founded on the principle that the more complex, higher-order information a generator can assimilate, the more semantically rich and realistic the generated images become.

In our research, we emphasize the significance of structural information within images. Numerous applications involve imagery featuring intricate topologies, such as biomedical images depicting neurons, membranes, or blood

17
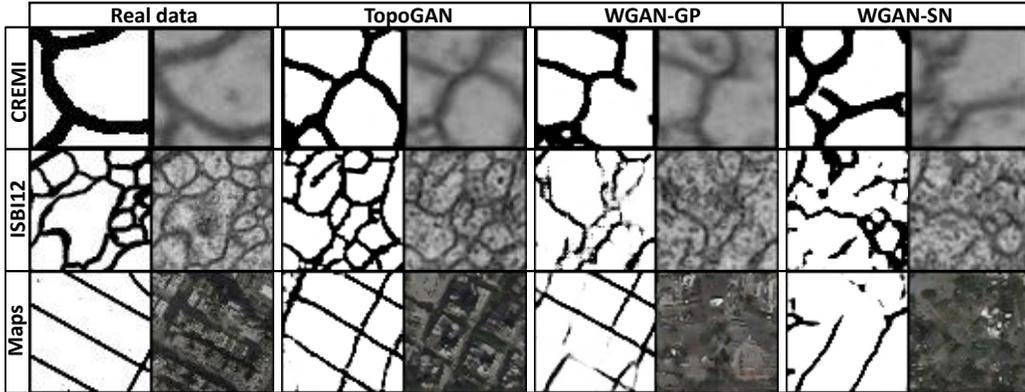
Figure 3.1: Sample images displaying structures such as neuron membranes and road networks from satellite imagery. Top to bottom: neuron images from CREMI and ISBI12, and road network images from Google Maps. From left to right: actual images, images generated by TopoGAN, WGAN-GP, and WGAN-SN. Each actual and synthetic image is accompanied by a textured version. Textures for synthetic images are applied using a separately-trained pix2pix network.

vessels, and satellite images showing road networks (Fig. 3.1). The topological characteristics of these structures—such as their connectivity and loopiness—carry essential semantic and functional information. Maintaining structural fidelity is paramount when synthetic images are used to train downstream applications like diagnostic tools that analyze the structural complexity of retinal vessels, navigation systems reliant on road network topology, or classifiers that assess neuron morphology and connectivity.

In this study, we introduce TopoGAN, a pioneering GAN model designed to learn and replicate the topology of real data. Topology in this context refers to the structural complexity indicated by the number of connected components and holes, which are globally challenging attributes to capture. Traditional GAN discriminators, which differentiate synthetic from real images based on CNN-based features, do not account for topological differences. Consequently, this limits the generator's ability to accurately learn and replicate the topology of real images. As depicted in Fig. 3.1, structures generated by conventional GANs, such as WGAN-GP and WGAN-SN, often appear fragmented and disconnected.

Our key technical innovation is the development of a *topological GAN loss*, which precisely aligns the synthetic and real image distributions based on their topological characteristics. Utilizing persistent homology theory [113],

we transform both synthetic and real images into a topological feature space to measure and quantify topological discrepancies as a loss. This loss is differentiable, allowing for optimization through backpropagation. Our topological GAN loss enhances the capabilities of the existing discriminator, enabling the generator to produce images that are not only realistic in terms of CNN-based features but also in topological attributes (Fig. 3.1). It's important to note that TopoGAN primarily focuses on generating binary images, or masks, that outline the structures of interest. Once these topologically accurate structures are generated, additional textural details can be applied using existing techniques such as pix2pix [68].

TopoGAN stands out as the inaugural generative model that effectively learns topology directly from real images, according to our findings. We have validated the effectiveness of TopoGAN through extensive testing across diverse datasets, including biomedical, satellite, and natural images. The success of our approach is quantified using the traditional FID score [167] alongside two innovative topology-aware metrics we developed based on persistent homology and the Betti number. Our results indicate that TopoGAN significantly surpasses conventional GAN models in these topology-specific evaluations. Additionally, we demonstrate that images synthesized with accurate topological features can enhance performance in subsequent applications such as image segmentation.

Our contributions within this research are multifaceted and significant:

- We introduce a novel topological GAN loss that quantifies the disparity between synthetic and real image distributions based on their topological characteristics. Unlike previous approaches that apply topological loss at the instance level [8], our methodology is the first to promote topological congruence across entire distributions.

- We confirm that this topological loss is differentiable, enabling its integration into the standard GAN training process, thus bridging the gap between traditional feature-based and topology-focused learning.

- We develop new topology-aware metrics that provide a more precise measure of a generator's performance in replicating topological features, offering a deeper insight into the quality of generated images beyond conventional metrics.

Through these innovations, TopoGAN not only advances the field of generative modeling by incorporating topological learning but also sets new standards for evaluating and understanding the complexities involved in synthesizing topologically accurate images.

3.2 Method

In our TopoGAN framework, we have engineered a mechanism that aligns the distribution of synthetic images with that of real images, effectively matching both on conventional image features and topological attributes. To achieve this, we have incorporated a novel loss term specifically for the generator, denoted as $L_{topo}(P_{data}, G)$. This *topological GAN loss* is designed to evaluate the proximity of the topology in images generated by the generator G to that of the actual images. By minimizing this loss, the generator is compelled to produce synthetic images that closely mirror the topology of the real images.

The overall loss function for the discriminator is outlined in Eq. (3.1). For the generator, the total loss is defined in Eq. (3.2) and comprises the traditional generator loss combined with the new topological GAN loss. Formally, we have

$$\arg\max_D \left[\mathbb{E}_{x \sim P_{data}} \log D(x) + \mathbb{E}_{z \sim P_z} \log(1 - D(G(z))) \right], \qquad (3.1)$$

$$\arg\min_G \left[\underbrace{\mathbb{E}_{z \sim P_z} \log(1 - D(G(z)))}_{\text{conventional generator loss}} + \lambda \underbrace{L_{topo}(P_{data}, G)}_{\text{topological GAN loss}} \right], \qquad (3.2)$$

where λ controls the weight of the topological GAN loss.

In our TopoGAN framework, the primary focus is on the generation of binary images, specifically masks that outline critical structures such as blood vessels, neuronal membranes, and road networks. The generator is designed to produce a real-valued grayscale image that represents these synthetic masks. These masks are interpreted by the discriminator as real-valued grayscale images with intensity values ranging from 0 to 1. Once these masks are generated, textural details are added using a separately trained pix2pix [68] network, which enhances the masks with realistic textures based on the underlying structure outlined by the mask.

The remainder of this section is dedicated to detailing the definition and optimization of the topological GAN loss. In Sec. 3.2.1, we delve into the method of extracting topological features, specifically the persistence diagram, from an input mask using persistent homology theory. This diagram is a crucial element in understanding the topological structure of the masks.

In Sec. 3.2.2 and Sec. 3.2.3, we articulate the specifics of the topological GAN loss. This involves comparing the distributions of persistence diagrams derived from both synthetic and real images. The objective of minimizing this loss is to align a synthetic persistence diagram more closely with its corresponding real diagram. Adjusting the persistence diagram in this manner encourages the development of structural masks that complete almost-loops,
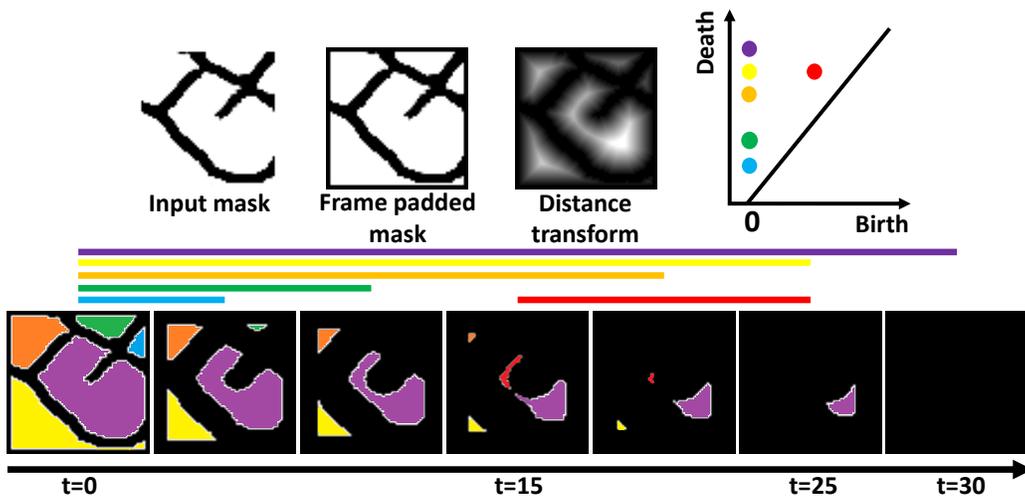
Figure 3.2: Illustration of persistent homology. Top row, from left to right: the initial input mask, framed to ensure all branches form discernible holes; the distance transform; and the resulting persistence diagram. Bottom row: a sequence of sublevel sets at various thresholds, showcasing the formation and filling of different holes. All original holes are born at $t = 0$, while an almost-hole (indicated in red in the diagram and sublevel sets) emerges at a later time ($t = 15$).

thus preventing the generation of images with incomplete topological features. This strategic approach is pivotal in teaching the generator to produce more accurate and topologically consistent images.

Additionally, as a novel technical contribution, we introduce two new topology-aware metrics in Sec. 3.2.4. These metrics are designed to evaluate the performance of the generative model by comparing the distributions within the topological feature space, offering a refined measure of how well the synthetic images match the topological characteristics of real images. These metrics provide a deeper insight into the effectiveness of the topological enhancements incorporated within the TopoGAN framework.

3.2.1 Persistent Homology: From Images to Topological Features

In our study, we utilize the theory of persistent homology to extract topological features from input masks, focusing on a comprehensive representation that includes both clear and nearly-formed topological structures—referred to as holes/loops and almost-holes/almost-loops, respectively. This process is

visualized in Fig. 3.2, where these concepts are introduced with further elaboration available in supplemental materials and a key reference for topological data analysis [113].

Considering a topological space $y \subseteq \mathbb{R}^2$, such as an image mask, the primary structures of interest are the holes (1-dimensional) and the connected components (0-dimensional). In this paper, our focus is primarily on the 1-dimensional topology, specifically the number of holes, quantified by the *Betti number*, β_y. An example from the CREMI dataset illustrates this, where a neuron membrane structure is framed to ensure complete analysis through 1-dimensional homology, as per the principles of algebraic topology [168].

In the depicted example (Fig. 3.2), there are five observable holes, giving a Betti number of $\beta_y = 5$. However, the Betti number traditionally captures only fully formed holes. Notably, there is a dangling branch in the middle of the image that nearly encloses an area, representing an almost-hole that the Betti number does not account for. To address this limitation and effectively incorporate these almost-formed structures into our analysis, we apply both the distance transform and the refined techniques of persistent homology [113]. We review the distance transform:

Definition 1. *The Distance Transform (DT) [169] generates a map D for each pixel p on a binary image I: $D(p) = \min_{q \in \Omega}\{||p - q|| \mid I(q) = 0\}$, in which Ω is the image domain.*

The process of extracting topological features from an image mask involves transforming the mask into a scalar function $f_y : \Omega \to \mathbb{R}^+$ across the entire image domain, where Ω is the image space. We then introduce the concept of *sublevel sets* for this function. A sublevel set is defined by applying a specific threshold t, resulting in $\Omega^t_{f_y} = \{x \in \Omega \mid f_y(x) \leq t\}$. This allows us to explore different aspects of the mask's topology by varying the threshold, which can convert almost-holes into complete holes at certain values.

The collection of all these sublevel sets forms what is known as a *filtration*. This filtration essentially illustrates the evolution of the mask as the threshold is progressively increased, revealing new topological features at various stages. Persistent homology analyzes this filtration to trace the life-cycle of each topological feature—be it holes, connected components, or more complex structures—across different threshold levels.

In Fig. 3.2, we observe the original five holes, which are present from the start (birth at $t = 0$) and are progressively filled as the threshold increases (death at different t values). An interesting feature, the almost-hole highlighted in red, emerges at a higher threshold level (birth at $t = 15$), when one of the purple holes divides into two, and it disappears at $t = 25$ as the filtration progresses and the gaps close.

These life events of holes are depicted in a *persistence diagram*, where each topological feature is represented as a point. The coordinates of these points are defined by the birth and death thresholds of the corresponding features. Thus, for our discussed example, the diagram contains five points originating at $birth = 0$ and various death times, along with an additional red point representing the almost-hole with a distinct birth time.

The utility of the persistence diagram is profound, as it provides a two-dimensional summary of the topological changes within the mask across various scales. It captures not just the existence of features, but their persistence over the range of sublevel sets, offering a deeper insight into the structural complexities of the image. This method is flexible and can be applied to any scalar function defined over a topological space, not just distance transforms, making it a powerful tool in topological data analysis.

3.2.2 Distance Between Diagrams and Topological GAN Loss

In the development of our topological GAN framework, we employ a refined technique to compare the topological features of synthetic and real images by transforming each input binary image y into a persistence diagram, $\mathrm{dgm}(f_y)$, using distance transform and persistent homology. This diagram represents the topological signature of y and serves as a crucial element in measuring topological fidelity. We define our topological GAN loss based on the discrepancies between the sets of diagrams derived from synthetic masks and those from real masks, utilizing concepts from optimal transport theory [170] to facilitate these comparisons. An illustration of this loss concept can be viewed in Fig. 3.3.

The distance between persistence diagrams is quantified using a version of the p-Wasserstein distance, known for its robust properties [171, 172]. This metric is traditionally used to measure the distance between two point sets represented on a 2D plane—each point in these sets corresponds to a topological feature (a hole or a connected component) in the persistence diagram.

However, for our specific application within the topological GAN framework, we modify the traditional p-Wasserstein distance to better suit our needs. We concentrate solely on the birth times of the topological features and disregard their death times. This adaptation focuses on the critical aspects of topological formation—particularly the birth of almost-holes, which are pivotal in understanding how well the synthetic image mimics the structural complexity of the real image. By projecting the points from both diagrams onto the birth axis, we compute their 1-Wasserstein distance, which calculates the optimal
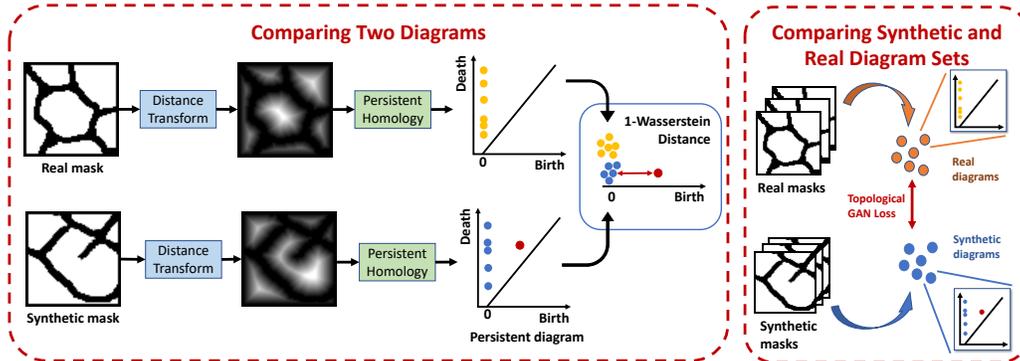
Figure 3.3: Our topology-processing pipeline. The process begins with a batch of both real and synthetic masks. Each mask undergoes a distance transform followed by persistent homology computation, resulting in its respective persistence diagram—a collection of 2D points. These diagrams from real and synthetic masks are compared using the 1-Wasserstein distance, focusing solely on the birth times. The loss is quantified as the matching distance between these sets of diagrams (real and synthetic), calculated through optimal transport.

matching distance between these two sets of points along the birth dimension, as depicted in Fig. 3.3.

This approach simplifies the analysis by focusing on the initial emergence of topological features, rather than their persistence or resolution. In practice, this means that the calculated distance primarily reflects the efforts required to close the gaps in synthetic images to match the real images' topology, particularly in terms of forming new holes. It is an effective measure for assessing how well the synthetic process reproduces the complex topological structures found in natural images. Through this modified metric, we aim to ensure that our generated images not only resemble the real ones visually but also replicate their intrinsic topological characteristics. Formally, the distance between two diagrams dgm_1 and dgm_2 is

$$\mathcal{W}_1(\mathrm{dgm}_1, \mathrm{dgm}_2) = \min_{\sigma \in \Sigma} \sum_{x \in \mathrm{dgm}_1} |b_x - b_{\sigma(x)}| = \sum_{x \in \mathrm{dgm}_1} |b_x - b_{\sigma^*(x)}|, \qquad (3.3)$$

in which Σ represents the set of all potential one-to-one correspondences between two diagrams, and σ^* denotes the optimal matching that can be selected. Here, b_x indicates the birth time of a point x in dgm_1. Similarly, $b_{\sigma(x)}$ and

$b_{\sigma^*(x)}$ are the birth times of x's match $\sigma(x)$ and optimal match $\sigma^*(x)$ in dgm_2.

To accommodate scenarios where the persistence diagrams dgm_1 and dgm_2 contain differing numbers of points, we employ a method that includes the diagonal line $b = d$ (where birth and death times are equal) within the diagrams. This approach allows points that cannot find a match in the opposing diagram to be paired with the diagonal, effectively handling the imbalance. This concept is not only a practical solution but also enhances the stability of the metric, ensuring that the distance calculation remains robust under variations in the diagram data [171].

The computation of matches is akin to the technique used in the sliced Wasserstein distance [173], though simplified in our application to use only one "slice" of the possible distributions, specifically where $d = 0$. This modification focuses on the birth times, streamlining the computation while maintaining effectiveness.

Topological GAN loss defined via matching persistence diagrams. The topological GAN loss is designed to quantitatively measure the difference between the topological features of synthetic and real images, captured in their respective persistence diagrams. Given the complexities and constraints of matching distributions, we devise a straightforward and computationally efficient loss function.

To compute this loss, we identify an optimal matching π^* between the sets of persistence diagrams from the synthetic images $\mathcal{D}_{syn}$ and the real images $\mathcal{D}_{real}$. Each pair of matched diagrams contributes to the overall loss through their diagram distance, denoted as $\mathcal{W}_1$ (referenced in Eq. (3.3)), which calculates the 1-Wasserstein distance between the birth time projections of the two diagrams. We have

$$L_{topo} = \sum\nolimits_{\text{dgm}_i \in \mathcal{D}_{syn}} \mathcal{W}_1\big(\text{dgm}_i, \pi^*(\text{dgm}_i)\big). \tag{3.4}$$

To identify the optimal matching π^* between synthetic and real diagram sets, we employ the optimal transport technique. Denote $\text{dgm}_i^s \in \mathcal{D}_{syn}$ and $\text{dgm}_j^r \in \mathcal{D}_{real}$. Let n_{syn} and n_{real} be the size of $\mathcal{D}_{syn}$ and $\mathcal{D}_{real}$. We solve Monge-Kantorovich's primal problem [170] to find the optimal transport plan:

$$\gamma^* = \min_{\gamma \in \Gamma} \sum_{i=1}^{n_{syn}} \sum_{j=1}^{n_{real}} \mathcal{W}_1(\text{dgm}_i^s, \text{dgm}_j^r) \cdot \gamma_{ij} \tag{3.5}$$

where $\Gamma = \big\{ \gamma \in \mathbb{R}_+^{n_{syn} \times n_{real}} | \gamma \mathbf{1}_{n_{real}} = 1/n_{syn} \cdot \mathbf{1}_{n_{syn}}, \gamma^\intercal \mathbf{1}_{n_{syn}} = 1/n_{real} \cdot \mathbf{1}_{n_{real}} \big\}$. $\mathbf{1}_n$ is an n-dimensional vector of all ones. Denote by γ^* the optimal solution to Eq. (3.5). We compute the optimal matching (π^*) by mapping the i-th

synthetic dgm_i^s to the best matched real diagram w.r.t. the optimal transportation plan, i.e., $\mathrm{dgm}_{h(i)}^r$ such that $h(i) = \arg\max_j \gamma_{ij}^*$ [174]. Formally, $\pi^*(\mathrm{dgm}_i^s) = \mathrm{dgm}_{h(i)}^r$.

3.2.3   Gradient of the Loss

To analyze the gradient of the topological GAN loss function, we commence by considering the equation of the loss, indicated as $L_{topo} = \sum_i L_{topo}^i$, where each individual component $L_{topo}^i = \mathcal{W}_1(\mathrm{DGM}_i, \pi^*(\mathrm{DGM}_i))$ represents the topological loss for each synthetic persistence diagram. The persistence diagram DGM_i is derived from the distance transform applied to the binary mask y_i, which in turn is obtained by thresholding the output of the generator $G(z_i)$.

To compute the gradient with respect to the generator G, it is essential to understand the dynamics of how the binary mask y_i evolves during training. The evolution of y_i can be visualized through the modifications in the synthetic mask and the associated persistence diagram across successive iterations, as illustrated in Fig. 3.4. Each training iteration modifies the mask y progressively towards the critical saddle point s of the distance transform function f. The function value $f(s)$ at this saddle point diminishes towards zero, prompting a corresponding movement of the dot x in the persistence diagram towards the left, thereby decreasing its birth time b_x. This movement effectively narrows the 1-Wasserstein distance between the synthetic and real persistence diagrams, optimizing the topological similarity.

Formally, by chain rule, we have $\frac{\partial L_{topo}^i}{\partial G} = \frac{\partial L_{topo}^i}{\partial \mathrm{dgm}_i} \cdot \frac{\partial \mathrm{dgm}_i}{\partial f_{y_i}} \cdot \frac{\partial f_{y_i}}{\partial G(z_i)} \cdot \frac{\partial G(z_i)}{\partial G}$. Next, we calculate each of the multiplicands.

Derivative of the loss with respect to persistence diagrams. Referring to Eqs. (3.3) and (3.4), the i-th term of the loss, L_{topo}^i, can be expressed as $L_{topo}^i = \sum_{x\in\mathrm{dgm}_i} |b_x - b_{\sigma^*(x)}| = \sum_{x\in\mathrm{dgm}_i} \mathrm{sign}(b_x - b_{\sigma^*(x)})(b_x - b_{\sigma^*(x)})$. This formulation relies on two key optimal mappings, π^* and σ^*. The mapping $\pi^* : \mathcal{D}_{syn} \to \mathcal{D}_{real}$ determines the best alignment through optimal transport between synthetic and real diagram sets, $\mathcal{D}_{syn}$ and $\mathcal{D}_{real}$. Similarly, σ^* provides a 1D optimal transport solution between the points of matched diagrams, mapping each point in dgm_i to its counterpart in $\pi^*(\mathrm{dgm}_i)$.

It is important to note that while the optimal transport plan (γ^* in Eq. (3.5)) adapts continuously with modifications in the synthetic diagrams, the mappings π^* and σ^* generally remain constant except at singular points, which form a set of measure zero. This constancy within a small neighborhood around the input allows us to assume fixed values for π^*, σ^*, and consequently for $\mathrm{sign}(b_x - b_{\sigma^*(x)})$ and $b_{\sigma^*(x)}$.

Therefore, the gradient of the loss relative to the birth times in each point
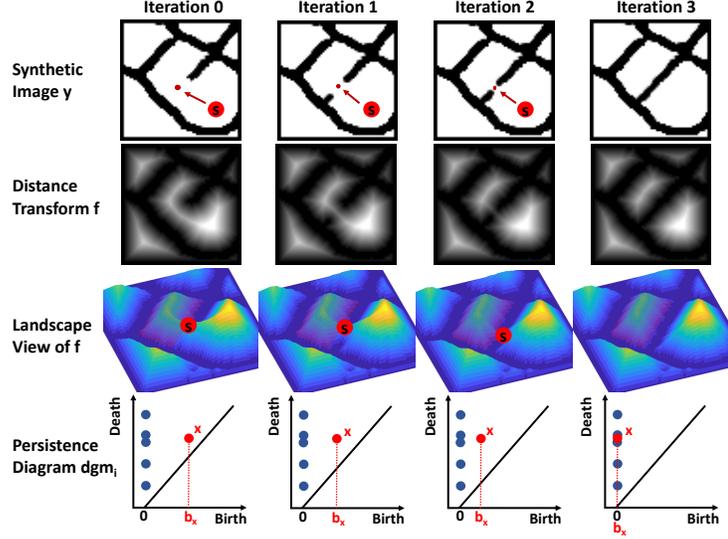
Figure 3.4: From top to bottom: the same synthetic image across various iterations, its distance transforms, landscape views of the distance transforms, and persistence diagrams. The red marker labeled s represents the saddle point; its function value determines the birth time of the near-hole x.

x in dgm_i is precisely given by: $\frac{\partial L^i_{topo}}{\partial b_x} = \mathrm{sign}(b_x - b_{\sigma^*(x)})$, and $\frac{\partial L^i_{topo}}{\partial d_x} = 0$.

This gradient direction, specifically $-\frac{\partial L^i_{topo}}{\partial \mathrm{dgm}_i} \frac{\partial \mathrm{dgm}_i}{\partial G}$, propels each point x in the synthetic diagram dgm_i horizontally towards its optimally matched point in the real diagram, $\sigma^*(x)$, as depicted in Fig. 3.4, emphasizing horizontal adjustments without vertical movement.

Derivative of the persistence diagram with respect to the distance transform. Focusing on the derivative of the birth time b_x with respect to the distance transform f_{y_i}, we note that $\frac{\partial b_x}{\partial f_{y_i}}$ is crucial. Observationally, the birth time b_x of an incipient hole in a filtration corresponds to the value at the saddle point s_x of f_{y_i}, which can be described as $b_x = \langle \delta_{s_x}, f_{y_i} \rangle$, where δ_{s_x} is the Dirac delta function centered at s_x. Therefore, the gradient is simply δ_{s_x}.

In practical terms, the negative gradient $-\frac{\partial L^i_{topo}}{\partial b_x} \frac{\partial b_x}{\partial f_{y_i}} \frac{\partial f_{y_i}}{\partial G}$ aims to adjust the function value at the saddle point $b_x = f_{y_i}(s_x)$, either increasing or decreasing it to align closer to the birth time $b_{\sigma^*(x)}$ of the corresponding point in the matched real diagram, as depicted in Fig. 3.4.

Derivative of the distance transform with respect to the synthetic image $G(z_i)$. The derivative of the distance transform f_{y_i} concerning the synthetic image $G(z_i)$ captures the relationship between adjustments in the

image and changes at the saddle point s_x. To modulate the distance transform value $f_{y_i}(s_x)$, it is necessary to affect the mask y_i at its nearest boundary point to s_x, labeled as r. Alterations to the synthetic image near the pixels around r will facilitate this change, leading to growth or reduction of the mask y_i towards the saddle point. This interaction, vital for guiding the evolution of the mask, is supported by additional calculations provided in the supplementary material and visual examples in Fig. 3.4.

3.2.4 Topology-Aware Metrics for GAN Evaluation

In this study, we introduce two innovative metrics designed to assess GAN performance with a focus on the topological attributes of generated images. Traditional methods like the Inception Score (IS) [163] and the Fréchet Inception Distance (FID) [167] rely on CNN-based image features extracted using a pre-trained Inception network, which may not preserve the topological characteristics of the images. This gap in evaluation motivates our development of topology-centered metrics.

The first metric, termed the *Betti Score*, quantifies the topological variance by utilizing the Betti number, which represents the count of holes within a mask. For both synthetic and real masks, histograms are computed and subsequently compared using the χ^2 distance, facilitating an intuitive yet profound measure of topological disparity.

