

Escuela de Negocios. Laboratorio de Neurociencia

Tipo de documento: Preprint

VesselGPT: Autoregressive Modeling of Vascular Geometry

Autorías ditellianas: Siless, Viviana; Iarussi, Emmanuel; Sinnona, Martín

Otras autorías: Feldman, Paula; Delrieux, Claudio

Fecha de publicación: 19/05/2025

Publicado originalmente en: Arxiv

¿Cómo citar este documento?

Feldman, P., et al. (2025). VesselGPT: Autoregressive Modeling of Vascular Geometry. Arxiv.

<https://doi.org/10.48550/arXiv.2505.13318>

El presente documento se encuentra alojado en el Repositorio Digital Universidad Torcuato Di Tella bajo una licencia Creative Commons Atribución 4.0, de acuerdo a lo indicado en la fuente original del documento

Dirección: <https://repositorio.utdt.edu/handle/20.500.13098/13402>

VesselGPT: Autoregressive Modeling of Vascular Geometry

Paula Feldman^{1,3}, Martin Sinnona³, Claudio Delrieux^{1,2}, Viviana Siless³, and Emmanuel Iarussi^{1,3}

¹ Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina

² Universidad Nacional del Sur, Bahía Blanca, Argentina

³ Universidad Torcuato Di Tella, Buenos Aires, Argentina
paulafeldman@conicet.gov.ar

Abstract. Anatomical trees are critical for clinical diagnosis and treatment planning, yet their complex and diverse geometry make accurate representation a significant challenge. Motivated by the latest advances in large language models, we introduce an autoregressive method for synthesizing anatomical trees. Our approach first embeds vessel structures into a learned discrete vocabulary using a VQ-VAE architecture, then models their generation autoregressively with a GPT-2 model. This method effectively captures intricate geometries and branching patterns, enabling realistic vascular tree synthesis. Comprehensive qualitative and quantitative evaluations reveal that our technique achieves high-fidelity tree reconstruction with compact discrete representations. Moreover, our B-spline representation of vessel cross-sections preserves critical morphological details that are often overlooked in previous parameterizations. To the best of our knowledge, this work is the first to generate blood vessels in an autoregressive manner. Code, data, and trained models will be made available.

Keywords: Vascular 3D model · Generative modeling · Neural Networks.

1 Introduction

Realistic 3D models of blood vessels are crucial for a number of medical applications ranging from diagnosis to surgical intervention. Accurate representations of hierarchical systems like blood vessels, renal tubules, and airways enable critical applications, including disease diagnosis [24], prognosis [20], and intervention simulation [16]. Additionally, they play a key role in surgical planning [15] and fluid dynamics simulations [30]. Beyond the medical domain, the ability to model complex branching structures extends to fields such as computer graphics, where applications include procedural generation of vegetation [17,7].

The choice of data representation for 3D vascular structures depends on the downstream task. However, obtaining high-fidelity reconstructions from patient scans is challenging, requiring expert knowledge and often leading to errors [13,19]. Despite advances in vessel segmentation, accurately capturing fine

vascular details remains difficult [1]. To address these limitations, various methods have been developed for synthesizing blood vessel geometries [34]. Furthermore, generative models provide a promising avenue for augmenting datasets, particularly for rare anatomical and pathological variations [4]. We identified two main approaches in the literature on generating vascular 3D models: model-based and data-driven. Model-based methods are fractal or space-filling algorithms using a set of fixed rules that include different branching parameters, generally adhering to hemodynamic laws with constraints related to flow and radius [11,29,25,23]. These methods often fail to capture the complexity and diversity of real anatomical data. Instead, data-driven methods model the data distribution without the need for hard-coded rules. The first method in this line was proposed by Wolterink et al. [33], who employed Generative Adversarial Networks but restricted the model to single-channel vessels. Subsequently, Sinha et al. [27] introduced a diffusion model to learn vessel tree distributions using signed distance functions (SDFs). Deo et al. [8] extended these ideas by leveraging latent diffusion models for aneurysm segmentation generation. Kuipers et al. [12] adopted a diffusion-based approach to generate vessel point clouds but required a post-processing step to reconstruct the centerlines. Prabhakar et al. [21] proposed a novel graph-based diffusion model capable of handling cycles, though it primarily targets capillary networks and focuses on generating the vessel topology rather than a 3D mesh.

