RELATIVE TOPOLOGY AS A SUPERVISORY SIGNAL FOR LONG-RANGE DEPENDENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

Topological data analysis provides principled descriptors of global structure beyond local visual appearance. Standard topological invariants, however, characterise an entire space and are therefore insensitive to which elements are selected by a query, limiting their use for query-dependent relational tasks. We address this mismatch using *relative homology*, which characterises a space with respect to a designated subspace. We introduce *relative topological supervision*, a training-only auxiliary objective in which the query determines the reference subspace and relative homology provides a target encoding the resulting query-dependent structural relation. The target is constructed from the training data and query without requiring annotations beyond those used for the downstream task; the auxiliary prediction head is discarded at inference. On the long-range same-path task in Pathfinder, relative topological supervision enables a plain feedforward convolutional network trained from scratch, without residual connections, recurrence, or attention, to solve the task, whereas matched controls using query-independent global topology or permuted topological targets fail. The method remains effective with compact models and limited training data, and we further demonstrate its applicability to retinal vessel segmentation. These results establish the usefulness of the tested query-dependent degree-zero construction across synthetic recognition and real-image segmentation, and motivate query-defined relative topology as a broader design principle.

1 INTRODUCTION

Topological data analysis (TDA) offers a principled language for describing global structure beyond local visual appearance. Homology characterises a space through topological features that remain invariant under continuous deformation, and its ranks, the Betti numbers, provide concise integer-valued summaries: β_0 counts connected components and β_1 counts independent loops (Hatcher, 2002). These global summaries complement deep visual models, which often emphasise local appearance, by providing structural information that spans the entire scene. Because they describe the topology of a space as a whole, Betti numbers are global topological invariants.

Connectedness is a topological property, but whether two specified locations belong to the same component is query-dependent and need not be retained by a global summary. Pathfinder poses the analogous visual question: the input specifies two marked locations and asks whether they belong to the same path. We refer to this selected pair as the query. Different queries on the same underlying scene can produce different labels even though its global topology remains unchanged. Answering the query requires preserving contour identity across widely separated locations while ignoring nearby distractors, making same-path recognition a direct test of *long-range dependency*. Pathfinder was introduced to isolate this form of perceptual grouping (Linsley et al., 2018) and was subsequently studied in raster-based visual models (Kim et al., 2020a) and flattened-sequence benchmarks (Tay et al., 2021). This query dependence exposes a fundamental mismatch: a global topological target remains constant across queries and therefore does not require a model to preserve which locations were selected. A topological target for a relational task must therefore encode not only global structure but also how the queried elements are situated within that structure.

Relative homology provides a classical language for expressing this dependence (Hatcher, 2002): rather than describing a space in isolation, it describes topology relative to a designated subspace.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Our key idea is to construct the reference subspace from the query, allowing the resulting topological quantity to encode how the queried locations relate to the scene. This places global and query-dependent topological supervision in a common mathematical framework. The novelty of our approach lies in turning this classical construction into a supervisory principle for relational geometric representation learning: relative homology describes how the scene is structured with respect to the queried elements, and predicting the resulting quantity trains the representation to preserve that relation. To our knowledge, prior work has not used a query-defined reference subspace to turn relative homology into a supervisory signal for learning relational geometric representations.

We demonstrate this concept on Pathfinder by taking the reference subspace to be the straight segment joining the two marked locations. The auxiliary target measures how adding this segment changes the number of foreground connected components. The target is computed deterministically from each training raster and requires no external annotations. During training, the network predicts this quantity in addition to the same-path label; the auxiliary prediction head is removed at inference. We call this training-only construction *relative topological supervision*.

Prior research has primarily treated long-range dependency as an architectural problem by changing how information propagates through the network (Linsley et al., 2018; Kim et al., 2020a; Gu et al., 2022; Li et al., 2023; Smith et al., 2023). We instead ask whether an ordinary feedforward network can learn the same relation when it is given a task-aligned supervisory signal. Because relative topological supervision changes only the training objective, it can be applied to different visual backbones without introducing a specialised inference-time operator.

Our experiments compare relative topological supervision with label-only training, a query-independent global topological target, and a permuted-target control that preserves the target marginal while breaking input–target correspondence. On a plain feedforward convolutional network, only the correctly paired relative target yields reliable same-path recognition: label-only training is unstable, and both controls remain near chance. With the same relative target, plain and residual convolutional stacks and Swin-T all learn the task, showing that the target is usable across distinct backbone families, although their data and parameter requirements differ substantially. A plain CNN reaches 98.15% mean accuracy using only $4{,}000$ training scenes, while a compact variant with $23{,}470$ parameters reaches 97.19% at $16{,}000$ scenes. The auxiliary target is computed once before training, and its prediction head adds no inference-time computation (Appendix D.5).

Finally, we examine whether the same degree-zero segment construction transfers beyond synthetic contour recognition to retinal-vessel segmentation. Retinal vessels form thin, branching structures in which small segmentation gaps can disrupt long-range continuity while producing only minor changes in pixelwise overlap (Shit et al., 2021). We evaluate on DRIVE (Staal et al., 2004), a public retinal-vessel segmentation benchmark, using the relative topological target as a training-only auxiliary task. Relative supervision yields higher mean Dice and IoU, although these overlap gains do not reach significance in the image-level paired test; the clearest supported effects are on clDice, Betti-1 error, thin-centreline recall, and junction fidelity.

Our main contributions can be summarised as follows:

(i) We introduce *relative topological supervision*, a training-only formulation in which a query-defined reference subspace turns relative homology into a task-aligned target for learning relational geometric representations.

(ii) Through matched controls on the two-dimensional Pathfinder same-path task, we show that query-aligned supervision enables a plain feedforward CNN without specialised long-range modules to learn the task reliably. Backbone, data, and capacity sweeps demonstrate applicability across convolutional and Transformer backbones and characterise sample and parameter efficiency.

(iii) We further evaluate the same degree-zero construction on DRIVE, a public retinal-vessel segmentation benchmark, where it improves structural fidelity without changing the inference architecture or its input.

2 Related work

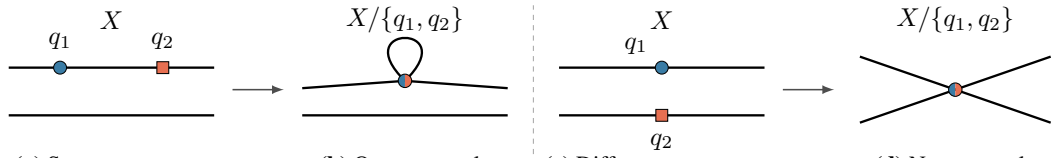
Topology-aware learning. Topology has been incorporated into deep learning in several roles. Persistent-homology summaries have been transformed into learnable features for classification

(Hofer et al., 2017; Kim et al., 2020b; Carrière et al., 2020), while topological constraints have been used to preserve the structure of latent representations (Moor et al., 2020). Topology has also served as a supervisory signal in knowledge distillation, where the global topology of teacher and student representations is aligned (Kim et al., 2024). In structured prediction, persistent-homology-based objectives penalise discrepancies between topological features of predicted and reference segmentations (Hu et al., 2019; Clough et al., 2022; Stucki et al., 2023), while clDice promotes continuity in tubular structures through agreement between segmentation masks and their centrelines (Shit et al., 2021). These approaches use topology as a learnable input representation, a constraint on latent geometry, a distillation signal, or an objective on the final structured prediction. Our approach uses a query-defined reference subspace to construct a relative topological target for auxiliary prediction, training the shared representation to capture structure relevant to the queried relation. This perspective connects topology-aware learning to Pathfinder, where previous work has primarily focused on architectural mechanisms for learning the same-path relation.