The second proposed metric leverages persistence diagrams, which capture both holes and near-holes, applying the kernel mean embedding method [175]. By defining a kernel suitable for these diagrams, each set of persistence diagrams—synthetic and real—is transformed into a vector in a Hilbert space, $\mathcal{H}$. The mean of these transformed sets is then computed as $\Phi(\mathcal{D}_{syn}) = \frac{1}{n_{syn}} \sum_{i=1}^{n_{syn}} \Phi(\text{dgm}_i^s)$ and $\Phi(\mathcal{D}_{real}) = \frac{1}{n_{real}} \sum_{i=1}^{n_{real}} \Phi(\text{dgm}_i^r)$. The topological discrepancy between the synthetic and real image sets is then quantified using the Maximum Mean Discrepancy (MMD), defined as $\text{MMD}(\mathcal{D}_{syn}, \mathcal{D}_{real}) = \| \Phi(\mathcal{D}_{syn}) - \Phi(\mathcal{D}_{real}) \|_{\mathcal{H}}$. This metric directly addresses the topological differences, offering a novel perspective on the efficacy and accuracy of GAN-generated imagery in replicating real-world topological features.

It is established that the sample-based Maximum Mean Discrepancy (MMD) converges to its theoretical counterpart, ensuring robustness and reliability for practical applications. In our analysis, we utilize the unbiased version of MMD as outlined in the supplemental material, based on the kernel mean embedding framework [175].

For the kernel function applied to persistence diagrams, several options are available, such as those discussed in recent studies [80, 81]. In this work, we

employ a Gaussian kernel defined by the 1-Wasserstein distance between diagrams, formulated as $k_{\mathcal{W}_1}(\mathrm{dgm}_i, \mathrm{dgm}_j) = \exp\left(-\frac{\mathcal{W}_1(\mathrm{dgm}_i, \mathrm{dgm}_j)}{\sigma^2}\right)$. This choice is motivated by the Gaussian kernel's effectiveness in capturing subtle differences in topological features through the sensitivity of the 1-Wasserstein distance.

Our proposed metrics, the Betti Score and the unbiased MMD, are specifically designed to assess the topological aspects of GAN-generated images. These tools provide a novel perspective on evaluating the quality of GAN outputs, complementing traditional measures such as the Fréchet Inception Distance (FID). To demonstrate the utility of these metrics, we will evaluate a TopoGAN model, analyzing its performance using FID, the unbiased MMD, and the Betti Score. This comprehensive evaluation will highlight the importance of considering topological properties in the assessment of GANs and showcase the added value of our topologically-informed metrics.

3.3 Experiments

TopoGAN is an advanced model based on the Wasserstein GAN with Gradient Penalty (WGAN-GP) framework, utilizing Deep Convolutional Generative Adversarial Networks (DCGANs) as its structural foundation. For a detailed exposition of TopoGAN's implementation, training methodology, and computational considerations, refer to Section B of the supplemental material. This model is assessed against two robust baseline GANs, namely WGAN-GP and Wasserstein GAN with Spectral Normalization (WGAN-SN), which are renowned for their ability to stabilize training and prevent mode collapse in GAN frameworks.

Datasets. To validate the efficacy of TopoGAN, we deploy it across a diverse set of datasets, each chosen to test different aspects of image segmentation capabilities:

- **CREMI** and **ISBI12** [176]: These datasets comprise neuron image segmentation challenges. We utilize 7500 and 1500 patches of size 64x64, respectively, extracted randomly from their segmentation masks to assess the performance in biological image segmentation.

- **Google Maps** (scraped by [68]) and **CMP Facade Database** [177]: These datasets include paired RGB images representing aerial photos to maps and facades to labels, respectively. For our purposes, RGB images are converted to grayscale, from which 4915 patches of size 64x64 are extracted from the maps, and 606 facade labels are resized to 128x128 to facilitate the study of urban image translation.

- **Retina Dataset**: Compiled from four distinct sources—**IOSTAR** (40

Table 3.1: Comparisons against baseline GANs on FID, unbiased MMD, and Betti score across five datasets. The standard deviations are based on 3 runs. We omit reporting unbiased MMD and Betti score of WGAN-SN on Retina as WGAN-SN fails to produce reasonable results.

	CREMI	ISBI12	Retina	Maps	Facade
			FID		
WGAN-GP	21.64±0.138	83.90±0.718	179.69±19.008	72.00±0.469	122.13±0.822
WGAN-SN	34.15±0.153	78.61±0.411	269.12±2.276	175.52±0.217	126.10±1.901
TopoGAN	**20.96**±0.195	**31.90**±0.248	**169.21**±21.976	**60.48**±0.467	**119.11**±0.874
			unbiased MMD		
WGAN-GP	0.142±0.014	0.558±0.010	1.735±0.050	0.482±0.007	0.137±0.004
WGAN-SN	0.326±0.016	0.602±0.006	-	0.724±0.005	0.166±0.005
TopoGAN	**0.134**±0.019	**0.405**±0.003	**1.602**±0.114	**0.471**±0.010	**0.080**±0.002
			Betti score		
WGAN-GP	0.236±0.003	0.908±0.104	0.541±0.188	0.223±0.010	0.176±0.006
WGAN-SN	0.125±0.002	1.775±0.039	-	0.255±0.020	0.142±0.017
TopoGAN	**0.015**±0.001	**0.802**±0.058	**0.457**±0.144	**0.177**±0.004	**0.124**±0.002

images) [178, 179], **DRIVE** (20 images) [180], **STARE** (20 images) [181], and **CHASE_DB1** (28 images) [182]—this dataset features 98 retina segmentations, each standardized to a resolution of 128x128. This dataset is crucial for evaluating the performance in medical image segmentation.

TopoGAN's performance is meticulously analyzed on these datasets to illustrate its practical potential, particularly in segmentation tasks where topological accuracy is paramount. This comprehensive dataset approach not only highlights TopoGAN's robustness across different image types but also underscores its adaptability to varying segmentation challenges.

Quantitative and qualitative results. We present a comprehensive analysis of TopoGAN's performance in comparison with two baseline models, WGAN-GP and WGAN-SN, across three distinct metrics: FID, unbiased MMD, and Betti score, as detailed in Table 3.1. The results underscore TopoGAN's superior capability in maintaining topological properties of the generated images, evidenced by its notable outperformance in the topology-centered metrics, unbiased MMD and Betti score, introduced in Sec. 3.2.4. This enhancement confirms that the topological GAN loss incorporated in TopoGAN effectively promotes structural and topological fidelity in the synthesized images.
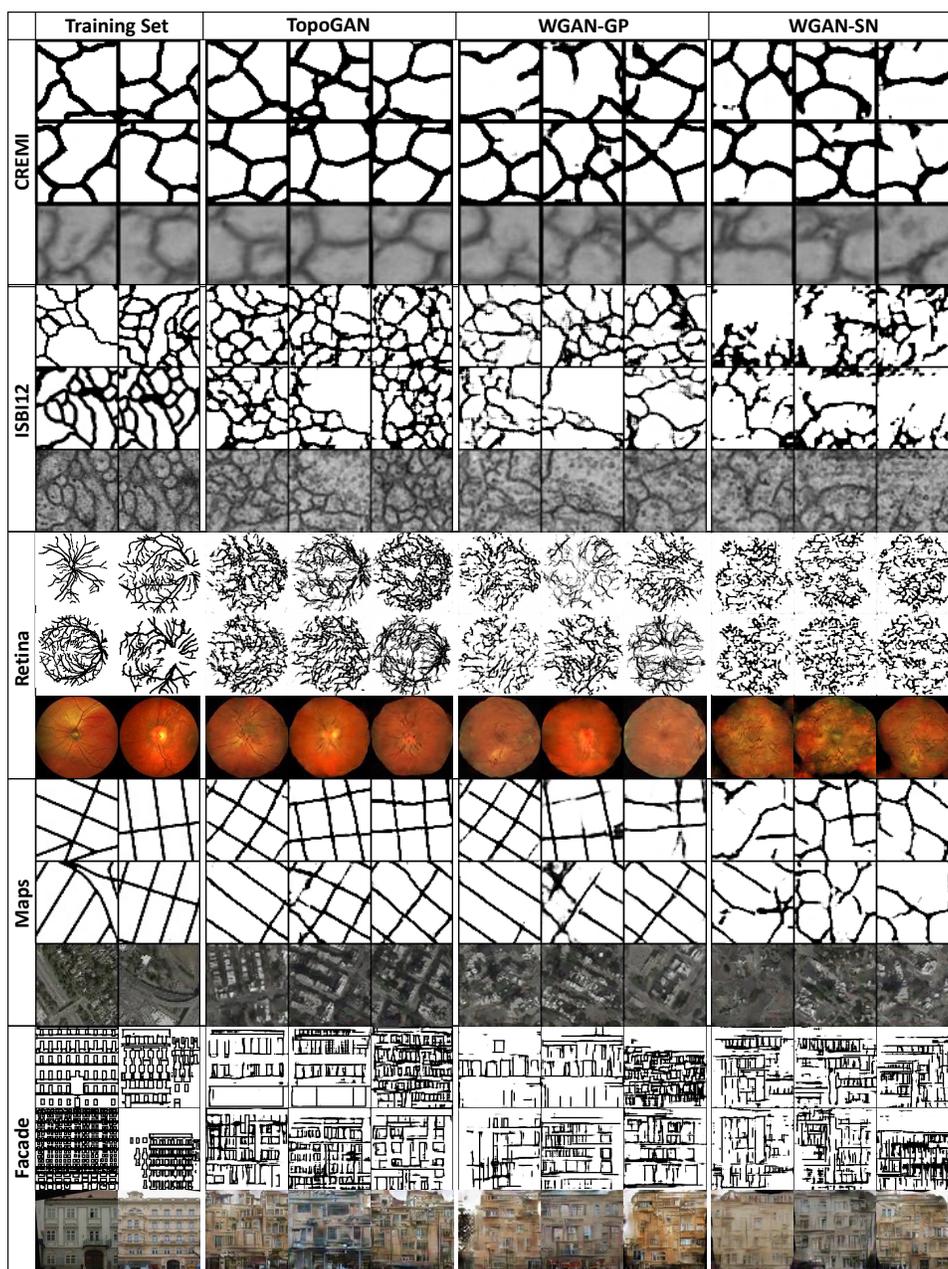
Figure 3.5: Qualitative comparisons across five datasets show real masks, followed by those generated by TopoGAN, WGAN-GP, and WGAN-SN. The third row for each dataset displays textured images corresponding to the synthetic masks above.

Additionally, TopoGAN also exhibits improved performance in the FID metric, suggesting that topological integrity not only preserves structural details but also aligns closely with human visual preferences in assessing image quality. More detailed comparisons of image topological quality across various training epochs are accessible in the supplemental material.

In Fig. 3.5, we display qualitative results using the same noise inputs to ensure fairness in comparing the different GAN methods. The masks generated by TopoGAN showcase clear boundaries and complete cycles, closely mirroring the topological characteristics of the actual data, as indicated by their Betti numbers. Additionally, TopoGAN shows commendable performance in rendering textures, which will be elaborated on later in the document. In contrast, the baseline models, WGAN-GP and WGAN-SN, often produce images with fragmented structural elements.

The **Retina** dataset poses a significant challenge to all models due to its limited size (only 98 images) and the intrinsic diversity in image geometry, resolution, aspect ratio, and contrast, sourced from multiple datasets. Despite these challenges, TopoGAN's ability to reproduce more accurate topological features stands out, further affirming its efficacy in complex segmentation tasks.

Segmentation application. TopoGAN's utility extends beyond image generation to practical applications in binary segmentation tasks. To assess this, we implemented a comparative study across multiple datasets where each dataset was used to train a segmentation network under three different conditions: using real training data, synthetic data generated by TopoGAN, and a combination of real and synthetic data. This setup allowed us to evaluate the effectiveness of synthetic data in enhancing segmentation performance when integrated with real datasets.

Segmentation performance was quantified using three metrics: pixel accuracy, Dice score, and Adapted Rand Index (ARI), with detailed results for the Dice score provided in Table 3.2 and additional metrics discussed in the supplemental material.

To create synthetic training pairs, a pix2pix [68] network—trained on actual data pairs—was used to transform GAN-generated masks into textured masks. These textured masks served as training inputs for a U-Net [183] based segmentation network. This approach leverages the strength of U-Net in handling spatial hierarchies and context, which is critical for accurate segmentation.

We conducted a rigorous evaluation using three-fold cross-validation to ensure the robustness of our findings, reporting both the mean and standard deviation of the Dice scores for the datasets involved. It's important to note

Table 3.2: Dice scores for segmentation networks tested on real data. Across each dataset, we trained 21 segmentation networks using real training data, synthetic data from TopoGAN and two baselines, and real data augmented with synthetic data. Results show the mean and standard deviation from a 3-fold cross-validation.

	CREMI	ISBI12	Retina
Real data	0.896±0.004	0.932±0.011	0.883±0.010
WGAN-GP	0.820±0.018	0.927±0.005	0.891±0.012
WGAN-SN	0.827±0.019	0.902±0.008	-
TopoGAN	0.851±0.011	0.933±0.006	0.892±0.013
WGAN-GP+real data	0.897±0.008	0.943±0.007	0.899±0.010
WGAN-SN+real data	0.900±0.004	0.905±0.054	-
TopoGAN+real data	**0.902**±0.006	**0.944**±0.008	**0.906**±0.014

that the segmentation studies were limited to the **CREMI**, **ISBI12**, and **Retina** datasets. The other datasets, namely **Google Maps** and **CMP Facade Database**, were excluded from this part of the study due to the absence of ground truth data necessary for segmentation tasks.

This structured approach not only highlights TopoGAN's potential in generating high-quality synthetic data that is useful for training segmentation models but also showcases how such synthetic data can be seamlessly integrated into existing workflows to improve segmentation accuracy across different medical and biological imaging scenarios.

3.4 Literature Review: GANs, Diffusion Models, and TopoGAN

3.4.1 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) were first introduced by Ian Goodfellow et al. in 2014 [184]. GANs consist of two neural networks, a generator and a discriminator, that are trained simultaneously through adversarial processes. The generator creates fake data samples, while the discriminator attempts to distinguish between real and fake samples. This setup leads to the generator producing increasingly realistic data over time.

One of the significant advancements in GANs is the development of Deep Convolutional GANs (DCGANs) [185]. DCGANs use convolutional layers to

improve the quality of generated images and have become a standard architecture for image generation tasks.

Another notable advancement is the introduction of Conditional GANs (cGANs) [186], which allow for the generation of data conditioned on specific inputs. This makes it possible to generate images of a particular class or style, significantly expanding the applicability of GANs.

The Wasserstein GAN (WGAN) [187] introduced a new loss function based on the Wasserstein distance, addressing the problem of mode collapse and improving the stability of GAN training. The WGAN-GP variant [188] further enhanced this approach by incorporating a gradient penalty to enforce the Lipschitz constraint.

In recent years, GANs have seen remarkable improvements in generating high-resolution images. The Progressive Growing of GANs (ProGAN) [189] and StyleGAN [190] architectures have set new benchmarks in image synthesis by progressively increasing the resolution during training and introducing style transfer capabilities within the network.

Recent works have also focused on improving the training dynamics and efficiency of GANs. Techniques such as spectral normalization [191] and self-attention mechanisms [192] have been proposed to stabilize GAN training and enhance the generation quality, especially in complex datasets.

Furthermore, the application of GANs has expanded beyond image generation. GANs are now being used in diverse fields such as text-to-image synthesis [193], 3D object generation [194], and even drug discovery [195].

3.4.2 Diffusion Models

Diffusion models have emerged as powerful generative models, particularly noted for their ability to generate high-quality data through a process of gradually denoising a sample from pure noise. The concept of diffusion probabilistic models was first introduced by Sohl-Dickstein et al. in 2015 [196]. These models use a forward process to gradually add noise to data and a reverse process to denoise it, thereby generating new samples.

A significant advancement in the field is the introduction of Denoising Diffusion Probabilistic Models (DDPMs) by Ho et al. [197]. DDPMs use a specific noise schedule and parameterization that result in high-quality sample generation. This model has set new benchmarks in image synthesis, demonstrating superior performance over GANs in certain tasks.

The development of improved training techniques and architectures has further enhanced diffusion models. Nichol and Dhariwal introduced improved denoising diffusion probabilistic models [198], which incorporated a more flexible noise schedule and improved model architectures, achieving even better

sample quality and faster training.

The application of diffusion models has expanded beyond image generation. Song and Ermon proposed the concept of score-based generative models [199], which unify score matching and generative modeling. This approach has been successfully applied to various data modalities, including images and audio.

Another notable contribution is the work on continuous-time diffusion models by Song et al. [200], where the authors extended the framework to continuous time, allowing for more flexible and potentially more powerful generative models. This approach, known as Score-Based Generative Modeling through Stochastic Differential Equations (SDEs), has opened new avenues for research in generative modeling.

Recent advancements have also focused on improving the computational efficiency of diffusion models. Techniques such as fast sampling methods [201] and optimization strategies [202] have been proposed to reduce the computational cost while maintaining high sample quality.

Diffusion models have shown remarkable potential in various applications, including text-to-image synthesis [203], image super-resolution [204], and even molecular generation [205]. These models continue to evolve, with ongoing research aimed at further enhancing their capabilities and expanding their applicability.

3.4.3 TopoGAN

Multiplex immunofluorescence (MxIF) imaging relies heavily on effective cell segmentation, which is challenging due to the use of multiple membrane markers that can vary in stain quality. Bao et al. proposes a deep learning-based membrane segmentation method that aggregates information from large-scale MxIF markers to improve topology preservation and segmentation accuracy, significantly outperforming traditional methods in clDiceSKEL and ARI metrics [206]. Arnavaz et al. [207] addresses topological consistency in segmentation and limited annotations in biomedical imaging by proposing a topological score to measure both topological and geometric consistency between predicted and ground truth segmentations. The topological score is applied to three scenarios involving U-net architectures, including a semi-supervised approach that utilizes un-annotated data, demonstrating effectiveness in segmenting tubular structures in the fetal pancreas from noisy live imaging confocal microscopy.

Segmentation networks often fail to learn global invariants like object shape and geometry when trained with standard loss functions. Ozcelik et al. introduces a topology-aware loss function that uses persistent homology and Vietoris-Rips filtration to penalize topological dissimilarities, improving seg-

mentation accuracy by modeling both shape and geometry simultaneously [208]. Experiments on 4327 CT images of aorta and great vessels demonstrate the effectiveness of this approach, showing superior performance compared to traditional methods.

In digital pathology, modeling the complex spatial context of cells is crucial for accurate classification and cancer diagnosis. Abousamra et al. introduces mathematical tools from spatial statistics and topological data analysis into a deep generative model, using them as conditional inputs and a differentiable loss to generate high-quality multi-class cell layouts [209]. The topology-rich cell layouts improve data augmentation and enhance the performance of downstream tasks like cell classification.

To the best of the author's knowledge, there are no existing works in the literature that improve upon or extend TopoGAN. The works citing TopoGAN primarily apply it to various tasks.

3.5 Conclusion

This study introduces TopoGAN, an innovative GAN methodology that explicitly focuses on learning the topological aspects of images from real data. We have developed a unique topological GAN loss, demonstrating its differentiability and seamless integration into standard GAN training processes. Alongside, we have proposed novel metrics specifically designed to assess topological discrepancies between generated and real images.

Empirical results from our study highlight TopoGAN's superior performance in capturing and replicating topological features compared to existing state-of-the-art GAN models. Our evaluations, both quantitative and qualitative, consistently show that images generated by TopoGAN maintain higher topological fidelity. In segmentation tasks, TopoGAN with solely synthetic data has shown results comparable to those obtained using real data, particularly evident in the **ISBI12** and **Retina** datasets. Moreover, when synthetic data is used to augment real data, there is a marked improvement in segmentation performance, underscoring the value of integrating synthetic data into training workflows.

The integration of TopoGAN and real data consistently outperformed other combinations, with WGAN-GP augmented with real data as a close second. Comprehensive details regarding the evaluation metrics, the training procedures for segmentation networks, and a complete set of results are available in the supplementary materials.

This contribution is poised to set a new benchmark in the field of generative modeling, particularly in applications where understanding and preserving the

underlying topology of the data is crucial.

Chapter 4

Topology-Guided CNN for Breast parenchyma learning on DCE-MRIs

4.1 Introduction

As the second application of persistent homology, this thesis utilizes topological structures derived from persistent homology as approximations of breast tissues to predict responses to breast cancer treatment. Breast cancer imaging's endeavor to precisely depict the intricacies of breast parenchyma, which is subject to dynamic changes due to angiogenesis, radiation, and chemotherapy, underscores a pivotal challenge. The deployment of 3D imaging technologies, notably MRI, stands out as a crucial advancement in capturing these complex transformations, offering profound implications for diagnosis, prognosis, and the orchestration of treatment strategies. Historically, the focus within cancer imaging research has gravitated towards the analysis of tumor texture and shape, inadvertently sidelining the rich data reservoir present in the tumor microenvironment. This oversight neglects the critical diagnostic and prognostic value embedded within the peritumoral stroma and parenchyma, where phenotypic variations are fuelled by immune responses, vascularity, and compositional diversity of the tissue. Notably, the patterns of fibroglandular tissue and the enhancements within the parenchyma are indicative of breast cancer risks and the efficacy of treatment protocols, emphasizing the need for advanced quantitative tools that delve into the tumor milieu and its environs, as observable in breast MRI scans.

The spectrum of methodologies proposed for the analysis of breast imagery is broad. Radiomic strategies, for instance, distil diagnostic and prognostic
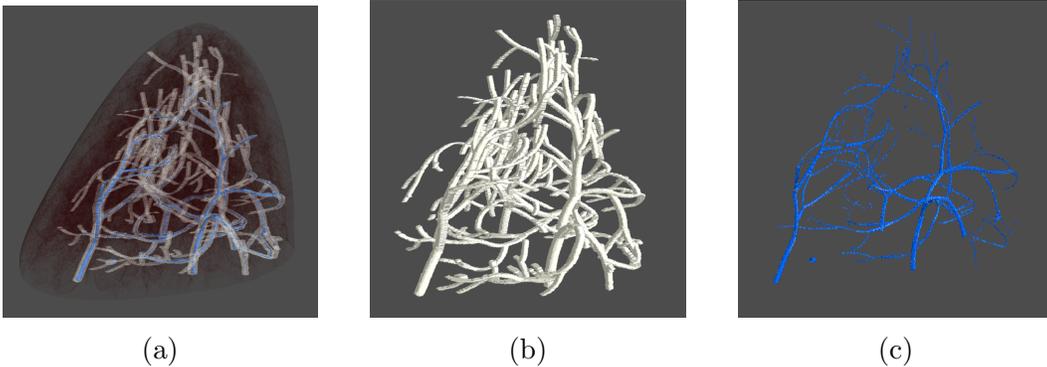
Figure 4.1: (a): 3D visualization of a phantom breast, showcasing glandular tissues (in white) and topological features (in blue); (b): Glandular tissues; (c): Topological features.

markers from both the tumor and its adjacent peritumoral areas through the lens of radiomic features [210, 211]. These features, though rooted in human-derived insights, strive to quantify diverse aspects such as texture within and around the tumor, geometrical attributes of vessels, among others. Despite their intuitiveness, radiomic approaches are hampered by two primary drawbacks: a shortfall in directly mapping the complex structural dynamics of the peritumoral stroma and parenchyma, and a rigidity in the handcrafted features that stifles their ability to capture the heterogeneous nature of breast parenchyma adequately, thus limiting their practical predictive capabilities albeit their high interpretability.

Conversely, data-driven methodologies, particularly convolutional neural networks (CNNs), have demonstrated substantial efficacy across various analytical domains by fostering an environment for end-to-end learning of feature representations [106, 212–217]. Although applying CNNs directly to MRI data presents a promising avenue [218–225], this approach is not without its challenges. Specifically, when the entirety of the MRI volume serves as the input, there's a risk of incorporating biologically extraneous or even misleading data that could skew the model's outcomes. Moreover, the architectural complexity and parameter density of 3D CNNs necessitate a considerable volume of training data — a requirement that often surpasses the feasible limits of controlled clinical studies, such as the I-SPY1 trial [226]. An additional limitation inherent to CNNs lies in the opacity of feature interpretability, as there is a disconnect between the learned features and the tangible structures of breast tissue they aim to represent.

In our study, we introduce *TopoTxR*, an innovative approach designed to surmount the challenges previously discussed. This methodology leverages
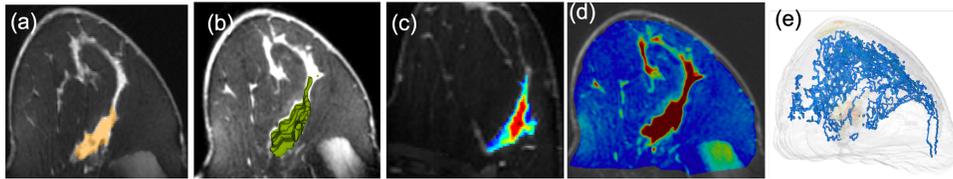
Figure 4.2: (a) A sample MRI image alongside radiomic features: (b) 3D tumor shape, (c) intratumoral texture (Haralick entropy), (d) overall breast texture (Haralick energy). Panel (e) presents topological structures identified by *TopoTxR*, highlighting the fibroglandular tissue geometry.

the principles of topology to discern the structures of the breast parenchyma, seamlessly integrating these sophisticated topological insights into deep convolutional neural networks (CNNs) to notably boost predictive accuracy. By concentrating on the parenchymal structures, our strategy ensures that predictive outcomes are anchored in biological realities, thus markedly elevating the efficacy of CNN models, even when training data is scarce.

At the heart of *TopoTxR* is the concept of persistent homology [113], a mathematical framework adept at identifying 1D (loops) and 2D (bubbles) topological features with a high degree of reliability [227]. These identified features are reflective of the breast's curvilinear tissue formations (such as ducts and vessels) and the spaces encircled by tissues and glands. This approach stands in contrast to traditional radiomics, offering a more nuanced and comprehensive view of the tumor microenvironment by providing a deeper structural context. Given that these topological structures are derived through unsupervised methods, the interpretative quality of these structures becomes paramount. Utilizing the phantom breast imaging dataset from VICTRE [228], we conduct both quantitative and qualitative evaluations to affirm that the topological features we extract are indeed representative of actual breast tissue structures. The visualization of these topological constructs, as showcased, effectively outlines the glandular textures within a phantom breast image, offering a vivid illustration of our method's capability to accurately model the complex biological architecture of the breast.

In this work, we unveil a groundbreaking topology-guided deep learning framework tailored for breast imaging analysis. Central to our methodology is the innovative use of topological structures to refine the model's focus towards voxels in close proximity to these critical tissue formations. By narrowing the attention to a select group of voxels with heightened biological significance, our model achieves efficient training with a limited dataset of MRI scans. Moreover, this focused approach opens avenues for linking the biological un-

derpinnings of various breast pathologies to their topological representations within tissue structures. Notably, our method enhances the predictive precision beyond what is achievable using Betti curves, as demonstrated in related research by [119]. Betti curves, while useful, capture a narrower spectrum of topological information, thus limiting their efficacy in comparison to the more detailed topological constructs we employ.