VesselVAE [9], which is conceptually closer to our approach, introduced a recursive variational neural network (RvNN) tailored for vascular geometry generation. However, like other parameterized methods, it represents vessels solely by their centerlines and radii. While this compact representation is widely used, it oversimplifies the vessel’s cross-sectional geometry, potentially leading to inaccuracies in simulations based on such data [10]. Moreover, the method is limited to relatively small vascular segments, a constraint that arises from the difficulty the RvNN architecture faces when scaling to deeper levels of recursion.

In light of these limitations, we propose a novel Transformer-based [32] approach for blood vessel synthesis. Although transformers have shown remarkable versatility in modeling sequential data, including geometric structures [26,28], their application to vascular generation remains largely unexplored. In this work we present a formulation that treats vascular trees as a sequence of nodes, leveraging the GPT-2 architecture to generate compact, anatomically realistic vessel structures. Additionally, we propose using a cross-sectional representation of the vessels’ channels based on B-splines. This approach allows a more nuanced surface model compared to methods that rely solely on a single radius value, overcoming a key limitation in previous techniques. Our approach proceeds in two stages. First, we use a vector-quantized autoencoder (VQ-VAE) [31] to learn a discrete vocabulary of node attributes, capturing essential geometric features in context. We then employ this learned codebook to train a GPT-2 model [22], enabling the autoregressive generation of 3D blood vessel structures. Finally, we show extensive experiments comparing our method with baseline techniques,

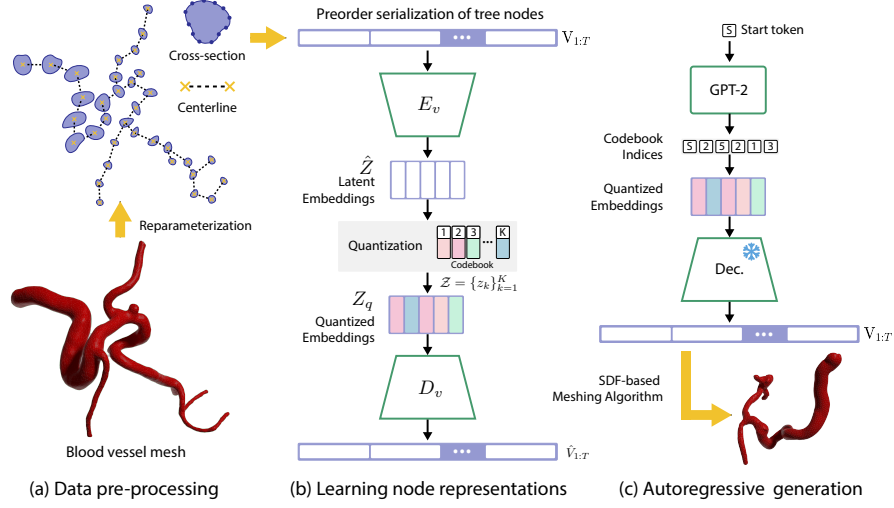


Fig. 1. VesselGPT overview. (a) We begin by reparameterizing the blood vessel meshes by computing their centerlines and fitting B-splines to the cross-sections. We store all parameters in a binary tree, where each node represents a centerline sample along with the corresponding weights of its cross-section spline. (b) We flatten the tree using pre-order traversal, then feed the resulting sequence into a Vector Quantized Variational Autoencoder (VQ-VAE). The encoder maps it to a discrete latent space, and the decoder learns to reconstruct the original sequence from the quantized embeddings. (c) We train a GPT model on the codebook index sequences (tokens) that the pretrained VQ-VAE encoder generates for each dataset sample. During inference, the autoregressive model generates token sequences, which are decoded back to the serialized tree data structured by means of the VQ-VAE decoder. Finally, a meshing algorithm reconstructs the vessel geometry from its parameterized representation.

demonstrating superior performance and validating the effectiveness of our approach.

2 Methods

We propose a sequence-based approach for autoregressively generating blood vessel trees as sequences of nodes (Figure 1). The proposed method consists of two main stages:

1. A Vector-Quantized Variational Autoencoder (VQ-VAE) trained to learn a discrete vocabulary of tokenized node embeddings.
2. A transformer model trained for vessel tree generation by performing autoregressive next-node prediction over the learned vocabulary of tokens.