Pathfinder and long-range architectures. Linsley et al. (2018) introduced two-dimensional Pathfinder with target paths comprising 6, 9, or 14 segments, using 150×150 images and approximately $900{,}000$ training examples for each path-length condition. Despite this scale, many of their tested feedforward CNNs struggled as the path length increased. Although several state-of-the-art feedforward architectures achieved high accuracy on the 14-segment condition, they required substantially more parameters than the hGRU. Their proposed horizontal gated recurrent unit (hGRU) solved the same condition with fewer parameters, providing a parameter-efficient recurrent approach to perceptual grouping. This parameter efficiency comes with repeated spatial computation: hGRU processes each image through multiple recurrent steps, requiring backpropagation through the unrolled computation during training. Subsequent work investigated which recurrent mechanisms support perceptual grouping and curve tracing (Kim et al., 2020a; Linsley et al., 2020). Tay et al. (2021) later recast Pathfinder as a sequence-classification problem by flattening each input image into a pixel sequence in Long Range Arena. Subsequent progress in this sequence setting has primarily come from architectural developments, including structured state-space models and long-convolution architectures (Gu et al., 2022; Romero et al., 2022; Li et al., 2023; Smith et al., 2023). Because these models receive one-dimensional pixel sequences whereas ours operate directly on two-dimensional images, the results are not directly comparable. This literature nevertheless provides the architectural context for our complementary question: whether a task-aligned supervisory signal enables a standard visual backbone to learn the same-path relation.

Self-pretraining and supervisory signals. Amos et al. (2024) showed that comparisons based on random initialisation can overstate differences between long-sequence architectures. Denoising pretraining on the downstream inputs, without an external corpus, substantially improved Transformers and narrowed their gap with specialised sequence models. Coser et al. (2025) further analysed these gains, highlighting the role of the attention parameters learned during pretraining. These findings establish the training signal, rather than architecture alone, as an important factor in long-range learning. Our approach shares this emphasis but differs in the signal and training procedure. Self-pretraining uses a generic reconstruction objective followed by fine-tuning, whereas relative topological supervision introduces a task-aligned, query-dependent auxiliary objective directly during supervised training. Relative topological supervision therefore provides structural supervision to the shared representation without requiring a separate pretraining phase.

Query conditioning. Visual models can incorporate conditioning information through spatial transformations, dynamically generated filters, feature-wise modulation, or explicit coordinate channels (Jaderberg et al., 2015; De Brabandere et al., 2016; Perez et al., 2018; Liu et al., 2018). Spatially indexed models can also preserve location by evaluating or sampling features at selected coordinates (Saito et al., 2019; Kirillov et al., 2020). These approaches change how a query or spatial location enters the input, feature computation, or readout. Relative topological supervision operates on a different axis: the query determines what structural relation is supervised, without requiring a query-specific inference-time operator. Query conditioning tells the model which locations are being asked about; relative topological supervision adds a training objective that depends on how those locations relate to the scene. In our Pathfinder experiments, the image markers provide the query, and the auxiliary target measures how the segment joining them changes foreground connectivity.



(a) Same component **(b)** One new cycle **(c)** Different components **(d)** No new cycle

Figure 1: Pairwise connectivity on a fixed foreground X with $\beta_0(X) = 2$ and $\beta_1(X) = 0$. Panels (a,c) show two choices of the query pair, with q_1 marked by a circle and q_2 by a square; panels (b,d) show the quotient $X/\{q_1, q_2\}$, in which the two points are identified. Identifying points in the same component adds one cycle, $\beta_1(X, \{q_1, q_2\}) - \beta_1(X) = 1$; identifying points in different components adds none.

3 From global to relative topology

This section identifies a limitation of global topological supervision for query-dependent tasks and develops an alternative based on relative homology. Let Ω denote the ambient spatial domain of a scene, $X \subseteq \Omega$ the task-relevant foreground structure, and q a query specifying locations or a spatial region in Ω. We first formulate a general principle for query-dependent topological supervision using relative homology and then derive the degree-zero target used in our experiments.

3.1 Homology as a global topological descriptor

Homology provides coordinate-free summaries of topological structure. It associates with X a vector space[1] $H_d(X)$. Its dimension, the d-th Betti number $\beta_d(X) = \dim H_d(X)$, provides a numerical summary of the topology of X. In particular, $\beta_0(X)$ counts connected components, whereas $\beta_1(X)$ counts independent one-dimensional cycles. These quantities are invariant under homotopy equivalence. In the continuous setting with no discretization error, rotations, reflections, and uniform rescalings by a nonzero factor therefore leave them unchanged.

For a fixed scene X, a global topological target is constant across queries, whereas the relation specified by q can change. This motivates constructing a target jointly from X and q.

3.2 Query-dependent relative construction

Relative homology studies a space together with a designated reference subspace. For a pair (Z, R) with $R \subseteq Z$, the relative homology group $H_d(Z, R)$ treats chains contained entirely in R as trivial, so relative cycles may have boundaries in R. For the finite cell-complex pairs considered here, with nonempty R represented as a subcomplex of Z, $H_d(Z, R) \cong \widetilde{H}_d(Z/R)$ (Hatcher, 2002, Proposition 2.22), where Z/R is obtained by collapsing R to a single point and $\widetilde{H}_d$ denotes reduced homology: for a nonempty space, it agrees with ordinary homology in positive degrees and has degree-zero dimension $\beta_0 - 1$. We write $\beta_d(Z, R) := \dim H_d(Z, R)$ for the corresponding relative Betti number.

For a query q, we construct a task-specific pair (Z_q, R_q) with $R_q \subseteq Z_q$, and use a scalar derived from its relative Betti numbers as a query-dependent topological supervisory signal. Figure 1 provides a simple and motivating example. Consider two distinct query points $q = (q_1, q_2) \in X^2$ and let $R_q = \{q_1, q_2\}$. Write $q_1 \sim_X q_2$ when the two points belong to the same connected component of X. The change in the number of independent loops tells us whether the two query points are connected (see Appendix A for the derivation):

$$\beta_1(X, R_q) - \beta_1(X) = \begin{cases} 1, & q_1 \sim_X q_2, \\ 0, & q_1 \not\sim_X q_2. \end{cases} \tag{1}$$

Figure 1(b,d) gives the corresponding quotient-space intuition: identifying points in the same component creates one additional cycle, whereas identifying points in different components does not. Thus, the relative β_1 change distinguishes the queried relation while the foreground X remains fixed.

[1]Throughout, homology is computed with coefficients in a fixed field.

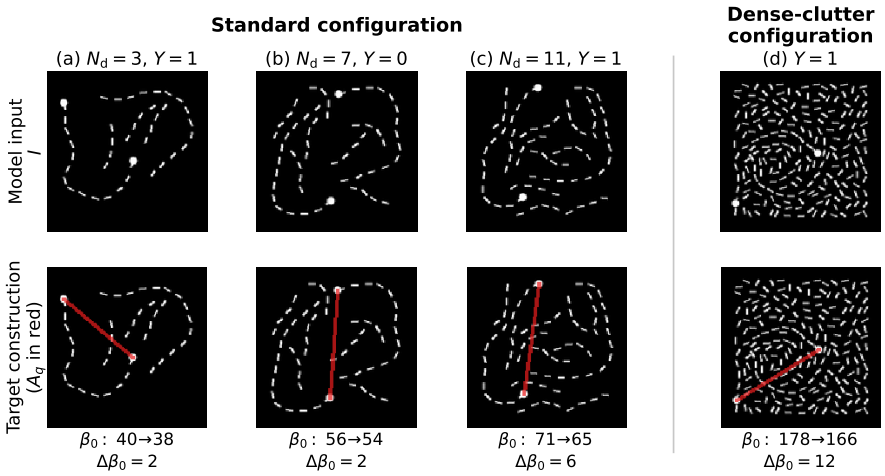


Figure 2: Panels (a–c) show standard configurations with different numbers of distractor paths and both same-path labels, while panel (d) shows the dense-clutter configuration. In each panel, the top image is the model input, whereas the bottom image overlays the query-induced segment A_q in red to visualise target construction. The segment is added to the rendered foreground X only when computing $\Delta\beta_0 = \beta_0(X) - \beta_0(X \cup A_q)$ and is never supplied to the model; the values below each column report $\beta_0(X) \to \beta_0(X \cup A_q)$ and the resulting $\Delta\beta_0$. Panels (a,b) have opposite same-path labels but the same relative target, whereas panels (a,c) have the same label but different relative targets, showing that $\Delta\beta_0$ is not a direct copy of Y.