Our topology-centric model is versatile, yet we particularly apply it to the prediction of responses to neoadjuvant chemotherapy (NAC) in breast cancer treatments. Accurately forecasting the pathological complete response (pCR) before initiating NAC can avert inefficacious treatments, thereby sparing patients from undue distress and reducing healthcare expenditures. Through empirical assessments on the I-SPY1 dataset [226] and an in-house proprietary dataset, our approach, *TopoTxR*, consistently outshines various benchmark models. This includes traditional radiomic methodologies, CNNs devoid of topological insights, and other cutting-edge strategies, underscoring the superior predictive prowess of our topologically informed approach.

In encapsulating our contributions, we highlight the following key aspects: - The adoption of persistent homology to delineate topological structures, offering a refined approximation of the breast's fibroglandular tissue architecture. - A thorough validation conducted with a phantom breast imaging dataset, substantiating the fidelity of our topological extractions in mirroring actual breast tissues. - The introduction of a topology-driven spatial attention mechanism that augments the focus and efficiency of 3D CNNs, thereby bolstering their predictive accuracy.

By bridging the gap between traditional engineered imaging features and fully autonomous data-driven CNN methodologies, our work presents a novel pathway in the predictive analysis of breast DCE-MRI, specifically targeting the prognostication of pCR.

4.2   Methodology

Our approach introduces a topological framework designed to significantly enhance the analysis of breast DCE-MRI images, with the dual objectives of: (1) extracting salient topological structures as precise representations of the underlying tissue structures, and (2) leveraging these extracted topological cues to inform and guide the training of a deep convolutional neural network (CNN). This methodology is visually encapsulated in Fig. 4.3, illustrating our comprehensive workflow.

The process begins with the identification of salient topological structures within the input image, utilizing the foundational principles of persistent ho-
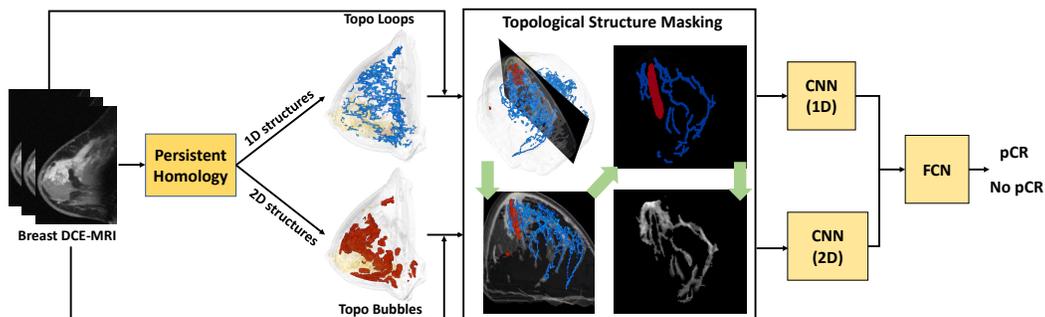
Figure 4.3: Our TopoTxR pipeline: Persistent homology extracts 1D and 2D topological features from breast MRI images. New images are created where voxels within these features retain their original MRI intensity values, and all other voxels are set to zero. Two images are generated, one each for 1D and 2D structures. These images are then processed by two separate 3D CNNs, followed by a fully connected network for predicting pCR response.

mology [113]. We concentrate on the extraction of 1D and 2D topological features, namely loops and bubbles, which are emblematic of significant tissue structures. The 1D features, or loops, are adept at capturing curvilinear structures such as ducts and vessels, integral to the organ's vascular and structural framework. Conversely, the 2D features, or bubbles, highlight voids that are encased by tissue and associated glands, further delineating spaces of interest within the tissue matrix. These topological constructs are invaluable, as they directly highlight tissue structures of paramount biological relevance. Our hypothesis posits that concentrating on these specific structures and their associated regions will furnish the model with contextually rich information conducive to accurate prediction.

Building upon this foundation, we introduce a novel topological-cycle-driven CNN architecture, meticulously designed to process MRIs alongside the identified topological structures. This method involves masking the MRI input in such a manner that only voxels corresponding to, or in the immediate vicinity of, the topological structures are retained for further processing. Subsequently, a 3D CNN is trained using these strategically masked images. This focused approach allows for effective model training even with a restricted dataset, by honing in on regions proximal to the tissue structures of interest.

It is critical to note that we distinguish between two primary types of topological structures: loops and bubbles. Accordingly, our architecture incorporates two distinct 3D CNNs, each dedicated to one of these topological types. The insights gleaned from each CNN are then amalgamated, feeding into a

series of fully connected layers tasked with the binary classification challenge of predicting pathological complete response (pCR) versus non-pCR. Empirical evidence supports the assertion that both types of topological features are indispensable, collectively contributing to a robust and nuanced structural understanding that underpins superior predictive accuracy.

This section will proceed by elucidating the underlying theory of persistent homology, followed by an introduction to the cycles that symbolize the topological structures under consideration. Finally, we will detail the operational mechanics of our topological-cycle-driven CNN, setting the stage for a deeper dive into the intricacies of this pioneering approach.

4.2.1 Background: Persistent Homology

In this foundational overview, we delve into the essence of persistent homology, a pivotal concept in topological data analysis, which serves as a powerful tool for uncovering and quantifying the inherent topological characteristics of data. For those seeking a deeper dive into the subject, the seminal work by Edelsbrunner and Harer [113] is an invaluable resource, offering comprehensive insights into the theoretical underpinnings and applications of persistent homology.
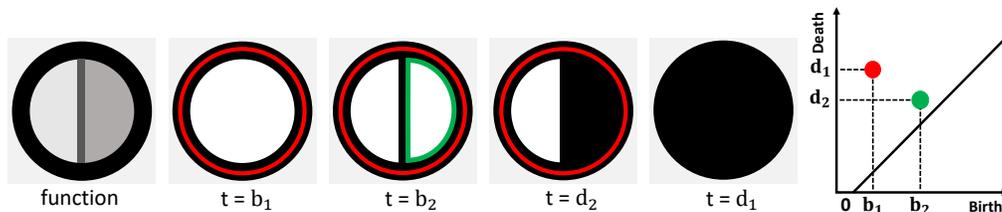
Figure 4.4: Left to right: Synthetic image f, sublevel sets at thresholds $b_1 < b_2 < d_2 < d_1$ (in black), and the 1D persistence diagram. A red loop, born at b_1 and dying at d_1, and a green loop, born at b_2 and dying at d_2, represent 1D topological features. These features correspond to the red and green points in the diagram.

Persistent homology is adept at extracting the topological footprint of a dataset observed through a scalar function. Given an image domain X and a real-valued function $f : X \to \mathbb{R}$, one can construct sublevel sets $X_t = \{x \in X : f(x) \leq t\}$, where t acts as a dynamic threshold modulating the evolution of these sets over time. This series of sublevel sets, $\mathcal{X} = \{X_t\}_{t \in \mathbb{R}}$, forms a filtration—a hierarchy of nested subsets of X that progressively reveals

the emergence and dissolution of topological features such as connected components, handles, and voids. The "birth" of a topological feature is marked by the threshold t at which it first appears in the filtration, whereas its "death" corresponds to the threshold at which it vanishes. The life-cycle of these features, from inception to dissolution, is meticulously cataloged in a persistence diagram, a scatter plot where each point (b, d) represents a feature with birth time b and death time d.

An illustrative example is provided in Fig. 4.4, showcasing how an example function f and its sublevel sets at varying thresholds lead to the creation and subsequent destruction of handles—topological structures represented as loops within the diagram. Each handle corresponds to a dot in the persistence diagram, positioned according to its birth and death times, offering a visual summary of the topological evolution captured by persistent homology. The "persistence" of a dot, the interval between its birth and death, quantifies the endurance of the corresponding topological feature within the dataset.

4.2.2 Persistence Cycles and Their Computation

Building upon this foundation, we introduce the concept of persistence cycles, geometric manifestations of the topological structures unearthed by persistent homology. These cycles, depending on their dimensionality (1D or 2D), represent loops or bubbles—each encapsulating essential aspects of the dataset's topological anatomy at the moment of their emergence. For instance, the loops depicted in Fig. 4.4 symbolize handles at their respective birth times, serving as tangible markers of the dataset's topological characteristics.

This exposition sets the stage for a deeper exploration of the computational techniques employed to identify and analyze these persistence cycles, offering a window into the intricate dance of topology that unfolds within complex datasets.

In the detailed exploration of topological data analysis within the context of image domains, we encounter a discrete landscape composed of vertices (akin to voxels), edges (which connect vertices), squares, and cubes. These elements are categorized by their dimensionality—vertices as 0-dimensional, edges as 1-dimensional, squares as 2-dimensional, and cubes as 3-dimensional entities. Collectively, a group of p-dimensional cells is termed a p-chain, serving as the building blocks for constructing complex topological structures.

The concept of a boundary operator is pivotal in understanding the architecture of these p-chains. For a given p-cell, σ, the boundary operator identifies its $(p-1)$-dimensional faces. The boundary of an edge, therefore, consists of its two endpoint vertices; a square's boundary is defined by the four edges that enclose it, and similarly, a cube's boundary is determined by the six

squares that form its exterior. The boundary of a p-chain, c, is mathematically expressed as $\partial(c) = \sum_{\sigma \in c} \partial(\sigma)$, where the summation operates under mod-2 arithmetic, highlighting the combinatorial nature of these constructions.
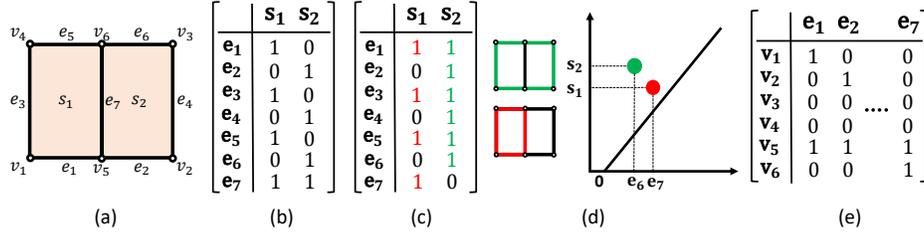
Figure 4.5: (a) Cubical complex example with cells ordered by function values. (b) 2D boundary matrix ∂. (c) Reduced boundary matrix. (d) Persistence diagram showing cycles corresponding to ∂. (e) 1D boundary matrix.

A p-chain assumes the status of a p-cycle when it encapsulates an empty boundary, representing a closed loop or shell devoid of any gaps. Such p-cycles constitute the null space of the boundary operator, essentially forming the kernel of this linear transformation. The equivalence of different cycles delineating the same topological structure—termed a homology class—underscores the flexibility in representing topological features within this framework. Each persistent homology class, visualized as a dot in the persistence diagram, can thus be represented by any one of its constituent cycles that emerge at the class's birth.

The representation of topological features by cycles is not rigidly fixed; various cycles can equivalently characterize the same homology class, offering multiple perspectives on the underlying topological essence. A pertinent aspect of this representation involves selecting the most concise representative cycle for each feature, typically the cycle with the minimum number of edges, to efficiently encapsulate the topological information. While the focus of this paper is on establishing a foundational approach for selecting standard representative cycles, the optimization for the shortest cycle remains an area earmarked for future exploration, as suggested by research aiming at heuristic and optimal cycle representations [84, 229]. This consideration opens avenues for refining the computational efficiency and interpretability of topological features extracted from image data, paving the way for more nuanced and effective applications in image analysis and beyond.

4.2.3 Computation of Persistent Homology and Representative Cycles

We initiate the process with a filtration function applied to a discretized representation of the image domain. An illustrative example of such a discretization for a 2D image is provided in Fig. 4.5(a), where we first organize all cells in ascending order based on their corresponding function values. The computation of persistence diagrams then proceeds by constructing *boundary matrices*, ∂_p, which map p-cells to their boundaries. An instance of the 1D and 2D boundary matrices for the specified complex and its filtration is showcased in Fig. 4.5, with the 1D boundary matrix effectively acting as the incidence matrix of the graph depicted in Fig. 4.5(e).

High-dimensional boundary matrices are defined in a similar manner, for example, the 2D boundary matrix illustrated in Fig. 4.5(b).

The persistence diagram is derived through the reduction of the boundary matrix, a process reminiscent of Gaussian elimination but tailored to prevent row or column perturbation. This reduction involves column operations on ∂ from left to right, with the reduced 2D boundary matrix presented in Fig. 4.5(c).

Upon the reduction of boundary matrices, each non-zero column signifies a persistent dot within the diagram, where the column itself embodies the cycle representing the associated topological structure. Our investigation encompasses both 1D and 2D cycles, corresponding to loops and bubbles, respectively. These extracted cycles are instrumental in directing 3D CNNs for subsequent analysis.

The computational effort required to derive topological cycles is commensurate with that of calculating persistent homology. Theoretically, this computation demands $O(n^\omega)$ time, where $\omega \approx 2.37$ denotes the exponent in matrix multiplication time—specifically, the time needed to multiply two $n \times n$ matrices. Here, n represents the number of voxels within an image. Practically, computing all cycles for an input image of size 256^3 takes approximately 5 minutes, as cited by [230].

4.2.4 Enhanced 3D CNN Leveraging Topological Insights

In the initial discussions within Section 4.2, we introduced the concept behind our enhanced 3D Convolutional Neural Network (CNN) that incorporates topological data for improved analysis. Here, we delve into the more nuanced aspects of this methodology.

The foundational step involves modifying the MRI image representation through the application of an inversion function $f = -I$. This inversion ensures that areas of significant tissue structures are highlighted by assigning them lower intensity values, facilitating their identification in subsequent analyses. Following this inversion, our focus shifts to the identification of topological cycles within the persistence diagram that exhibit substantial persistence. It is widely accepted in the field that features associated with lower persistence are predominantly noise artifacts; thus, our selection process prioritizes cycles with higher persistence. These are deemed to embody the most critical structures, accurately reflecting genuine tissue configurations. The selection threshold for these cycles is determined by a hyperparameter, which is finely adjusted based on empirical evidence.

Subsequently, we construct two distinct binary 3D masks, each corresponding to either 1D or 2D topological cycles identified earlier. These masks undergo a slight dilation process to not only encompass the identified structures themselves but also their immediate surroundings, thereby capturing a more comprehensive view of the area of interest. Rather than employing these masks in their binary form for the prediction of pathological complete response (pCR), we opt to reintegrate the original intensity values of the foreground voxels within these masks. This approach effectively utilizes the cycle masks to highlight relevant areas within the MRI images, as illustrated in the *Topological Structure Masking* segment depicted in Fig. 4.3. Masked images generated for both 1D and 2D cycles are then fed into their respective CNNs for further processing. To ensure uniformity, all masked MRIs are standardized to a dimension of $256 \times 256 \times 256$.

Our analytical framework employs two parallel CNNs, each designed to handle the data derived from cycles and bubbles separately, yet sharing a common architectural blueprint. This architecture comprises five layers of 3D convolution, succeeded by layers of batch normalization and LeakyReLU activation. The feature maps generated by these convolutional networks are then merged and transformed into a cohesive feature vector. This vector subsequently navigates through a fully connected network, consisting of three layers, culminating in the final pCR prediction. The inclusion of ReLU activation, batch normalization, and a dropout layer (positioned within the second fully connected layer) ensures robustness and generalization of the model. Training of the integrated network model employs a stochastic gradient descent (SGD) optimizer alongside a cross-entropy loss function, emphasizing an end-to-end training approach that harmonizes all components towards achieving superior predictive performance.

Table 4.1: Comparison of proposed method vs. baseline methods across four metrics: accuracy, AUC, specificity, sensitivity. Last row p-values calculated between baseline MRI and TopoTxR.

	Accuracy	AUC	Specificity	Sensitivity
	Without Feature Selection			
Radiomics	0.517±0.086	0.536±0.098	0.557±0.058	0.477±0.176
PD	0.529±0.071	0.537±0.078	0.543±0.075	0.515±0.151
Radiomics+PD	0.533±0.080	0.538±0.095	0.567±0.065	0.5±0.175
	With Feature Selection			
Radiomics	0.563±0.085	0.593±0.098	0.552±0.180	0.575±0.081
PD	0.549±0.081	0.567±0.097	0.551±0.167	0.547±0.071
Radiomics+PD	0.563±0.093	0.587±0.099	0.592±0.178	0.534±0.087
	3D CNN			
MRI	0.633±0.200	0.621±0.102	0.570±0.322	0.673±0.354
TopoTxR (MRI+Topo)	**0.851**±0.045	**0.820**±0.035	**0.736**±0.086	**0.904**±0.068
p-value	0.0625	0.0625	0.3750	0.1875

4.3 Experimental Results

We conducted an assessment of our proposed methodology on the challenge of pCR prediction, utilizing the ISPY-1 post-contrast DCE-MRI dataset [226]. This dataset includes a cohort of 162 patients, among which 47 achieved pCR (average age = 48.8 years) and 115 did not achieve pCR (average age = 48.5 years). The evaluation framework was structured around a 5-fold cross-validation approach. Metrics for evaluation comprised accuracy, area under the curve (AUC), specificity, and sensitivity, with reporting on both the mean and standard deviation for these metrics.

In the optimization phase, critical hyperparameters such as the learning rate, momentum, weight decay factor, batch size, and dropout rate were fine-tuned through a grid search mechanism. This search was conducted within a 3-fold cross-validation on a selectively held-out validation subset.

The performance of our approach was benchmarked against various established methods: - **Radiomics**: A radiomic signature encompassing 92 dimensions was computed [231], focusing exclusively on features extracted from the tumor region. A classifier was subsequently trained on this signature. -

Table 4.2: Ablation study results. All numbers are reported from 5-fold cross validations.

	Accuracy	AUC	Specificity	Sensitivity
	Persistence Threshold			
90% Remain	0.826±0.069	0.783±0.063	0.675±0.1115	0.891±0.084
60% Remain	**0.851**±0.021	0.793±0.028	0.647±0.073	**0.939**±0.017
	Dimension			
Dimension 1	0.718±0.068	0.697±0.025	0.639±0.149	0.754±0.161
Dimension 2	0.756±0.036	0.691±0.013	0.520±0.116	0.863±0.103
	Dilation Radius			
Radius 2	0.721±0.036	0.673±0.024	0.569±0.037	0.777±0.055
Radius 4	0.677±0.023	0.603±0.007	0.442±0.063	0.764±0.054
Radius 8	0.646±0.034	0.569±0.040	0.399±0.057	0.737±0.033

PD: Features based on the persistence diagrams (PDs) of input MRI images were used to train a classifier. While several classifier options were explored, the sliced Wasserstein kernel distance was employed for PDs as a feature vector [81]. - **Radiomics+PD**: This approach integrated features from both the radiomic signature and PDs, with a classifier trained on the combined feature set. - **With feature selection**: Feature selection techniques, specifically Mutual Information Difference (MID) and Mutual Information Quotient (MIQ), were applied to all methods mentioned above. An exhaustive search among various combinations of feature selection schemes and classifiers (including Random Forests, Linear Discriminant Analysis, Quadratic Discriminant Analysis, and SVM) was conducted to identify optimal configurations. The most favorable outcomes are reported. - **MRI**: A direct application of a 3D CNN to the original DCE-MRIs was also evaluated.

Quantitative results. The integration of Radiomics and PD features exhibited enhanced performance when combined with a Random Forest classifier, as detailed in Table 4.1. It was noted that the straightforward implementation of a 3D CNN (referred to as the MRI method) underperformed, likely due to the limited size of the dataset. Our novel approach, *TopoTxR* (MRI+Topo), surpassed all compared baseline methodologies in effectiveness. Given the dataset's inherent imbalance, we also present the specificity and sensitivity metrics for each classifier. Future investigations will aim to further mitigate
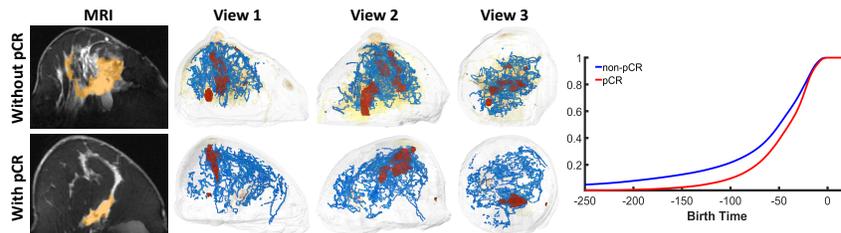
Figure 4.6: Qualitative analysis of patients with and without pCR. First column shows breast DCE-MRI slices with tumors highlighted in orange (tumor masks not used in TopoTxR creation). Columns 2-4 depict 3D visualizations of topological structures from various angles, with 1-D (loops) in blue and 2-D (bubbles) in red. Top row indicates absence of pCR; bottom row shows presence of pCR. Rightmost column presents the cumulative density function of the birth times of topological structures.

the effects of data imbalance.

Ablation study. The concept of persistence within topological data analysis—defined as the lifespan from a structure's emergence to its dissolution—serves as a filter to exclude transient structures, typically attributed to noise, that might otherwise detract from the analysis's accuracy. We examined the influence of varying persistence thresholds, setting these so as to retain 90%, 60%, and 30% of the topological structures, respectively. As indicated in Table 4.2, the preservation of 30% of the structures (as also referenced in the results for TopoTxR in Table 4.1) facilitates an optimal balance, enhancing the relevance and impact of the retained topological features. Furthermore, evaluations were conducted to gauge the predictive value of using exclusively 1D (loops) or 2D (bubbles) structures. While each demonstrated superiority over baseline models, neither approached the efficacy of the comprehensive *TopoTxR* approach, underscoring the complementary nature of the 1D and 2D topological information. An additional aspect of our analysis involved the exploration of the effects of dilation applied to the topological structure masks, with optimal outcomes achieved in the absence of any dilation. The reported performance of *TopoTxR* in Table 4.1 reflects the implementation with 30% of the structures maintained, leveraging both 1D and 2D structures, without any dilation.

4.3.1 Discussion and TopoTxR Feature Interpretation

The topological structures identified by TopoTxR provide meaningful representations of breast tissue configurations, essential for interpreting learning outcomes and generating novel biological insights. These insights derive from focusing the analysis directly on the topological features and their surrounding areas.

Fig. 4.6 displays topographical representations of the topological structures identified by TopoTxR, offering different perspectives for a representative DCE-MRI scan from each patient group. Observations reveal that the structures (both 1D and 2D) tend to be sparse in cases exhibiting pathological complete response (pCR) and relatively dense in non-pCR cases. In the accompanying MRI images, it is noted that the breast with pCR exhibits scattered fibroglandular density with minimal background parenchymal enhancement. Conversely, the non-pCR breast displays a more heterogeneous fibroglandular density coupled with moderate background parenchymal enhancement. These observations suggest that the features extracted by *TopoTxR* adeptly capture the intricate fibroglandular structures, potentially serving as indicators of treatment response.

Further analysis focuses on comparing the topological behavior between the two patient groups. Given that the imaging is processed using an inverted function $f = -I$, the birth time of a topological structure, which typically denotes the threshold at which a cycle appears, inversely reflects the brightness of the structure. In Fig. 4.6 (right), the cumulative density function (CDF) of the birth times for the topological structures from pCR (depicted in red) and non-pCR (depicted in blue) patients is plotted. The CDFs illustrate that the tissue structures in pCR patients generally manifest as less bright (or less visible) compared to those in non-pCR patients, corroborating the qualitative observations previously mentioned. A Kolmogorov-Smirnov test [232] conducted to statistically compare these CDFs yielded a *p-value* of 0.0002, confirming a significant distinction in the distribution of the birth times between the pCR and non-pCR groups, thereby reinforcing the potential of *TopoTxR* features as predictors of therapeutic efficacy.

4.4 Literature Review: TopoTxR

In breast radiology, predicting pathological Complete Response (pCR) after neoadjuvant chemotherapy is crucial for personalized treatment and prognosis. Du et al. introduce a technique that uses persistent homology to extract multi-dimensional topological representations from 3D data, significantly speeding

up computation [233]. The extracted topological information is distilled into deep neural networks using response-based knowledge distillation, improving pCR prediction accuracy from 85.1% to 90.5% and reducing computation time by 66% on a public breast DCE-MRI dataset.

Modern deep neural networks excel in medical image analysis but often overlook key anatomical structures like connected components and loops. Peng et al. introduce PHG-Net, a persistent homology-guided approach that incorporates topological features into medical image classification [234]. By computing cubical persistence diagrams and extracting topological features with a lightweight PH module, which is then fused with feature maps from CNNs or Transformers, PHG-Net enhances the classification performance. Evaluations on three public datasets show significant improvements over state-of-the-art methods.

Precision medicine aims to provide personalized care based on individual patient characteristics rather than generalized therapies. Radiomics and pathomics extract qualitative and quantitative data from radiology and pathology images, respectively, but can be enhanced by incorporating mathematical methods from differential geometry and algebraic topology. Geometry offers precise local measurements, such as curvature, to identify abnormalities, while topology captures essential features like connected components and holes, providing a robust shape descriptor. Integrating these mathematical tools can lead to more nuanced diagnostics and a comprehensive understanding of medical images, advancing the field of precision medicine [235].

Incorporating scientific research into clinical practice via clinical informatics, including genomics, proteomics, bioinformatics, and biostatistics, enhances patient treatment. Computational pathology, a growing subspecialty, aims to integrate whole slide images, multi-omics data, and health informatics, playing a crucial role in cancer diagnosis. Iqbal et al. review existing computational and digital pathology methods for breast cancer diagnosis, with a focus on deep learning [236]. It covers public datasets, existing deep learning methods, and publicly available code repositories, concluding with challenges and future directions for deep learning-based diagnosis.

4.5 Conclusion

In this study, we introduce a pioneering topological biomarker, *TopoTxR*, which harnesses the substantial geometric data intrinsic to structural MRI to enhance the efficacy of convolutional neural networks (CNNs). Utilizing the principles of persistent homology, we extract meaningful 1D cycles and 2D bubbles from breast DCE-MRIs. These topological structures are strategically

employed to focus the neural networks' attention, optimizing the analysis and interpretation of the imaging data. Our findings robustly demonstrate that *TopoTxR*, when applied to treatment-naive imaging, effectively predicts the pathological complete response (pCR), underscoring its potential as a significant advancement in the realm of medical imaging and cancer treatment prediction. This approach not only enhances the predictive accuracy but also contributes to a deeper understanding of the underlying biological processes, potentially guiding more personalized and effective treatment strategies.

Chapter 5

Hardware Acceleration of Boundary Matrix Reduction

5.1 Introduction

The second part of this thesis focuses on the computation of persistent homology. Initially, we introduce a hardware design aimed at accelerating boundary matrix reduction.

The realm of Topological Data Analysis (TDA) delves into understanding data's inherent topological features—such as connected components, tunnels, and voids—that offer a holistic, intuitive depiction of data characteristics. At the core of TDA, the concept of persistent homology [2, 113] stands out by mapping the evolution of data's topological features across various scales via a *filter function*. This function, which could be an image's intensity or a density measure, allows for the exploration of *sublevel sets*—regions under a predefined threshold value. Persistent homology examines these sets across a range of thresholds, forming a *filtration*, and observes the emergence and dissolution of topological features. The cumulative insights from this analysis are represented in a persistence diagram, plotting the lifespan of these features on a bidimensional plane, marking their inception and resolution times.

An illustrative case is provided in Fig. 5.1, showcasing the application of persistent homology to an image from the MNIST dataset [237]. Here, the evolution of the digit "8" through various threshold levels (t_0 to t_6) is captured, revealing the formation and subsequent filling of its upper and lower loops at different stages. This example not only demonstrates the power of persistent homology in capturing significant topological changes but also emphasizes its potential for a deeper understanding of complex data structures.

