Anatomica: Localized Control over Geometric and Topological Properties for Anatomical Diffusion Models

Supplementary Material

6. Overview

Methodological Details In Sec. 7, we detail the methods relating to diffusion model training, substructure parsing, and geo-topological measurement.
Experimental Details In Sec. 8, we provide additional details on dataset creation, task setup, and evaluation metrics.
Ablations In Sec. 9, we study the influence of various hyperparameters such as individual loss weightings, decoding resolutions, and softmax temperature for geometric and topological guidance.

7. Methodological Details

7.1. Variational Autoencoder

For this study, we adapt the voxel map VAE architecture specified by Kadry et al. [30], which consists of a convolutional encoder and decoder. All architectural and training hyperparameters can be found in tables 5 and 6.
Decoder Architecture We introduce two variants of Anatomica for latent diffusion guidance, with the primary difference being the decoder architecture that converts latent grid representation $\mathbf{z} \in \mathbb{R}^{c\times h\times w\times d}$ into a voxel grid representation $\hat{\mathbf{V}} \in \mathbb{R}^{C\times\alpha\times\beta\times\gamma}$ that can be anatomically characterized. **Anatomica-V** uses a convolutional decoder that mirrors the encoder, where the latent grid can only be decoded to full voxel resolution. On the other hand, **Anatomica-L** uses a neural field-based decoder that takes as input an arbitrary point grid $\mathbf{X} \in \mathbb{R}^{\alpha\times\beta\times\gamma\times 3}$ and returns for each point, the probability vector denoting the most likely anatomical class.
Neural Field Decoder Our decoder $\mathcal{F}$ decodes voxel maps with neural fields by first applying a bottleneck convolution to the latent grid representation in order to aggregate features within a local neighborhood. We then use a set of query points to interpolate into the latent grid representation using the slice operator $\mathcal{T}^l[\mathbf{X}]$ to create a set of latent points which are then point-wise concatenated with random Fourier features [36, 43]. These features are then fed into a multi-layer perceptron (MLP) which consists of several hidden layers and finally outputs a logit vector for each query point. The logit vectors are then softmaxed to produce a segmentation probability vector.
Training We train all autoencoders with a combination of Dice-Cross Entropy reconstruction loss and KL divergence loss [28]. For neural field decoder training, we decode back to the full resolution global voxel grid with a fully discretized domain $\mathbf{X}^q \in \mathbb{R}^{H\times W\times D\times 3}$.

Table 5. Autoencoder architecture hyperparameters

Conv. Encoder (shared)	**Value**
Num. Channels	[64, 128, 256]
Num. Res. Blocks	2
Final Downscaling Factor	4
Bottleneck Dim	3
Conv. Decoder (Anatomica-V)	**Value**
Num. Channels	[64, 128, 256]
Num. Res. Blocks	2
Final Upscaling Factor	4
Neural Field Decoder (Anatomica-L)	**Value**
Bottleneck Conv. Channels	64
Positional Encoding Dim	10
Positional Encoding Bandwidth	1
MLP Hidden Dim	128
MLP Num Layers	3
Normalization	LayerNorm
Activation	ReLU

Table 6. Autoencoder training hyperparameters

Hyperparameter	**Value**
Learning Rate	1×10^{-5}
Epochs	40
Batch Size	1
Dice-CE Loss Weight	1
KL Loss Weight	1×10^{-6}

7.2. Latent Diffusion Model

Architecture & Training For latent diffusion model architecture and training, we follow the formulation and architecture specified by Kadry et al. [30]. All architectural and training hyperparameters can be found in table 7. Our denoising model D_θ is parametrized in a skip-connection manner with a U-Net $\mathcal{K}_\theta$ with a convolutional encoder and decoder through the following relation:

$$D_\theta(\mathbf{z}_\sigma;\sigma) = c_{\text{skip}}(\sigma)\,\mathbf{z}_\sigma + c_{\text{out}}(\sigma)\,\mathcal{K}_\theta\big(c_{\text{in}}(\sigma)\,\mathbf{z}_\sigma;\, c_{\text{noise}}(\sigma)\big) \tag{10}$$

Where (c_{skip},c_{out},c_{in},c_{noise}) are noise-level-dependent scaling coefficients [32], and σ is the noise level. We use the same hyperparameters for the scaling coefficients as in Karras et al. [32], but sample our noise level $p(\sigma)$ from a lognormal distribution with different parameters (see table 7).

Sampling Once the denoiser has been sufficiently trained, we define a specific noise level schedule governing the reverse process, in which the initial noise level, σ, starts at σ_{max} and decreases to σ_{min}:

$$\sigma_i = \left(\sigma_{\text{max}}^{\frac{1}{\rho}} + \frac{i}{N-1}(\sigma_{\text{min}}^{\frac{1}{\rho}} - \sigma_{\text{max}}^{\frac{1}{\rho}})\right)^{\rho} \tag{11}$$

where ρ, σ_{min} and σ_{max} are hyperparameters defined in table 7. We specifically use the stochastic sampling method proposed in Karras et al. [32] (see Tab. 7 for hyperparameters).

Table 7. Diffusion model hyperparameters

Training	**Value**
lr	2.5×10^{-5}
Epochs	50
Batch Size	1
Num. Channels	[64, 128, 196]
Num. Res. Blocks	2
Num. Attn. Heads	1
Attn. Res.	8, 4, 2
σ_{data}	1
$p(\sigma)$ mean	1
$p(\sigma)$ std	1.2
Sampling	**Value**
σ_{min}	1×10^{-2}
σ_{max}	80
ρ	1

7.3. Substructure Parsing

Anatomica revolves around parsing substructures through the use of selection vectors and control domains, enabling the measurement of anatomical properties for specified tissues within localized regions of interest. Selection vectors are binary vectors $\mathbf{u} \in \{0,1\}^C$ which select a subset of tissues from a voxel grid $\mathbf{V}$ through the Boolean subset operator $\mathcal{U}[\mathbf{u}]$ (see Algorithm 1). By varying the Boolean selection vector, we enable the measurement of anatomic structures that are composed of multiple tissue types. Control domains are instantiated as point grids $\mathbf{X} \in \mathbb{R}^{\alpha \times \beta \times \gamma \times 3}$ that are used to parse substructures from grid-like representations such as voxel grids $\mathbf{V}$ using the voxel slicing operator $\mathcal{T}^s[\mathbf{X}]$ with **V-parsing** (see Algorithm 2). Alternatively, we can parse substructures directly from the latent representation $\mathbf{z}$ using the latent slicing operator $\mathcal{T}^l[\mathbf{X}]$ with **L-parsing** (see Algorithm 3). To obtain control domains, we first define a template domain $\mathbf{X}^{\text{temp}} \in \mathbb{R}^{\alpha \times \beta \times \gamma \times 3}$ as a point grid centered at $\mathbf{0}$, with a grid size $\mathbf{g} = [\alpha, \beta, \gamma]$. We then apply a spatial transformation defined by affine transformation parameters $\mathbf{A} = [\mathbf{R}, \mathbf{s}, \mathbf{t}]$ to obtain anatomically relevant control domains.