More generally, the relative pair, homological degree, and scalar target can be chosen according to the task. These choices define a broader design space; all relative-supervision experiments in this work evaluate the single degree-zero segment construction defined below. In Pathfinder same-path recognition in Section 4 and retinal-vessel segmentation in Section 5, we choose the following construction. Define $A_q = \mathrm{conv}\{q_1, q_2\}$, the segment joining the queried locations, and the pair $(Z_q, R_q) = (X \cup A_q, A_q)$. We define the auxiliary target as

$$\Delta\beta_0(X; q_1, q_2) := \dim \widetilde{H}_0(X) - \dim H_0(X \cup A_q, A_q) = \beta_0(X) - \beta_0(X \cup A_q). \tag{2}$$

The second equality holds since A_q is nonempty and connected. This number has an intuitive interpretation: if A_q intersects $k \geq 1$ connected components of X, then $\Delta\beta_0 = k - 1 \geq 0$. This target need not determine the Pathfinder same-path label, as illustrated in Section 4. Appendix A derives the motivating connectivity identity and the experimental degree-zero target.

4 Experiments on two-dimensional Pathfinder

We use two-dimensional Pathfinder to test whether query-dependent relative topological supervision enables a plain feedforward convolutional network trained from random initialisation to learn long-range same-path recognition.

4.1 Task and dataset

Building on Pathfinder and Path-X (Linsley et al., 2018; Tay et al., 2021), we construct a two-dimensional same-path recognition benchmark with a controlled number of distractors. Each scene contains two target paths of 14 locally aligned segments and $N_d \in \{3, \ldots, 11\}$ shorter distractor paths, rendered as a single-channel 128×128 image I (see Figure 2). Two query markers are placed at target-path endpoints, and the binary task label is $Y = 1$ if and only if they belong to the same generated path. Because paths are rendered with gaps, this same-path relation is not identical to connectedness of the rendered foreground X. The generated path count $2 + N_d$ is used only to stratify the dataset splits.

Table 1: Supervisory-signal and self-pretraining controls on the plain feedforward convolutional stack (w=32) using 4,000 training scenes. Values are mean ± sample standard deviation of validation-selected test same-path accuracy over three seeds. The fourth column counts seeds exceeding 90% validation accuracy within 200 epochs.

Encoder init.	Auxiliary target	Test accuracy (%)	Successful runs
Random	No auxiliary signal	61.30 ± 6.53	$0/3$
	Global $\beta_0(X)$ control	50.93 ± 0.95	$0/3$
	Permuted relative $\Delta\beta_0$	50.67 ± 1.18	$0/3$
	Relative $\Delta\beta_0$ (ours)	$\mathbf{98.15 \pm 0.06}$	$\mathbf{3/3}$
Self-pretrained	No auxiliary signal	67.70 ± 26.62	$1/3$
	Relative $\Delta\beta_0$ (ours)	$\mathbf{98.19 \pm 0.32}$	$\mathbf{3/3}$

We construct a 32,000-scene training pool with fixed 900-scene validation and test sets. Primary experiments use the nested 4,000-scene subset, while scaling experiments use nested training subsets from 250 to 32,000 scenes. Appendix B details generation, splits, augmentation, and shortcut audits.

4.2 Relative target construction and learning setup

For each input image I, we detect the query locations q_1, q_2 using circular-template correlation combined with a foreground distance transform and spatial peak selection. We threshold I to obtain the rendered foreground X. Following equation 2, we rasterise the segment $\mathrm{conv}\{q_1, q_2\}$ as a finite-width capsule on the pixel grid, denote the resulting connector by A_q, and compute

$$\Delta\beta_0(X; q_1, q_2) = \beta_0(X) - \beta_0(X \cup A_q)$$

using 8-connected pixel adjacency. The target is derived entirely from I, without additional annotation or generator metadata; Appendix B gives the precise rasterisation rule. Figure 2 visualises this image-level construction.

Our default model is a five-stage plain feedforward convolutional stack (base channel width $w = 32$; 656,804 parameters; architecture details in Appendix C). A classification head predicts Y, while a second head predicts the non-negative relative target via Softplus. We optimise

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(Y, \widehat{Y}) + \lambda_\Delta \mathcal{L}_{\mathrm{Huber}}(\Delta\beta_0, \widehat{\Delta\beta_0}), \qquad \lambda_\Delta = 0.5. \tag{3}$$

The auxiliary head is discarded at inference.

Unless stated otherwise, primary plain-stack experiments use random initialisation and a fixed 200-epoch budget, with checkpoints selected by validation same-path accuracy (Appendix C.2).

4.3 Supervision and pretraining controls

Holding backbone, data, augmentation, number of downstream training epochs, and seeds fixed, Table 1 compares label-only training, global $\beta_0(X)$, permuted relative targets, and the proposed relative target, together with the matched label-only and relative conditions after self-pretraining on the same images. Here, self-pretraining uses masked image reconstruction on the same training rasters, followed by supervised fine-tuning of the encoder with newly initialised prediction heads; the reconstruction decoder is discarded. Appendix C.4 details the masking procedure, reconstruction architecture, and pretraining and fine-tuning protocols. At 4,000 scenes, relative supervision raises accuracy from $61.30 \pm 6.53\%$ to $98.15 \pm 0.06\%$ and succeeds in all three seeds; the global and permuted controls remain near chance, although global $\beta_0(X)$ is learned to a final-checkpoint test mean absolute error (MAE) of 0.731. After self-pretraining, label-only training reaches $67.70 \pm 26.62\%$ with one successful seed, whereas the relative target reaches $98.19 \pm 0.32\%$ with three, separating query-dependent supervision from an additional head or generic pretraining. Appendices D.1 and D.2 give the control definitions, per-seed results, and further analysis.

4.4 Data and model scaling

Backbone families and training-set size. With the relative target and random initialisation, the plain stack, the residual stack, and Swin-T all learn the task, so relative supervision is not tied to

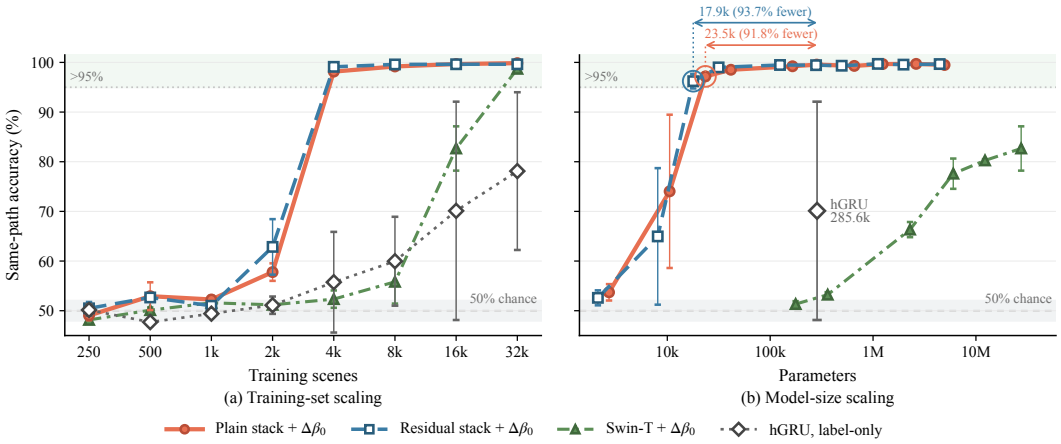


Figure 3: Training-set and model-size scaling. The plain stack, residual stack, and Swin-T use relative $\Delta\beta_0$ supervision; hGRU is a label-only recurrent reference (Appendix C.3). (a) Validation-selected test same-path accuracy against the number of training scenes. (b) The same accuracy against the number of parameters at 16,000 training scenes; hGRU is shown at its fixed reference size. The annotations compare the smallest plain and residual stacks exceeding 95% mean validation-selected test accuracy with the hGRU parameter count. Points and error bars report the mean and sample standard deviation over three seeds.

one architecture family (Figure 3a). The amount of data they need differs. On nested subsets, both convolutional stacks show a sharp transition between 2,000 and 4,000 training scenes, where their mean accuracies rise from below 63% to above 98%, whereas Swin-T reaches $82.67 \pm 4.46\%$ at 16,000 scenes and $98.70 \pm 0.17\%$ at 32,000. The hGRU is included as a label-only architectural reference and does not receive relative $\Delta\beta_0$ supervision; it is therefore not a fourth backbone in the matched relative-supervision comparison. The original study used 150×150 images and approximately 900,000 training examples per path-length condition, whereas our benchmark uses 128×128 scenes, varies the number of distractor paths as $N_d \in \{3, \ldots, 11\}$, and contains at most 32,000 training scenes. Under its prescribed training protocol, hGRU remains highly seed-sensitive on our benchmark, reaching $70.11 \pm 21.98\%$ at 16,000 scenes and $78.11 \pm 15.88\%$ at 32,000.