Recent years have seen the advent of topology-inspired methodologies which
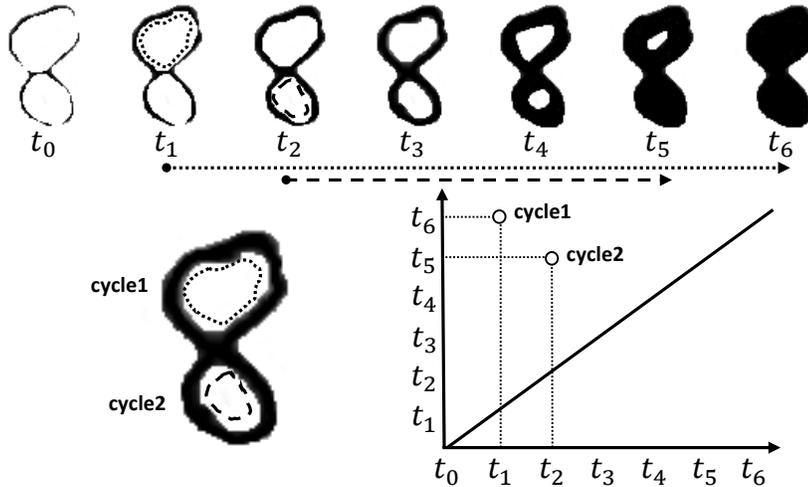
Figure 5.1: Illustration of a persistence diagram (bottom right) calculated from an image (bottom left) extracted from the MNIST dataset. The inverse of the initial image serves as the input. Displayed in the top row are the sublevel sets and the filtration process.

have found successful application across a broad spectrum of domains. These include, but are not limited to, molecular biology [238, 239], signal processing [240], sensor network design [241], robotics [242], shape recognition [243], computer graphics [244], and geometric modeling [245]. In the realm of biomedical image analysis, such topological techniques have been employed to investigate the overarching structures within both structural and functional MRI datasets [246–248]. By their very nature, topological invariants offer robustness against noise and deformation, ensuring the preservation of topological features without alteration through stretching or compressing. A particularly valuable characteristic of persistence diagrams is their Lipschitz continuity relative to the filter function that generates them [171], underscoring their stability and reliability in representing the topological attributes of data.

The structure of this document is laid out as follows: Section 2 introduces the foundational aspects of persistent homology theory. In Section 3, we delve into the specifics regarding the boundary matrix and the process of boundary matrix reduction. The subsequent Section 4 unveils a hardware-centric solution designed to significantly expedite the reduction process. We then scrutinize the efficacy of our proposed hardware solution through performance assessments conducted on two distinct datasets: MNIST and The Mammographic Image Analysis Society (MIAS) [249]. For demonstrative purposes, the MIAS dataset undergoes a reduction in resolution from 1024×1024 to 32×32, whilst preserving the original aspect ratio.

$$\partial_1 = \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 0 & 1 & 0 & 1 & 0 \\ v_2 & 0 & 0 & 1 & 1 & 0 & 1 \\ v_3 & 0 & 1 & 0 & 1 & 1 & 0 \\ v_4 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \rightarrow \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 1 & 1 & 0 & 1 & 0 \\ v_2 & 0 & 0 & 1 & 1 & 0 & 1 \\ v_3 & 0 & 1 & 0 & 1 & 1 & 0 \\ v_4 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \rightarrow \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 1 & 1 & 1 & 1 & 0 \\ v_2 & 0 & 0 & 1 & 1 & 0 & 1 \\ v_3 & 0 & 1 & 0 & 0 & 1 & 0 \\ v_4 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}$$

$$\rightarrow \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 1 & 1 & 0 & 1 & 0 \\ v_2 & 0 & 0 & 1 & 0 & 0 & 1 \\ v_3 & 0 & 1 & 0 & 0 & 1 & 0 \\ v_4 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \rightarrow \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 1 & 1 & 0 & 0 & 0 \\ v_2 & 0 & 0 & 1 & 0 & 0 & 1 \\ v_3 & 0 & 1 & 0 & 0 & 0 & 0 \\ v_4 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \cdots \rightarrow R_1 = \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 1 & 1 & 0 & 0 & 0 \\ v_2 & 0 & 0 & 1 & 0 & 0 & 0 \\ v_3 & 0 & 1 & 0 & 0 & 0 & 0 \\ v_4 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Figure 5.2: The $1-$dimensional boundary matrix, ∂_1, is derived from the simplicial complex showcased in the top left of Fig. 2.3, which also includes a defined filter function. In ∂_1, rows are associated with vertices (0-simplices) and columns with edges (1-simplices). The initial steps of the boundary matrix reduction are illustrated, with the reduction halting at R_1, which represents the reduced outcome of ∂_1.

The section on Persistent Homology introduces fundamental concepts necessary for understanding the core ideas of this paper, focusing on simplices, simplicial complexes, boundary operators, and filtrations. Due to space constraints, detailed discussions on cycles, chain groups, and homology groups are omitted, with readers directed to [113, 250, 251] for comprehensive information.

Simplicial complex

Defined within the framework of this study is the concept of a d-dimensional simplex, σ, which represents the convex hull formed by $d+1$ affinely independent vertices. For 3D datasets, the entities of 0-, 1-, 2-, and 3-dimensions are identified respectively as vertex, edge, triangle, and tetrahedron (illustrated in the top right of Fig. 2.3). A simplicial complex, K, is characterized as a finite collection of simplices that adhere to two primary conditions: (1) every face of a simplex within K also belongs to K, and (2) the intersection of any two simplices in K is either null or a shared face of both simplices.

Boundary Operator Definition

The boundary operator conceptually unravels a d-dimensional simplex into its constituent $(d-1)$-dimensional faces. For instance, the boundary of an edge

(considered a 1-simplex) is represented by the union of its endpoint vertices (0-simplices). Similarly, a triangle's boundary is composed of its enclosing edges, while a tetrahedron's boundary encompasses the surrounding triangles. This operation, applied to simplicies, effectively breaks down a d-simplex into a collection of $(d-1)$-simplicies, as visually demonstrated in the secondary row of Fig. 2.3, where boundaries of 1-, 2-, and 3-dimensional simplicies are elucidated.

Filtration

The concept of Filtration emerges from considering a topological space X and a real-valued function f defined over X, facilitating the construction of sublevel sets $X_t = \{x \in X : f(x) \leq t\}$, where t represents a threshold dictating the evolution of sublevel sets. This process initiates with an empty set and progressively encompasses the entirety of the topological space X as t advances from $-\infty$ to $+\infty$. The resultant ordered series of sets, evolving in response to the increment in t, is recognized as a filtration induced by the function f.

Algorithm 1 Boundary matrix reduction

1: **procedure** INITIALIZATION
2:     $R \leftarrow$ boundary matrix ∂
3:     $low_R() \leftarrow -1$
4:     **for** $i = 1$ to n **do**
5:     **if** column i has 1 **then**
6:         $low_R(i) \leftarrow$ row index of the last 1 in column i of R
7:         **endif**
8:     **endfor**
9:     **for** $i = 1$ to n **do**
10:     **while** $\exists i' < i$ with $low_R(i') = low_R(i)$ **do**
11:         add column i' to column i
12:         update $low_R(i)$
13:         **endwhile**
14:     **endfor**

5.2   Boundary Matrix Reduction

The computation of the persistence diagram necessitates a filter function assigned to simplicies. Illustrated in the top left of Fig. 2.3, filtration function values are annotated adjacent to their respective simplicies, such as vertices,
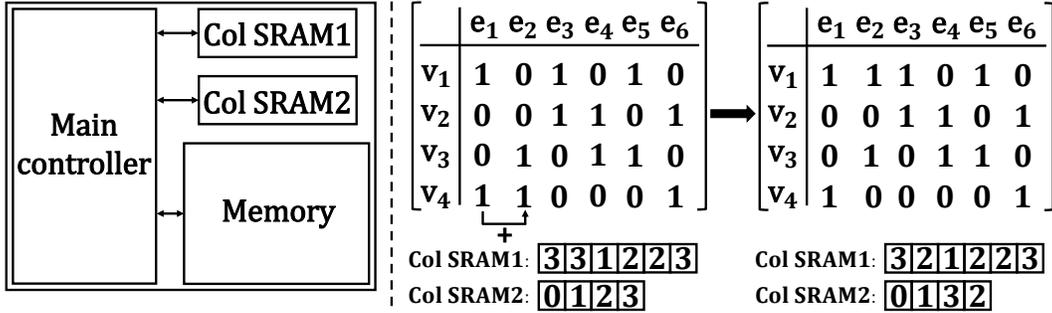
Figure 5.3: Left: architecture of the proposed hardware implementation of boundary matrix reduction. Right: example Col SRAM updates of the first reduction step in Fig. 5.2.

edges, and faces. Sorting these simplices typically by ascending filter function values facilitates the creation of a boundary matrix ∂. This binary matrix encodes the boundary operator, where an entry $\partial(u, v) = 1$ signifies that the simplex σ_u, corresponding to row u, constitutes part of the boundary of simplex σ_v, associated with column v.

The goal of boundary matrix reduction is to transform ∂ into another binary matrix R, employing column-wise operations on ∂. Each step involves the modification of a current column by adding it to a previously reduced column to its left, iterating from left to right until the reduction criteria are met. Specifically, the process concludes either when the rightmost column of R minimizes the row index of its last non-zero entry or when that column itself becomes entirely zero. For clarity, we introduce $low_R(i)$ as the notation for the row index of the lowest non-zero entry in column i of R, or -1 if column i is entirely zero. To reduce a column i, we search for another column j where $low_R(i) = low_R(j)$ and $j < i$, and proceed to add column j to column i until either column i becomes zero or no suitable column j can be found. Note that these additions are performed under $\mathbb{Z}_2$ arithmetic, meaning that $1 + 1 = 0$.

For illustrative purposes, we consider the reduction of the 1-dimensional boundary matrix ∂_1, derived from the simplicial complex shown in the top left of Fig. 2.3. The initial steps of this reduction process are visualized in Fig. 5.2, culminating in the reduced matrix R_1. Here, $low_{R_1}(i) \neq low_{R_1}(j)$ for any two distinct non-zero columns i and j, as depicted in the lower right of Fig. 5.2. The pseudo-code for this boundary matrix reduction process is presented in Algorithm 1, commencing with an initial scan of the boundary matrix to assign appropriate indices to $low_R()$ (executed in the first for-loop), followed by the reduction steps as previously outlined.

58

5.3 Innovative Hardware Solution for Boundary Matrix Optimization

In the quest for efficient computation of the persistence diagram, Section 3 elucidates the complexity involved in reducing a boundary matrix, which is replete with columnar manipulations. Such operations are not only resource-intensive but also become increasingly cumbersome with the growth of the boundary matrix size. This enlargement exacerbates the frequency of cache misses during the reduction phase, thereby elevating the challenges associated with memory management during computational processes. Furthermore, the task of identifying pairs of columns i and j, where $low_R(i) = low_R(j)$, escalates into a significant hurdle, amplifying the computational and power demands on both the software and hardware fronts. To address these bottlenecks, this section introduces a groundbreaking hardware accelerator designed specifically for streamlining the boundary matrix reduction. This pioneering solution holds the promise of catapulting the processing speeds by an unprecedented factor of 20,000 to 30,000, showcasing remarkable efficiency enhancements on datasets such as MNIST and MIAS.

The subsequent paragraphs will delve into the operational intricacies of each component within the proposed hardware architecture, as delineated in Fig. 5.3.

1. Memory: the memory module stores boundary matrix. Currently, only on-chip SRAM is considered. The architecture can be easily extended to DRAM when applied to a larger dataset.

2. Col SRAM1: Col SRAM1 stores the index of the lowest 1 in each column (i.e. $low_R()$).

3. Col SRAM2: Col SRAM2 stores the number columns which share the same $low_R()$.

4. Main Controller: the main controller module is responsible for the control of the entire hardware including reading and writing of the SRAMs.

5.4 Optimization of Boundary Matrix Reduction Through Hardware Acceleration

This section presents an illustrative example of the initial step in boundary matrix reduction as depicted in Fig. 5.2, with operational flow detailed in Fig. 5.3. Specifically, Col SRAM1 represents the index of the lowest 1 in column

59

Table 5.1: Comparisons of processing time between software and hardware implementations.

Dataset	Dimensions	SW Runtime	HW Runtime	HW/SW Speedups
MNIST	2-dim	2224 ms	1.10 ms	2022x
	1-dim	2639 ms	0.13 ms	20300x
MIAS	2-dim	4087 ms	1.51 ms	2706x
	1-dim	4816 ms	0.22 ms	21891x

i (denoted as $low_R(i)$), and Col SRAM2 tracks the count of columns sharing identical $low_R()$ values. As the boundary matrix reduction unfolds, these SRAMs are updated in tandem, significantly reducing the search duration for eligible column pairs due to the pre-stored data within the SRAMs.

For the practical realization of this hardware mechanism, the Memory module incorporates 24 SRAM units, each boasting a capacity of 1536 entries of 32-bit width. This circuitry is fabricated using advanced 28 nm CMOS technology, designed to occupy a minimal area of 0.5 mm^2 while consuming a modest 20 mW of power at an operational frequency of 1 GHz.

To evaluate the efficacy of this hardware solution, 10 instances from both the MNIST and MIAS datasets were subjected to software and hardware-based reduction for 1− and 2− dimensional boundary matrices. The software implementation, denoted as SW in Table 5.1 for simplicity, was developed in C++ and compiled under a 64−bit Windows environment using Visual Studio 2015. This baseline software accepts filtration matrices as input to generate reduced boundary matrices as output. The performance metrics, as shown in Table 5.1, were derived from a computing environment equipped with an Intel Core i7-9700K 3.6GHz CPU and 8GB DDR4 RAM. The table aggregates the runtime data across 10 samples for each dataset, highlighting the remarkable acceleration achieved with the hardware solution, particularly for 1-dimensional boundary matrices, thereby showcasing the transformative potential of dedicated hardware accelerations in the computation of persistence homology.

Chapter 6

Efficient Computation of Euler Characteristic Curves with GPU on Image Data

6.1 Introduction

In this section of the thesis, we propose an endeavor to accelerate the computation of the Euler characteristic curve, a simple yet powerful topological descriptor.

Topological Data Analysis (TDA) represents an innovative field where datasets are scrutinized using topological instruments, primarily persistent homology, to describe the multifaceted shapes data can take. This methodological approach distinguishes itself by examining data across various scales rather than confining its analysis to a single scale. It achieves this through the concept of a filtration, a technique that chronicles the evolution of data shapes by progressively analyzing their structure at increasing granularity levels. Filtrations adapt to the data type at hand, employing Alpha-shape filtration for point-cloud data situated in three-dimensional space, Vietoris–Rips filtration for analyzing high-dimensional metric data through pairwise distances, and cubical filtration for exploring two- or three-dimensional grayscale imaging data. Amidst the expanding applicability of TDA methods, our focus sharpens on imaging data, acknowledging the promising advancements persistent homology has made in recent years as documented by several studies [9, 23, 32, 252].

At the core of these advancements, persistent homology emerges as a profound topological descriptor, offering unparalleled insights into the filtration-based analysis of data. Its integration with cutting-edge deep learning methodologies has unveiled potential for a symbiotic relationship, although chal-

lenges persist. The seamless fusion of persistent homology with deep learning paradigms remains elusive; a significant hurdle is the demanding computational resources and extensive processing time required for persistent homology calculations on practical datasets. This contrasts starkly with the streamlined, highly optimized computations that characterize modern deep learning pipelines, many of which harness the parallel processing prowess of graphical processing units (GPUs) to realize neural network architectures and facilitate large-scale simulations. The sophistication of existing software for persistent homology lags, particularly in processing imaging data, a gap only recently addressed by initiatives like Zhang et al.'s GPU implementation for Vietoris–Rips filtrations derived from point-cloud data, which does not extend to imaging data [159].

In light of these observations, our investigation pivots towards the Euler characteristic curve (ECC), a simpler yet expressively rich topological descriptor tailored for imaging data analysis. The efficacy of ECC in delivering meaningful topological insights across a spectrum of imaging applications [133–135] positions it as a critical tool in our analytical arsenal. This research not only highlights the ECC's versatility but also underscores our groundbreaking achievement in computing ECC with unparalleled speed, enabling the processing of 3D images of size 512^3 at an unprecedented rate of 30 times per second. Our innovative streaming strategy further extends our capability to process immense images, sizes of 4096^3 voxels and beyond, a feat made possible despite the intrinsic limitations posed by GPU memory constraints. These breakthroughs hint at the potential for ECC computations to integrate seamlessly with contemporary image processing workflows, marking a significant stride towards the harmonization of TDA with modern computational techniques.

Machine Learning Integration. A key area of interest is the incorporation of ECC computation within machine learning frameworks, notably convolutional neural networks (CNNs), which have revolutionized fields ranging from computer vision to biomedical image processing, and even computational astrophysics. Despite the promising integration of related topological descriptors in similar machine learning contexts [136, 253, 254], the swift computation of such descriptors remains a formidable challenge. It's imperative to note that our focus is predominantly on surmounting the barrier of accelerated computation, sidestepping other challenges like gradient optimization, which fall outside our current investigative scope.

The advent of CNNs marked a paradigm shift in the 1980s, propelling large-scale applications into feasibility through the advent of modern GPUs. A critical analysis of CNN implementations reveals the performance impediments

tied to CPU-bound operations for image processing tasks, even in scenarios of high optimization [255]. This bottleneck, primarily attributed to the extensive data transfer overhead between GPU memory and main memory (RAM), fueled our motivation to develop a GPU-centric ECC implementation. In doing so, ECC computation becomes an integral part of the processing pipeline, efficiently handling images already resident in GPU memory, thereby mitigating data transfer delays and enhancing overall processing throughput.

Physical and Astrophysical Simulations. Our research extends to scenarios involving large images or volumetric data, particularly emanating from physical and astrophysical simulations increasingly run on GPUs. These simulations generate large 3D volumes, motivating our pursuit of performing ECC computations *in vivo*—directly within the simulation process, leveraging the simulated volume's residence in GPU memory for real-time analysis.

Contributions. The cornerstone of this paper is the development of a streaming ECC computation method optimized for GPU implementation, tailored specifically for imaging data analysis. While the underlying algorithm remains straightforward, our contribution is deeply rooted in an

implementation meticulously refined for contemporary GPU architectures. This involves adapting the computation to fit massive parallelism, ensuring efficient use of limited GPU memory resources, and designing the algorithm to circumvent data transfer latency—a primary bottleneck in computational efficiency. Our implementation strategy, designed to exploit GPU/CPU asynchronous operations and parallelism fully, is elaborated in detail, providing insights into our methodological choices without assuming prior GPU technical knowledge.

By demystifying these GPU-centric computational strategies, we aspire to broaden the adoption of GPU technologies in computational geometry and topology software development. We are optimistic that the techniques employed here could catalyze further advancements in persistence computations, particularly in the realm of image analysis, fostering a deeper integration of TDA methodologies with contemporary computational practices.

Outline. Organized for clarity and depth, this paper begins with a brief overview of ECC and its conceptual underpinnings (Section 6.2). We then explore the current landscape of ECC applications and implementations (Section 2.4.1), followed by a detailed exposition of our GPU implementation strategy (Section 6.3), with advanced optimizations discussed in an Appendix (Section 6.3.3). Comprehensive experiments are presented and analyzed in Section 6.4, evaluating our methodology against established goals. The paper concludes with a summary and reflections on our findings and contributions.

6.2 Foundational Concepts

This section unfolds the foundational elements requisite for a topological approach to analyzing imaging data, alongside an introduction to the essentials of GPU programming for computational acceleration.

6.2.1 Cubical Filtrations from Imaging Data

Our methodology processes d-dimensional grayscale images, essentially d-dimensional arrays populated with real-valued intensities. These individual elements, termed pixels within 2D contexts and voxels for 3D or higher dimensions, are uniformly referred to as voxels here. A pivotal operation within our framework is *thresholding*, a technique that isolates voxels beneath a specified intensity threshold t. This operation is critical for structurally analyzing the topological evolution of the image as the threshold varies.

Algorithm 2 Iterative Calculation of the Vector of Changes in Euler Characteristic (VCEC)

Require: $\boldsymbol{I}$: an input image
Ensure: $\boldsymbol{VCEC}$: the vector delineating the Euler characteristic's alterations.

1: Formulate $\boldsymbol{VCEC}$ as a blank array
2: **for all** voxels v within $\boldsymbol{I}$ **do**
3:     **for all** faces c of v **do**
4:         **if** c is attributable to v **then**
5:             Adjust $\boldsymbol{VCEC}$[value of v in I] by $(-1)^{\text{dimension of face } c}$

In alignment with pioneering works such as [140], our approach begins by defining the fundamental unit of our analysis: the elementary interval, which can be either an inclusive range $[k, k+1]$ or a singular value $[k, k]$ for any integer k. From this, we construct elementary (cubical) cells as combinations of these intervals, with their dimensionality determined by the count of inclusive ranges. This structure allows for a comprehensive discussion about the various forms these cells can take, such as vertices, edges, squares, and cubes, corresponding to dimensions $0, 1, 2, 3$, and beyond. In this setup, a cell a is considered a face of another cell b if a is fully contained within b, and conversely, a coface if b is contained within a. The notion of *voxels* pertains to the cells of highest dimension within our cubical framework, with each voxel's intensity directly influencing its associated cells. This hierarchical value assignment forms the basis for constructing a *cubical complex*, $K_{\leq t}$, through

thresholding, with the ensuing collection of these complexes as the threshold t varies constituting a *cubical filtration*, meticulously indexed by t.

This expanded discussion not only sets the stage for a deeper topological examination of imaging data but also prepares the ground for leveraging GPU programming to accelerate the computation, promising a novel synthesis of topological analysis and computational efficiency.

6.2.2 Expounding on the Euler Characteristic Curve

Given the foundational structure previously outlined, the Euler Characteristic Curve (ECC) emerges as a pivotal concept within the realm of cubical filtrations. Formally articulated as:

$$ECC_i = \chi(K_{\leq t_i}) = \sum_j (-1)^j c_j(K_{\leq t_i}) = \sum_j (-1)^j \beta_j(K_{\leq t_i}) \qquad (6.1)$$

where t_i represents the threshold values ranked from smallest to largest within the image, $c_j(.)$ quantifies the j-dimensional cells, and $\beta_j(.)$ refers to the j-dimensional Betti numbers. The correlation of ECC with the spatial topology is cemented through the Euler–Poincare formula, as chronicled by [256], underscoring the ECC's integral role in mapping the topological landscape of the data.

It's pertinent to highlight that Betti numbers, denoting the ranks of the *cubical homology groups* as per [140], serve as a bridge to understanding the topological complexity of the cubical complex $K_{\leq t}$. Within the context of three-dimensional complexes, these numbers provide a count of the connected components, tunnels, and voids, hence portraying the ECC as an amalgamation of disparate topological dimensions. This framework also accommodates the notion of *persistent homology*, as explicated in [257] and further applied in this domain by [258]. The interplay between persistent homology, Betti curves, and the ECC is vividly depicted in Fig. 6.2, offering visual insights into their interconnectedness.

6.2.3 ECC Computation Revisited

The traditional method for calculating ECC involves explicit enumeration of the Euler characteristic at every threshold, a process bearing the time complexity of $O(mn)$, with m representing the distinct grayscale values in the image and n the voxel count, presuming the image's dimensionality remains constant.

Pioneered by Snidaro and Foresti [132], the inaugural algorithm proposed
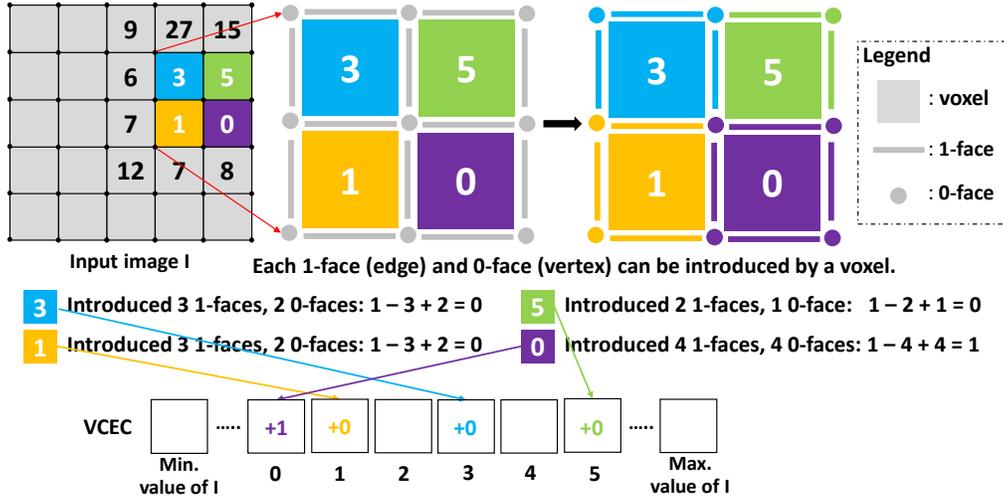
Figure 6.1: Visualization of Algorithm 2 applied to a 2×2 image section. Each voxel, representing a 2-cell, along with its associated 1-faces and 0-faces, contributes to updating the VCEC based on the voxel's specific filtration value (refer to line 5 in Algorithm 2).

an $O(n)$ complexity framework, albeit its intricate nature and challenges in extending beyond two-dimensional analyses. In contrast, our methodology is inspired by a substantially more straightforward algorithm [143] that conceptualizes an image as a cubical filtration.

Vector of Changes in Euler Characteristic (VCEC) Analysis. Central to our strategy is the computation of the Vector of Changes in Euler Characteristic (VCEC) - a series where $VCEC_0 = ECC_0$ and each subsequent $VCEC_i = ECC_i - ECC_{i-1}$ for $i > 0$. Given that ECC_i equates to the cumulative sum of $VCEC_j$ up to index i, we employ dynamic programming to efficiently derive ECC within a temporal frame of $O(m)$, leveraging parallel algorithms on GPUs for practical speed enhancements, as suggested by [259]. This refined approach not only simplifies ECC calculation but also aligns with advancements in GPU computing to facilitate rapid, scalable analysis of topological data characteristics.

In essence, each $VCEC_i$ value articulates the Euler characteristic for cell groups at threshold t_i. While these groups constitute a chain complex, allowing discussions on homology groups, they typically don't form a cubical complex, limiting the intuitive application of homology group ranks. These could be understood in terms of relative homology, yet such an interpretation isn't imperative for our purposes.