Algorithm 1 Boolean Subset Operator

Require: $\mathbf{V} \in \mathbb{R}^{C \times H \times W \times D}$ ▷ Voxel map
Require: $\mathbf{u} \in \{0,1\}^C$ ▷ Boolean selection vector
$\hat{\mathbf{S}} \leftarrow \mathbf{0}$ ▷ Initialize
for tissue i where $\mathbf{u}_i = 1$ **do**
  $\hat{\mathbf{S}} \leftarrow \max(\hat{\mathbf{S}}, \mathbf{V}_i)$ ▷ Union via maximum
end for
return $\hat{\mathbf{S}} \in \mathbb{R}^{H \times W \times D}$

Algorithm 2 Voxel Substructure Parsing (V-parsing)

Require: $\mathbf{V} \in \mathbb{R}^{C \times H \times W \times D}$ ▷ Voxel map
Require: $\mathbf{u} \in \{0,1\}^C$ ▷ Selection vector
Require: $\{\mathbf{X}_k\}_{k=1}^K$ where $\mathbf{X}_k \in \mathbb{R}^{\alpha \times \beta \times \gamma \times 3}$ ▷ Control domains
Subset Tissues
$\hat{\mathbf{S}} \leftarrow \mathcal{U}[\mathbf{u}](\mathbf{V}) \in \mathbb{R}^{H \times W \times D}$ ▷ Boolean subset
Parse Substructures
for $k = 1, \ldots, K$ **do**
  $\mathbf{S}_k \leftarrow \mathcal{T}^s[\mathbf{X}_k](\hat{\mathbf{S}}) \in \mathbb{R}^{\alpha \times \beta \times \gamma}$ ▷ Voxel slice
end for
return $\{\mathbf{S}_k\}_{k=1}^K$

Algorithm 3 Latent Substructure Parsing (L-parsing)

Require: $\mathbf{z} \in \mathbb{R}^{c \times h \times w \times d}$ ▷ Latent representation
Require: $\mathbf{u} \in \{0,1\}^C$ ▷ Selection vector
Require: $\{\mathbf{X}_k\}_{k=1}^K$ where $\mathbf{X}_k \in \mathbb{R}^{\alpha \times \beta \times \gamma \times 3}$ ▷ Control domains
Parse Substructures
for $k = 1, \ldots, K$ **do**
  $\mathbf{z}_k \leftarrow \mathcal{T}^l[\mathbf{X}_k](\mathbf{z}) \in \mathbb{R}^{c \times \alpha \times \beta \times \gamma}$ ▷ Latent slice
  $\mathbf{S}_k \leftarrow (\mathcal{U}[\mathbf{u}] \circ \mathcal{F}[\mathbf{X}_k])(\mathbf{z}_k) \in \mathbb{R}^{\alpha \times \beta \times \gamma}$ ▷ Decode
end for
return $\{\mathbf{S}_k\}_{k=1}^K$

7.4. Control Domains

Anatomica supports several methods for defining control domains $\mathbf{X}_k$, each useful for probing different anatomical or geometric properties. In this study, we primarily compute control domain parameters from real anatomical voxel maps and measure geometric properties within such domains to define targets for diffusion guidance. This approach is not limited to guidance use-cases, and can potentially be used

for other machine-learning tasks that use differentiable loss functions. We now detail the algorithmic procedures for computing control domain parameters across different coordinate systems from anatomical voxel maps.

Global Domain Computation The global control domain can be used to measure properties over the entire voxel grid without geometric feature extraction at a variable spatial resolution. We compute global domains through Algorithm 4.

Algorithm 4 Global Domain Computation

Require: (α, β, γ) with $\alpha \approx \beta \approx \gamma$ ▷ Volumetric grid size

Set Affine Parameters
$\mathbf{R} \leftarrow \mathbf{I} \in \mathbb{R}^{3\times3}$ ▷ Rotation
$\mathbf{t} \leftarrow \mathbf{0} \in \mathbb{R}^3$ ▷ Translation
$\mathbf{s} \leftarrow \mathbf{1} \in \mathbb{R}^3$ ▷ Scale
return $\mathbf{A} = [\mathbf{R}, \mathbf{s}, \mathbf{t}]$

Cartesian Domain Computation Cartesian domains enable the measurement of anatomical properties within localized bounding boxes that contain structures of interest. We compute Cartesian domains through Algorithm 5.

Algorithm 5 Cartesian Domain Computation

Require: $\mathbf{V} \in \mathbb{R}^{C\times H\times W\times D}$ ▷ Voxel map
Require: $\mathbf{u} \in \{0,1\}^C$ ▷ Tissue selection vector
Require: (α, β, γ) with $\alpha \approx \beta \approx \gamma$ ▷ Volumetric grid size

Extract Bounding Box
$\hat{\mathbf{S}} \leftarrow \mathcal{U}[\mathbf{u}](\mathbf{V}), \tilde{\mathbf{S}} \leftarrow \mathbb{I}[\hat{\mathbf{S}} > 0.9]$ ▷ Subset & binarize
$\mathbf{r}^{\text{upper}}, \mathbf{r}^{\text{lower}} \leftarrow \text{ExtractLimits}(\tilde{\mathbf{S}})$ where $\mathbf{r}^{\text{upper}}, \mathbf{r}^{\text{lower}} \in \mathbb{R}^3$
Set Affine Parameters
$\mathbf{R} \leftarrow \mathbf{I} \in \mathbb{R}^{3\times3}$ ▷ Rotation
$\mathbf{t} \leftarrow (\mathbf{r}^{\text{upper}} + \mathbf{r}^{\text{lower}})/2 \in \mathbb{R}^3$ ▷ Translation
$\mathbf{s} \leftarrow (\mathbf{r}^{\text{upper}} - \mathbf{r}^{\text{lower}}) \oslash [\alpha, \beta, \gamma]^T \in \mathbb{R}^3$ ▷ Scale
return $\mathbf{A} = [\mathbf{R}, \mathbf{s}, \mathbf{t}]$

Interface Domain Computation Interface domains enable the measurement of local anatomical properties at the interfacial region between two or more structures, such as valve annuli or branch points. We compute interface domains through Algorithm 6.

Algorithm 6 Interface Domain Computation

Require: $\mathbf{V} \in \mathbb{R}^{C\times H\times W\times D}$ ▷ Voxel map
Require: $\mathbf{u}^A, \mathbf{u}^B \in \{0,1\}^C$ ▷ Tissue selection vectors
Require: (α, β, γ) with $\alpha \ll \beta \approx \gamma$ ▷ Planar grid size
Require: $k_{\text{dil}}, \mathbf{R}^r \in \mathbb{R}^{3\times1}$ ▷ Kernel size, ref vector