Model capacity. Because hGRU was introduced as a parameter-efficient recurrent solution to Pathfinder, Figure 3b also includes the label-only hGRU at its fixed reference size of 285,637 parameters. We vary capacity within each family while keeping the relative target and the 16,000-scene training split fixed and using the prescribed optimisation protocol for each family: the plain and residual stacks are scaled by their base channel width w, which sets every stage width, and Swin-T by its embedding dimension, with depth and number of attention heads reduced for the smaller variants. The smallest tested plain and residual stacks whose mean validation-selected test accuracy exceeds 95% contain 23,470 (w=6) and 17,881 (w=3) parameters, respectively, about one twenty-eighth of the parameter counts of the corresponding default models; smaller variants are inaccurate or unstable (Figure 3b). All tested Swin-T variants remain below 90%; the full model has 27.5M parameters, about 1,500× the smallest successful residual stack. Appendix D.3 reports the complete data- and model-scaling results.

Computational cost. At the 16,000-scene scale, using the same training split, seeds, batch size, precision, and 200-epoch budget, the recorded mean training time was 12.6 ± 0.8 minutes for the plain stack trained with relative supervision and 4.47 ± 0.07 hours for the label-only hGRU, corresponding to a 21.3× ratio in recorded mean wall-clock time (Appendix D.5).

Dense clutter. To test robustness under increased perceptual clutter, we also evaluate scenes with a dense field of additional isolated segments. On the dense-clutter configuration in Figure 2(d), whose rendered foreground contains $\beta_0(X) = 173.8 \pm 20.8$ components, the plain stack trained on 16,000 scenes reaches $95.15 \pm 0.39\%$ with the relative target and $56.52 \pm 11.00\%$ with labels alone over

three seeds; every relative run and no label-only run crosses 90% validation accuracy. Appendix D.4 provides the setup and per-seed results.

Together, these results support the importance of what the auxiliary objective asks the representation to preserve. Predicting global topology or permuted relative targets does not yield reliable same-path recognition, whereas the correctly paired relative target succeeds across seeds, with or without the tested reconstruction pretraining. The scaling results also show that this signal does not eliminate architectural differences: convolutional models succeed with substantially less data and fewer parameters than the tested Swin variants. Its benefit persists under increased clutter, supporting its usefulness beyond the standard scene configuration.

5 Application to biological imaging: Retinal vessel segmentation

We next test whether the same degree-zero segment construction transfers from synthetic same-path recognition to retinal-vessel segmentation in real fundus images. Retinal vessels form thin, branching networks in which small gaps can leave pixelwise overlap nearly unchanged while breaking vessel continuity and distorting branching structure (Shit et al., 2021).

Experimental setup. We evaluate on the DRIVE benchmark (Staal et al., 2004), using a $16/4$ split of its 20 training images for training and validation and the official 20-image test set for evaluation. Each full fundus image is resized to 512×512 without patch cropping. We train an ImageNet-initialised ResNet-18 U-Net for 60 epochs over three matched seeds; Appendix E gives the full protocol.

For auxiliary supervision, we construct training-only queries by sampling 50 pairs of vessel locations from the expert-annotated mask of each training image. For all four conditions, the sampled locations are rendered as query markers during training so that query presentation is held fixed across controls. The segment joining each pair defines the reference subspace and is used only to compute the relative $\Delta\beta_0$ target from the annotation mask; the segment itself is not supplied to the model. This construction requires no labels beyond the standard vessel-segmentation annotations.

We compare segmentation-only training with global (β_0, β_1) supervision, permuted relative targets, and relative $\Delta\beta_0$ supervision. All conditions share the same binary cross-entropy and soft-Dice segmentation objective. For the three auxiliary conditions, the auxiliary loss has weight 0.05. At inference, the auxiliary head is discarded and segmentation is produced from the clean, unmarked fundus image, with metrics computed inside the field-of-view mask.

We evaluate overlap with Dice, global topology with absolute Betti-number errors, and structural continuity with clDice, thin-centreline recall, and junction F_1. Thin-centreline recall is computed on ground-truth vessel centrelines with radius at most $1.5\,\mathrm{px}$, and junctions are matched within a $3\,\mathrm{px}$ tolerance. For paired tests, each metric is first averaged across the three matched seeds within each test image, and the 20 held-out images are treated as the independent units.

Results. Table 2 summarises segmentation and structural fidelity on the held-out DRIVE images. Relative $\Delta\beta_0$ supervision attains the best mean on all six reported metrics. Relative to segmentation-only training, it improves Dice by 0.20 percentage points, clDice by 0.70 points, thin-centreline recall by 2.00 points, and junction F_1 by 1.76 points, while reducing Betti-0 and Betti-1 errors by 2.13 and 2.52 per image, respectively.

The clDice improvement, Betti-1-error reduction, and gains in thin-centreline recall and junction F_1 each yield an unadjusted image-level $p < 0.002$ under the paired tests.

Neither control matches this pattern of mean improvements: global supervision reduces clDice and thin-centreline recall relative to segmentation-only training, whereas the permuted target increases Betti-0 error and yields smaller improvements in the remaining structural metrics than relative supervision.

Together, these results provide real-image evidence that the benefit of this degree-zero construction is not restricted to synthetic same-path recognition. Figure 4 visualises a representative held-out example. Additional metrics and test details are provided in Appendix E.

Table 2: DRIVE results using a ResNet-18 U-Net. Dice, clDice, thin-centreline (Thin CL) recall, and junction F_1 are reported as percentages; Betti errors are absolute Betti-number errors. Values are mean $\pm$ sample standard deviation across three matched seeds. Bold indicates the best mean, and † denotes a two-sided paired Wilcoxon signed-rank test $p < 0.05$ against segmentation-only training.

Objective	Dice ↑	clDice ↑	Betti-0 error ↓	Betti-1 error ↓	Thin CL recall ↑	Junction F1 ↑
Segmentation only	79.95±0.23	80.97±0.10	40.13±3.73	18.27±1.82	62.13±1.43	61.95±2.09
Global (β_0, β_1)	79.96±0.26	80.75±0.48	40.23±6.30	18.08±1.19	61.11±0.29	62.66±1.31
Permuted $\Delta\beta_0$	80.05±0.12	81.34±0.06	41.18±4.31	17.07±1.01	62.77±3.68	62.40±2.75
Relative $\Delta\beta_0$	**80.15±0.23**	**81.67±0.37**†	**38.00±1.00**	**15.75±1.17**†	**64.13±3.13**†	**63.71±2.13**†

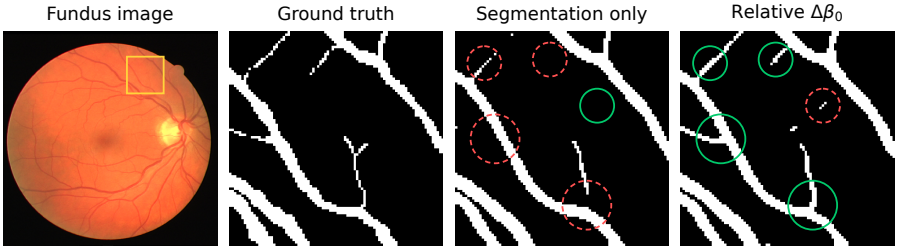


Figure 4: A qualitative example from the held-out DRIVE test set. From left to right: the full fundus image with the selected region, the reference vessel mask, the segmentation-only prediction, and the prediction trained with relative $\Delta\beta_0$ supervision. Green circles mark ground-truth centreline fragments recovered only with relative supervision; red dashed circles mark fragments recovered only with segmentation-only training. Table 2 summarises results over all 20 test images.

6 Conclusion and Future Directions

Long-range dependency is a question not only of architecture but also of what a network is taught to preserve. A global invariant of a fixed scene is constant across queries; relativising topology to a query-defined subspace instead yields a training-only $\Delta\beta_0$ target that enables a plain feedforward convolutional network to learn two-dimensional Pathfinder without specialised long-range modules. Matched controls support the role of query-dependent content, while backbone, data-, and capacity-scaling experiments identify regimes in which the signal is effective or insufficient. The same degree-zero segment construction improves retinal-vessel structural fidelity without changing inference, showing that its benefit transfers across the two evaluated domains.