Identification of Faces by Voxels. A face is deemed to be *introduced by a*
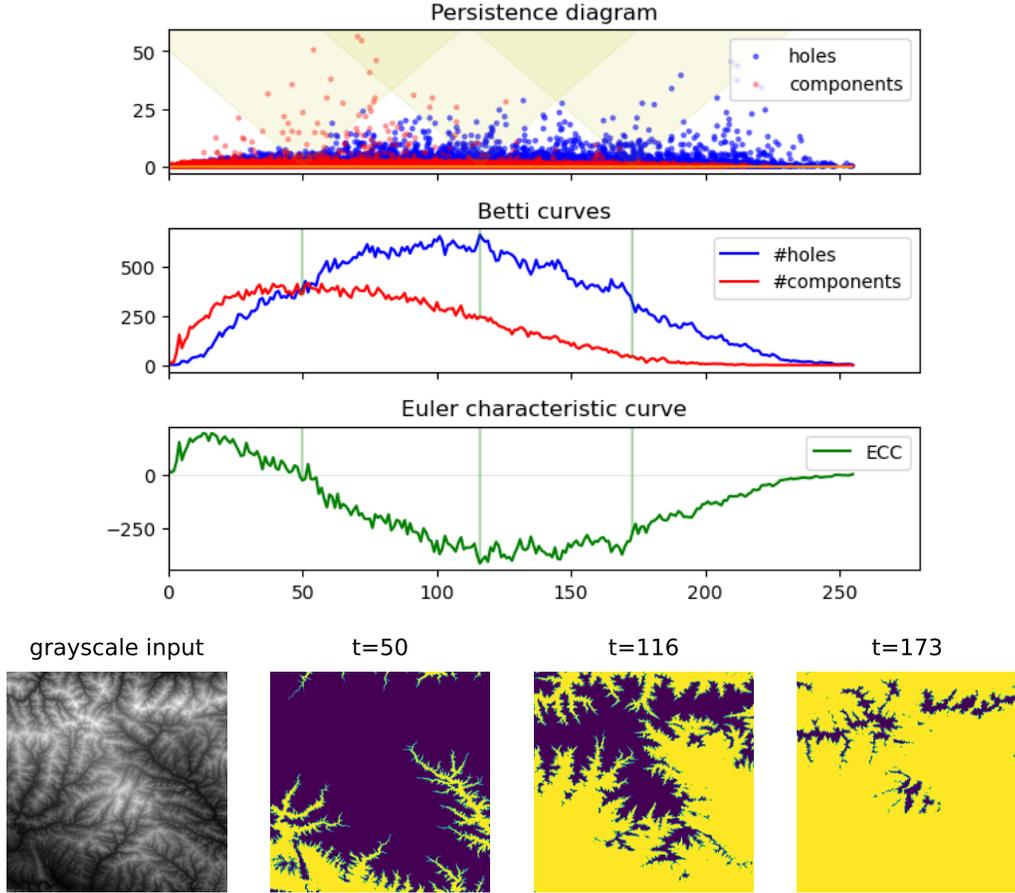
Figure 6.2: The bottom part of this figure displays an input image alongside its three distinct thresholding stages, with grayscale intensities indicative of terrain heights. At the top, three corresponding plots, aligned along a common x-axis denoting the image's threshold levels, illustrate the persistence diagram, intentionally rotated for enhanced clarity. At every threshold level, the diagram identifies the duration of topological entities: connected components (in red) and cavities (in blue). Highlighted sections within these plots pinpoint the existence of these topological structures at the specified thresholds, further delineated as Betti curves beneath. The Euler Characteristic Curve (ECC) emerges from the differential analysis of these curves, underscoring ECC's dependency on the enumeration of topological attributes, whereas persistence homology also evaluates their significance.

voxel if it possesses the minimum value among all voxels encompassing that face, effectively inheriting its value from the voxel. This process necessitates

a method for breaking ties consistently: in cases where voxels share values, preference is given to the voxel with a lexically earlier position within the dataset. This procedure facilitates swift computation, obviating the need for direct index comparisons, thereby streamlining the analysis.

Developing a Sequential Algorithm. To lay the groundwork for our exploration into VCEC computation within images, we propose a foundational sequential algorithm (referenced in Algorithm 2 and illustrated in Fig. 6.1). A significant advantage of this approach is its elimination of the necessity to directly manage information pertaining to cells of lower dimensions, simplifying the process. This streamlined algorithm not only serves as a precursor to more complex computations but also establishes a methodological baseline that will inform the development of our more advanced GPU-based algorithm.

6.2.4 Expanding on GPU Computations

Diving deeper into the realm of Graphical Processing Units (GPUs), this segment elucidates the critical components and functionalities of GPUs that are paramount for executing our computational tasks, with a detailed exploration set to follow in Section 6.3.

Originating as accelerators for 2D graphical rendering and subsequently evolving to facilitate 3D scene rendering, GPUs have transcended their initial purpose. The inherent parallel nature of pixel processing, where each pixel's computation is both independent and similar to that of its peers, makes these tasks ideally suited for parallelization. Today, GPUs have emerged as versatile tools for a broad spectrum of general-purpose computing tasks. Despite this broad utility, not every computational challenge is aptly met by GPU processing; however, the calculation of the Euler Characteristic Curve (ECC) distinctly benefits from the parallel computational capabilities GPUs offer.

Our focus narrows to Nvidia's GPU hardware and the CUDA programming framework, chosen for its widespread adoption and robust support community. The CUDA model allows programmers to define a *kernel*—a function designed to run concurrently across a multitude of threads, leveraging the massive parallel processing power inherent to GPUs. For instance, the Nvidia RTX 2070 GPU, utilized in our experiments, provides up to 2304 individual threads for computation, embodying Nvidia's Turing architecture. It's important to note that the architectural nuances and programming strategies for GPUs have evolved, moving towards more intuitive and accessible programming paradigms. This evolution reflects not only in the increased flexibility and efficiency of GPU programming but also in how these advances can be harnessed to enhance the computation of topological features like ECC in imaging

data.

By integrating these insights into GPU technology with our algorithmic strategies, we aim to push the boundaries of what's possible with ECC computations, making full use of the parallel processing prowess GPUs offer. This exploration is not just about leveraging the raw power of GPUs but about marrying this power with sophisticated topological analysis techniques to unlock new potential in the processing and understanding of complex imaging data.

6.3 Execution on GPU

The execution strategy deployed on the GPU is delineated in Algorithm 3. To enhance clarity, the exemplified code is deliberately condensed, emphasizing the manipulation of 2D grayscale images, where pixel intensities span from 0 to 255. The computation's architectural design is initially overviewed, followed by a comprehensive explication of the execution specifics.

6.3.1 Execution Challenges

In migrating to a GPU-based framework, we confronted several pivotal challenges:

1. The translation from a CPU-centric algorithm to one optimized for GPU utilization necessitated a fundamental reorganization to tap into the extensive parallelism offered by GPUs. This transition was not trivial, as it involved scaling from a modest number of concurrent threads to managing thousands simultaneously, requiring a novel computation paradigm.

2. In contrast to the CPU implementation's reliance on a straightforward lock-free approach to mitigate concurrency issues, the GPU adaptation demanded meticulous attention to race conditions and synchronization intricacies. This was paramount not only for ensuring computational accuracy but also for fine-tuning performance through strategic synchronization granularity.

This section elaborates on the techniques and methodologies employed to address these challenges, ensuring efficient utilization of GPU capabilities for algorithm execution.

Algorithm 3 Implementation of the VCEC on GPU for a 2D image

```
 1: __constant__ int image_width, image_height;
 2: const int num_bins = 256;
 3: __global__ void vcec_kernel(cudaTextureObject_t voxels, int* vcec_global)
 4: {
 5:     __shared__ int vcec_local[num_bins];
 6:     const int thread_number = blockDim.x * threadIdx.y + threadIdx.x;
 7:
 8:     if (thread_number < num_bins)
 9:         vcec_local[thread_number] = 0;
10:     __syncthreads();
11:
12:     const int ix = blockDim.x * blockIdx.x + threadIdx.x + 1;
13:     const int iy = blockDim.y * blockIdx.y + threadIdx.y + 1;
14:     if (ix >= image_width + 1 || iy >= image_height + 1) return;
15:
16:     int change = 1;
17:     int c = tex2D<float>(voxels, ix, iy);
18:     int t = tex2D<float>(voxels, ix, iy - 1);
19:     int b = tex2D<float>(voxels, ix, iy + 1);
20:     int l = tex2D<float>(voxels, ix - 1, iy);
21:     int r = tex2D<float>(voxels, ix + 1, iy);
22:
23:     // Vertices
24:     change += (c < l && c < t && c < tex2D<float>(voxels, ix - 1, iy - 1));
25:     change += (c < t && c <= r && c < tex2D<float>(voxels, ix + 1, iy - 1));
26:     change += (c < l && c <= b && c <= tex2D<float>(voxels, ix - 1, iy + 1));
27:     change += (c <= b && c <= r && c <= tex2D<float>(voxels, ix + 1, iy + 1));
28:     // Edges
29:     change -= ((c < t) + (c < l) + (c <= r) + (c <= b));
30:
31:     atomicAdd(&vcec_local[c], change);
32:     __syncthreads();
33:     if (thread_number < num_bins)
34:                             atomicAdd(&vcec_global[thread_number], vcec_local[thread_number]);
35: }
```

(2) Leveraging the GPU memory hierarchy and managing resource constraints were paramount. GPUs are known for their complex memory architecture and limited memory capacity. This necessitates a deliberate integration of memory considerations into the algorithmic framework. We developed an optimized multi-tier caching system, informed by the specific access patterns encountered when handling cubical complexes. By analyzing these patterns, we significantly reduced reliance on slower memory types, instead utilizing the GPU's texture memory. This memory type benefits from a cache optimized for spatial locality, caching data for a voxel and its immediate neighbors in both 2D and 3D spaces, thereby enhancing performance. (3) Another challenge was maintaining efficient streaming operations to process data sets larger than the GPU's memory without compromising performance. We achieved this through a streamlined pipeline approach, allowing computation and memory transfers to occur concurrently, facilitating the handling of extensive data sets.

6.3.2 Computational Framework

The compute kernel, as defined in the C++ code in Algorithm 3, represents the core computation performed by each thread. Moving forward, we adopt a thread-centric perspective, focusing on the activities of an individual thread while acknowledging that these actions occur in parallel across many threads.

Individual Thread. A single thread is responsible for processing one voxel, executing the operations outlined in lines 3–5 of Algorithm 2. Specifically, it assesses the voxel's faces, determines those introduced by the voxel, and modifies the VCEC vector accordingly.

Thread Blocks. Threads are organized into blocks, processing the image in segmented rectangular sections. These sections are addressed simultaneously, allowing for a scalable and efficient processing model. Refer to Fig. 2.2 for a schematic representation.

6.3.3 Optimizations and Efficient Memory Usage

Tailoring computations to align seamlessly with the GPU architecture not only leverages its computational power but also necessitates a strategic approach to memory management. GPUs are distinguished by a layered memory hierarchy, each layer offering distinct size limitations and access speeds. Unlike traditional CPU-centric programming, where memory management is largely abstracted away, effective GPU programming demands a meticulous allocation and utilization of these diverse memory resources.

71

In the following sections, we delve into our implementation strategy, highlighting the rationale behind our choices and pinpointing potential bottlenecks. We initiate our discussion with a theoretical naive approach that directly transposes Algorithm 2 onto a GPU framework. Through a progressive refinement process, we address and rectify inefficiencies, culminating in the optimized approach detailed in Algorithm 3.

Managing Input Image Data. The transfer of image data from the system's main memory (RAM) to the GPU's *global memory* marks the first step in preparing for GPU-based processing. Global memory, owing to its substantial capacity, is typically the only viable storage option on a GPU for accommodating images of practical dimensions. However, it's crucial to acknowledge that exceptionally large images may surpass the global memory's capacity limits. While our initial discussions presuppose the image's compatibility with global memory constraints, we later address and devise solutions for managing larger-than-memory images in Section 6.3.4, ensuring a comprehensive approach to GPU-based image processing.

This exploration into optimization techniques aims not just to adapt our computational methodology to the GPU's strengths but to do so in a way that maximizes efficiency and performance. By carefully navigating the intricacies of GPU memory management and iteratively refining our algorithm, we pave the way for a robust and scalable implementation capable of handling the demands of topological data analysis on large-scale imaging datasets.

Navigating Race Conditions for Efficient Memory Use. The practice of storing the Vector of Changes in Euler Characteristic (VCEC) within the global memory as a simple array of 256 integers introduces a critical challenge: *race conditions*. Given the parallel nature of GPU operations, multiple threads concurrently updating a single memory location can lead to these conditions. To counteract this, modern GPUs implement *atomic operations*, such as *atomicAdd*, guaranteeing the integrity of updates to VCEC. Nonetheless, the serialization of updates to the same memory address negates a fundamental GPU benefit—its inherent capacity for massive parallel processing.

Leveraging Register Memory to Counteract Serialization. A strategic response to the challenge of race conditions involves utilizing *registers* for interim voxel contribution tallies. Registers, representing the pinnacle of GPU memory speed, are exclusively accessible to individual threads, effectively eliminating the concern of concurrent updates. While global VCEC updates still require *atomicAdd*, the frequency of these updates is significantly reduced—limited to a singular update per thread. This approach not only mitigates the adverse effects of serialization, inherent in handling 2D or 3D data (which would otherwise necessitate 9 or 27 updates per voxel, respectively),

72

but also capitalizes on the GPU's parallel processing strengths, ensuring that the computation of VCEC remains both efficient and scalable.
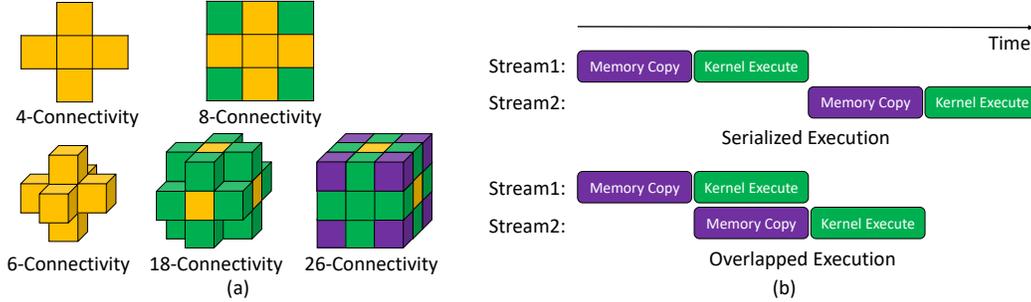
Figure 6.3: (a) Displays the assortment of voxel connectivity patterns, crucial for understanding cubical complex structures. (b) Demonstrates a strategy where computational tasks and memory transfers are conducted in parallel, showcasing a significant reduction in total processing time. This approach exemplifies how synchronizing data-intensive operations with memory transactions can optimize workflow efficiency on GPUs.

Enhancing Efficiency with Shared Memory. While leveraging registers marks a substantial leap forward, achieving the epitome of optimization necessitates employing *shared memory*. As a GPU memory type, shared memory features lower latency than global memory but doesn't quite match the speed of registers. Its defining characteristic is the communal access it offers to threads within the same block, allowing each thread to contribute to a block-local Vector of Changes in Euler Characteristic (VCEC), rather than to the global VCEC. This block-local VCEC mirrors the structure of its global counterpart, consisting of an array of 256 integers.

To circumvent potential race conditions during updates, *atomicAdd* operations remain essential. However, the likelihood of these conditions materially decreases as only threads from the same block interact with specific shared memory locations. By judiciously selecting the block size—a parameter under our control—we can fine-tune performance. This strategy benefits from the reduced latency of shared memory and, on contemporary GPUs, the enhanced efficiency of atomic operations performed within this space, a contrast to older GPU models which might necessitate a nuanced approach for integrating results.

Concurrent Initialization and Aggregation. Initializing the shared memory and assimilating its contents into the global VCEC necessitates a parallel approach. Each thread assumes responsibility for a unique location within the shared-memory array, ensuring that the initialization and final aggregation

processes are distributed evenly across the block's threads. This method not only streamlines the update process to the global VCEC but also emphasizes the adaptability and precision in utilizing shared memory to optimize computational workflows.

Ensuring Coherence with Block-level Synchronization. The asynchronous nature of thread execution within a block introduces the necessity for stringent synchronization measures. Without these, threads risk operating on yet-to-be-initialized memory segments or contributing incomplete local results to the global VCEC. By instituting *block-level synchronizing barriers*, we guarantee that all threads within a block proceed from one operational phase to the next in unison, thereby maintaining the integrity and reliability of the computational process.

Through these meticulous optimizations—leveraging shared memory, parallelizing initialization and finalization steps, and enforcing block-level synchronization—we not only address inherent challenges but also unlock new levels of computational efficiency and accuracy, paving the way for advanced GPU-based topological data analysis.

Strategic Neighbor Access. In our approach to constructing the cubical complex, we adopt a dynamic methodology for determining the presence of lower-dimensional cells. Rather than storing these cells directly, we analyze the input voxels to infer which cells each voxel introduces. This process is dependent on a comparison between a voxel and its immediate neighbors, applying a methodical approach to breaking ties between equally valued voxels. For practical purposes, we engage with the 8-connectivity model in two dimensions (expanding to 26-connectivity in three dimensions) as illustrated in Fig. 6.3(a). This model negates the need for an explicit representation of vertices and faces with varying filtration values, thus conserving memory by circumventing the additional data storage otherwise required.

Leveraging the Texture Cache for Efficient Data Access. One of the critical challenges in GPU-based computations lies in the high latency associated with global memory access. To address this, we utilize the GPU's *texture cache*, a specialized caching mechanism designed to prefetch and store the values of neighboring voxels. Despite being occasionally referred to as *texture memory*, it functions primarily as a caching layer above the global memory, significantly reducing access times. When a voxel's value is fetched through this cache, as demonstrated in line 16, the system automatically caches the values of its neighbors in a dedicated low-latency memory space, comparable in speed to shared memory.

This caching strategy is specifically optimized to exploit the spatial locality inherent in ECC computations, ensuring that data access patterns are

as efficient as possible. Given the parallel processing of neighboring voxels, there's a high probability that their values are pre-cached, thereby minimizing the need for repeated global memory accesses. While this does not guarantee the elimination of all uncached accesses—owing to the cache's finite size—it significantly enhances the efficiency of our computations, ensuring that, on average, the necessity for direct global memory retrieval is kept to a minimum. This optimization exemplifies our commitment to maximizing computational efficiency through innovative use of GPU resources.

Mitigating the Risks of Register Overflow. Our computational strategy meticulously allocates voxel values to *registers*, the swiftest tier of GPU memory. This allocation is particularly critical for 2D and 3D inputs, where the spatial connectivity of voxels dictates their frequency of access and, consequently, their storage in registers. For instance, in 2D scenarios, voxels connected through 4-connectivity are implicated in multiple comparison operations, warranting their cache in registers for rapid access. Conversely, voxels associated through 8-connectivity, which are referenced less frequently, are not stored in registers to conserve this limited resource, relying instead on the texture cache for efficient access. This selective caching is critical to optimizing computational efficiency and ensuring the judicious use of registers.

However, the allure of leveraging registers for every computational variable can lead to *register spilling*, a counterproductive phenomenon where an excess of data allocated to registers overflows into local memory. This overflow is particularly insidious because local memory, despite its name, resides within the much slower global memory, resulting in significant performance degradation. This pitfall underscores the necessity of balanced register allocation, especially given the hardware-imposed limits on the number of registers per block and the configurable nature of block sizes.

Addressing the Challenges of Branching and Looping. While GPUs afford a considerable degree of programming flexibility, their efficiency peaks when operating under a SIMD (single instruction, multiple data) paradigm. This implies that algorithms designed with a linear execution flow, devoid of conditional branching or looping based on input data, are ideally suited for GPU execution. Branching and looping introduce computational divergence within thread execution, undermining the parallel processing capabilities of GPUs and, by extension, the overall performance of the algorithm.

By adopting a strategic approach to register use, carefully managing data access patterns, and minimizing branching and looping, our methodology aims to leverage the inherent strengths of GPU architecture. This ensures that our computational solutions not only achieve high levels of efficiency and speed but also sidestep the common pitfalls associated with GPU programming, paving

the way for advanced, optimized topological data analysis.

Navigating GPU Thread Scheduling with Warps. The GPU's approach to thread scheduling introduces the concept of *warps*, which are essentially groups of 32 threads operating within each block. This grouping mechanism is central to understanding GPU performance, particularly when dealing with conditional branching. When an *if* statement causes the thread execution path to branch, it can lead to *intra-warp branching*, where the warp's threads may follow different execution paths. This divergence forces the paths to be executed serially rather than in parallel, potentially causing non-branching threads to idle and diminishing the overall parallel execution efficiency. To circumvent such issues, our implementation strategies include surrounding our data with a protective layer of voxels set to infinite values, thereby reducing the likelihood of divergent branching caused by boundary checks. Additionally, we refine the logic for updating variables to minimize branching, such as by incorporating the outcome of logical expressions directly into variable updates, further optimizing our use of GPU resources.

Optimizing Data Access with Constant Memory. Efficiently managing data access within GPU kernels is crucial for performance, especially for repeatedly accessed data like image dimensions. Repeatedly fetching such information from the slower global memory can significantly hamper performance. To address this, we leverage GPU's *constant memory* for storing such frequently accessed data. This specialized memory type is designed for the rapid *broadcast* of its contents to numerous threads, significantly reducing access latency compared to global memory. By storing essential parameters like the width and height of the image in constant memory, we ensure that these values are swiftly and efficiently available to all threads, enhancing our algorithm's overall efficiency and performance on the GPU. This strategic use of constant memory is illustrative of our broader approach to GPU programming, where understanding and exploiting the unique characteristics of GPU memory types can lead to substantial performance gains.

6.3.4 Streaming for Large 3D Image Processing

Addressing the challenge of processing 3D images too large for conventional GPU memory storage, we introduce a *chunking* strategy that divides the input into manageable sections for sequential processing. This method not only accommodates large datasets but also leverages CUDA's asynchronous capabilities, allowing simultaneous data transfers and kernel executions. This overlapping of operations effectively mitigates the latency traditionally associated with moving data between GPU and system RAM, as demonstrated in
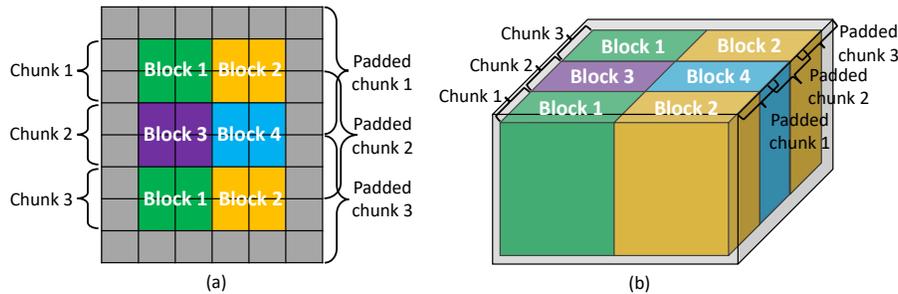
Figure 6.4: Illustration of Chunking Mechanism in 2D and 3D Settings. This figure highlights the concept of chunking as a strategy for partitioning the input into manageable segments, ensuring a disjoint decomposition. The introduction of padding around these chunks is critical; it extends beyond the chunk's original boundaries to include adjacent information, thereby facilitating accurate computations. This padding ensures that every cell, particularly those at the chunk borders, is accounted for precisely once, preserving the integrity of computational results.

Fig. 6.3.

For an input image with dimensions $(w_0, w_1, \dots)$, we segment the image along its primary axis. This division results in c *chunks*, each with dimensions up to $(\frac{w_0}{c}, w_1, \dots)$. Such segmentation ensures that each chunk aligns with contiguous memory addresses, adhering to the row-major order storage convention in C++. To ensure each voxel within a chunk has full access to its surrounding neighbors for accurate processing, we introduce a one-voxel padding around each chunk, as depicted in Fig. 6.4. This padding either carries the actual neighboring voxel values or is set to positive infinity to maintain computational integrity.

Upon division, each chunk, inclusive of its padding, is sequentially loaded into GPU memory for processing. Depending on the chunk's size, one or more CUDA blocks may be assigned for its computation. As computations complete, resources are dynamically reallocated to subsequent chunks, ensuring continuous processing.

Maximizing Efficiency through Concurrent Operations. Modern CUDA devices are equipped with multiple engines dedicated to specific tasks—typically including two copy engines for host-to-device and device-to-host transfers, alongside a kernel engine for executing computation tasks. Utilizing pinned (non-pageable) host memory and launching tasks in separate CUDA streams facilitates the concurrent execution of these operations, devoid of interdependencies. This approach enables the simultaneous loading of chunks onto the device, execution of computation kernels, and transfer of results back to the

host. By judiciously selecting the number of chunks c, we significantly reduce the overhead associated with data transfer. Fig. 6.3(b) exemplifies a scenario where concurrent transfers and kernel executions virtually negate the kernel execution time for a chunk when $c = 2$, showcasing the strategic advantage of this streaming and overlapping methodology in large-scale 3D image processing.

6.4 Experimental Setup and Methodology

For our experimental evaluation, we opted for the C++ programming language, utilizing the compiler provided with Visual Studio 2019 (v142) and adhering to the C++14 standard for compiling both our CPU and GPU-based implementations. The hardware platform for these experiments was a contemporary desktop configuration, consisting of an Intel Core i7-9700K CPU equipped with 8 physical cores (hyper-threading was disabled for the tests), 16GB of RAM for ample processing memory, and a Sabrent Rocket Q 2TB NVMe PCIe M.2 2280 SSD for rapid data access and storage. Graphical processing was handled by a NVIDIA RTX 2070 graphics card, which boasts 8GB of GDDR6 memory, providing a robust framework for assessing the performance of our computational methods. This setup represents a well-equipped but still accessible workstation, typical of a modern computational environment.

Table 6.1: In the comparative analysis presented through this table, we scrutinize the overall execution durations for both CPU and GPU-based methodologies. This comprehensive timing assessment encompasses all phases of execution, inclusive of disk input/output activities as well as the preliminary overheads incurred during the initialization of GPU-related computational processes. Particularly, the emphasis on the final two columns of the table serves to illuminate performance metrics under conditions where computational data is pre-loaded into GPU memory, offering insight into the efficiency gains achievable in optimized GPU memory utilization scenarios.

	Input size(B)	CPU overall	GPU overall	Overall speedup	CPU disk read	GPU disk read	GPU over-head	GPU exec. (kernel)	GPU kernel Gvox/s
				Uniform Noise					
4096^3	256G	37.72m	9.10m	4.14x	7.30m	9.08m	0.67s	0.20m	5.62
2048^3	32G	4.86m	0.71m	6.77x	0.99m	0.71m	0.41s	0.03m	5.61
1024^3	4G	36.85s	5.63s	6.55x	6.85s	5.20s	0.37s	0.16s	6.57
512^3	512M	4.97s	0.85s	5.86x	1.00s	0.64s	0.19s	0.02s	6.55
				Gaussian Random Field					
512^3	512M	4.93s	0.86s	5.75x	0.90s	0.66s	0.19s	20.88ms	6.43
256^3	64M	0.63s	0.24s	2.58x	0.13s	0.09s	0.15s	2.64ms	6.35
128^3	8M	0.11s	0.12s	0.86x	0.02s	0.01s	0.12s	0.35ms	6.00
8192^2	256M	1.47s	0.53s	2.75x	0.44s	0.36s	0.16s	6.64ms	10.10
4096^2	64M	0.38s	0.21s	1.84x	0.12s	0.08s	0.14s	1.74ms	9.64
2048^2	16M	0.09s	0.18s	0.55x	0.04s	0.03s	0.12s	0.45ms	9.32
				VICTRE					
287 359 202	79.3M	0.59s	0.30s	1.98x	0.16s	0.13s	0.14s	3.85ms	5.41
440 518 488	424M	2.99s	0.77s	3.87x	0.98s	0.45s	0.24s	20.65ms	5.39
359 359 299	147M	1.11s	0.36s	3.02x	0.29s	0.15s	0.16s	7.13ms	5.40
434 446 384	283M	1.96s	0.53s	3.70x	0.79s	0.30s	0.18s	13.72ms	5.42
				CMB					
1500 750	1.07M	0.03s	0.12s	0.22x	0.01s	0.01s	0.11s	0.15ms	7.40
3000 1500	4.29M	0.09s	0.15s	0.61x	0.04s	0.02s	0.13s	0.44ms	10.16
6400 3200	19.5M	0.37s	0.25s	1.49x	0.13s	0.08s	0.14s	1.94ms	10.56

Table 6.2: In our analysis, we present average timings derived from processing a variable number of files, aiming to illustrate how performance scales with workload size. This data unequivocally demonstrates the impact of GPU overhead, which becomes significantly less burdensome when spread over multiple samples, especially noticeable when dealing with singular, smaller files. A critical observation from this table is the predominant limitation imposed on GPU efficiency by disk input/output operations, highlighting a key bottleneck in achieving optimal GPU performance.