2.1 Codebook learning

Autoregressive generative models produce token sequences by conditioning each new token on those previously produced. In order to tokenize blood vessel structures, we first convert them into a sequential representation. Specifically, we model each vessel as a binary tree and serialize it using a preorder traversal, resulting in an ordered vector of node attributes—including coordinates, spline coefficients, and control points. Moreover, we define a special *zero attribute* vector for absent child nodes, ensuring that the original tree structure can be accurately reconstructed from this serialized representation during decoding.

Using the coordinates and spline coefficients directly as tokens for autoregressive generation is not optimal, mainly because it fails to effectively capture geometric patterns. This is because such a representation lacks information about neighboring vessel nodes and does not incorporate any priors from the distribution of vascular structures. To address this challenge, we propose to learn a quantized embedding from a collection of vessel trees. The task formulation is specified as follows: Let $V_{1:T} = (v_1, \dots, v_T)$ represent a sequence of ground-truth blood vessel nodes, where each node $v_t \in \mathbb{R}^m$ is associated with an m -dimensional attribute vector that includes its spatial coordinates x, y, z along with the coefficients and control points defining the spline. The goal of this task is to sequentially predict the synthesized blood vessel nodes $\hat{V}_{1:T}$, approximating the original sequence $V_{1:T}$.

To construct the node vocabulary, we employ a vector-quantized autoencoder (VQ-VAE) [31], which consists of an encoder (E_v), a decoder (D_v), and a discrete codebook $\mathcal{Z} = \{z_k\}_{k=1}^K$, where $z_k \in \mathbb{R}^{d_z}$, and d_z is the dimension of the codebook vectors. The VQ-VAE is trained to self-reconstruct realistic vessel trees. Specifically, given vessel tree data in the continuous domain $V_{1:T}$, the VQ-VAE encoder (E_v), which incorporates a transformer layer [32], first encodes the continuous sequence of vessel nodes into latent features $\hat{Z} \in \mathbb{R}^{T \times d_z}$. These latent features $\hat{Z}$ are then quantized to Z_q via an element-wise quantization function Q_v , which maps each feature vector to its closest entry in the codebook:

$$Z_q = Q_v(\hat{Z}) := \arg \min_{z_k \in \mathcal{Z}} \|\hat{z}_t - z_k\|_2 \quad \forall t \in [1, T]. \quad (1)$$

The quantized representation Z_q is subsequently reconstructed into continuous vessel tree representations $\hat{V}_{1:T}$ by the VQ-VAE decoder (D_v), which has a symmetric architecture to the encoder. The VQ-VAE model is trained using the following objective function:

$$\mathcal{L}_{\text{VQ}} = \|V_{1:T} - \hat{V}_{1:T}\|_1 + \|\text{sg}(\hat{Z}) - Z_q\|_2^2 + \lambda \|\hat{Z} - \text{sg}(Z_q)\|_2^2, \quad (2)$$

where the first term enforces reconstruction consistency, the second and third terms update the codebook via the stop-gradient operator sg [3], and λ is a weighting factor.

2.2 Autoregressive vessel sequence prediction

The second component in our model is a GPT-style decoder-only transformer that takes advantage of quantized embedding from the previous stage. Given a sequence of tokens computed from the nodes of a vessel tree, the transformer is trained to predict the codebook index of the next embedding in the sequence. Once trained, the transformer can be auto-regressively sampled to predict sequences of embeddings. These tokens can then be decoded into the tree geometry of novel vessel trees.

The input of the transformer consists of the codebook indices of the embeddings $e(Z_q)$, obtained by encoding the input sequence using the frozen encoder of stage 1. The list of codebook indices is prefixed and suffixed with predefined start and end tokens. The features then pass through a stack of multi-headed self-attention layers, where the transformer is trained to predict the next quantized codebook entry in the sequence. Specifically, we maximize the *log* probability of the training sequences with respect to the transformer parameters θ :

$$\prod_{t=1}^T p(e^t \mid e^{<t}; \theta), \quad (3)$$

where e^t represents the codebook index corresponding to the vessel node at step t , and $e^{<t}$ denotes the sequence of tokens up to step $t - 1$.