Extract Interface Regions
$\hat{\mathbf{S}}^A \leftarrow \mathcal{U}[\mathbf{u}^A](\mathbf{V}), \hat{\mathbf{S}}^B \leftarrow \mathcal{U}[\mathbf{u}^B](\mathbf{V})$ ▷ Subset
$\hat{\mathbf{S}}^A_{\text{dil}} \leftarrow \text{maxpool}_{k_{\text{dil}}}(\hat{\mathbf{S}}^A), \hat{\mathbf{S}}^B_{\text{dil}} \leftarrow \text{maxpool}_{k_{\text{dil}}}(\hat{\mathbf{S}}^B)$ ▷ Dilate
$\mathbf{M} \leftarrow \min(\hat{\mathbf{S}}^A_{\text{dil}}, \hat{\mathbf{S}}^B_{\text{dil}})$ ▷ Combine
$\hat{\mathbf{S}}^A_{\text{int}} \leftarrow \hat{\mathbf{S}}^A \odot \mathbf{M}, \hat{\mathbf{S}}^B_{\text{int}} \leftarrow \hat{\mathbf{S}}^B \odot \mathbf{M}$ ▷ Mask interface
Compute Interface Frame Orientations
$\mathbf{p}^A, \mathbf{p}^B \leftarrow \text{Centroid}(\hat{\mathbf{S}}^A_{\text{int}}), \text{Centroid}(\hat{\mathbf{S}}^B_{\text{int}})$ ▷ (Alg. 12)
$\mathbf{R}^\alpha \leftarrow (\mathbf{p}^B - \mathbf{p}^A)/\|\mathbf{p}^B - \mathbf{p}^A\| \in \mathbb{R}^{3\times1}$ ▷ Interface vector
$\mathbf{R}^\beta, \mathbf{R}^\gamma \leftarrow \text{Orthonorm.}(\mathbf{R}^\alpha, \mathbf{R}^r) \in \mathbb{R}^{3\times1}$ ▷ (Alg. 11)
Set Affine Parameters
$\mathbf{R} \leftarrow [\mathbf{R}^\alpha, \mathbf{R}^\beta, \mathbf{R}^\gamma] \in \mathbb{R}^{3\times3}$ ▷ Rotation
$\mathbf{s} \leftarrow [\alpha/H, \beta/W, \gamma/D]^T \in \mathbb{R}^3$ ▷ Scale
$\mathbf{t}^A \leftarrow \mathbf{p}^A \in \mathbb{R}^3, \mathbf{t}^B \leftarrow \mathbf{p}^B \in \mathbb{R}^{3\times1}$ ▷ Translation
return $\mathbf{A}^A = [\mathbf{R}, \mathbf{s}, \mathbf{t}^A], \mathbf{A}^B = [\mathbf{R}, \mathbf{s}, \mathbf{t}^B]$

Curvilinear Domain Computation Curvilinear domains enable the measurement of cross-sectional anatomical properties along tubular structures such as blood vessels. We compute curvilinear domains through Algorithm 7. For skeletonization, we follow the methods and hyperparameters detailed in Kadry et al. [28] for non-differentiable hard skeletonization.

Algorithm 7 Curvilinear Domain Computation

Require: $\mathbf{V} \in \mathbb{R}^{C\times H\times W\times D}$ ▷ Voxel map
Require: $\mathbf{u} \in \{0,1\}^C$ ▷ Tissue selection vector
Require: (α, β, γ) with $\alpha \ll \beta \approx \gamma$ ▷ Planar grid size
Require: $\mathbf{i}_{\text{sub}}, \mathbf{R}^r \in \mathbb{R}^{3\times1}$ ▷ Subsampling Indices, Ref vector

Extract Centerline
$\hat{\mathbf{S}} \leftarrow \mathcal{U}[\mathbf{u}](\mathbf{V}), \tilde{\mathbf{S}} \leftarrow \mathbb{I}[\hat{\mathbf{S}} > 0.9]$ ▷ Subset & binarize
$\mathbf{C} \leftarrow \text{Skeletonize}(\tilde{\mathbf{S}})$ where $\mathbf{C} \in \mathbb{R}^{N_{\text{center}}\times3}$
Compute Curvilinear Frames
$\mathbf{F}^\alpha \leftarrow \text{FiniteDifference}(\mathbf{C}) \in \mathbb{R}^{N_{\text{center}}\times3}$ ▷ Tangent vectors
$\mathbf{F}^\beta_0, \mathbf{F}^\gamma_0 \leftarrow \text{Orthonorm.}(\mathbf{F}^\alpha_0, \mathbf{R}^r)$ ▷ (Alg. 11)
$\mathbf{F}^\beta, \mathbf{F}^\gamma \leftarrow \text{ParallelTransport}(\mathbf{F}^\alpha, \mathbf{F}^\beta_0, \mathbf{F}^\gamma_0)$ ▷ (Alg. 10)
$\mathbf{C}^{\text{sub}} \leftarrow \text{Subsample}(\mathbf{C}, \mathbf{i}_{\text{sub}})$ where $\mathbf{C}^{\text{sub}} \in \mathbb{R}^{N_{\text{planes}}\times3}$
$\mathbf{R}^\alpha, \mathbf{R}^\beta, \mathbf{R}^\gamma \leftarrow \text{Subsample}(\mathbf{F}^\alpha, \mathbf{F}^\beta, \mathbf{F}^\gamma, \mathbf{i}_{\text{sub}}) \in \mathbb{R}^{N_{\text{planes}}\times3}$
Set Affine Parameters
for domain $k = 1, \ldots, N_{\text{planes}}$ **do**
  $\mathbf{R}_k \leftarrow [\mathbf{R}^\alpha_k, \mathbf{R}^\beta_k, \mathbf{R}^\gamma_k] \in \mathbb{R}^{3\times3}$ ▷ Rotation
  $\mathbf{s}_k \leftarrow [\alpha/H, \beta/W, \gamma/D]^T \in \mathbb{R}^3$ ▷ Scale
  $\mathbf{t}_k \leftarrow \mathbf{C}^{\text{sub}}_k \in \mathbb{R}^3$ ▷ Translation
end for
return $\{\mathbf{A}_k\}_{k=1}^{N_{\text{planes}}}$

Spherical Domain Computation Spherical domains enable the measurement of radial anatomical properties of

shell-like structures such as myocardial walls. We compute spherical domains through Algorithm 8. Instead of sampling equidistant points in polar and azimuthal space, we compute equally distributed points on the sphere surface using the Fibonacci lattice algorithm [18].

Algorithm 8 Spherical Domain Computation

Require: $\mathbf{V} \in \mathbb{R}^{C\times H\times W\times D}$ ▷ Voxel map
Require: $\mathbf{u} \in \{0,1\}^C$ ▷ Tissue selection vector
Require: (α,β,γ) with $\alpha \approx \beta \ll \gamma$ ▷ Ray-like grid size
Require: N_{rays}, N_q, $\mathbf{R}^r \in \mathbb{R}^{3\times 1}$ ▷ Number of rays, query ray resolution, ref vector