	Input size(B)	GPU overall avg. [ms]	GPU disk read avg. [ms]
		Uniform Noise	
$128^2 \times 1$	64K	119.83	0.69
$128^2 \times 100$	6.25M	1.77	0.46
$128^2 \times 1000$	62.5M	0.66	0.45
$128^2 \times 10000$	625M	0.52	0.42
		Gaussian Random Field	
$128^3 \times 1$	8M	124.68	12.02
$128^3 \times 10$	80M	28.13	13.86
$128^3 \times 100$	800M	15.38	13.82
$128^3 \times 1000$	8000M	11.96	11.67

Datasets Utilized in the Study. Our analysis incorporates a diverse array of datasets, blending both synthetic creations and data captured from real-world phenomena:

- The cosmic microwave background (CMB) imaging data is sourced from astrophysical observations, mapping the early universe. We utilize a planar projection of the original spherical data at various resolutions, with each image comprising no more than 256 distinct values.

- From the Virtual Imaging Clinical Trials for Regulatory Evaluation (VIC-TRE) initiative, we employ simulated 3D imaging of breast tissue, generating a series of 20 volumetric breast images. These images are notable for their limited palette of 11 unique values, emphasizing the project's focus on realism in medical imaging simulations.

- Our study further extends to 70 samples of 2D Gaussian Random Fields (GRF) across 7 distinct sizes (10 iterations per size) and 30 samples of 3D GRFs, organized into 3 size categories (with 10 iterations each).

These images are characterized by a spectrum of 1024 unique values, offering insights into statistical properties over space.

- For assessments on a larger scale, we generate datasets through uniform distribution sampling across each voxel, referred to herein as uniform noise, to test the scalability and performance under varied data conditions.

With the exception of the CMB dataset—which is encoded as 8-bit unsigned integers—the rest of the data is formatted in binary as 32-bit IEEE 754 floating-point numbers, ensuring precision and consistency across our computational evaluations.

Evaluating Voxel Processing Efficiency. Our primary interest lies in quantifying the computational efficiency through the metric of *voxel throughput*, essentially the volume of image data (in pixels or voxels) that can be processed per second. We denote this measure in GVox/s, representing billions (10^9) of voxels processed each second, with temporal metrics recorded in milliseconds (ms, 10^{-3} seconds).

6.4.1 Analyzing Single Image Processing Performance

For our evaluation involving single images stored on disk, we benchmark against CHUNKYEuler by Heiss and Wagner [143], a leading CPU-based solution for parallel streaming ECC computations. This tool represents the apex of current CPU capabilities for handling data sizes within our experimental scope. Our tests utilize the full capacity of all eight CPU cores available.

Comprehensive Execution Time Assessment. In this scenario, our analysis captures the total execution time, incorporating the duration required to read the image data from disk storage; detailed results are compiled in Table 6.1. Notably, for image files under 16MB, the CPU implementation demonstrates superior speed, attributed to the initial overhead encountered in launching GPU processes. Conversely, for larger files exceeding 0.5GB, the GPU-based method outpaces the CPU version by a factor of 4 to 6 times. However, it's important to underscore that disk I/O operations consume a substantial portion of the GPU execution time, ranging from 75% to 99.7%.

Efficient Streaming for Large-Scale Data Handling. Our approach successfully processes image files far exceeding the limits of the 8GB GPU memory and 16GB system RAM through an innovative streaming algorithm. Overcoming this significant challenge, particularly highlighted in Section 6.4, allowed us to process an image of 4096^3 dimensions, equivalent to 0.25TB of data. This

accomplishment underscores the effectiveness of our streaming methodology in managing data sizes beyond conventional memory capacities.

Understanding GPU Overhead. The previously discussed overhead is attributed to several GPU-related activities, including device initialization, shutdown procedures, and memory allocations specific to our computational framework. This initial overhead is observed to range between 100 and 700 milliseconds and incurs only once. This characteristic underpins the GPU's enhanced efficiency not only for processing substantial datasets but also when dealing with multiple smaller datasets collectively. The subsequent sections will delve into this aspect further.

6.4.2 Exploratory Study: Batch Image Processing

Our exploration extends to batch processing scenarios involving multiple image files, particularly emphasizing smaller file sizes that previously posed challenges due to the inherent GPU overhead. As depicted in Table 6.2, this overhead becomes less significant when the GPU processes several files concurrently, leading to a preference for GPU processing over CPU methods in batch scenarios. However, it's crucial to acknowledge that this does not represent an optimal condition for GPU utilization, given that performance continues to be constrained by disk input/output operations.

Future Directions. This observation paves the way for potentially innovative strategies, such as adopting compressed image formats to alleviate disk I/O time constraints. Such approaches hold promise for further enhancing processing efficiency and will be a focal point of our forthcoming research endeavors.

6.4.3 Focused Study: GPU-Centric Processing Framework

Transitioning to a scenario that aligns closely with our core objectives, we investigate an operational context where image data resides and undergoes processing exclusively within GPU memory. This setup simulates end-to-end GPU-centric pipelines, notably those involving advanced convolutional neural network (CNN) applications. Our aim is to ascertain whether our ECC kernel can seamlessly integrate into such GPU-dedicated pipelines without introducing significant performance impediments. Moreover, this evaluation seeks to confirm the absence of unanticipated bottlenecks within our computational architecture, ensuring the viability of our approach for high-speed, GPU-exclusive analytical workflows.

Table 6.3: This table delineates the execution times associated with a pipeline that iteratively conducts ECC and Gaussian smoothing operations. The pivotal insight derived from this analysis is the dominance of the overall computational duration by the execution of these two kernels when the process is averaged over several iterations. Such a pattern emphatically confirms the absence of extraneous bottlenecks within the pipeline, particularly validating the efficiency of our ECC computational process. It's noteworthy that the image data is loaded from disk only once at the onset, rendering the disk loading duration a singular expense in the context of this pipeline's execution timeline.

	Overall [ms]	Overall avg. [ms]	ECC mem. avg. [ms]	ECC exec. avg. [ms]	Gaussian exec. avg. [ms]	Disk read [ms]
			Uniform Noise			
(ECC+Gaussian) × 1	137.16	137.16	0.28	0.16	1.55	7.72
(ECC+Gaussian) × 10	172.80	17.28	0.06	0.15	0.20	7.38
(ECC+Gaussian) × 100	149.96	1.50	0.03	0.13	0.09	7.81
(ECC+Gaussian) × 1000	352.02	0.35	0.03	0.12	0.07	7.22
(ECC+Gaussian) × 1000	2786.64	0.28	0.03	0.17	0.07	7.57

Pipeline Configuration. We devise a pipeline comprising two key stages aimed at enhancing image analysis efficiency: (1) ECC computation; followed by (2) application of a Gaussian smoothing filter. These steps are cyclically executed on an image preloaded into GPU memory, with the process iterating up to 10,000 times on a 1024^2 Gaussian Random Field (GRF) image. After every iteration, the Vector of Changes in the Euler Characteristic (VCEC) is transferred to RAM for ECC computation and further processing.

Gaussian Smoothing Filter. The Gaussian smoothing filter is implemented as a discrete Gaussian convolution, leveraging its separability for optimized performance through a specially designed GPU kernel. The Gaussian kernel is set to a width of 13 pixels, as depicted in Fig. 6.5, to ensure effective smoothing.

Identifying Potential Performance Bottlenecks. Given that the image is initially loaded into the GPU memory, the disk read time is effectively distributed over numerous kernel executions. This distribution also applies to the initial GPU overhead. As demonstrated, the process of transferring the VCEC from GPU to RAM and its subsequent CPU processing (column "ECC mem" in Table 6.3) does not significantly affect performance. The primary contributors to the overall processing time are the executions of the kernel

Figure 6.5: Images at consecutive steps in the smoothing pipeline.

operations themselves.

Comparative Performance Analysis. This setup enables a direct performance comparison between the ECC kernel and the Gaussian smoothing kernel. According to Table 6.3, both kernels exhibit comparable throughput rates, with the Gaussian kernel achieving up to 2.5 times faster performance. Nonetheless, its influence on the comprehensive performance within a CNN framework is expected to be minimal, considering that convolution operations typically account for less than half of a convolutional layer's total processing time.

ECC Kernel Efficacy. The efficacy of the ECC kernel is underscored by its processing throughput, ranging between 5 and 10 GVox/s. This throughput facilitates the efficient handling of:

1. Volumetric 3D images of 512^3 voxels at a frequency of 30Hz;
2. High-resolution 2D images with 8K dimensions (7680×4320 pixels) at a frequency of 120Hz.

These metrics underscore the ECC kernel's capability to support high-frequency analysis across various image resolutions, enhancing the utility of GPU-accelerated image processing pipelines.

Table 6.4: The table presents the averaged total and kernel-specific execution times across different sizes of binary data inputs. In this context, "binary data" implies an input grid consisting solely of 0s and 1s. The binary nature of the data leads to more sequential accesses within the histogram operations of the GPU kernels, which is evidenced by the extended GPU total and kernel execution times in comparison to those observed with Gaussian Random Field (GRF) data. This increase in processing time highlights the impact of data structure on the efficiency of GPU computations.

	GPU overall avg. [ms] bin.—GRF—(bin.-GRF)/GRF	GPU exec. (kernel) [ms] bin.—GRF—(bin.-GRF)/GRF	GPU kernel Gvox/s bin.—GRF—(GRF-bin.)/GRF
512^3	867.09 — 859.22 — 0.92%	29.44 — 20.88 — 40.99%	4.56 — 6.43 — 29.08%
256^3	244.16 — 243.52 — 0.26%	3.70 — 2.64 — 40.15%	4.53 — 6.35 — 28.66%
128^3	149.04 — 124.68 — 19.53%	0.48 — 0.35 — 37.14%	4.37 — 6.00 — 27.16%
8192^2	559.27 — 534.71 — 4.59%	15.53 — 6.64 — 133.88%	4.32 — 10.10 — 57.23%
4096^2	236.76 — 210.41 — 12.52%	3.97 — 1.74 — 128.16%	4.23 — 9.64 — 56.12%
2048^2	190.13 — 178.13 — 6.73%	1.04 — 0.45 — 131.11%	4.03 — 9.32 — 56.76%

6.4.4 Assessment of Binary Data Handling and Exploration of Optimizations

The utilization of *atomicAdd* function is crucial for preventing race conditions when concurrent threads target the same bin in the VCEC array, leading to sequential access patterns that can undermine parallelism. This issue becomes more pronounced with inputs having limited unique values, causing numerous threads to converge on identical memory locations. In this context, we employ binary data—essentially a grid comprised solely of 0s and 1s—as a stringent test case for our algorithm. We further delve into two potential optimization strategies: (1) a 2-round VCEC accumulation process and (2) a warp-level VCEC computation approach, aiming to curb the serial access bottleneck to the VCEC array. It's noteworthy, however, that these strategies were ultimately not incorporated into our final implementation due to the marginal performance gains observed.

Binary Data Performance Insights. The performance implications for binary inputs are delineated in Table 6.4, juxtaposed with selected data from Table 6.1 to facilitate a comparative analysis with Gaussian random fields (GRF) data. For binary data, the GPU *kernel execution* time sees a significant uptick—approximately 130% in 2D scenarios and 40% in 3D contexts. Despite being a challenging test case, the GPU kernel maintains commendable efficiency, processing over 4 Gvoxels per second. Nonetheless, the exploration into mitigating the effects of serialized VCEC writes is warranted.

2-round VCEC Accumulation Exploration. A novel 2-round VCEC ac-

cumulation approach was prototyped to address the challenge of serialized access during concurrent thread writes. This technique, designed to enhance performance in situations characterized by a multitude of thread blocks and a constrained range of values, modifies the traditional method where each block directly contributes to a singular global VCEC vector. Instead, contributions are first made to intermediary vectors which are subsequently merged to construct the global vector. This strategy, though innovative, was not adopted in our final implementation due to its limited impact on overall performance enhancement.

Table 6.5 presents our experimental findings, where the kernel execution time for the 2-round scheme encompasses both the ECC computation and the aggregation of intermediate vectors. The subscript percentages denote the proportion of intermediate VCEC vectors relative to the total number of thread blocks. For context, results from our original GPU implementation on both Gaussian Random Fields (GRF) and binary data are recapitulated in Table 6.4. While the 2-round scheme tends to augment overall overhead, resulting in prolonged GPU and kernel execution times for GRF data, a notable exception is observed with 2D datasets of size 8192×8192. Here, the scheme demonstrates an enhancement in performance attributable to the sizable nature of the input and its comparatively limited value spectrum, featuring 600 unique values versus the 1000+ found in 3D datasets sized 512^3. Conversely, for binary data, the reduced conflict from concurrent writes to bins for voxel values 0 and 1 under the 2-round VCEC accumulation scheme is manifested through marked improvements in kernel execution time.

The introduction of a hyper-parameter dictating the ratio of intermediate vectors to thread blocks, alongside an additional kernel for vector aggregation, characterizes the 2-round VCEC accumulation scheme. Notwithstanding, the supplementary kernel's overhead generally surpasses the computational benefits in the context of moderate-sized inputs. Consequently, enhanced performance is predominantly confined to particular conditions, such as expansive inputs with restricted value distributions.

6.4.5 Optimization through Warp-Level VCEC Computation

Utilizing warp-level primitives within CUDA, our approach significantly enhances parallelism within each warp. This optimization allows threads to individually access registers, manage distinct memory addresses, and pursue divergent control flow trajectories, streamlining computational efficiency. Furthermore, inter-warp data exchange is directly facilitated through registers,

circumventing the need for shared memory's additional load and store operations, as well as the overhead of using an extra register for address storage. This method not only refines the execution efficiency but also simplifies the data handling process, making it a superior alternative to traditional shared memory utilization.

Table 6.5: This table presents a comparison of average kernel execution times across three distinct implementations when applied to Gaussian Random Field (GRF) and binary data. The implementations are as follows: (1) the original GPU implementation, (2) an optimized GPU approach employing a 2-round VCEC accumulation strategy with a 3% ratio of intermediate vectors to thread blocks ($GPU_{3\%}$), and (3) a variation of the 2-round VCEC accumulation with a heightened 30% ratio. Additionally, the fourth column, denoted as GPU_{warp}, showcases the execution times utilizing warp-level VCEC computation specifically for binary data. This comparative analysis aims to elucidate the performance impacts of various optimization strategies on the kernel execution times within different data contexts.

	GRF exec. (kernel) [ms]	Binary exec. (kernel) [ms]
	GPU — $GPU_{3\%}$ — $GPU_{30\%}$	GPU — $GPU_{3\%}$ — $GPU_{30\%}$ — GPU_{warp}
512^3	20.88 — 22.12 — 23.07	29.44 — 29.43 — 29.25 — 22.8
256^3	2.64 — 2.75 — 2.76	3.70 — 3.67 — 3.67 — 2.91
128^3	0.35 — 0.38 — 0.39	0.48 — 0.47 — 0.48 — 0.49
8192^2	6.64 — 6.44 — 6.56	15.53 — 14.28 — 14.26 — 14.87
4096^2	1.74 — 1.76 — 1.85	3.97 — 3.74 — 3.75 — 3.24
2048^2	0.45 — 0.45 — 0.46	1.04 — 0.96 — 0.95 — 0.72

We explored the efficiency of warp-level VCEC computation for binary data, implementing a two-pass method that aggregates changes in the Euler Characteristic separately for voxels valued at 0 and 1. This method employs warp-level reductions, culminating with the first thread of each warp atomically updating the local VCEC. Subsequent to both passes, these local results are atomically integrated into the global VCEC. The outcomes, denoted as GPU_{warp} in our comparative analysis, revealed this approach outperforms the 2-round VCEC accumulation for binary datasets.

This warp-level strategy mandates that threads alternate between simulating a zero-change in Euler characteristic and executing actual reductions, a process potentially costly in worst-case scenarios involving diverse voxel values. Despite its advantages for binary data, this approach's efficacy diminishes with inputs characterized by a broader spectrum of values, especially when dealing

with floating-point data. The method showcases limited applicability, primarily beneficial for datasets with minimal unique value counts, thereby restricting its general utility across more varied or complex data types.

Table 6.6: This table displays the average execution times for images constructed by stitching together varying numbers of base images. Each base image measures 64 by 64 pixels in two dimensions and 64 by 64 by 64 voxels in three dimensions. The data clearly demonstrates that as the number of base images stitched together increases, the processing time required per base image decreases. For these tests, we utilized images generated from Gaussian random fields.

	Stitch factor	GPU overall avg. [ms]	GPU disk read avg. [ms]	GPU overall avg. / stitch factor [ms]
64^2	1	113.12	0.86	113.12
128^2	4	124.24	1.12	31.06
256^2	16	129.50	1.50	8.09
512^2	64	130.36	2.46	2.03
1024^2	256	137.00	6.05	0.53
64^3	1	113.81	8.56	113.81
128^3	8	124.68	12.02	15.58
256^3	64	206.34	75.44	3.22
512^3	512	984.64	797.54	1.92

In summary, the 2-round VCEC accumulation method demonstrated a modest improvement in kernel execution time for binary data, achieving an average speedup of 4.7%. However, it introduced an average slowdown of 5.6% for Gaussian Random Field (GRF) data, attributing to the added overhead. Conversely, warp-level VCEC computation on binary data yielded a notable speedup of 20.9% on average, suggesting potential benefits for specific datasets. Despite these findings, given the marginal improvements and the broader applicability of our method to a wide range of data with diverse values, we opted against integrating these two techniques into our final implementation.

6.4.6 Case study: Batch processing with image stitching

Exploring a novel approach to batch processing, this case study investigates the effects of stitching multiple base files into a single, larger image for GPU analysis. Utilizing base images of size 64x64 in 2D and 64x64x64 in 3D, our

experiments, as shown in Table 6.6, indicate that larger stitching factors significantly reduce the processing time per base image. This method effectively spreads the one-time GPU overhead across numerous base images, enhancing GPU resource utilization and effectively minimizing latency with larger stitched images. Our results confirm that the overhead associated with stitching is negligible, offering a viable strategy for batch processing large datasets.

6.4.7 Dependence on dimension

Interestingly, our findings reveal that the algorithm's performance remains consistent across different image dimensions, underscoring the effectiveness of our caching strategy. The seamless retrieval of neighboring voxels from the cache mitigates the impact of having a higher number of neighbors (8 in 2D vs 26 in 3D), thus eliminating any significant performance difference between two and three-dimensional images. This caching efficacy, however, is limited to dimensions three or lower due to the constraints of the texture cache, indicating that our approach's advantages may not extend into higher-dimensional spaces.

6.5 Discussion

This work unveils a novel GPU-based methodology for efficiently computing the Euler Characteristic Curve (ECC) from imaging data. Our developed software stands out for its exceptional functionality and high-speed performance, offering significant advantages:

- **Accelerated Processing:** Our GPU-based approach facilitates rapid image processing, achieving an impressive throughput of over 4×10^9 voxels per second when the data is stored in GPU memory. This capability is particularly beneficial in the realm of convolutional neural networks (CNNs), where swift data processing is paramount.

- **Advanced Streaming Capabilities:** The algorithm effectively manages large-scale datasets by leveraging streaming, thus overcoming the inherent limitations of GPU memory capacity. This adaptability ensures the processing of extensive images without performance degradation.

- **Comprehensive Topological Analysis:** By calculating the ECC, the software captures vital topological information, demonstrating successful application across various domains and contributing to a deeper understanding of the data's inherent structure.

Looking ahead, we identify promising directions for future research and application. Our objectives are two-pronged:

1. *Integration into Convolutional Neural Networks:* We aim to seamlessly incorporate ECC calculations within CNN workflows. Bridging the current gap in efficiency, we aspire to embed topological methods firmly within mainstream machine learning paradigms, enhancing model interpretability and performance.

2. *Exploration of Persistent Homology:* Building on our specialized knowledge in GPU handling of cubical filtrations, we are motivated to explore the full potential of persistent homology in this context. Anticipating the development of pioneering GPU algorithms for persistence analysis, we hope to set new benchmarks in the field of topological data analysis.

GPU algorithms can also be applied to other fields like clustering method [260]. In conclusion, our work not only presents a breakthrough in topological data processing but also lays the groundwork for substantial advancements in machine learning and data analysis techniques.

6.6 Literature Review: Efficient Computation of Euler Characteristic Curves with GPU on Image Data

Inspired by our GPU computation of Euler Characteristic Curves, Choi et al. present a GPU-friendly and computationally efficient algorithm to enumerate toric seeds up to simplicial isomorphism, focusing on seeds with Picard number 4 [261]. This work extends the classification of toric manifolds with small Picard numbers, building on the foundational classifications by Kleinschmidt (1988) and Batyrev (1991) for Picard number≤ 3.

Malott el al. examine the dimensional Euler Characteristic Curve (ECC), an efficient alternative to Persistent Homology (PH) for Topological Data Analysis (TDA) [262]. It introduces an improved algorithm for computing the dimensional ECC and interprets the results as proximity-series representations of topological features. The study explores methods to compare and classify ECC curves, demonstrating their effectiveness and scalability in classifying MRA Brain Artery scans, addressing limitations of PH in handling large data sets.

Drawing inspiration from my work, Laky et al. present parallel algorithms and software implementations for fast computation of the Euler characteristic

(EC) for high-resolution 2D and 3D field data using vertex contributions [263]. The proposed methods significantly outperform existing tools like GUDHI and match the speed of CHUNKYEuler, while also being capable of calculating other topological descriptors such as perimeter, area, and volume. Tested on synthetic data and real applications, the algorithms also include low-memory versions for handling large data objects. All data and software are shared as open-source code.

Dłotko et al. explore Euler characteristic curves for 1-parameter filtrations and Euler characteristic profiles for multiparameter filtrations as alternatives to persistent homology in topological data analysis [264]. These Euler characteristic–based approaches overcome persistent homology's limitations, such as difficulty in distribution, generalization to multifiltrations, and computational intensity for large datasets. The paper presents efficient algorithms for distributed computation, demonstrates their stability, and showcases their practical applicability through multiple use cases, making them robust tools for big data analysis.

Munch provides a comprehensive overview of the Euler Characteristic Transform (ECT), a powerful shape representation method. ECT encodes an embedded shape using sub-level sets of a direction-based function, returning the Euler characteristics of these sets [265]. Demonstrated to be injective on the space of embedded simplicial complexes, ECT has found applications across various disciplines, including plant morphology and protein structural analysis. The article explains the main idea using a simple leaf example and surveys the key concepts, theoretical foundations, and available applications of ECT.

Chapter 7

Conclusion and Future Work

This thesis first introduces innovative applications of topology in deep learning frameworks, highlighting the integration of topological insights to improve model robustness and interpretability. In Chapter 3, we discuss the development of a Topology-Aware Generative Adversarial Network (GAN), which incorporates topological loss functions to guide the training process, thereby enhancing the generative model's ability to produce images with accurate topological features. Another application is detailed in Chapter 4, where we present a novel approach for identifying topological biomarkers in Breast Dynamic Contrast-Enhanced Magnetic Resonance Imaging (DCE-MRI). This method leverages topological data analysis to discern subtle patterns and structures in medical images that are indicative of underlying biological processes, offering potential advancements in diagnostic imaging. Collectively, these chapters showcase the potent integration of topological methods into high-performance computing and deep learning, setting a new frontier in both fields.

Additionally, we embark on a detailed exploration of hardware-accelerated techniques for computing persistent homology, a key tool in topological data analysis. The investigation is structured into three distinct sections. The first section, presented in Chapter 5, delves into the development of a hardware accelerator specifically designed for efficient boundary matrix reduction, a critical step in calculating persistent homology. The second section, discussed in Chapter 6, focuses on GPU computation of Euler Characteristic Curves, leveraging the parallel processing capabilities of GPUs to enhance computational efficiency and scalability. The third section extends the discussion to a comprehensive GPU-based framework for persistent homology computation, aiming to significantly reduce computation time and resource consumption.

Despite the substantial progress made in the field of topological data analysis (TDA), there remains an acute need for fast and reliable tools capable of managing the complexities and scale of modern datasets. TDA can find

applications in fields such as 3D vision [266, 267]. As we advance, my research will focus on developing solutions that not only enhance computational efficiency but also expand the accessibility and application of TDA. Below are the directions I plan to explore in the next stage of my work:

- **Efficient Persistent Homology Computation Tool:** A paramount objective is to create a tool that significantly outpaces current solutions in computing persistent homology. This tool will be designed to deliver accurate results reliably across a wide range of data sizes and precision levels. Importantly, it will be user-friendly, easy to install, and compile, ideally packaged within a Conda environment to facilitate widespread adoption and ease of use.

- **Visualization Tool for Morse Complexes:** Another key development will be a dedicated visualization tool tailored for Morse complexes. This tool aims to serve both the topological data analysis and the broader visualization communities. It will provide intuitive and insightful visual representations of complex topological structures, making it easier for researchers and practitioners to interpret and communicate their results.

- **Deep Learning Framework for Topology Computation:** I aim to develop a deep learning framework capable of integrating gradient propagation directly through topological computation algorithms. This innovative framework will bridge the gap between deep learning and topological analysis, enabling new classes of algorithms that can learn from and leverage topological data in a fundamentally new way.

- **Efficient Boundary Matrix Reduction Algorithm on GPU:** To address the computational bottlenecks in TDA, particularly in the reduction of boundary matrices, I will work on developing an algorithm optimized for GPU execution. This effort will focus on harnessing the parallel processing power of GPUs to accelerate the reduction process, thereby enabling faster and more scalable computations that are crucial for handling large-scale topological data.

Through these initiatives, I aim to push the boundaries of what's possible in topological data analysis, making it faster, more accurate, and more applicable to a variety of real-world problems.

Bibliography

[1] Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction.* American Mathematical Society, 2010. ISBN 978-0-8218-4925-5.

[2] Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, Nov 2002. ISSN 1432-0444. doi: 10.1007/s00454-002-2885-2. URL https://doi.org/10.1007/s00454-002-2885-2.

[3] Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. The structure and stability of persistence modules. *arXiv preprint arXiv:1207.3674*, 21, 2012.

[4] Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, Aarti Singh, et al. Confidence sets for persistence diagrams. *The Annals of Statistics*, 42(6):2301–2339, 2014.

[5] Gunnar Carlsson, Afra Zomorodian, Anne Collins, and Leonidas Guibas. Persistence barcodes for shapes. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, page 124–135, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 3905673134. doi: 10.1145/1057432.1057449. URL https://doi.org/10.1145/1057432.1057449.

[6] Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures, 2020.

[7] T. Yung Kong. *Digital Topology*, pages 73–93. Springer US, Boston, MA, 2001. ISBN 978-1-4615-1529-6. doi: 10.1007/978-1-4615-1529-6_3. URL https://doi.org/10.1007/978-1-4615-1529-6_3.