Our experiments focus on a segment-based degree-zero target. Extending the framework to other query-defined subspaces and higher homological degrees, for example to characterise regions separated by planar networks, is a natural direction for future work.

Reproducibility Statement

Section 3 defines the target of Equation 2 and Section 4 states the task, target construction, learning objective, and model-selection protocol. Appendix B gives dataset construction, the exact-symmetry verification, and the integrity and shortcut audits; Appendix C gives architectures and parameter counts; Appendix C.2 provides the complete optimisation settings, including architecture- and initialisation-specific learning rates. Appendix C.3 documents the hGRU reimplementation and its architecture-specific optimisation settings. Every reported number is an aggregate over per-seed run records; per-seed values are given for the headline conditions. Section 6 discusses extensions to other relative constructions. Anonymised source code for generation, training and evaluation is included in the supplementary material.

Ethics Statement

This work uses synthetic images generated by the procedure described in Section 4 and publicly released, de-identified retinal fundus images from DRIVE (Staal et al., 2004), used under the dataset's

research terms. We collected no new human-participant data. We see no direct dual-use concern. We adhere to the ICLR Code of Ethics.

AI USE STATEMENT

In this work, we used generative AI tools (LLM-based coding and writing assistants) for tasks with required disclosure: assisting in the design of experiments and controls, cleaning and reformatting experimental metadata. We also used them for tasks with recommended disclosure: implementing and editing software code, drafting and revising text, summarising related literature, and creating scientific figures. We did not use them to generate the synthetic datasets themselves (which are produced by a deterministic procedural generator written for this work), to formulate mathematical derivations, or for translation. We have reviewed all AI-assisted work: every numerical value reported in this paper is derived from stored per-run records independently of the drafting process, and implementation code was verified by targeted audits described in Appendix B and C.3. The authors take full responsibility for the final content of this work, including text, claims, and artefacts.

REFERENCES

Ido Amos, Jonathan Berant, and Ankit Gupta. Never train from scratch: Fair comparison of long-sequence models requires data-driven priors. In *International Conference on Learning Representations (ICLR)*, 2024.

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. In *International Conference on Artificial Intelligence and Statistics*, pp. 2786–2796. PMLR, 2020.

James R. Clough, Nicholas Byrne, Ilkay Oksuz, Veronika A. Zimmer, Julia A. Schnabel, and Andrew P. King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12): 8766–8778, 2022.

Omar Coser, Loredana Zollo, Paolo Soda, and Antonio Orvieto. Towards understanding self-pretraining for sequence classification. In *ICML 2025 Workshop on Methods and Opportunities at Small Scale (MOSS)*, 2025.

Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations (ICLR)*, 2022.

Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. *Advances in Neural Information Processing Systems*, 30, 2017.

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Jungeun Kim, Junwon You, Dongjin Lee, Ha Young Kim, and Jae-Hun Jung. Do topological characteristics help in knowledge distillation? In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24674–24693. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/kim24aj.html`.

Junkyung Kim, Drew Linsley, Kalpit Thakkar, and Thomas Serre. Disentangling neural mechanisms for perceptual grouping. In *International Conference on Learning Representations (ICLR)*, 2020a.

Kwangho Kim, Jisu Kim, Manzil Zaheer, Joon Kim, Frederic Chazal, and Larry Wasserman. PLLay: Efficient topological layer based on persistent landscapes. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15965–15977. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/b803a9254688e259cde2ec0361c8abe4-Paper.pdf.

Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Yuhong Li, Tianle Cai, Yi Zhang, Deming Chen, and Debadeepta Dey. What makes convolutional models great on long sequence modeling? In *International Conference on Learning Representations (ICLR)*, 2023.

Drew Linsley, Junkyung Kim, Vijay Veerabadran, Charles Windolf, and Thomas Serre. Learning long-range spatial dependencies with horizontal gated recurrent units. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Drew Linsley, Junkyung Kim, Alekh Ashok, and Thomas Serre. Recurrent neural circuits for contour detection. In *International Conference on Learning Representations (ICLR)*, 2020.

Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the CoordConv solution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *International Conference on Machine Learning*, pp. 7045–7054. PMLR, 2020.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.

David W. Romero, David M. Knigge, Albert Gu, Erik J. Bekkers, Efstratios Gavves, Jakub M. Tomczak, and Mark Hoogendoorn. Towards a general purpose CNN for long range dependencies in ND. *arXiv preprint arXiv:2206.03398*, 2022.

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

Suprosanna Shit, Johannes C. Paetzold, Anjany 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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Jimmy T.H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Joes Staal, Michael D. Abramoff, 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.

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 (ICML)*, pp. 32698–32727, 2023.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long Range Arena: A benchmark for efficient Transformers. In *International Conference on Learning Representations (ICLR)*, 2021.

Adam Van Etten, Dave Lindenbaum, and Todd M. Bacastow. SpaceNet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01404*, 2018.

SUPPLEMENTARY MATERIAL

A DERIVATIONS FOR THE RELATIVE CONSTRUCTIONS

All homology groups are taken over the fixed coefficient field specified in Section 3.

Pairwise connectivity. Let $R = \{q_1, q_2\} \subset X$, where q_1 and q_2 are distinct. Because $H_1(R) = 0$ and $H_0(R) \cong \mathbb{F}^2$, the long exact sequence of the pair (X, R) contains

$$0 \longrightarrow H_1(X) \longrightarrow H_1(X, R) \longrightarrow H_0(R) \xrightarrow{\iota_*} H_0(X).$$

Exactness gives

$$\beta_1(X, R) - \beta_1(X) = \dim \ker \iota_*.$$

If q_1 and q_2 lie in the same connected component of X, their two generators in $H_0(R)$ map to the same component class in $H_0(X)$, so $\dim \ker \iota_* = 1$. If they lie in different components, the two images are independent and $\dim \ker \iota_* = 0$. This proves Equation equation 1.

Degree-zero target. For the experimental construction, let $Z_q = X \cup A_q$ and $R_q = A_q$, where A_q is nonempty and connected and intersects X. The degree-zero part of the long exact sequence gives

$$H_0(A_q) \longrightarrow H_0(Z_q) \longrightarrow H_0(Z_q, A_q) \longrightarrow 0.$$

The image of $H_0(A_q)$ is one-dimensional, and hence

$$\dim H_0(X \cup A_q, A_q) = \beta_0(X \cup A_q) - 1.$$

Together with $\dim \widetilde{H}_0(X) = \beta_0(X) - 1$, this yields

$$\dim \widetilde{H}_0(X) - \dim H_0(X \cup A_q, A_q) = \beta_0(X) - \beta_0(X \cup A_q),$$

which establishes the relative-homology interpretation of the target in Equation equation 2. If A_q intersects k connected components of X, its addition merges those k components into one, so $\beta_0(X \cup A_q) = \beta_0(X) - k + 1$ and $\Delta\beta_0 = k - 1 \geq 0$.

B PATHFINDER TASK, CORPUS, AND SHORTCUT AUDITS

Generation and task. Each scene contains two target paths of 14 locally aligned segments and $N_{\mathrm{d}} \in \{3, \ldots, 11\}$ shorter distractor paths, rendered on a single-channel 128×128 canvas. Paths are drawn with a stroke width of $1.5\,\mathrm{px}$, and the query markers are discs of radius $3\,\mathrm{px}$ placed at target-path endpoints. The same-path label is $Y = 1$ if and only if the two markers belong to the same generated path. A blank border of at least $10\,\mathrm{px}$ prevents the geometric augmentation from clipping the rendered foreground.

Corpus and nested subsets. The standard-configuration corpus contains a 32,000-scene training pool and fixed validation and test sets of 900 scenes each. The training subsets are strictly nested:

$$\mathcal{S}_{250} \subset \mathcal{S}_{500} \subset \mathcal{S}_{1000} \subset \mathcal{S}_{2000} \subset \mathcal{S}_{4000} \subset \mathcal{S}_{8000} \subset \mathcal{S}_{16000} \subset \mathcal{S}_{32000}.$$

The primary experiments use $\mathcal{S}_{4000}$, whereas the scaling experiments use the complete nested sequence. The primary subset and both held-out sets are exactly balanced within every generated path-count stratum. In the full training pool, the positive and negative counts differ by at most two scenes within any stratum. All standard-configuration experiments use the same broken-line rendering with visible query markers.