Generate Radial Directions
$\hat{\mathbf{S}} \leftarrow \mathcal{U}[\mathbf{u}](\mathbf{V})$ ▷ Subset tissues
$\mathbf{p} \leftarrow \text{Centroid}(\hat{\mathbf{S}}) \in \mathbb{R}^3$ ▷ (Alg. 12)
$\mathbf{R}^\gamma \leftarrow \text{FibonacciLattice}(N_{\text{rays}}, \mathbf{p})$ where $\mathbf{R}^\gamma \in \mathbb{R}^{N_{\text{rays}}\times 3}$
$\mathbf{R}^\beta, \mathbf{R}^\alpha \leftarrow \text{Orthonorm.}(\mathbf{R}^\gamma, \mathbf{R}^r) \in \mathbb{R}^{N_{\text{rays}}\times 3}$ ▷ (Alg. 11)
Find Wall Centroids and Set Affine Parameters
for domain $k = 1, \ldots, N_{\text{rays}}$ **do**
  $\mathbf{X}_k^{\text{ray}} \leftarrow \text{MakeQueryRay}(\mathbf{p}, \mathbf{R}_k^\gamma, N_q)$ where $\mathbf{X}_k^{\text{ray}} \in \mathbb{R}^{1\times 1\times N_q\times 3}$
  $\mathbf{S}_k^{\text{ray}} \leftarrow \mathcal{T}^s[\mathbf{X}_k^{\text{ray}}](\hat{\mathbf{S}})$ ▷ Slice along ray
  $\mathbf{p}_{\text{wall},k} \leftarrow \text{Centroid}(\mathbf{S}_k^{\text{ray}}) \in \mathbb{R}^3$ ▷ (Alg. 12)
  $\mathbf{R}_k \leftarrow [\mathbf{R}_k^\alpha, \mathbf{R}_k^\beta, \mathbf{R}_k^\gamma] \in \mathbb{R}^{3\times 3}$ ▷ Rotation
  $\mathbf{s}_k \leftarrow [\alpha/H, \beta/W, \gamma/D]^T \in \mathbb{R}^3$ ▷ Scale
  $\mathbf{t}_k \leftarrow \mathbf{p}_{\text{wall},k} \in \mathbb{R}^3$ ▷ Translation
end for
return $\{\mathbf{A}_k\}_{k=1}^{N_{\text{rays}}}$

Cylindrical Domain Computation Cylindrical domains enable the measurement of radial anatomical properties of walled tubular structures such as coronary arteries. We compute cylindrical domains through Algorithm 9. We compute cylindrical domains by defining equidistant ray centers along the z-axis and equally sampling the polar directions according to predefined sampling resolutions.

Algorithm 9 Cylindrical Domain Computation

Require: $\mathbf{V} \in \mathbb{R}^{C\times H\times W\times D}$ ▷ Voxel map
Require: $\mathbf{u} \in \{0,1\}^C$ ▷ Tissue selection vector
Require: (α,β,γ) with $\alpha \approx \beta \ll \gamma$ ▷ Ray-like grid size
Require: N_z, N_θ, N_q, $\mathbf{R}^r \in \mathbb{R}^{3\times 1}$ ▷ Z-levels, angles, query ray resolution, ref vector

Generate Cylindrical Directions
$\hat{\mathbf{S}} \leftarrow \mathcal{U}[\mathbf{u}](\mathbf{V})$ ▷ Subset tissues
$\mathbf{R}^\gamma \leftarrow \text{CylindricalLattice}(N_z, N_\theta)$ where $\mathbf{R}^\gamma \in \mathbb{R}^{N_{\text{rays}}\times 3}$, $N_{\text{rays}} = N_z \times N_\theta$
$\mathbf{R}^\beta, \mathbf{R}^\alpha \leftarrow \text{Orthonorm.}(\mathbf{R}^\gamma, \mathbf{R}^r) \in \mathbb{R}^{N_{\text{rays}}\times 3}$ ▷ (Alg. 11)
Find Wall Centroids and Set Affine Parameters
for domain $k = 1, \ldots, N_{\text{rays}}$ **do**
  $\mathbf{X}_k^{\text{ray}} \leftarrow \text{MakeQueryRay}(\mathbf{R}_k^\gamma, N_q)$ where $\mathbf{X}_k^{\text{ray}} \in \mathbb{R}^{1\times 1\times N_q\times 3}$
  $\mathbf{S}_k^{\text{ray}} \leftarrow \mathcal{T}^s[\mathbf{X}_k^{\text{ray}}](\hat{\mathbf{S}})$ ▷ Slice along ray
  $\mathbf{p}_{\text{wall},k} \leftarrow \text{Centroid}(\mathbf{S}_k^{\text{ray}}) \in \mathbb{R}^3$ ▷ (Alg. 12)
  $\mathbf{R}_k \leftarrow [\mathbf{R}_k^\alpha, \mathbf{R}_k^\beta, \mathbf{R}_k^\gamma] \in \mathbb{R}^{3\times 3}$ ▷ Rotation
  $\mathbf{s}_k \leftarrow [\alpha/H, \beta/W, \gamma/D]^T \in \mathbb{R}^3$ ▷ Scale
  $\mathbf{t}_k \leftarrow \mathbf{p}_{\text{wall},k} \in \mathbb{R}^3$ ▷ Translation
end for
return $\{\mathbf{A}_k\}_{k=1}^{N_{\text{rays}}}$

Parallel Transport Procedure For curvilinear coordinate systems, we aim to maintain consistent frame orientations as we move along the centerline. To do this, we apply parallel transport by propagating an initial orthonormal frame along a centerline using the Rodrigues rotation formula.

Algorithm 10 ParallelTransport

Require: $\mathbf{F}^1 \in \mathbb{R}^{N_{\text{center}}\times 3}$ ▷ Normalized tangent vectors
Require: $\mathbf{F}_0^2, \mathbf{F}_0^3 \in \mathbb{R}^3$ ▷ Initial normalized frame vectors
for $i = 1, \ldots, N_{\text{center}} - 1$ **do**
  $\mathbf{a}_i \leftarrow (\mathbf{F}_{i-1}^1 \times \mathbf{F}_i^1)/\|\mathbf{F}_{i-1}^1 \times \mathbf{F}_i^1\|$ ▷ Rotation axis
  $\theta_i \leftarrow \cos^{-1}(\mathbf{F}_{i-1}^1 \cdot \mathbf{F}_i^1)$ ▷ Rotation angle
  $\mathbf{F}_i^2, \mathbf{F}_i^3 \leftarrow \text{Rodrigues}(\mathbf{F}_{i-1}^2, \mathbf{F}_{i-1}^3, \mathbf{a}_i, \theta_i)$
end for
return $\mathbf{F}^2, \mathbf{F}^3 \in \mathbb{R}^{N_{\text{center}}\times 3}$

Orthonormalization Procedure For interface, curvilinear, spherical, and cylindrical coordinate systems, we wish to compute a set of orthonormal frame vectors from an initial vector. To do this, we define an arbitrary reference vector $\mathbf{R}^r$ and compute orthonormal frame vectors from a primary direction vector by taking successive cross products. For numerical stability, we use an alternate reference vector if the reference and initial vectors are perfectly aligned.

Algorithm 11 Orthonormalization

Require: $\mathbf{U}^0 \in \mathbb{R}^{3\times 1}$ ▷ Primary direction vector
Require: $\mathbf{R}^r \in \mathbb{R}^{3\times 1}$ ▷ Reference vector
$\mathbf{U}^1 \leftarrow (\mathbf{U}^0 \times \mathbf{R}^r)/\|\mathbf{U}^0 \times \mathbf{R}^r\| \in \mathbb{R}^{3\times 1}$ ▷ Second frame vector
$\mathbf{U}^2 \leftarrow \mathbf{U}^0 \times \mathbf{U}^1 \in \mathbb{R}^{3\times 1}$ ▷ Third frame vector
return $\mathbf{U}^1, \mathbf{U}^2$

7.5. Geometric Measurement & Guidance

Scale Standardization of Mass We normalize the measured mass m_k by the total number of voxels in the control domain $\alpha\beta\gamma$ in order to remain invariant to control domain discretization, allowing us to maintain similar geometric loss weightings across different discretization levels.