[8] Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In *Advances in Neural Information Processing Systems*, pages 5658–5669, 2019.

[9] Pengxiang Wu, Chao Chen, Yusu Wang, Shaoting Zhang, Changhe Yuan, Zhen Qian, Dimitris Metaxas, and Leon Axel. Optimal Topological Cycles and Their Application in Cardiac Trabeculae Restoration. In *In International Conference on Information Processing in Medical Imaging (IPMI), 2017*, pages 80–92, 05 2017. doi: 10.1007/978-3-319-59050-9_7.

[10] Roland Kwitt, Stefan Huber, Marc Niethammer, Weili Lin, and Ulrich Bauer. Statistical topological data analysis-a kernel perspective. In *Advances in neural information processing systems*, pages 3070–3078, 2015.

[11] Eleanor Wong, Sourabh Palande, Bei Wang, Brandon Zielinski, Jeffrey Anderson, and P Thomas Fletcher. Kernel partial least squares regression for relating functional brain network topology to clinical measures of behavior. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 1303–1306. IEEE, 2016.

[12] Frédéric Chazal, Leonidas J Guibas, Steve Y Oudot, and Primoz Skraba. Persistence-based clustering in riemannian manifolds. *Journal of the ACM (JACM)*, 60(6):41, 2013.

[13] Xiuyan Ni, Novi Quadrianto, Yusu Wang, and Chao Chen. Composing tree graphical models with persistent homology features for clustering mixed-type data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2622–2631. JMLR. org, 2017.

[14] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research*, 18(1):218–252, 2017.

[15] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research*, 16(1):77–102, 2015.

[16] Kush R Varshney and Karthikeyan Natesan Ramamurthy. Persistent topology of decision boundaries. In *2015 IEEE International Conference*

on *Acoustics, Speech and Signal Processing (ICASSP)*, pages 3931–3935. IEEE, 2015.

[17] Qi Zhao, Ze Ye, Chao Chen, and Yusu Wang. Persistence enhanced graph neural network. In *International Conference on Artificial Intelligence and Statistics*, pages 2896–2906, 2020.

[18] Zuoyu Yan, Tengfei Ma, Liangcai Gao, Zhi Tang, and Chao Chen. Link prediction with persistent homology: An interactive view. In *International Conference on Machine Learning*, 2021. URL `https://api.semanticscholar.org/CorpusID:235422112`.

[19] Pengxiang Wu, Songzhu Zheng, Mayank Goswami, Dimitris Metaxas, and Chao Chen. A topological filter for learning with label noise. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[20] Scott Kulp, Chao Chen, Dimitris Metaxas, and Leon Axel. Ventricular blood flow analysis using topological methods. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pages 663–666. IEEE, 2015.

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

[22] Shahira Abousamra, Minh Hoai, Dimitris Samaras, and Chao Chen. Localization in the crowd with topological constraints. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 872–881. AAAI Press, 2021. doi: 10.1609/AAAI.V35I2.16170. URL `https://doi.org/10.1609/aaai.v35i2.16170`.

[23] 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*, page 118–136. Springer-Verlag, 2020. doi: 10.1007/978-3-030-58580-8_8.

[24] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Mandar Dixit. Connectivity-optimized representation learning via persistent homology. In *International Conference on Machine Learning*, pages 2751–2760, 2019.

[25] Christoph Hofer, Florian Graf, Bastian Rieck, Marc Niethammer, and Roland Kwitt. Graph filtration learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4314–4323. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/hofer20b.html`.

[26] 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*, pages 2573–2582, 2019.

[27] Yixiong Lin, Mengru Shi, Dawei Xiang, Peisheng Zeng, Zhuohong Gong, Haiwen Liu, Quan Liu, Zhuofan Chen, Juan Xia, and Zetao Chen. Construction of an end-to-end regression neural network for the determination of a quantitative index sagittal root inclination. *Journal of Periodontology*, 93(12):1951–1960, 2022.

[28] Caitao Zhan, Mohammad Ghaderibaneh, Pranjal Sahu, and Himanshu Gupta. Deepmtl: Deep learning based multiple transmitter localization. In *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 41–50, 2021.

[29] Caitao Zhan, Mohammad Ghaderibaneh, Pranjal Sahu, and Himanshu Gupta. Deepmtl pro: Deep learning based multiple transmitter localization and power estimation. *Pervasive Mob. Comput.*, 82(C), jun 2022.

[30] Jun Wu, Xuesong Ye, Chengjie Mou, and Weinan Dai. Fineehr: Refine clinical note representations to improve mortality prediction. In *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–6, 2023. doi: 10.1109/ISDFS58141.2023.10131726.

[31] Jun Wu, Xuesong Ye, and Yanyuet Man. Bottrinet: A unified and efficient embedding for social bots detection via metric learning. In *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–6, 2023. doi: 10.1109/ISDFS58141.2023.10131839.

[32] Fan Wang, Saarthak Kapse, Steven Liu, Prateek Prasanna, and Chao Chen. TopoTxR: A Topological Biomarker for Predicting Treatment Response in Breast Cancer. In *Information Processing in Medical Imaging - 27th International Conference, IPMI*, volume 12729 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 2021. doi: 10.1007/978-3-030-78191-0_30.

[33] Jiachen Yao, Yikai Zhang, Songzhu Zheng, Mayank Goswami, Prateek Prasanna, and Chao Chen. Learning to segment from noisy annotations: A spatial correction approach. In *International Conference on Learning Representations*, 2023.

[34] Weimin Lyu, Songzhu Zheng, Tengfei Ma, and Chao Chen. A study of the attention abnormality in trojaned BERTs. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4727–4741, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.348. URL https://aclanthology.org/2022.naacl-main.348.

[35] Weimin Lyu, Songzhu Zheng, Lu Pang, Haibin Ling, and Chao Chen. Attention-enhancing backdoor attacks against BERT-based models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.716. URL https://aclanthology.org/2023.findings-emnlp.716.

[36] Weimin Lyu, Songzhu Zheng, Tengfei Ma, Haibin Ling, and Chao Chen. Attention hijacking in trojan transformers, 2022.

[37] Weimin Lyu, Songzhu Zheng, Haibin Ling, and Chao Chen. Backdoor attacks against transformers with attention enhancement. In *ICLR 2023 Workshop on Backdoor Attacks and Defenses in Machine Learning*, 2023. URL https://openreview.net/forum?id=QmrHMnzJG0-.

[38] Weimin Lyu, Xiao Lin, Songzhu Zheng, Lu Pang, Haibin Ling, Susmit Jha, and Chao Chen. Task-agnostic detector for insertion-based backdoor attacks, 2024.

[39] Songzhu Zheng, Yikai Zhang, Lu Pang, Weimin Lyu, Mayank Goswami, Anderson Schneider, Yuriy Nevmyvaka, Haibin Ling, and Chao Chen.

On the existence of a trojaned twin model. In *ICLR 2023 Workshop on Backdoor Attacks and Defenses in Machine Learning*, 2023.

[40] Weimin Lyu, Xinyu Dong, Rachel Wong, Songzhu Zheng, Kayley Abell-Hart, Fusheng Wang, and Chao Chen. A multimodal transformer: Fusing clinical notes with structured ehr data for interpretable in-hospital mortality prediction. In *AMIA Annual Symposium Proceedings*, volume 2022, page 719. American Medical Informatics Association, 2022.

[41] Xinyu Dong, Rachel Wong, Weimin Lyu, Kayley Abell-Hart, Jianyuan Deng, Yinan Liu, Janos G Hajagos, Richard N Rosenthal, Chao Chen, and Fusheng Wang. An integrated lstm-heterorgnn model for interpretable opioid overdose risk prediction. *Artificial intelligence in medicine*, 135:102439, 2023.

[42] Chen Li, Xiaoling Hu, Shahira Abousamra, and Chao Chen. Calibrating uncertainty for semi-supervised crowd counting. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16685–16695, 2023. doi: 10.1109/ICCV51070.2023.01534.

[43] Chen Li, Xiaoling Hu, and Chao Chen. Confidence estimation using unlabeled data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sOXU-PEJSgQ.

[44] Lujain Alsaleh, Chen Li, Justin L. Couetil, Ze Ye, Kun Huang, Jie Zhang, Chao Chen, and Travis S. Johnson. Spatial transcriptomic analysis reveals associations between genes and cellular topology in breast and prostate cancers. *Cancers*, 14(19), 2022.

[45] Qiaomu Miao, Sinhwa Kang, Stacy Marsella, Steve DiPaola, Chao Wang, and Ari Shapiro. Study of detecting behavioral signatures within deepfake videos. *arXiv preprint arXiv:2208.03561*, 2022.

[46] Qiaomu Miao, Minh Hoai, and Dimitris Samaras. Patch-level gaze distribution prediction for gaze following. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 880–889, January 2023.

[47] Jiaxiang Ren, Heng Fan, Jie Yang, and Haibin Ling. Detection of trabecular landmarks for osteoporosis prescreening in dental panoramic radiographs. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 2194–2197, 2020. doi: 10.1109/EMBC44109.2020.9175281.

[48] Yasha Singh, Vivek Atulkar, Jiaxiang Ren, Jie Yang, Heng Fan, Longin Jan Latecki, and Haibin Ling. Osteoporosis prescreening and bone mineral density prediction using dental panoramic radiographs. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 2700–2703, 2021. doi: 10.1109/EMBC46164.2021.9630183.

[49] Jiaxiang Ren, Kicheon Park, Yingtian Pan, and Haibin Ling. Self-supervised bulk motion artifact removal in optical coherence tomography angiography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20617–20625, June 2022.

[50] Yingtian Pan, Kicheon Park, Jiaxiang Ren, Nora D Volkow, Haibin Ling, Alan P Koretsky, and Congwu Du. Dynamic 3d imaging of cerebral blood flow in awake mice using self-supervised-learning-enhanced optical coherence doppler tomography. *Communications Biology*, 6(1):298, 2023.

[51] Zhilin Zou, Kicheon Park, Jiaxiang Ren, Wensheng Cheng, Sophia Liu, Yingtian Pan, Congwu Du, and Haibin Ling. Weakly supervised detection of cell activation. In Barjor S. Gimi and Andrzej Krol, editors, *Medical Imaging 2024: Clinical and Biomedical Imaging*, volume 12930, page 129301J. International Society for Optics and Photonics, SPIE, 2024. doi: 10.1117/12.3005371. URL https://doi.org/10.1117/12.3005371.

[52] Haoxiang Wang, Yite Wang, Ruoyu Sun, and Bo Li. Global convergence of maml and theory-inspired neural architecture search for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9797–9808, 2022.

[53] Yite Wang, Dawei Li, and Ruoyu Sun. Ntk-sap: Improving neural network pruning by aligning training dynamics. In *The Eleventh International Conference on Learning Representations*, 2022.

[54] Yite Wang, Jing Wu, Naira Hovakimyan, and Ruoyu Sun. Balanced training for sparse gans. *Advances in Neural Information Processing Systems*, 36, 2024.

[55] Yite Wang, Jiahao Su, Hanlin Lu, Cong Xie, Tianyi Liu, Jianbo Yuan, Haibin Lin, Ruoyu Sun, and Hongxia Yang. Lemon: Lossless model expansion. In *The Twelfth International Conference on Learning Representations*, 2023.

[56] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[57] Xiaomin Li, Vangelis Metsis, Huangyingrui Wang, and Anne Hee Hiong Ngu. Tts-gan: A transformer-based time-series generative adversarial network. In *Artificial Intelligence in Medicine: 20th International Conference on Artificial Intelligence in Medicine, AIME 2022, Halifax, NS, Canada, June 14–17, 2022, Proceedings*, page 133–143, Berlin, Heidelberg, 2022. Springer-Verlag.

[58] Xiaomin Li, Anne Hee Hiong Ngu, and Vangelis Metsis. Tts-cgan: A transformer time-series conditional gan for biosignal data augmentation, 2022.

[59] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[60] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[61] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Machine Learning*, 2018.

[62] Huidong Liu, GU Xianfeng, and Dimitris Samaras. A two-step computation of the exact gan wasserstein distance. In *International Conference on Machine Learning*, pages 3165–3174, 2018.

[63] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Which training methods for gans do actually converge? In *International Conference on Machine Learning*, 2018.

[64] Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. Improving generalization and stability of generative adversarial networks. In *International Conference on Learning Representations*, 2019.

[65] Huidong Liu, Xianfeng Gu, and Dimitris Samaras. Wasserstein gan with quadratic transport cost. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[66] Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J. Guibas, and Gunnar E. Carlsson. A topology layer for machine learning. *CoRR*, abs/1905.12200, 2019. URL `http://arxiv.org/abs/1905.12200`.

[67] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.

[68] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[69] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[70] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4570–4580, 2019.

[71] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[72] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[73] Guo-Jun Qi, Liheng Zhang, Hao Hu, Marzieh Edraki, Jingdong Wang, and Xian-Sheng Hua. Global versus localized generative adversarial nets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1517–1525, 2018.

[74] Jean Kossaifi, Linh Tran, Yannis Panagakis, and Maja Pantic. Gagan: Geometry-aware generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 878–887, 2018.

[75] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2427–2436, 2019.

[76] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.

[77] Mohsen Ghafoorian, Cedric Nugteren, Nóra Baka, Olaf Booij, and Michael Hofmann. El-gan: Embedding loss driven generative adversarial networks for lane detection. In *European Conference on Computer Vision*, pages 256–272. Springer, 2018.

[78] Jyh-Jing Hwang, Tsung-Wei Ke, Jianbo Shi, and Stella X Yu. Adversarial structure matching for structured prediction tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4056–4065, 2019.

[79] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4741–4748, 2015.

[80] Genki Kusano, Yasuaki Hiraoka, and Kenji Fukumizu. Persistence weighted gaussian kernel for topological data analysis. In *International Conference on Machine Learning*, pages 2004–2013, 2016.

[81] Mathieu Carriere, Marco Cuturi, and Steve Oudot. Sliced wasserstein kernel for persistence diagrams. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 664–673. JMLR. org, 2017.

[82] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems*, pages 1634–1644, 2017.

[83] 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*, pages 16–28. Springer, 2019.

[84] Pengxiang Wu, Chao Chen, Yusu Wang, Shaoting Zhang, Changhe Yuan, Zhen Qian, Dimitris Metaxas, and Leon Axel. Optimal topological cycles and their application in cardiac trabeculae restoration. In *International Conference on Information Processing in Medical Imaging*, pages 80–92. Springer, 2017.

[85] Agata Mosinska, Pablo Márquez-Neila, Mateusz Koziński, and Pascal Fua. Beyond the pixel-wise loss for topology-aware delineation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[86] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[87] Yufan He, Aaron Carass, Yihao Liu, Bruno M. Jedynak, Sharon D. Solomon, Shiv Saidha, Peter A. Calabresi, and Jerry L. Prince. Fully convolutional boundary regression for retina oct segmentation. In Dinggang Shen, Tianming Liu, Terry M. Peters, Lawrence H. Staib, Caroline Essert, Sean Zhou, Pew-Thian Yap, and Ali Khan, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 120–128. Springer International Publishing, 2019. ISBN 978-3-030-32239-7.

[88] Yulei Qin, Mingjian Chen, Hao Zheng, Yun Gu, Mali Shen, Jie Yang, Xiaolin Huang, Yue-Min Zhu, and Guang-Zhong Yang. Airwaynet: A voxel-connectivity aware approach for accurate airway segmentation using convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 212–220. Springer, 2019.

[89] Karthikeyan Natesan Ramamurthy, Kush Varshney, and Krishnan Mody. Topological data analysis of decision boundaries with application to model selection. In *International Conference on Machine Learning*, pages 5351–5360, 2019.

[90] Qi Zhao and Yusu Wang. Learning metrics for persistence-based summaries and applications for graph classification. In *Advances in Neural Information Processing Systems*, pages 9859–9870, 2019.

[91] Valentin Khrulkov and Ivan Oseledets. Geometry score: A method for comparing generative adversarial networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2621–2629, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[92] 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.

[93] Elizabeth Cain, Ashirbani Saha, Michael Harowicz, Jeffrey Marks, Paul Marcom, and Maciej Mazurowski. Multivariate ML models for prediction of pCR to NAC in BCa using mri features: a study using an independent validation set. *BCa Research and Treatment*, 2019.

[94] Subramani Mani, Yukun Chen, Xia Li, Lori Arlinghaus, A Bapsi Chakravarthy, Vandana Abramson, Sandeep R Bhave, Mia A Levy, Hua Xu, and Thomas E Yankeelov. Machine learning for predicting the response of breast cancer to neoadjuvant chemotherapy. *JAMIA*, 2013.

[95] Nathaniel Braman, Prateek Prasanna, Jon Whitney, Salendra Singh, Niha Beig, Maryam Etesami, David DB Bates, Katherine Gallagher, B Nicolas Bloch, Manasa Vulchi, et al. Association of peritumoral radiomics with tumor biology and pathologic response to preoperative targeted therapy for HER2 (ERBB2)–positive breast cancer. *JAMA network open*, 2(4), 2019.

[96] Lars Grimm. Breast mri radiogenomics: Current status and research implications. *Journal of Magnetic Resonance Imaging*, 2015.

[97] Michael Z. Liu, Simukayi Mutasa, Peter Chang, Maham Siddique, Sachin Jambawalikar, and Richard Ha. A novel cnn algorithm for pathological complete response prediction using an i-spy trial breast mri database. *Magnetic Resonance Imaging*, 2020.

[98] Kavya Ravichandran, Nathaniel Braman, Andrew Janowczyk, and Anant Madabhushi. A deep learning classifier for prediction of pathological complete response to neoadjuvant chemotherapy from baseline breast dce-mri. In *Medical imaging 2018: computer-aided diagnosis*, volume 10575, pages 79–88. SPIE, 2018.

[99] Richard Ha, Christine Chin, Jenika Karcich, Michael Liu, Peter Chang, Simukayi Mutasa, Eduardo Sant, Ralph Wynn, Eileen Connolly, and Sachin Jambawalikar. Prior to initiation of chemotherapy, can we predict breast tumor response? deep learning convolutional neural networks approach using a breast mri tumor dataset. *Journal of Digital Imaging*, 2018. doi: 10.1007/s10278-018-0144-1.

[100] Yu-Hong Qu, Hai-Tao Zhu, Kun Cao, Xiao-Ting Li, Meng Ye, and Ying-Shi Sun. Prediction of pathological complete response to neoadjuvant chemotherapy in breast cancer using a deep learning (dl) method. *Thoracic Cancer*, 2020. doi: 10.1111/1759-7714.13309.

[101] Hongyi Duanmu, Pauline Boning Huang, Srinidhi Brahmavar, Stephanie Lin, Thomas Ren, Jun Kong, Fusheng Wang, and Tim Q Duong. Prediction of pathological complete response to neoadjuvant chemotherapy in breast cancer using deep learning with integrative imaging, molecular and demographic data. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23*, pages 242–252. Springer, 2020.

[102] Shahira Abousamra, Danielle Fassler, Jiachen Yao, Rajarsi R Gupta, Tahsin Kurc, Luisa Escobar-Hoyos, Dimitris Samaras, Kenneth Shroyer, Joel Saltz, and Chao Chen. Unsupervised stain decomposition via inversion regulation for multiplex immunohistochemistry images. In *Medical Imaging with Deep Learning*, 2023. URL `https://openreview.net/pdf?id=J0VD-I2IOOg`.

[103] Jiachen Yao, Nina Hagemann, Qiaojie Xiong, Jianxu Chen, Dirk M. Hermann, and Chao Chen. Topological analysis of mouse brain vasculature via 3d light-sheet microscopy images, 2024.

[104] Thijs Kooi, Geert Litjens, Bram Van Ginneken, Albert Gubern-Mérida, Clara I Sánchez, Ritse Mann, Ard den Heeten, and Nico Karssemeijer. Large scale deep learning for computer aided detection of mammographic lesions. *Medical image analysis*, 35:303–312, 2017.

[105] Dina Abdelhafiz, Clifford Yang, Reda Ammar, and Sheida Nabavi. Deep CNN for mammography: advances, challenges and applications. *BMC bioinformatics*, 2019.

[106] Abdulhamit Subasi, Aayush Dinesh Kandpal, Kolla Anant Raj, and Ulas Bagci. Breast cancer detection from mammograms using artificial intelligence. In *Applications of Artificial Intelligence in Medical Imaging*, pages 109–136. Elsevier, 2023.

[107] Kenji Akazawa, Yasuhiro Tamaki, Tetsuya Taguchi, Yoshio Tanji, Yasuo Miyoshi, Seung Jin Kim, Satsuki Ueda, Tetsu Yanagisawa, Yoshinobu Sato, Shinichi Tamura, et al. Preoperative evaluation of residual tumor extent by three-dimensional magnetic resonance imaging in breast cancer patients treated with neoadjuvant chemotherapy. *The breast journal*, 12 (2):130–137, 2006.

[108] Jason D Keune, Donna B Jeffe, Mario Schootman, Abigail Hoffman, William E Gillanders, and Rebecca L Aft. Accuracy of ultrasonography

and mammography in predicting pathologic response after neoadjuvant chemotherapy for breast cancer. *The American Journal of Surgery*, 199 (4):477–484, 2010.

[109] Viviana Londero, Massimo Bazzocchi, Chiara Del Frate, Fabio Puglisi, Carla Di Loreto, Giuliana Francescutti, and Chiara Zuiani. Locally advanced breast cancer: comparison of mammography, sonography and mr imaging in evaluation of residual disease in women receiving neoadjuvant chemotherapy. *European radiology*, 14:1371–1379, 2004.

[110] Juan Herrada, Revathy B Iyer, E Neely Atkinson, Nour Sneige, Aman U Buzdar, and Gabriel N Hortobagyi. Relative value of physical examination, mammography, and breast sonography in evaluating the size of the primary tumor and regional lymph node metastases in women receiving neoadjuvant chemotherapy for locally advanced breast carcinoma. *Clinical cancer research: an official journal of the American Association for Cancer Research*, 3(9):1565–1569, 1997.

[111] Sarah J Vinnicombe, A David MacVicar, Rosemary L Guy, John P Sloane, Trevor J Powles, Graham Knee, and Janet E Husband. Primary breast cancer: mammographic changes after neoadjuvant chemotherapy, with pathologic correlation. *Radiology*, 198(2):333–340, 1996.

[112] Gunter von Minckwitz, Serban Dan Costa, Wolfgang Eiermann, Jens-Uwe Blohmer, Augustinos Harjanto Tulusan, Christian Jackisch, and Manfred Kaufmann. Maximized reduction of primary breast tumor size using preoperative chemotherapy with doxorubicin and docetaxel. *Journal of clinical oncology*, 17(7):1999–1999, 1999.

[113] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[114] Hyekyoung Lee, Hyejin Kang, Moo K Chung, Bung-Nyun Kim, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE Trans Med Imaging*, 2012.

[115] Suprosanna Shit, Johannes Christian Paetzold, Anjany Kumar Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien P. W. Pluim, Ulrich Bauer, and Bjoern H Menze. cldice - a novel topology-preserving loss function for tubular structure segmentation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16555–16564, 2020. URL https://api.semanticscholar.org/CorpusID:232428560.

[116] Nico Stucki, Johannes C Paetzold, Suprosanna Shit, Bjoern Menze, and Ulrich Bauer. Topologically faithful image segmentation via induced matching of persistence barcodes. In *International Conference on Machine Learning*, pages 32698–32727. PMLR, 2023.

[117] Hongxiao Wang, Gang Huang, Zhuo Zhao, Liang Cheng, Anna Juncker-Jensen, Máté Levente Nagy, Xin Lu, Xiangliang Zhang, and Danny Z Chen. Ccf-gnn: A unified model aggregating appearance, microenvironment, and topology for pathology image classification. *IEEE Transactions on Medical Imaging*, 2023.

[118] Suraj Mishra, Ya Xing Wang, Chuan Wei, Danny Chen, and X. Hu. Vtg-net: A cnn based vessel topology graph network for retinal artery/vein classification. *Frontiers in Medicine*, 8:750396, 11 2021. doi: 10.3389/fmed.2021.750396.

[119] 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 *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part II*, pages 56–65. Springer, 2022.

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

[121] 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*, 2019.

[122] 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*, pages 118–136. Springer, 2020.

[123] 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.

[124] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[125] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Mask-guided contrastive attention model for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1179–1188, 2018.

[126] Keith J Worsley. The geometry of random images. *Chance*, 9(1):27–40, 1996.

[127] J Richard Gott III, Adrian L Melott, and Mark Dickinson. The sponge-like topology of large-scale structure in the universe. *The Astrophysical Journal*, 306:341–357, 1986.

[128] Dmitri I. Novikov, Hume A. Feldman, and Sergei F. Shandarin. Minkowski functionals and cluster analysis for CMB maps. *International Journal of Modern Physics D*, 08(03):291–306, 1999.

[129] Robert J. Adler and A. M. Hasofer. Level Crossings for Random Fields. *The Annals of Probability*, 4(1):1 – 12, 1976. doi: 10.1214/aop/1176996176. URL https://doi.org/10.1214/aop/1176996176.

[130] A Odgaard and HJG Gundersen. Quantification of connectivity in cancellous bone, with special emphasis on 3-D reconstructions. *Bone*, 14(2): 173–182, 1993.

[131] Berthold Horn, Berthold Klaus, and Paul Horn. *Robot vision*. The MIT Press, 1986.

[132] L. Snidaro and G. L. Foresti. Real-time thresholding with Euler numbers. *Pattern Recogn. Lett.*, 24(9–10):1533–1544, 2003. ISSN 0167-8655. doi: 10.1016/S0167-8655(02)00392-6. URL https://doi.org/10.1016/S0167-8655(02)00392-6.

[133] Omer Bobrowski and Primoz Skraba. Homological percolation and the Euler characteristic. *Phys. Rev. E*, 101:032304, Mar 2020. doi: 10.1103/PhysRevE.101.032304. URL https://link.aps.org/doi/10.1103/PhysRevE.101.032304.

[134] Lorin Crawford, Anthea Monod, Andrew X Chen, Sayan Mukherjee, and Raúl Rabadán. Predicting clinical outcomes in glioblastoma: an application of topological and functional data analysis. *Journal of the American Statistical Association*, 115(531):1139–1150, 2020.

[135] Erik J Amézquita, Michelle Y Quigley, Tim Ophelders, Jacob B Landis, Daniel Koenig, Elizabeth Munch, and Daniel H Chitwood. Measuring

hidden phenotype: quantifying the shape of barley seeds using the Euler characteristic transform. *in silico Plants*, 4(1), 12 2021. ISSN 2517-5025. doi: 10.1093/insilicoplants/diab033. URL `https://doi.org/10.1093/insilicoplants/diab033`.

[136] Wenyi Lin, Kyle Hasenstab, Guilherme Cunha, and Armin Schwartzman. Comparison of handcrafted features and convolutional neural networks for liver MR image adequacy assessment. *Scientific Reports*, 10, 11 2020. doi: 10.1038/s41598-020-77264-y. URL `https://doi.org/10.1038/s41598-020-77264-y`.

[137] Alexander Smith and Victor M. Zavala. The Euler characteristic: A general topological descriptor for complex data. *Computers & Chemical Engineering*, 154:107463, 2021. ISSN 0098-1354. doi: https://doi.org/10.1016/j.compchemeng.2021.107463. URL `https://www.sciencedirect.com/science/article/pii/S0098135421002416`.