Once trained, the transformer can autoregressively generate a sequence of tokens; starting with a learned start token and continuing until a stop token is reached using beam sampling. The codebook embeddings indexed by the generated token sequence are then decoded by D_v to reconstruct the structure of the vessel.

2.3 3D mesh synthesis.

To reconstruct a 3D mesh representation of a blood vessel from a set of splines, we employ a three-step approach that involves centerline fitting, signed distance field (SDF) generation, and surface extraction through marching cubes. First, we fit a B-spline to each branch of the centerline and obtain a smooth parametric representation. Next, we construct a signed distance field (SDF) that encapsulates the vessel’s shape. The cross-sectional geometry of the vessel is characterized by a set of B-splines defining the radius at different points along the centerline. The SDF is then built by interpolating these cross-sectional shapes along the vessel’s trajectory, yielding a continuous representation of the vessel. Finally, we apply the marching cubes algorithm [18] to the generated SDF and obtain a triangulated surface mesh.

3 Experimental Setup

Materials. Our model was trained on the publicly available Aneurisk dataset⁴ [2]. This dataset comprises 100 vessel segments reconstructed from 3D angiographic images, representing both healthy vasculature and aneurysms. We transformed the 3D meshes into binary tree representations, and obtained the vessel centerline with the *network extraction* script from the VMTK toolkit⁵. The centerline points are obtained by calculating the ratio of the sphere step to the local maximum radius, using a user-defined advancement ratio. For each sample along the centerline, the contour of the blood vessel conduit was delineated by fitting a B-spline to the surface points located on the plane perpendicular to the centerline. Data augmentation was performed by resampling the trees at various rates and applying random rotations to the entire tree. Additionally, non-binary trees were converted to binary form, and any trees containing loops were excluded. This procedure ultimately yielded 528 binary trees from the original 3D meshes.

To serialize a binary tree, we traverse it in preorder and store each node’s attributes sequentially. The tree can be incomplete, meaning that some nodes have fewer than two children or none. To preserve the structure during deserialization, we insert zero vectors in place of these non-existing children, ensuring that the hierarchy can be accurately reconstructed. However, since the VQ-VAE learns an optimal representation for each tree based on the entire dataset, these zero vectors are often mapped to different token sequences. As a result, after reconstructing the unquantized tree from the decoder, threshold values lower than 10^{-2} to zero to ensure non-existing nodes are accurately predicted.

Implementation details. For extracting the centerline, we configured the VMTK script with an advancement ratio of 1.05. The script generates several cross-sectional views at bifurcation points; in these instances, we opted for the cross-section with the smallest area to ensure it aligns correctly with the principal direction of the centerline. All attributes were normalized to the range $[-1, 1]$, and the root node of each tree was positioned at the origin.

Training. For Stage 1, we defined the codebook vector dimension as 64 and utilized 16 tokens per tree node. The batch size was set to 1, and we applied the ADAM optimizer with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning rate of 1×10^{-4} . Training the model on a single NVIDIA RTX A4500 requires approximately 3 hours. For Stage 2, we maintained the same hyperparameters, learning rate, and optimizer. Training time for this stage is approximately 24 hours.

Metrics. We benchmarked our method with state-of-the-art approaches using established metrics for assessment, consistent with previous works [35,26,27]: We use Minimum Matching Distance (MMD) to evaluate quality, Coverage (COV) to measure diversity, and 1-Nearest Neighbor Accuracy (1-NNA) to assess plausibility. To compute these point-based metrics, we randomly sample 1000 points from all baseline generated meshes.