Local to Global Transformation of Moments Our geometric moment formulation can be sensitive to control domain discretization and coordinate system choice. For example, a substructure can encompass 80% of the control domain, while the control domain itself occupies only a small region of the global domain, resulting in a large measured mass m_k. Likewise, a localized control domain can yield a centroid $\mathbf{p}_k$ that lies at the center of the control domain but at the periphery of the global domain. We therefore aim to express our geometric measurements in a manner that is invariant to control domain choice. This is important when applying MSE-based geometric loss functions across different tasks due to varying scales. Geometric moments are first computed in normalized local grid coordinates on $[0, 1]^3$. The local centroid is therefore converted to a centered displacement by subtracting $\frac{1}{2}\mathbf{1}$ and then mapped to global coordinates using the forward control-domain transformation parameters $\mathbf{A}_k = [\mathbf{R}_k, \mathbf{s}_k, \mathbf{t}_k]$.

Stabilizing Covariance Normalization As we normalize the covariance matrix by the trace, we stabilize the gradient in the case of empty substructures by adding a small epsilon (1e-9) to the diagonal of the covariance matrix.

Adaptive Mass Weighting To avoid centroid and covariance gradient explosion in the case of near-empty segmentations, we adaptively weight the centroid and covariance losses by the mass of the substructure. Below a predefined threshold, we set the centroid and covariance weightings $\lambda_1 = \lambda_2 = 0$. This mass threshold is determined on a task-by-task basis, where we multiply the average mass in the real dataset for the task by a factor of 0.1.

Algorithm 12 Geometric Measurement

Require: $\mathbf{S}_k \in \mathbb{R}^{\alpha\times\beta\times\gamma}$ ▷ Substructure
Require: $\mathbf{R}_k \in \mathbb{R}^{3\times 3}$ ▷ Rotation matrix
Require: $\mathbf{s}_k \in \mathbb{R}^3$ ▷ Scale vector
Require: $\mathbf{t}_k \in \mathbb{R}^3$ ▷ Translation vector

Compute Local Moments
$m_k^{\text{raw}} \leftarrow \text{ComputeMass}(\mathbf{S}_k)$ ▷ (Eq. 7)
$\mathbf{p}_k^{\text{local}} \leftarrow \text{ComputeCentroid}(\mathbf{S}_k, m_k^{\text{raw}})$ ▷ (Eq. 7)
$\mathbf{\Sigma}_k^{\text{local}} \leftarrow \text{ComputeCovariance}(\mathbf{S}_k, \mathbf{p}_k^{\text{local}}, m_k^{\text{raw}})$ ▷ (Eq. 7)
$m_k^{\text{local}} \leftarrow m_k^{\text{raw}}/(\alpha\beta\gamma)$ ▷ Normalize by voxel count
Local to Global Transformation
$\mathbf{J}_k \leftarrow \mathbf{R}_k \,\text{diag}(\mathbf{s}_k)$ ▷ Rotation-scale matrix
$m_k^{\text{global}} \leftarrow m_k^{\text{local}} \cdot |\det(\mathbf{J}_k)|$ ▷ Transform mass
$\mathbf{d}_k^{\text{local}} \leftarrow \mathbf{p}_k^{\text{local}} - \frac{1}{2}\mathbf{1}$ ▷ Local displacement from center
$\mathbf{d}_k^{\text{global}} \leftarrow \mathbf{J}_k \mathbf{d}_k^{\text{local}}$ ▷ Transform displacement
$\mathbf{p}_k^{\text{global}} \leftarrow \mathbf{t}_k + \mathbf{d}_k^{\text{global}}$ ▷ Transform centroid
$\mathbf{\Sigma}_k^{\text{global}} \leftarrow \mathbf{J}_k \mathbf{\Sigma}_k^{\text{local}} \mathbf{J}_k^T$ ▷ Transform covariance
return $(m_k^{\text{global}}, \mathbf{p}_k^{\text{global}}, \mathbf{\Sigma}_k^{\text{global}})$

7.6. Topological Measurement & Guidance

We partition the persistence set into disjoint sets $\mathcal{Y}_k$ and $\mathcal{Z}_k$ consisting of points that should be preserved or suppressed based on a topological prior $\mathcal{B}_k = [B_{k,0}, B_{k,1}, B_{k,2}] \in \mathbb{N}^3$, which specifies the desired features for the components, loops, and voids, respectively. For each dimension, we sort the points by persistence and select the top $B_{k,i}$ points for each dimension i specified by the prior.

7.7. Parallelization

For our geometric measurement operations, we take advantage of parallel GPU computation. We parallelize across different batch indices, constraints, and substructures. When computing control domains, some domain types allow for invalid domains, such as in the case of spherical ray domains, where the ray may not intersect with the substructure. We handle these invalid domains by masking out the computed loss. The only exceptions are the skeletonization step for curvilinear control domains and persistent homology computation, as no GPU-compatible implementations are publicly available, and CPU-parallelization over several cores did not provide significant speedups.

8. Experimental Details

8.1. Baselines

Explicit Conditioning To ensure that the elements of $\mathbf{G}_{\text{exp}}$ are roughly between 0 and 1, we min-max normalize the masses m_k, centroids $\mathbf{p}_k$, and normalized covariances $\mathbf{\Sigma}_k^n$ with values calculated from the real dataset (Tab. 8). The LDM input channel count is increased to accommodate the concatenated input. This method does not readily permit the use of dropout to train a diffusion model in an unconditional

Table 8. Normalizing constants for geometric moments during explicit conditioning across different tasks.

	Geometric Control Task			
Parameter	**RV**	**Mitral**	**Aortic**	**Myo**
Mass Min m_k	3.19×10^{-3}	3.67×10^{-4}	0	0
Mass Max m_k	1.3×10^{-2}	1.36×10^{-3}	8.59×10^{-4}	1.95×10^{-5}
Centroid Min $\mathbf{p}_k$	0	0	-7.81×10^{-3}	0
Centroid Max $\mathbf{p}_k$	1	1	0.91	0.64
Covariance Min $\mathbf{\Sigma}_k$	-1×10^{-4}	-8.66×10^{-4}	-5.59×10^{-4}	2.88×10^{-4}
Covariance Max $\mathbf{\Sigma}_k$	1×10^{-2}	2.34×10^{-3}	1.56×10^{-3}	8.03×10^{-4}

manner because the null condition is defined as zero, which is equivalent to the minimum moment values.

Implicit Conditioning To compute the ellipsoidal distance map, we use the centroids $\mathbf{p}_k$ and non-normalized covariances $\mathbf{\Sigma}_k$ for each component to compute the Mahalanobis distance [7] for each voxel position. We then apply a shifted sigmoid transform to constrain the outputs between 0 and 1, and subsequently concatenate the resulting grid to the latents. To enable unconditional generation, we randomly drop out each substructure channel of $\mathcal{G}$imp with a probability of 0.1.