Generated paths and raster components. Because each path is rendered as separated segments, the number of generated paths is not the number of connected components in the raster. The generated path count is

$$2 + N_{\mathrm{d}} \in [5, 13],$$

whereas the rendered foreground satisfies

$$\beta_0(X) \in [38, 72]$$

over the full corpus. Writing $b = 2 + N_{\mathrm{d}}$, the observed raster count is $4b + 20$ on 97.32% of scenes, $4b + 19$ on 2.65%, and $4b + 18$ on 0.03%. The global control in Table 1 predicts the pixel-level component count $\beta_0(X)$, not the generated path count b. Although the two quantities are strongly associated in this corpus, they should not be identified.

Relative-target computation. The relative target is computed from the input image I rather than from the latent generated paths. We obtain the rendered foreground X by thresholding the normalised intensity of I at 0.1, and compute

$$\Delta\beta_0(X; q_1, q_2) = \beta_0(X) - \beta_0(X \cup A_q).$$

Here, the query locations q_1 and q_2 are the centres of the query markers detected in I. The query-induced subspace A_q is rasterised at native resolution as the analytic capsule of radius $1.5\,\mathrm{px}$ around the closed segment joining these centres: a pixel is included if its centre lies within $1.5\,\mathrm{px}$ of the segment. The capsule is evaluated directly at pixel centres, without supersampling, interpolation, or subsequent thresholding. Both $\beta_0(X)$ and $\beta_0(X \cup A_q)$ are computed using 8-connected foreground labelling. Across all $33{,}800$ standard-configuration scenes, $\Delta\beta_0$ lies in $[0, 12]$ with median 4. The coordinate-derived and image-derived target fields agree on all scenes; every stored target is non-negative and equals the corresponding before–after component-count difference.

Integrity checks. A metadata-level audit covers all $33{,}800$ standard-configuration scenes and verifies the split sizes, strict nesting of all training subsets, absence of sample-identifier overlap between splits, label balance, and the stored before–after target identity. A separate raster-level audit covers $17{,}800$ scenes—the nested $16{,}000$-scene training subset and the complete 900-scene validation and test sets—and does not import the target renderer. Independently recomputed same-path labels, component counts, and relative targets agree with the stored values on all $17{,}800$ scenes (100%); the recovered marker centres lie within $1.5\,\mathrm{px}$ of the stored locations in every case. Exchanging the two query locations preserves both the capsule and the target on 100% of the audited scenes. Under the dihedral transformations used for augmentation, the capsule is equivariant and the target is invariant on 100% of the audited scenes.

Shortcut checks. The scalar baselines evaluated here achieve at most 52% test accuracy. The best train-fitted threshold on the distance between the queried points scores 51.11%; on the generated path count, 50.00%; on the amount of foreground ink, 49.78%; on the compressed file size, 50.11%. The auxiliary target itself is not a shortcut to the label: the best threshold on $\Delta\beta_0$ alone reaches 51.56% and a per-value majority lookup 51.44%, and a logistic regression on all six scalars together, $\Delta\beta_0$ included, reaches 52.00%.

Augmentation. All standard experiments apply a uniformly sampled element of the dihedral group D_4, followed by an integer translation of at most $8\,\mathrm{px}$. The transformations are applied by padding and cropping after tensor conversion, without interpolating the $1.5\,\mathrm{px}$ strokes. The enforced $10\,\mathrm{px}$ border margin prevents clipping. These transformations preserve the same-path label, the component counts, and the relative target. Because the target is invariant under augmentation, it is computed once and cached rather than recomputed during training.

C Pathfinder architectures and training protocols

C.1 Feedforward backbones

Plain stack. The plain backbone comprises five double-convolution stages with output channel widths $w, 2w, 2w, 4w, 4w$, where w denotes the base channel width. Each stage applies a 3×3 stride-2 convolution followed by a 3×3 stride-1 convolution, with batch normalisation and ReLU after each convolution, reducing the spatial resolution as $128^2 \to 64^2 \to 32^2 \to 16^2 \to 8^2 \to 4^2$. Global average pooling averages each channel over the final 4×4 feature map, yielding a feature vector of dimension $4w$, which is passed to the prediction heads used by each experimental condition. The backbone contains no residual connections, recurrence, or attention. We use $w = 32$ by default, giving $656{,}804$ parameters in the reported implementation; complete parameter counts for the capacity sweep are given in Table 5.

Residual stack. The residual backbone begins with a 5×5 stride-2 convolution, followed by batch normalisation and ReLU, which reduces the 128×128 input to 64×64 with w channels. Four basic residual blocks, each containing two 3×3 convolutions, operate at this resolution. Three subsequent stages reduce the spatial resolution to 32^2, 16^2, and 8^2, with output channel widths $2w$, $4w$, and $6w$,

respectively. Each stage contains two residual blocks; its first block performs stride-2 downsampling and uses a 1×1 projection on the skip path when required. Global average pooling over the final 8×8 feature map yields a feature vector of dimension $6w$, which is passed to the prediction heads used by each experimental condition. Thus, eight residual-branch convolutions operate at 64×64 before further downsampling, providing substantially more high-resolution processing than the plain stack. We use $w = 16$ by default, giving $495{,}540$ parameters in the reported implementation; complete parameter counts for the capacity sweep are given in Table 5.

ImageNet-initialised backbones. We additionally evaluate single-channel adaptations of TorchVision ResNet-18 ($11{,}171{,}779$ parameters) and Swin-T ($27{,}518{,}589$ parameters), initialised from the default ImageNet-1K weights. For ResNet-18, the first convolution is replaced by a single-channel layer whose weights are initialised by averaging the pretrained RGB kernels across the input-channel dimension. For Swin-T, the same averaging procedure is applied to the patch-embedding projection, with its pretrained bias retained. All remaining backbone weights are preserved, while the original ImageNet classification layer is replaced by randomly initialised task heads.

C.2 Optimisation and model selection

Learning-rate assignment. The randomly initialised plain and residual stacks use AdamW with base learning rate 10^{-3}, whereas the ImageNet-initialised ResNet-18 and Swin-T models use 10^{-4}. Within each architecture and initialisation setting, the learning rate is held fixed across all supervisory conditions and is not tuned separately for the proposed target or its controls. These experiments use weight decay 10^{-4}, a five-epoch linear warmup, and cosine annealing to 10^{-6}. The optimisation settings used for Swin-T from random initialisation are described below.

Swin-T from random initialisation. All randomly initialised Swin-T models use AdamW with learning rate 10^{-4}. Weight decay 0.05 is applied to two-dimensional weight tensors, while norm, bias, and relative-position-bias parameters are excluded from weight decay. We use a ten-epoch linear warmup followed by cosine annealing to 10^{-6}, gradient clipping at 1.0, and no stochastic depth. This protocol is fixed across all Swin-T training-set and capacity-scaling experiments.

Training budget and checkpoint selection. Unless stated otherwise, Pathfinder models are trained for a fixed budget of 200 epochs without early stopping. We avoid patience-based early stopping because some runs exhibit delayed transitions from chance-level performance. For each run, we select the checkpoint with the highest validation same-path accuracy and report the corresponding test performance. All conditions within a matched comparison use the same training budget and checkpoint-selection rule.

C.3 Recurrent reference baseline

Reimplementation. We reimplemented the hGRU cell from the reference code released by Linsley et al. (2018). In a direct unit test with identical inputs and weights, the reference and our implementation produced identical cell outputs (maximum absolute difference 0). Our implementation also matches the reference parameter count of $285{,}637$. We additionally verified that all trainable parameters receive finite, non-zero gradients and that the reference kernel-symmetrisation constraint is applied during optimisation.