⁴ <https://github.com/permfl/AneuriskData>

⁵ <http://www.vmtk.org/vmtkscripts/vmtknetworkextraction>

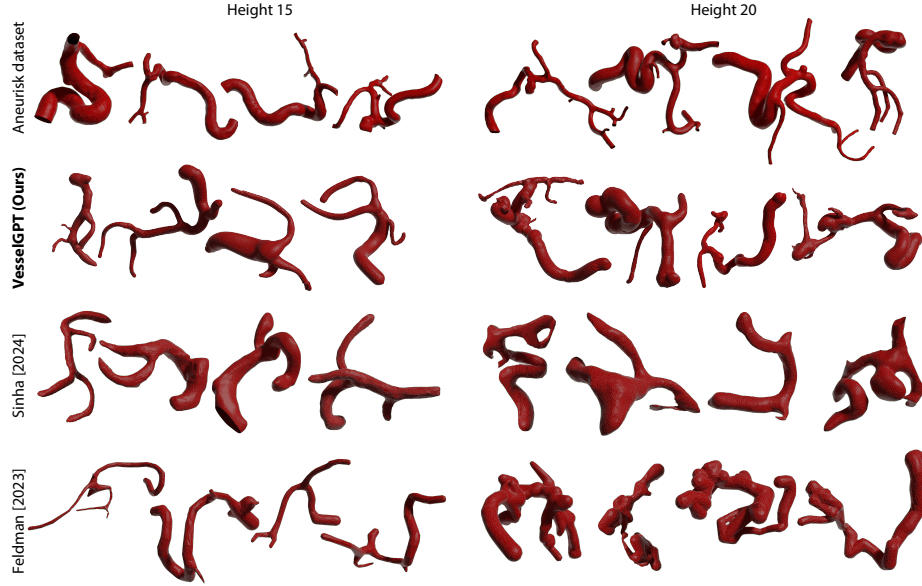


Fig. 2. Comparison of vessel geometries generated by VesselGPT (Ours) and baseline methods trained on the Aneurisk dataset. Sinha et al. [27] produce valid vessels but lack diversity, resulting in simpler, shallower structures. Feldman et al. [9] performs well up to height 15 but struggles with deeper trees. In contrast, our method generates more diverse and realistic vessel meshes.

To further assess the quality of our generated samples, we compared their distribution to that of the training dataset using vessel-specific metrics. These metrics are well-established in 3D blood vessel modeling and have shown to effectively capture key vascular properties [5,14,9,27]. Specifically, we evaluated branch-wise tortuosity and total vessel centerline length. Tortuosity, measured using the distance metric [6], is particularly relevant due to its clinical significance, as it quantifies the twisting complexity of individual vessel branches. Total vessel length, on the other hand, has been widely used to distinguish between healthy and pathological vasculature. Finally, to assess how closely the generated vessel structures align with real ones, we quantify the similarity between their distributions using cosine similarity.

4 Results

We compare our method to leading vasculature generation approaches. Sinha et al. [27] uses diffusion over the space of INRs, focusing primarily on representation rather than the generation of novel structures. Feldman et al. [9] employs a recursive approach to generate vessel sequences but faces limitations with deeper

trees and simplifies radius parametrization to a single value per centerline point, losing fine details. The results are presented in Tab. 1. Both our model and the baselines were trained with two different datasets: the Aneurisk trees trimmed at height 15 and 20. For the MMD metric, while a low value indicates resemblance, an extremely small value may suggest memorization. Our model outperforms the method by [27] in both COV and 1-NNA. Although VesselVAE achieves a higher COV score, qualitative evaluation highlights its difficulties with deeper trees and its tendency to oversimplify vessel radii using circular cross-sections, preventing it from capturing fine details. While these metrics are commonly used in the literature, they fail to capture key characteristics of the data. Furthermore, Fig. 2 provides example renders for all methods and training datasets, complementing the quantitative analysis.

Table 1. Quantitative results for novel tree generation comparing our method against baseline approaches. $\uparrow$ indicates that higher values are better, $\downarrow$ indicates that lower values are better, and for 1-NNA, the optimal score is 0.5.

	Aneurisk Height 15			Aneurisk Height 20		
	MMD $\downarrow$	COV $\uparrow$	1-NNA	MMD $\downarrow$	COV $\uparrow$	1-NNA
Sinha et al.[27]	0.43	0.103	0.	0.418	0.104	0.
Feldman et al.[9]	0.014	0.49	0.238	.013	0.45	0.126
Ours	0.14	0.31	0.097	0.14	0.41	0.17

For the metric characterization analysis, we utilized state-of-the-art vascular metrics to assess the generated blood vessels. We computed histograms of total length, and tortuosity for both the real and generated vessel sets. To quantify the similarity between these distributions, we used cosine similarity. Given that all values are positive, cosine similarity ranges from 0 to 1. Our analysis yielded a total length similarity of 0.88 and a tortuosity similarity of 0.97 for trees of height 20, while for trees of height 15, these values were 0.87 and 0.96, respectively. These results indicate a strong agreement between the real and generated distributions, demonstrating the realism of our synthesized vessels. The slightly lower length similarity is due to some model-generated samples containing only a few tokens, this only occurs in less than 5% of the generated samples.