8.2. Datasets

Cardiac Dataset For our study, we utilize TotalSegmentator v2 [46] to create the cardiac segmentations, with 596 3D segmentations manually selected based on segmentation quality assessment. Cardiac structures include the myocardium (Myo), left and right atria (LA & RA), left and right ventricles (LV & RV), aorta (Ao), and pulmonary artery (PA), were segmented using a specialized TotalSegmentator model trained on sub-millimeter resolution data. For the inferior vena cava (IVC), superior vena cava (SVC), and pulmonary veins (PV), we retain the labels from the original dataset. This results in 11 channels per segmentation. To ensure anatomical validity, we perform topological filtration on all structures except the pulmonary veins, where we extract only the largest connected component. The resulting segmentations are standardized by resampling to a uniform voxel resolution of 2mm and subsequently cropped to a fixed range. The crop center is determined from the union of all four chamber segmentations, and the crop length is set to 128 voxels for each side.

Aortic Dataset For the aorta dataset, we extract labels directly from the original TotalSegmentator v2 [46] segmentations, without applying a specialized model, resulting in 450 3D segmentations manually selected based on segmentation quality assessment. The labels include the main aortic trunk and the ascending branches, which comprise the brachiocephalic trunk (BCT), left common carotid artery (LCCA), right common carotid artery (RCCA), left subclavian artery (LSCA), and right subclavian artery (RSCA), for a total of 7 channels per segmentation. All segmentations are resampled to an isotropic voxel size of 2 mm and cropped to a spatial size of 128^3 using a crop center determined from the center of all combined tissues.

Table 9. Task-specific hyperparameters and configurations for geometric control tasks.

	Geometric Control Task			
Parameter	**RV**	**Mitral**	**Aortic**	**Myo**
Domain	Cartesian	Interface	Curvilinear	Spherical
Selection Vector	[RV]	[LV], [LA]	[Ao]	[Myo]
Num. Substructures	1	2	5	4
Grid Resolution	[64,64,64]	[4,32,32]	[1,32,32]	[4,4,16]
Mass Threshold	10^{-5}	10^{-4}	10^{-6}	10^{-6}
λ_{geo}	1	1	1	1
λ_0 (Mass)	10^7	10^9	10^9	10^9
λ_1 (Centroid)	10^5	10^6	10^5	10^5
λ_2 (Covariance)	10^4	10^4	10^3	10^4

Spinal Dataset For the spinal dataset, we utilize the CT-Spine1K dataset [9] and extract all vertebral body segmentations, resulting in 784 3D segmentations. The segmentations include 7 cervical vertebrae (C1–C7), 12 thoracic vertebrae (T1–T12), and 5 lumbar vertebrae (L1–L5), for a total of 25 channels per segmentation. To ensure spatial consistency and anatomical completeness, all segmentations are first resampled to an isotropic voxel spacing of $1\,\text{mm}$. The center of the crop box is determined from the union (voxelwise sum) of all vertebral structures in each scan, and a fixed crop of 128^3 voxels is applied for each case.

Coronary Dataset For the coronary dataset, we extract coronary artery-related labels from the DISRUPT-CAD dataset [45], consisting of 120 patients with approximately 375 OCT frames in the longitudinal (z) direction. The segmentations include lumen (Lu), calcium (Ca), and vessel wall (Ve), for a total of 4 channels per segmentation. Training samples are generated by resampling the x and y directions to 128×128 pixels while preserving the original z resolution, then randomly cropping 128 consecutive frames along the z-axis from each patient scan. This yields approximately 360 unique 3D segmentations of size 128^3 with an isotropic in-plane voxel spacing of approximately $0.1\,\text{mm}$.

8.3. Tasks

Geometric Control Tasks We detail the task-specific hyperparameters and configurations for the geometric control tasks in Tab. 9.

Topological Control Tasks We detail the task-specific hyperparameters and configurations for the topological control tasks in Tab. 10.

Multiscale Control We detail the task-specific hyperpa-

Table 10. Task-specific hyperparameters and configurations for topological control tasks.

Parameter	Topological Control Task			
	Atrial Separation	Branch Connectivity	Vert. Connectivity	Calcium Count
Domain	Global	Global	Global	Global
Selection Vector	[LA, RA]	All Tissues	[T6–T10]	[Ca]
Num. Substructures	1	1	1	1
Grid Resolution	[64,64,64]	[64,64,64]	[64,64,64]	[64,64,64]
Softmax Value	4	4	4	4
λ_{topo}	5	1	5	50
Prior B0	2	1	1	2
Prior B1	0	0	9	0
Prior B2	0	0	0	0

Table 11. Task-specific hyperparameters and configurations for multiscale control tasks. Hyperparameters marked with a slash / indicate smaller and larger domain configurations, respectively.

Parameter	Spinal	Aorta	Myo Wall	Vessel Wall
Domain	Cartesian	Curvilinear	Spherical	Cylindrical
Selection Vector	[T5–T10]/[T6–T8]	[Ao]	[Myo]	[Ca, Ve]
Num. Substructures	1	5	16	16
Grid Resolution	[64,64,64]	[1,32,32]	[1,32,32]	[1,32,32]
Mass Threshold	10^{-4}	10^{-6}	10^{-6}	10^{-6}
λ_{geo}	1	1	1	1
λ_0 (Mass)	10^7	10^9	10^9	10^9
λ_1 (Centroid)	10^5	10^5	10^5	10^5
λ_2 (Covariance)	10^4	10^3	10^4	10^4
Domain Grid	[64,64,64]	[1,16,16]/[16,16,16]	[4,4,16]/[8,8,16]	[4,4,32]/[16,16,32]

rameters and configurations for the multiscale control tasks in Tab. 11. For the spinal task, we achieve multiscale control by changing the selection vector to include fewer or more vertebral bodies. For all other tasks, we change the control domain grid resolution along specified axes.

Partial Decoding To study partial decoding resolution, we used a Cartesian domain with different resolutions. For Anatomica-L, both coarse and local L-parsing used grid resolutions of 32^3, 64^3, and 128^3 for low, medium, and high resolutions respectively. For Anatomica-V, we used global decoding with a fixed resolution of 128^3. We measured speed in terms of the maximum number of label maps sampled per second using the maximum allowable batch size on a single GPU. We used an A100 with 40 GB of memory for benchmarking. For geometric guidance, the wall clock time was approximately 50 seconds per sample for the highest decoding resolution with a convolutional decoder.

8.4. Evaluation

Frechet Morphological Distance To compute the morphological features, the features are normalized by the mean and standard deviation of the real data.

Pointcloud evaluation metrics: To compute the point cloud metrics, we calculate NNA for every tissue label using 256 points sampled using farthest point sampling. The metric is then averaged over the number of components. To compute the pointcloud distances, we approximate Earth Mover's Distance (EMD) through the Sinkhorn divergence [16].

Topological Precision To compute the Betti numbers, we take the argmax of the predicted segmentation and compute persistent homology. For a binary segmentation, the barcodes are 1 or 0 depending on the existence of the structure. We then take the sum of barcodes per dimension as the Betti number. The topological precision is then the fraction of samples with the correct Betti number per dimension.

9. Ablation Studies

9.1. Geometric Guidance Ablations

We aim to study the influence of individual geometric loss weights on the geometric fidelity and generation quality. We specifically examine the influence of *disentangled* geometric guidance, where, for example, we only constrain the centroid but let size and shape free to vary. To do this, we sweep over the composite geometric loss weighting λ_{geo} for all tasks, and apply different combinations of loss weightings $[\lambda_0, \lambda_1, \lambda_2]$ to activate or deactivate different geometric loss terms (see Tab. 12). We sample 128 samples for each experiment, with 100 sampling steps.