[138] Leo M. Betthauser. *Topological Reconstruction of Grayscale Images*. PhD thesis, University of Florida, 2018. URL `https://people.clas.ufl.edu/peterbubenik/files/Betthauser_Thesis.pdf`.

[139] Qitong Jiang, Sebastian Kurtek, and Tom Needham. The Weighted Euler Curve Transform for Shape and Image Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[140] Tomasz Kaczyński, Konstantin Mischaikow, and Marian Mrozek. Computational homology. *Bull. Amer. Math. Soc*, 43:255–258, 2006.

[141] Djemel Ziou and Madjid Allili. Generating cubical complexes from image data and computation of the Euler number. *Pattern Recognition*, 35(12):2833 – 2839, 2002. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/S0031-3203(01)00238-2. URL `http://www.sciencedirect.com/science/article/pii/S0031320301002382`. Pattern Recognition in Information Systems.

[142] Eitan Richardson and Michael Werman. Efficient Classification Using the Euler Characteristic. *Pattern Recogn. Lett.*, 49(C):99–106, nov 2014. ISSN 0167-8655. doi: 10.1016/j.patrec.2014.07.001. URL `https://doi.org/10.1016/j.patrec.2014.07.001`.

[143] Teresa Heiss and Hubert Wagner. Streaming algorithm for Euler characteristic curves of multidimensional images. In Michael Felsberg, An-

ders Heyden, and Norbert Krüger, editors, *Computer Analysis of Images and Patterns - 17th International Conference, CAIP 2017, Ystad, Sweden, August 22-24, 2017, Proceedings, Part I*, volume 10424 of *Lecture Notes in Computer Science*, pages 397–409. Springer, 2017. doi: 10.1007/978-3-319-64689-3_32. URL `https://doi.org/10.1007/978-3-319-64689-3_32`.

[144] A. Verri, C. Uras, P. Frosini, and M. Ferri. On the use of size functions for shape analysis. In *[1993] Proceedings IEEE Workshop on Qualitative Vision*, pages 89–96, 1993. doi: 10.1109/WQV.1993.262946.

[145] Konstantin Mischaikow and Marian Mrozek. Chaos in the lorenz equations: A computer-assisted proof. *Bulletin of the American Mathematical Society*, 32(1):66–72, January 1995. ISSN 0273-0979. doi: 10.1090/S0273-0979-1995-00558-6.

[146] Tomasz. Kaczynski, Konstantin Michael. Mischaikow, and Marian. Mrozek. *Computational homology.* Applied mathematical sciences ; v. 157. Springer, New York, 2004. ISBN 0387408533.

[147] Fan Wang, Hubert Wagner, and Chao Chen. GPU Computation of the Euler Characteristic Curve for Imaging Data. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry (SoCG 2022)*, volume 224 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 64:1–64:16, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-227-3. doi: 10.4230/LIPIcs.SoCG.2022.64. URL `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.SoCG.2022.64`.

[148] Fan Wang, Chunhua Deng, Bo Yuan, and Chao Chen. Hardware acceleration of persistent homology computation. In *Large-Scale Annotation of Biomedical Data and Expert Label Synthesis and Hardware Aware Learning for Medical Imaging and Computer Assisted Intervention: International Workshops, LABELS 2019, HAL-MICCAI 2019, and CuRIOUS 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13 and 17, 2019, Proceedings*, page 81–88, Berlin, Heidelberg, 2019. Springer-Verlag. ISBN 978-3-030-33641-7. doi: 10.1007/978-3-030-33642-4_9. URL `https://doi.org/10.1007/978-3-030-33642-4_9`.

[149] Paul Bendich, Herbert Edelsbrunner, and Michael Kerber. Computing robustness and persistence for images. *IEEE Transactions on*

Visualization and Computer Graphics, 16(6):1251–1260, 2010. doi: 10.1109/TVCG.2010.139.

[150] Vanessa Robins, Peter John Wood, and Adrian P. Sheppard. Theory and algorithms for constructing discrete morse complexes from grayscale digital images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33 (8):1646–1658, aug 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.95. URL https://doi.org/10.1109/TPAMI.2011.95.

[151] Hubert Wagner, Chao Chen, and Erald Vuçini. *Efficient Computation of Persistent Homology for Cubical Data*, pages 91–106. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-23175-9. doi: 10.1007/978-3-642-23175-9_7. URL https://doi.org/10.1007/978-3-642-23175-9_7.

[152] The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015. URL http://gudhi.gforge.inria.fr/doc/latest/.

[153] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Distributed computation of persistent homology. In *Proceedings of the Meeting on Algorithm Engineering & Expermiments*, page 31–38, USA, 2014. Society for Industrial and Applied Mathematics.

[154] David Günther, Jan Reininghaus, Hubert Wagner, and Ingrid Hotz. Efficient computation of 3d morse-smale complexes and persistent homology using discrete morse theory. *The Visual Computer*, 28(10):959–969, 2012. URL http://dblp.uni-trier.de/db/journals/vc/vc28.html#GuntherRWH12.

[155] Konstantin Mischaikow and Vidit Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete Comput. Geom.*, 50(2):330–353, sep 2013. ISSN 0179-5376. doi: 10.1007/s00454-013-9529-6. URL https://doi.org/10.1007/s00454-013-9529-6.

[156] Hubert Wagner. Slice, Simplify and Stitch: Topology-Preserving Simplification Scheme for Massive Voxel Data. In Erin W. Chambers and Joachim Gudmundsson, editors, *39th International Symposium on Computational Geometry (SoCG 2023)*, volume 258 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 60:1–60:16, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-273-0. doi: 10.4230/LIPIcs.SoCG.2023.

60. URL `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.SoCG.2023.60`.

[157] Attila Gyulassy, Peer-Timo Bremer, Bernd Hamann, and Valerio Pascucci. A practical approach to morse-smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1619–1626, 2008. doi: 10.1109/TVCG.2008.110.

[158] Olaf Delgado-Friedrichs, Vanessa Robins, and Adrian Sheppard. Skeletonization and partitioning of digital images using discrete morse theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3): 654–666, 2015. doi: 10.1109/TPAMI.2014.2346172.

[159] Simon Zhang, Mengbai Xiao, and Hao Wang. GPU-Accelerated Computation of Vietoris-Rips Persistence Barcodes. In *36th International Symposium on Computational Geometry (SoCG 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[160] Varshini Subhash, Karran Pandey, and Vijay Natarajan. A gpu parallel algorithm for computing morse-smale complexes. *IEEE Transactions on Visualization and Computer Graphics*, 29(9):3873–3887, 2023. doi: 10.1109/TVCG.2022.3174769.

[161] Shuang Liu, Olivier Bousquet, and Kamalika Chaudhuri. Approximation and convergence properties of generative adversarial learning. In *Advances in Neural Information Processing Systems*, pages 5545–5553, 2017.

[162] Youssef Mroueh, Tom Sercu, and Vaibhava Goel. McGan: Mean and covariance feature matching GAN. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2527–2535, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[163] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[164] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016.

[165] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pages 2203–2213, 2017.

[166] Youssef Mroueh and Tom Sercu. Fisher gan. In *Advances in Neural Information Processing Systems*, pages 2513–2523, 2017.

[167] 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. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6626–6637. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7240-gans-trained-by-a-two-time-scale-update-rule-converge-to-a-local-nash-e pdf`.

[168] James R Munkres. *Elements of algebraic topology*. CRC Press, 2018.

[169] Ricardo Fabbri, Luciano Da F Costa, Julio C Torelli, and Odemir M Bruno. 2d euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys (CSUR)*, 40(1):1–44, 2008.

[170] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[171] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1): 103–120, 2007.

[172] David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have l p-stable persistence. *Foundations of computational mathematics*, 10(2):127–139, 2010.

[173] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

[174] G Peyré and M Cuturi. Computational optimal transport foundations and trends in machine learning 11 (2019) 355, 1803.

[175] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, March 2012.

[176] Ignacio Arganda-Carreras, Srinivas C. Turaga, Daniel R. Berger, Dan Cireşan, Alessandro Giusti, Luca M. Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M. Buhmann, Ting Liu, Mojtaba Seyedhosseini, Tolga Tasdizen, Lee Kamentsky, Radim Burget, Vaclav Uher, Xiao Tan, Changming Sun, Tuan D. Pham, Erhan Bas, Mustafa G. Uzunbas, Albert Cardona, Johannes Schindelin, and H. Sebastian Seung. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in Neuroanatomy*, 9:142, 2015. ISSN 1662-5129. doi: 10.3389/fnana.2015.00142. URL `https://www.frontiersin.org/article/10.3389/fnana.2015.00142`.

[177] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *German Conference on Pattern Recognition*, pages 364–374. Springer, 2013.

[178] Samaneh Abbasi-Sureshjani, Iris Smit-Ockeloen, Jiong Zhang, and Bart Ter Haar Romeny. Biologically-inspired supervised vasculature segmentation in slo retinal fundus images. In Mohamed Kamel and Aurélio Campilho, editors, *Image Analysis and Recognition*, pages 325–334, Cham, 2015. Springer International Publishing. ISBN 978-3-319-20801-5.

[179] S. Abbasi-Sureshjani, I. Smit-Ockeloen, E. Bekkers, B. Dashtbozorg, and B. t. H. Romeny. Automatic detection of vascular bifurcations and crossings in retinal images using orientation scores. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 189–192, April 2016. doi: 10.1109/ISBI.2016.7493241.

[180] Joes Staal, Michael D Abràmoff, Meindert Niemeijer, Max A Viergever, and Bram Van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging*, 23(4):501–509, 2004.

[181] A. D. Hoover, V. Kouznetsova, and M. Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical Imaging*, 19(3):203–210, March 2000. doi: 10.1109/42.845178.

[182] Christopher G. Owen, Alicja R. Rudnicka, Robert Mullen, Sarah A. Barman, Dorothy Monekosso, Peter H. Whincup, Jeffrey Ng, and Carl Paterson. Measuring Retinal Vessel Tortuosity in 10-Year-Old Children: Validation of the Computer-Assisted Image Analysis of the Retina (CA-IAR) Program. *Investigative Ophthalmology and Visual Science*, 50(5):

2004–2010, 05 2009. ISSN 1552-5783. doi: 10.1167/iovs.08-3018. URL
https://doi.org/10.1167/iovs.08-3018.

[183] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[184] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[185] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[186] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[187] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[188] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[189] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[190] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[191] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[192] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363, 2019.

[193] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069, 2016.

[194] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, pages 82–90, 2016.

[195] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

[196] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265, 2015.

[197] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

[198] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8162–8171, 2021.

[199] Yang Song and Stefano Ermon. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[200] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[201] Alexia Jolicoeur-Martineau, Simon Piché-Taillefer, Milad Khayatkhoei, and Ioannis Mitliagkas. Adversarial score matching and improved sampling for image generation. *arXiv preprint arXiv:2009.05475*, 2021.

[202] Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

[203] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2022.

[204] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Lopes, Simon Niklaus, Jonathan Ho, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*, 2022.

[205] Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. *arXiv preprint arXiv:2203.17003*, 2022.

[206] Shunxing Bao, Can Cui, Jia Li, Yucheng Tang, Ho Hin Lee, Ruining Deng, Lucas W. Remedios, Xin Yu, Qi Yang, Sophie Chiron, Nathan Heath Patterson, Ken S. Lau, Qi Liu, Joseph T. Roland, Lori A. Coburn, Keith T. Wilson, Bennett A. Landman, and Yuankai Huo. Topological-preserving membrane skeleton segmentation in multiplex immunofluorescence imaging. In John E. Tomaszewski and Aaron D. Ward, editors, *Medical Imaging 2023: Digital and Computational Pathology*, volume 12471, page 124710B. International Society for Optics and Photonics, SPIE, 2023. doi: 10.1117/12.2654087. URL https://doi.org/10.1117/12.2654087.

[207] Kasra Arnavaz, Oswin Krause, Jelena Miskovic Krivokapic, Silja Heilmann, Jakob Andreas Bærentzen, Pia Nyeng, and Aasa Feragen. Semi-supervised, topology-aware segmentation of tubular structures from live imaging 3d microscopy, May 2021.

[208] Seher Ozcelik, Sinan Unver, Ilke Ali Gurses, Rustu Turkay, and Cigdem Gunduz-Demir. Topology-aware loss for aorta and great vessel segmentation in computed tomography images, 2024.

[209] Shahira Abousamra, Rajarsi Gupta, Tahsin Kurc, Dimitris Samaras, Joel Saltz, and Chao Chen. Topology-guided multi-class cell context generation for digital pathology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3323–3333, June 2023.

[210] Ashirbani Saha, Michael Harowicz, and Maciej Mazurowski. Breast cancer mri radiomics: An overview of algorithmic features and impact of inter-reader variability in annotating tumors. *Medical Physics*, 45, 04 2018. doi: 10.1002/mp.12925.

[211] Joost JM Van Griethuysen, Andriy Fedorov, Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina GH Beets-Tan, Jean-

Christophe Fillion-Robin, Steve Pieper, and Hugo JWL Aerts. Computational radiomics system to decode the radiographic phenotype. *Cancer research*, 77(21):e104–e107, 2017.

[212] Jeremy R Burt, Neslisah Torosdagli, Naji Khosravan, Harish RaviPrakash, Aliasghar Mortazi, Fiona Tissavirasingham, Sarfaraz Hussein, and Ulas Bagci. Deep learning beyond cats and dogs: recent advances in diagnosing breast cancer with deep neural networks. *The British journal of radiology*, 91(1089):20170545, 2018.

[213] Sofia Jarkman, Micael Karlberg, Milda Pocevičiūtė, Anna Bodén, Péter Bándi, Geert Litjens, Claes Lundström, Darren Treanor, and Jeroen van der Laak. Generalization of deep learning in digital pathology: Experience in breast cancer metastasis detection. *Cancers*, 14(21):5424, 2022.

[214] Alexander Tack, Anirban Mukhopadhyay, and Stefan Zachow. Knee menisci segmentation using convolutional neural networks: data from the osteoarthritis initiative. *Osteoarthritis and cartilage*, 26(5):680–688, 2018.

[215] Ziyu Su, Muhammad Khalid Khan Niazi, Thomas E Tavolara, Shuo Niu, Gary H Tozbikian, Robert Wesolowski, and Metin N Gurcan. Bcr-net: A deep learning framework to predict breast cancer recurrence from histopathology images. *Plos one*, 18(4):e0283562, 2023.

[216] Jun Zhang, Ashirbani Saha, Zhe Zhu, and Maciej Mazurowski. Hierarchical convolutional neural networks for segmentation of breast tumors in mri with application to radiogenomics. *IEEE Transactions on Medical Imaging*, PP:1–1, 08 2018. doi: 10.1109/TMI.2018.2865671.

[217] Qikui Zhu, Bo Du, Baris Turkbey, Peter L Choyke, and Pingkun Yan. Deeply-supervised cnn for prostate segmentation. In *2017 international joint conference on neural networks (IJCNN)*, pages 178–184. IEEE, 2017.

[218] Michael Z Liu, Simukayi Mutasa, Peter Chang, Maham Siddique, Sachin Jambawalikar, and Richard Ha. A novel cnn algorithm for pathological complete response prediction using an i-spy trial breast mri database. *Magnetic resonance imaging*, 73:148–151, 2020.

[219] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127, 2019.

[220] Qing Lyu, Sanjeev Namjoshi, Emory McTyre, Umit Topaloglu, Richard Barcus, Michael Chan, Christina Cramer, Waldemar Debinski, Metin Gurcan, Glenn Lesser, Hui-Kuan Lin, Reginald Munden, Boris Pasche, Kiran K Solingapuram Sai, Roy Strowd, Stephen Tatter, Kounosuke Watabe, Wei Zhang, Ge Wang, and Christopher Whitlow. A transformer-based deep-learning approach for classifying brain metastases into primary organ sites using clinical whole-brain mri images. *Patterns*, 3:100613, 11 2022. doi: 10.1016/j.patter.2022.100613.

[221] H. Li, A. Menegaux, B. Schmitz-Koep, A. Neubauer, F.J.B. Bäuerlein, S. Shit, C. Sorg, B. Menze, and D. Hedderich. Automated claustrum segmentation in human brain mri using deep learning. *Human Brain Mapping*, 42(18):5862–5872, 2021. ISSN 1097-0193. doi: 10.1002/hbm. 25655. URL https://eurekamag.com/research/079/334/079334358. php.

[222] Maciej Mazurowski, Mateusz Buda, Ashirbani Saha, and Mustafa Bashir. Deep learning in radiology: an overview of the concepts and a survey of the state of the art. *Journal of Magnetic Resonance Imaging*, 49, 02 2018. doi: 10.1002/jmri.26534.

[223] Ali Nawaz, Syed Muhammad Anwar, Rehan Liaqat, Javid Iqbal, Ulas Bagci, and Muhammad Majid. Deep convolutional neural network based classification of alzheimer's disease using mri data. In *2020 IEEE 23rd International Multitopic Conference (INMIC)*, pages 1–6. IEEE, 2020.

[224] Mehmet Ufuk Dalmış, Geert Litjens, Katharina Holland, Arnaud Setio, Ritse Mann, Nico Karssemeijer, and Albert Gubern-Mérida. Using deep learning to segment breast and fibroglandular tissue in mri volumes. *Medical physics*, 44(2):533–546, 2017.

[225] Amin Ranem, John Kalkhof, Caner Özer, Anirban Mukhopadhyay, and Ilkay Oksuz. Detecting respiratory motion artefacts for cardiovascular mris to ensure high-quality segmentation. In *International Workshop on Statistical Atlases and Computational Models of the Heart*, pages 447–456. Springer, 2022.

[226] D Newitt and N Hylton. Multi-center breast dce-mri data and segmentations from patients in the i-spy 1/acrin 6657 trials. *Cancer Imaging Arch. doi*, 2016.

[227] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 263–271, 2005.

[228] Aldo Badano, Christian G. Graff, Andreu Badal, Diksha Sharma, Rong-ping Zeng, Frank W. Samuelson, Stephen J. Glick, and Kyle J. My-ers. Evaluation of Digital Breast Tomosynthesis as Replacement of Full-Field Digital Mammography Using an In Silico Imaging Trial. *JAMA Network Open*, 1(7):e185474–e185474, 11 2018. ISSN 2574-3805. doi: 10.1001/jamanetworkopen.2018.5474. URL https://doi.org/10.1001/jamanetworkopen.2018.5474.

[229] Xudong Zhang, Pengxiang Wu, Changhe Yuan, Yusu Wang, Dimitris N Metaxas, and Chao Chen. Heuristic search for homology localization problem and its application in cardiac trabeculae reconstruction. In *IJCAI*, pages 1312–1318, 2019.

[230] Nikola Milosavljević, Dmitriy Morozov, and Primoz Skraba. Zigzag per-sistent homology in matrix multiplication time. In *SoCG*, pages 216–225, 2011.

[231] Joost JM Van Griethuysen, Andriy Fedorov, Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina GH Beets-Tan, Jean-Christophe Fillion-Robin, Steve Pieper, and Hugo JWL Aerts. Compu-tational radiomics system to decode the radiographic phenotype. *Cancer research*, 2017.

[232] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *J Am Stat Assoc*, 1951.

[233] Shiyi Du, Qicheng Lao, Qingbo Kang, Yiyue Li, Zekun Jiang, Yan-feng Zhao, and Kang Li. Distilling knowledge from topological rep-resentations for pathological complete response prediction. In *Medi-cal Image Computing and Computer Assisted Intervention – MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part II*, page 56–65, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-16433-0. doi: 10.1007/978-3-031-16434-7_6. URL https://doi.org/10.1007/978-3-031-16434-7_6.

[234] Yaopeng Peng, Hongxiao Wang, Milan Sonka, and Danny Z. Chen. Phg-net: Persistent homology guided medical image classification. In *Proceed-ings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7583–7592, January 2024.

[235] Richard M. Levenson, Yashbir Singh, Bastian Rieck, Quincy A. Hath-away, Colleen Farrelly, Jennifer Rozenblit, Prateek Prasanna, Bradley

Erickson, Ashok Choudhary, Gunnar Carlsson, and Deepa Sarkar. Advancing precision medicine: Algebraic topology and differential geometry in radiology and computational pathology. *Laboratory Investigation*, 104(6):102060, 2024. ISSN 0023-6837. doi: https://doi.org/10.1016/j.labinv.2024.102060. URL `https://www.sciencedirect.com/science/article/pii/S0023683724017380`.

[236] Muhammad Shahid Iqbal, Waqas Ahmad, Roohallah Alizadehsani, Sadiq Hussain, and Rizwan Rehman. Breast cancer dataset, classification and detection using deep learning. *Healthcare*, 10(12), 2022. ISSN 2227-9032. URL `https://www.mdpi.com/2227-9032/10/12/2395`.

[237] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[238] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*, SCG '06, pages 119–126, New York, NY, USA, 2006. ACM. ISBN 1-59593-340-9. doi: 10.1145/1137856.1137877. URL `http://doi.acm.org/10.1145/1137856.1137877`.

[239] Jacob E. Goodman and Joseph O'Rourke, editors. *Handbook of Discrete and Computational Geometry*. CRC Press, Inc., Boca Raton, FL, USA, 1997. ISBN 0-8493-8524-5.

[240] Jose A. Perea and John Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15:799–838, 2015.

[241] R. Ghrist and A. Muhammad. Coverage and hole-detection in sensor networks via homology. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 254–260, April 2005. doi: 10.1109/IPSN.2005.1440933.

[242] Vin De Silva and Robert Ghrist. Blind swarms for coverage in 2-d. In *In Proceedings of Robotics: Science and Systems*, page 01, 2005.

[243] C. Li, M. Ovsjanikov, and F. Chazal. Persistence-based structural recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2003–2010, June 2014. doi: 10.1109/CVPR.2014.257.

[244] Tamal K. Dey, Kuiyu Li, Jian Sun, and David Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3d models. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, pages 45:1–45:9, New York, NY, USA, 2008. ACM. ISBN 978-1-4503-0112-1. doi: 10.1145/1399504. 1360644. URL http://doi.acm.org/10.1145/1399504.1360644.

[245] Herbert Edelsbrunner. Surface Tiling with Differential Topology. In Mathieu Desbrun and Helmut Pottmann, editors, *Eurographics Symposium on Geometry Processing 2005*. The Eurographics Association, 2005. ISBN 3-905673-24-X. doi: 10.2312/SGP/SGP05/009-011.

[246] Moo K. Chung, Peter Bubenik, and Peter T. Kim. Persistence diagrams of cortical surface data. In Jerry L. Prince, Dzung L. Pham, and Kyle J. Myers, editors, *Information Processing in Medical Imaging*, pages 386–397, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-02498-6.

[247] Moo K. Chung, Jamie L. Hanson, Jieping Ye, Richard J. Davidson, and Seth D. Pollak. Persistent homology in sparse regression and its application to brain morphometry. *IEEE Transactions on Medical Imaging*, 34 (9):1928–1939, 9 2015. ISSN 0278-0062. doi: 10.1109/TMI.2015.2416271.

[248] Hyekyoung Lee, Hyejin Kang, Moo K. Chung, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE Transactions on Medical Imaging*, 31:2267–2277, 2012.

[249] John Suckling. The mammographic image analysis society digital mammogram database"exerpta medica. *Exerpta Medica International Congress Series*, 1069, 01 1994.

[250] James R. Munkres. *Elements of Algebraic Topology*. Addison Wesley Publishing Company, 1984. ISBN 0201045869. URL http://www.worldcat.org/isbn/0201045869.

[251] Steve Oudot. Persistence theory - from quiver representations to data analysis. In *Mathematical surveys and monographs*, 2015.

[252] Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[253] Lukman Hakim, Muthu Subash Kavitha, Novanto Yudistira, and Takio Kurita. Regularizer based on Euler characteristic for retinal blood vessel segmentation. *Pattern Recognition Letters*, 149:83–90,

2021. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2021.05. 023. URL `https://www.sciencedirect.com/science/article/pii/ S0167865521002075`.

[254] Talha Qaiser, Yee-Wah Tsang, Daiki Taniyama, Naoya Sakamoto, Kazuaki Nakane, David Epstein, and Nasir Rajpoot. Fast and accurate tumor segmentation of histology images using persistent homology and deep convolutional features. *Medical Image Analysis*, 55:1–14, 2019. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2019.03. 014. URL `https://www.sciencedirect.com/science/article/pii/ S1361841518302688`.

[255] Xiaqing Li, Guangyan Zhang, H Howie Huang, Zhufan Wang, and Weimin Zheng. Performance analysis of GPU-based convolutional neural networks. In *2016 45th International conference on parallel processing (ICPP)*, pages 67–76. IEEE, 2016.

[256] David S. Richeson. *Euler's Gem: The Polyhedron Formula and the Birth of Topology*. Princeton University Press, Princeton, 2008. ISBN 9781400838561. doi: doi:10.1515/9781400838561. URL `https://doi. org/10.1515/9781400838561`.

[257] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 454–463. IEEE, 2000.

[258] Hubert Wagner, Chao Chen, and Erald Vuçini. Efficient computation of persistent homology for cubical data. In *Topological methods in data analysis and visualization II*, pages 91–106. Springer, 2012.

[259] Shubhabrata Sengupta, Mark Harris, Michael Garland, et al. Efficient parallel scan algorithms for GPUs. *NVIDIA, Santa Clara, CA, Tech. Rep. NVR-2008-003*, 1(1):1–17, 2008.

[260] Jiaxiang Ren, Shengjie Zhao, Kai Yang, and Brian Nlong Zhao. A novel and robust face clustering method via adaptive difference dictionary. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 627–632, 2017. doi: 10.1109/ICMEW.2017.8026269.

[261] Suyoung Choi, Hyeontae Jang, and Mathieu Vallée. GPU Algorithm for Enumerating PL Spheres of Picard Number 4: Application to Toric Topology. In Wolfgang Mulzer and Jeff M. Phillips, editors, *40th International Symposium on Computational Geometry (SoCG 2024)*, volume 293 of *Leibniz International Proceedings in Informatics (LIPIcs)*,

pages 41:1–41:15, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-316-4.

[262] Nicholas O. Malott and Philip A. Wilsey. Scalable homology classification through decomposed euler characteristic curves. In *2023 IEEE International Conference on Big Data (BigData)*, pages 768–777, 2023. doi: 10.1109/BigData59044.2023.10386966.

[263] Daniel J. Laky and Victor M. Zavala. A fast and scalable computational topology framework for the euler characteristic. *ArXiv*, abs/2311.11740, 2023. URL `https://api.semanticscholar.org/CorpusID:265295572`.

[264] Paweł Dłotko and Davide Gurnari. Euler characteristic curves and profiles: a stable shape invariant for big data problems. *GigaScience*, 12: giad094, 11 2023. ISSN 2047-217X. doi: 10.1093/gigascience/giad094. URL `https://doi.org/10.1093/gigascience/giad094`.

[265] Elizabeth Munch. An invitation to the euler characteristic transform, 2023.

[266] Wen-Yan Lin, Fan Wang, Ming-Ming Cheng, Sai-Kit Yeung, Philip H.S. Torr, Minh N. Do, and Jiangbo Lu. Code: Coherence based decision boundaries for feature correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):34–47, 2018. doi: 10.1109/TPAMI.2017.2652468.

[267] Fan Wang, Aditi Nayak, Yogesh Agrawal, and Roy Shilkrot. Hierarchical image link selection scheme for duplicate structure disambiguation. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 221. BMVA Press, 2018. URL `http://bmvc2018.org/contents/papers/0718.pdf`.