5 Conclusions

We introduce VesselGPT, a novel method for generating blood vessels in an autoregressive manner by sequentially sampling from a learned codebook. Our method is capable of learning a discrete optimal representation of blood vessels in an embedding space, that allows synthesizing high quality novel structures. An interesting future direction would be to evaluate the generated structures in a downstream task, such as blood flow simulations. Assessing how well the synthesized vasculature performs in practical applications would provide deeper insights

into its clinical and scientific relevance. Furthermore, integrating domain-specific constraints or leveraging physics-informed models could enhance the utility and realism of the generated structures in real-world scenarios. Our approach demonstrates promising advancements in 3D blood vessel geometry synthesis, paving the way for enhanced clinical applications that could assist healthcare professionals in diagnosis, treatment planning, and surgical interventions.

Acknowledgements This project was supported by grants from Salesforce, USA (Einstein AI 2020), National Scientific and Technical Research Council (CONICET), Argentina (PIP 2021-2023 GI - 11220200102981CO), and Universidad Torcuato Di Tella, Argentina.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alblas, D., Brune, C., Wolterink, J.M.: Deep-learning-based carotid artery vessel wall segmentation in black-blood mri using anatomical priors. In: Medical Imaging 2022: Image Processing. vol. 12032, pp. 237–244. SPIE (2022)
2. Aneurisk-Team: Aneuriskweb project. Web Site (2012)
3. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
4. van Breugel, B., Liu, T., Oglic, D., van der Schaar, M.: Synthetic data in biomedicine via generative artificial intelligence. *Nature Reviews Bioengineering* pp. 1–14 (2024)
5. Bullitt, E., Gerig, G., Aylward, S., Joshi, S., Smith, K., Ewend, M., Lin, W.: Vascular attributes and malignant brain tumors. In: Proc. Med. Image Comput. Comput.-Assist. Interv. (MICCAI). pp. 671–679. Springer (2003)
6. Bullitt, E., Gerig, G., Pizer, S.M., Lin, W., Aylward, S.R.: Measuring tortuosity of the intracerebral vasculature from mra images. *IEEE transactions on medical imaging* **22**(9), 1163–1171 (2003)
7. Cuntz, H., Forstner, F., Borst, A., Häusser, M.: One rule to grow them all: a general theory of neuronal branching and its practical application. *PLoS computational biology* **6**(8), e1000877 (2010)
8. Deo, Y., Lin, F., Dou, H., Cheng, N., Ravikumar, N., Frangi, A., Lassila, T.: Few-shot learning in diffusion models for generating cerebral aneurysm geometries. In: *Lecture Notes in Computer Science*. Springer Nature (2024)
9. Feldman, P., Fainstein, M., Siless, V., Delrieux, C., Iarussi, E.: Vesselvae: Recursive variational autoencoders for 3d blood vessel synthesis. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. pp. 67–76. Springer (2023)
10. Ferrero-Aprato, C.: Impact of circular cross-section idealization in image-based computational hemodynamic models of coronary arteries. Ph.D. thesis, Politecnico di Torino (2019)