Effect of Guidance Weight In Fig. 8, we see that increasing geometric guidance weight when all loss weightings are activated (Full) improves geometric fidelity up to a certain weight, after which sample quality degrades, decreasing geometric fidelity. This is especially pronounced in the case of centroid-only guidance for the mitral valve and myocardium wall tasks. For generation quality, we see similar trends where increasing guidance weights can reduce FMD up to a certain guidance weight.

Effect of Disentangled Guidance In Fig. 8, we demonstrate that our framework supports disentangled geometric guidance across all tasks. For instance, centroid-only guidance achieves centroid fidelity comparable to full guidance, without significantly affecting mass fidelity, shape fidelity, or generation quality as measured by FMD.

Table 12. Loss weight configurations for geometric guidance ablation study.

Guidance Loss	λ_0	λ_1	λ_2
Full	✓	✓	✓
Mass Only	✓	×	×
Centroid Only	×	✓	×
Covariance Only	×	×	✓

9.2. Topological Guidance Ablations

We aim to study the influence of topological loss weightings, softmax temperature, and partial decoding strategy on topological fidelity. We first sample 64 segmentations for several combinations of guidance weight and softmax temperature and evaluate topological fidelity for every combination (Fig. 9). We then sample 128 samples for various coarse decoding resolutions and guidance weights while evaluating topological fidelity (Fig. 10) and sampling speed (Tab. 13).

Effect of Guidance Weight We see in Fig. 9 that increasing guidance weights broadly improves topological fidelity but can decrease fidelity with excessively large guidance weights.

Effect of Softmax Temperature Similarly, in Fig. 9, we see that increasing softmax temperature can improve topological fidelity for the same guidance weight, but also improves robustness against the negative effects of exceedingly high guidance weights. The atrial separation task is an exception to this, where topological precision for loops and voids is maximized by using a softmax temperature of 1.

Effect of Partial Decoding Strategy We see in Fig. 10 that applying partial decoding with increased resolution can significantly improve topological fidelity at an increased computational cost. We find that the benefits of increased decoding resolution varies based on the topological feature and task. For example, the number of extra loops in the atrial separation task is minimized at a decoding resolution of 128, while the number of extra components for the aortic branch task is invariant after a decoding resolution of 32. We also see from Tab. 13 that a decoding resolution of 64 represents a good trade-off between computational cost and topological fidelity, providing a speedup of 11x over the next highest resolution. For topological guidance, the wall-clock time was approximately 420 seconds per sample for the highest decoding resolution with a convolutional decoder.

Table 13. **Topological sampling speed comparison for partial decoding strategies**. Speed is measured in terms of sampled label maps per second using the maximum allowable batch size on a single GPU, normalized to the slowest method.

Methodology			
Approach	**Domain**	**Res.**	**Speed (↑)**
Anatomica-L	Coarse	16	**32.00**
		32	26.25
		64	11.00
		128	1.14
Anatomica-V	Global	128	1.00