Architecture and reference initialisation. Our single-layer hGRU uses 25 feature channels, 15×15 horizontal kernels, and $T = 8$ recurrent steps. The 7×7 input convolution is initialised with the reference Gabor filters and remains trainable. Following Linsley et al. (2018), the inhibitory and excitatory horizontal kernels $(w_{\mathrm{inh}}, w_{\mathrm{exc}})$ and the 1×1 gate weights (u_1, u_2) are orthogonally initialised. The trainable interaction coefficients are initialised as $\alpha = 0.1$, $\gamma = 1.0$, $\kappa = 0.5$, $\omega = 0.5$, and $\mu = 1.0$. The gate biases use Chronos initialisation,

$$b_1 \sim \log\big(\mathrm{Unif}(1, T-1)\big), \qquad b_2 = -b_1,$$

which distributes the initial gate time constants across the recurrent window.

Optimisation. Preliminary runs showed that hGRU was sensitive to the optimisation settings: AdamW with weight decay and cosine scheduling led to chance-level predictions. We therefore train the label-only hGRU reference with NAdam at a constant learning rate of 10^{-3}, without weight decay or a learning-rate scheduler. We use a batch size of 32, gradient clipping at 1.0, and bfloat16 precision; float16 is avoided for numerical stability through the unrolled recurrence. These architecture-specific settings are fixed across all hGRU data-scale experiments.

C.4 Self-pretraining protocol

Previous work has shown that self-pretraining on the downstream training data, using masked or autoregressive sequence-prediction objectives, can substantially improve performance on long-range sequence benchmarks, including Pathfinder, without requiring external data (Amos et al., 2024; Coser et al., 2025). Motivated by these results, we adapt the same training-only principle to our two-dimensional setting using image-space masked reconstruction. This control tests whether generic reconstruction pretraining on the same training rasters can account for, or complement, the benefit of relative topological supervision.

Architecture and masking. The masked autoencoder uses the same five-stage plain-stack encoder as the downstream model, with width $w = 32$, no residual connections, and $656{,}288$ encoder parameters. A five-stage U-Net-style decoder bilinearly upsamples the 4×4 bottleneck representation through spatial resolutions 8×8, 16×16, 32×32, 64×64, and 128×128. At the first four resolutions, the upsampled representation is concatenated with the corresponding encoder feature map and processed by two 3×3 convolution–batch-normalisation–ReLU layers. The decoder contains $757{,}025$ parameters, giving $1{,}413{,}313$ parameters for the complete autoencoder. Only the encoder is transferred to downstream models.

Input rasters are normalised to $[-1, 1]$ and partitioned into non-overlapping 16×16-pixel patches, giving an 8×8 grid of 64 patches. For each scene, 32 of the 64 patches are sampled uniformly without replacement and replaced by the normalised background value -1. The reconstruction loss is the mean squared error evaluated only at masked pixel locations. A final 1×1 convolution followed by $\tanh$ constrains the reconstruction to the normalised input range.

Pretraining protocol. For each seed, the autoencoder is pretrained exclusively on the same $4{,}000$ training rasters used for downstream learning, under the standard broken-line rendering with query markers. Class labels and auxiliary targets are not used, and no external data are introduced. The same dihedral transformations and integer translations used in the downstream experiments are applied before masking. Pretraining runs for 200 epochs with batch size 32 using AdamW ($\mathrm{lr} = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 10^{-4}), a five-epoch linear warmup, and cosine annealing to 10^{-6}. Optimisation is performed in single precision. The final encoder checkpoint from each seed is used to initialise the corresponding downstream model, while the decoder is discarded.

Downstream fine-tuning and comparison design. After pretraining, the decoder is discarded and the encoder weights are used to initialise the downstream plain-stack model. All encoder parameters are then fine-tuned jointly with the task head or heads for 200 epochs. We compare the four combinations of $\{\text{random}, \text{self-pretrained}\}$ initialisation and $\{\text{label-only}, \text{relative topological}\}$ supervision. All conditions use the same $4{,}000$-scene training split, validation and test sets, augmentation, training budget, and nominal AdamW settings. The relative auxiliary loss is assigned weight $\lambda_\Delta = 0.5$ when enabled and zero otherwise.

D Complete Pathfinder results and analysis

D.1 Supervisory-signal controls

We compare relative topological supervision with three controls designed to test the importance of the auxiliary signal's content:

1. **label-only**: the same-path label is used without an auxiliary loss, so $\lambda_\Delta = 0$.

Table 3: Per-seed validation-selected test same-path accuracy (%) for supervisory-signal and self-pretraining controls on the plain stack ($w = 32$) with $4{,}000$ training scenes. A successful run exceeds 90% validation accuracy within the 200-epoch downstream-training budget.

Encoder init.	Supervision	Seed 23	Seed 7	Seed 42	Mean ± sample SD	Success
Random	Label-only	53.78	64.56	65.56	61.30 ± 6.53	$0/3$
	Global $\beta_0(X)$	50.00	50.89	51.89	50.93 ± 0.95	$0/3$
	Permuted relative $\Delta\beta_0$	49.78	50.22	52.00	50.67 ± 1.18	$0/3$
	Relative $\Delta\beta_0$	**98.22**	**98.11**	**98.11**	$\mathbf{98.15 \pm 0.06}$	$\mathbf{3/3}$
Self-pretrained	Label-only	52.22	98.44	52.44	67.70 ± 26.62	$1/3$
	Relative $\Delta\beta_0$	**98.00**	**98.00**	**98.56**	$\mathbf{98.19 \pm 0.32}$	$\mathbf{3/3}$

2. **global** $\beta_0(X)$: the auxiliary head predicts the query-independent raster component count $\beta_0(X)$. This condition tests whether learning a global topological invariant is sufficient without relating that invariant to the query.
3. **permuted relative** $\Delta\beta_0$: the relative targets in the training split are reassigned among training images using a deterministic seed-specific permutation. This preserves the empirical target distribution, auxiliary head, and loss term while destroying the correct image–target pairing. Validation and test targets are not permuted.
4. **relative** $\Delta\beta_0$: the proposed query-dependent target defined in §3.

The four random-initialisation conditions use the same $4{,}000$-scene split, plain-stack backbone, augmentation, training budget, and nominal optimisation settings. Their aggregate results are reported in Table 1, and Table 3 provides the per-seed results together with the two self-pretraining conditions.

The label-only condition varies substantially across seeds and never reaches the success threshold. The global control learns its auxiliary target to a final test MAE of 0.7306 ± 0.1779, yet same-path accuracy remains at chance. Permuting the relative targets likewise leaves performance at chance, showing that the additional head and loss term are not sufficient by themselves. Only the correctly paired relative target succeeds across all three seeds.

D.2 Self-pretraining results

Table 3 reports the per-seed results for the self-pretrained conditions alongside their randomly initialised counterparts. Self-pretraining does not make label-only training reliable: only one of three runs reaches the success threshold, yielding $67.70 \pm 26.62\%$ validation-selected test accuracy. With relative topological supervision, all three runs succeed from both random and self-pretrained initialisation, reaching $98.15 \pm 0.06\%$ and $98.19 \pm 0.32\%$, respectively. Thus, within the measured setting, self-pretraining neither reliably substitutes for the query-aligned target nor provides an evident test-accuracy improvement once relative supervision is present. The mean epoch at which validation accuracy first exceeds 90% is 97.7 from random initialisation and 91.0 after self-pretraining.

D.3 Complete data- and model-scaling results

Figure 3 summarises the principal training-set and model-size scaling trends. Table 4 reports the complete training-set sweep from 250 to 32,000 scenes, and Table 5 reports the complete model-size sweep with the training-set size fixed at 16,000 scenes. The latter table includes the Swin-T capacity sweep and the label-only hGRU reference shown at its fixed parameter count in the main figure.

Training-set scaling. The two convolutional stacks exhibit a sharp transition between $2{,}000$ and $4{,}000$ training scenes. Their mean validation-selected test accuracies increase from 57.78% and 62.85%, respectively, at $2{,}000$ scenes to 98.15% and 99.11% at $4{,}000$, with all three runs of each architecture exceeding 90% validation accuracy at the larger scale. Swin-T trained from random initialisation requires substantially more data, reaching $82.67 \pm 4.46\%$ at $16{,}000$ scenes and $98.70 \pm 0.17\%$ at $32{,}000$. The label-only hGRU reference remains highly variable, reaching $70.11 \pm 21.98\%$ at $16{,}000$ scenes and $78.11 \pm 15.88\%$ at $32{,}000$, with only one of three runs crossing the validation threshold at either scale.