11. Hamarneh, G., Jassi, P.: Vascusynth: Simulating vascular trees for generating volumetric image data with ground-truth segmentation and tree analysis. *Computerized medical imaging and graphics* **34**(8), 605–616 (2010)
12. Kuipers, T.P., Konduri, P.R., Marquering, H., Bekkers, E.J.: Generating cerebral vessel trees of acute ischemic stroke patients using conditional set-diffusion. In: *Medical Imaging with Deep Learning* (2024)
13. Lan, H., Updegrove, A., Wilson, N.M., Maher, G.D., Shadden, S.C., Marsden, A.L.: A re-engineered software interface and workflow for the open-source simvascular cardiovascular modeling package. *J. Biomech. Eng.* **140**(2), 024501 (2018)
14. Lang, S., Müller, B., Dominietto, M.D., Cattin, P.C., Zanette, I., Weitkamp, T., Hieber, S.E.: Three-dimensional quantification of capillary networks in healthy and cancerous tissues of two mice. *Microvascular research* **84**(3), 314–322 (2012)
15. Lawaetz, J., Kristensen, J.S.S., Nayahangan, L.J., Van Herzeele, I., Konge, L., Eiberg, J.P.: Simulation based training and assessment in open vascular surgery: a systematic review. *Eur. J. Vasc. Endovasc. Surg.* **61**(3), 502–509 (2021)
16. Le Bras, A., Boustia, F., Janot, K., Le Pabic, E., Ouvrard, M., Fougereou-Leurent, C., Ferre, J.C., Gauvrit, J.Y., Eugene, F.: Rehearsals using patient-specific 3d-printed aneurysm models for simulation of endovascular embolization of complex intracranial aneurysms: 3d sim study. *Journal of Neuroradiology* (2023)
17. Lee, J.J., Li, B., Benes, B.: Latent l-systems: Transformer-based tree generator. *ACM Transactions on Graphics* **43**(1), 1–16 (2023)
18. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353 (1998)
19. Mou, L., Lin, J., Zhao, Y., Liu, Y., Ma, S., Zhang, J., Lv, W., Zhou, T., Frangi, A.F., Zhao, Y.: Costa: A multi-center multi-vendor tof-mra dataset and a novel cerebrovascular segmentation network. *IEEE Trans. Med. Imaging* (2024)
20. Murthy, V.L., Naya, M., Foster, C.R., Hainer, J., Gaber, M., Dorbala, S., Charytan, D.M., Blankstein, R., Di Carli, M.F.: Coronary vascular dysfunction and prognosis in patients with chronic kidney disease. *JACC Cardiovasc. Imaging* **5**(10) (2012)
21. Prabhakar, C., Shit, S., Musio, F., Yang, K., Amiranashvili, T., Paetzold, J.C., Li, H.B., Menze, B.: 3d vessel graph generation using denoising diffusion. *arXiv preprint arXiv:2407.05842* (2024)
22. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. *OpenAI blog* **1**(8), 9 (2019)
23. Rauch, N., Harders, M.: Interactive Synthesis of 3D Geometries of Blood Vessels. In: Theisel, H., Wimmer, M. (eds.) *Eurographics 2021 - Short Papers*. The Eurographics Association (2021)
24. Roman, M.J., Kizer, J.R., Best, L.G., Lee, E.T., Howard, B.V., Shara, N.M., Devreux, R.B.: Vascular biomarkers in the prediction of clinical cardiovascular disease: the strong heart study. *Hypertension* **59**(1), 29–35 (2012)
25. Schneider, M., Reichold, J., Weber, B., Székely, G., Hirsch, S.: Tissue metabolism driven arterial tree generation. *Medical image analysis* **16**(7), 1397–1414 (2012)
26. Siddiqui, Y., Alliegro, A., Artemov, A., Tommasi, T., Sirigatti, D., Rosov, V., Dai, A., Nießner, M.: Meshgpt: Generating triangle meshes with decoder-only transformers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 19615–19625 (2024)
27. Sinha, A., Hamarneh, G.: Representing anatomical trees by denoising diffusion of implicit neural fields. *arXiv preprint arXiv:2403.08974* (2024)

28. Sung-Bin, K., Chae-Yeon, L., Son, G., Hyun-Bin, O., Ju, J., Nam, S., Oh, T.H.: Multitalk: Enhancing 3d talking head generation across languages with multilingual video dataset. arXiv preprint arXiv:2406.14272 (2024)
29. Talou, G.D.M., Safaei, S., Hunter, P.J., Blanco, P.J.: Adaptive constrained constructive optimisation for complex vascularisation processes. *Scientific Reports* **11**(1), 1–22 (2021)
30. Taylor, C.A., Petersen, K., Xiao, N., Sinclair, M., Bai, Y., Lynch, S.R., UpdePac, A., Schaap, M.: Patient-specific modeling of blood flow in the coronary arteries. *Computer Methods in Applied Mechanics and Engineering* **417**, 116414 (2023)
31. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. *Advances in neural information processing systems* **30** (2017)
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
33. Wolterink, J.M., Leiner, T., Isgum, I.: Blood vessel geometry synthesis using generative adversarial networks. arXiv preprint arXiv:1804.04381 (2018)
34. Wu, J., Hu, Q., Ma, X.: Comparative study of surface modeling methods for vascular structures. *Computerized Medical Imaging and Graphics* **37**(1), 4–14 (2013)
35. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 4541–4550 (2019)