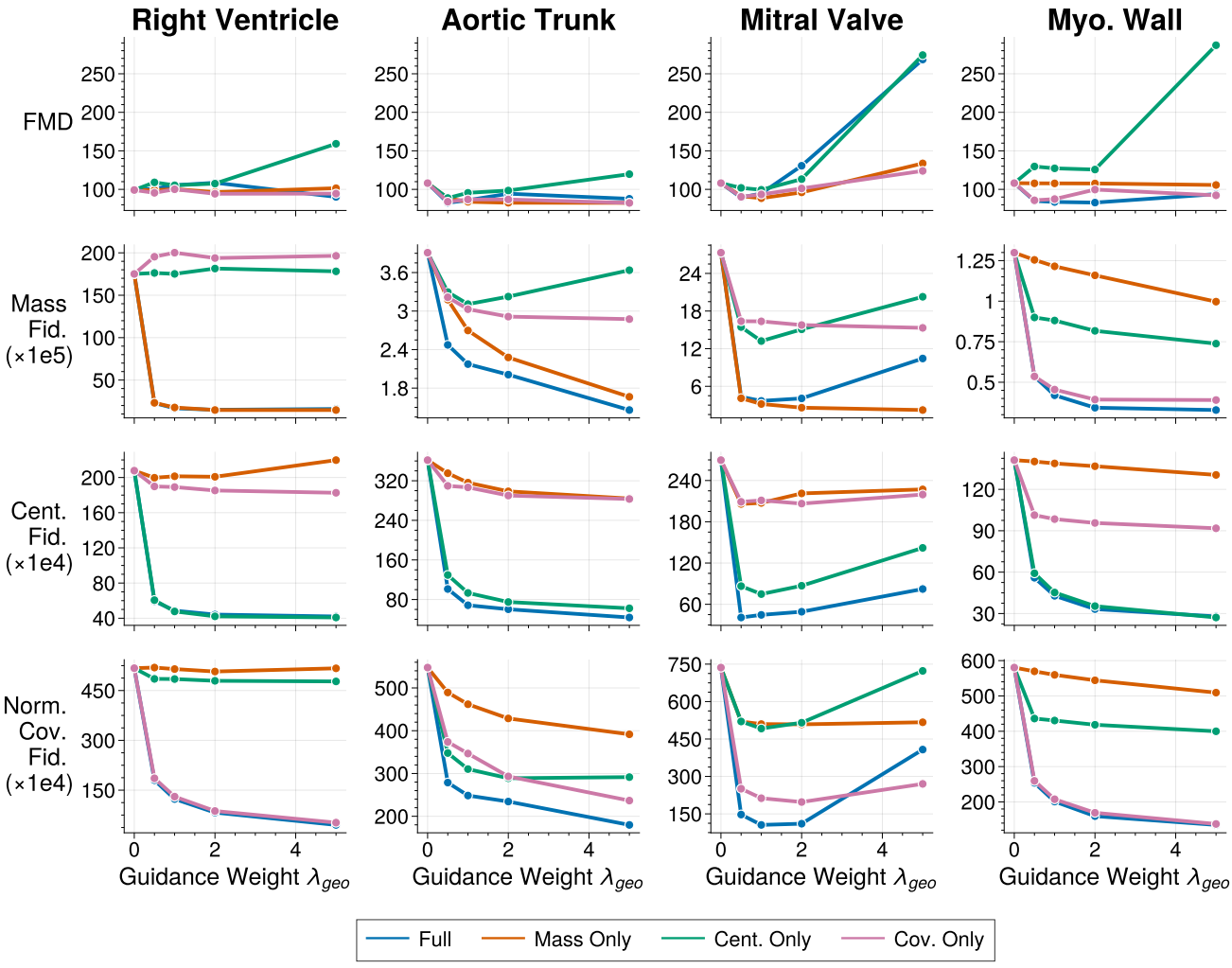

Figure 8. **Geometric guidance and disentangled guidance ablation study.**

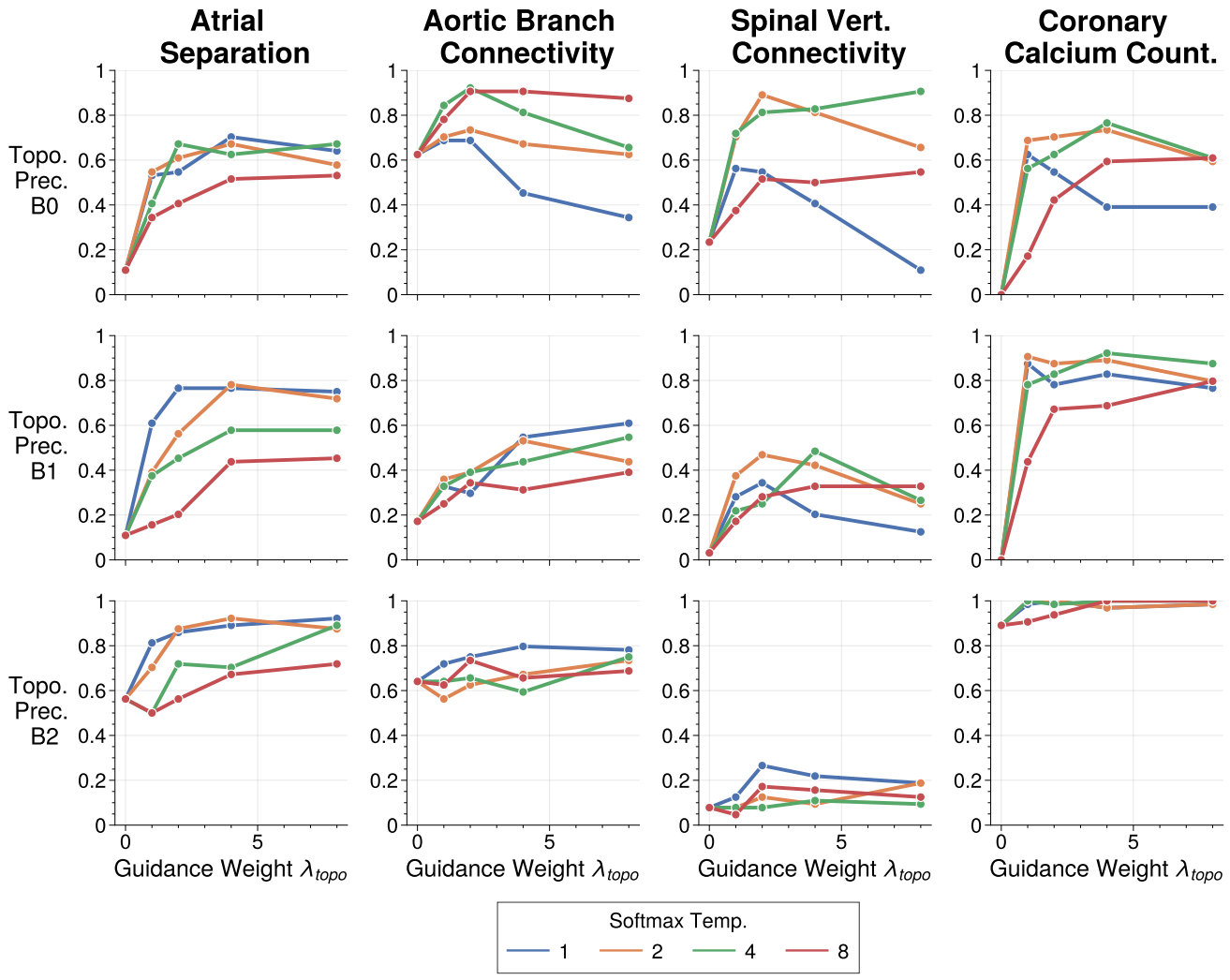

Figure 9. **Topological guidance and softmax temperature ablation study.**

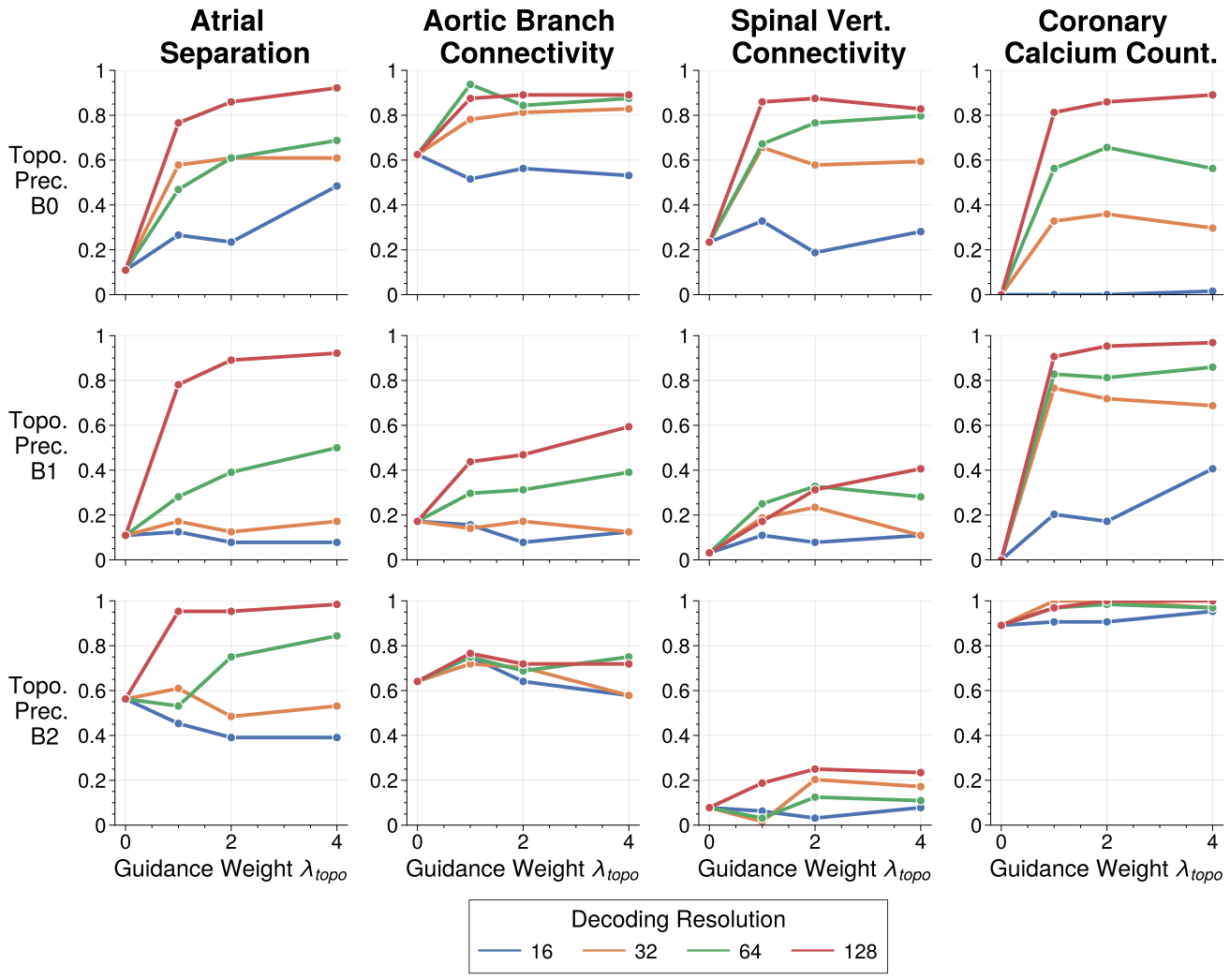

Figure 10. **Topological guidance and partial decoding resolution ablation study.**