Table 4: Complete training-set scaling results on the standard broken-line rendering with query markers. The plain and residual stacks use relative $\Delta\beta_0$ supervision ($\lambda_\Delta = 0.5$) with AdamW at learning rate 10^{-3}, a five-epoch warmup, and cosine decay to 10^{-6}. Swin-T uses the scratch-training protocol described in Appendix C.2. The hGRU reference uses label-only training with NAdam at learning rate 10^{-3}, without weight decay or a learning-rate scheduler. All runs use a 200-epoch budget, batch size 32, bfloat16 precision, and seeds $23, 7, 42$. Values are mean $\pm$ sample standard deviation. "Test (sel.)" denotes test accuracy at the checkpoint selected by validation same-path accuracy. The final column reports the mean first epoch exceeding 90% validation accuracy; a parenthetical count is shown when fewer than three runs reach the threshold.

Architecture	**Train scenes**	**Test (sel.) (%)**	**Test (final) (%)**	**First $\geq 90\%$ val.**
Plain stack ($w = 32$, 656,804)	250	49.04 ± 0.94	50.70 ± 0.28	never
	500	52.93 ± 2.79	54.81 ± 2.52	never
	1,000	52.26 ± 0.55	52.96 ± 1.85	never
	2,000	57.78 ± 1.76	57.74 ± 0.71	never
	4,000	98.15 ± 0.06	98.33 ± 0.11	97.7
	8,000	99.19 ± 0.17	99.41 ± 0.06	47.7
	16,000	99.67 ± 0.38	99.85 ± 0.06	25.7
	32,000	99.85 ± 0.06	99.89 ± 0.11	14.3
Residual stack ($w = 16$, 495,540)	250	50.48 ± 1.28	51.30 ± 1.56	never
	500	52.67 ± 0.78	53.33 ± 1.49	never
	1,000	51.00 ± 0.11	49.96 ± 0.55	never
	2,000	62.85 ± 5.59	63.11 ± 4.94	never
	4,000	99.11 ± 0.22	99.22 ± 0.00	61.3
	8,000	99.59 ± 0.17	99.63 ± 0.13	31.7
	16,000	99.63 ± 0.06	99.81 ± 0.06	18.7
	32,000	99.63 ± 0.13	99.81 ± 0.06	11.3
Swin-T (random, 27.5M)	250	48.15 ± 0.34	48.04 ± 0.38	never
	500	50.11 ± 0.56	50.15 ± 0.81	never
	1,000	51.59 ± 1.41	50.81 ± 1.44	never
	2,000	51.19 ± 0.76	51.70 ± 1.67	never
	4,000	52.33 ± 1.75	53.41 ± 1.41	never
	8,000	55.81 ± 4.38	56.33 ± 5.57	never
	16,000	82.67 ± 4.46	82.41 ± 4.72	never
	32,000	98.70 ± 0.17	98.93 ± 0.28	77.7
hGRU (285,637, label-only)	250	50.15 ± 1.14	50.44 ± 0.78	never
	500	47.70 ± 0.78	48.41 ± 2.33	never
	1,000	49.41 ± 0.67	50.78 ± 0.29	never
	2,000	51.11 ± 1.75	50.15 ± 0.46	never
	4,000	55.74 ± 10.14	55.67 ± 9.14	never
	8,000	59.96 ± 7.31	59.70 ± 6.40	never
	16,000	70.11 ± 21.98	67.85 ± 23.31	86.0 (1/3)
	32,000	78.11 ± 15.88	78.04 ± 15.85	50.0 (1/3)

Table 5: Complete model-size scaling results with the training-set size fixed at 16,000 scenes. The plain, residual, and randomly initialised Swin-T models use relative $\Delta\beta_0$ supervision with $\lambda_\Delta = 0.5$; Swin-T uses the scratch-training protocol described in Appendix C.2. The hGRU row is a label-only recurrent reference. Values are mean $\pm$ sample standard deviation over seeds 23, 7, 42, and "Test (sel.)" denotes test accuracy at the validation-selected checkpoint. The default-width plain and residual models were independently rerun as part of the capacity sweep rather than reused from the training-set sweep.

Model	**Width**	**Parameters**	**Test (sel.) (%)**	**Test (final) (%)**	**First $\geq 90\%$ val.**
Plain stack	$w = 2$	2,714	53.70 ± 1.65	52.48 ± 0.67	never
	$w = 4$	10,536	74.04 ± 15.43	74.96 ± 14.02	never
	$w = 6$	23,470	97.19 ± 0.78	97.30 ± 0.71	91.7
	$w = 8$	41,516	98.52 ± 0.23	98.78 ± 0.19	55.3
	$w = 16$	164,820	99.22 ± 0.40	99.70 ± 0.06	34.3
	$w = 21$	283,420	99.56 ± 0.22	99.59 ± 0.13	29.7
	$w = 32$	656,804	99.30 ± 0.23	99.74 ± 0.06	25.3
	$w = 44$	1,240,496	99.67 ± 0.00	99.70 ± 0.13	24.7
	$w = 64$	2,622,276	99.70 ± 0.23	99.93 ± 0.06	23.7
	$w = 88$	4,955,196	99.48 ± 0.32	99.81 ± 0.06	23.7
Residual stack	$w = 1$	2,115	52.59 ± 1.51	51.89 ± 1.28	never
	$w = 2$	8,074	64.96 ± 13.75	65.70 ± 14.22	never
	$w = 3$	17,881	96.22 ± 1.47	96.56 ± 1.07	100.7
	$w = 4$	31,536	99.00 ± 0.33	99.30 ± 0.36	38.7
	$w = 8$	124,636	99.48 ± 0.17	99.56 ± 0.11	21.7
	$w = 12$	279,304	99.44 ± 0.22	99.74 ± 0.06	20.3
	$w = 16$	495,540	99.33 ± 0.44	99.78 ± 0.11	19.0
	$w = 24$	1,112,716	99.70 ± 0.06	99.78 ± 0.00	19.0
	$w = 32$	1,976,164	99.56 ± 0.22	99.93 ± 0.06	16.7
	$w = 48$	4,441,876	99.67 ± 0.11	99.81 ± 0.13	19.7
Swin-T (random)	12	176,823	51.33 ± 0.59	50.78 ± 1.35	never
	16	362,286	53.22 ± 0.51	52.74 ± 2.12	never
	32	2,287,295	66.33 ± 1.53	66.19 ± 2.41	never
	48	6,020,469	77.59 ± 3.05	78.59 ± 3.14	never
	64	12,250,879	80.26 ± 0.26	80.63 ± 1.51	never
	96	27,518,589	82.67 ± 4.46	82.41 ± 4.72	never
hGRU (label-only)	—	285,637	70.11 ± 21.98	67.85 ± 23.31	86.0 (1/3)

Model-size scaling. With $16{,}000$ training scenes, the smallest tested plain and residual stacks whose mean validation-selected test accuracy exceeds 95% contain $23{,}470$ and $17{,}881$ parameters, respectively. These models contain 91.8% and 93.7% fewer parameters than the $285{,}637$-parameter hGRU reference. The immediately smaller variants are both less accurate and substantially more variable across seeds, indicating an architecture-dependent capacity transition. Above approximately 10^5 parameters, both convolutional families remain near 99% across the tested widths. By contrast, randomly initialised Swin-T improves from $51.33 \pm 0.59\%$ with $176{,}823$ parameters to $82.67 \pm 4.46\%$ with 27.5 million parameters, and no tested width exceeds the 90% validation threshold at this data scale. Thus, relative supervision supports highly parameter-efficient convolutional models but is not by itself sufficient in every low-capacity or backbone regime.

Effect of pretrained initialisation. With relative $\Delta\beta_0$ supervision and $4{,}000$ training scenes, ImageNet-initialised ResNet-18 and Swin-T reach $99.04 \pm 0.06\%$ and $98.81 \pm 0.17\%$, respectively. Under the same relative target and data scale, Swin-T trained from random initialisation reaches only $52.33 \pm 1.75\%$. Initialisation therefore has a substantial effect on the sample efficiency of this backbone even when the supervisory target is held fixed.

D.4 Dense-clutter stress test

We additionally evaluate the higher-clutter configuration illustrated in Figure 2(d). Each 128×128 scene contains two target paths, eight distractor paths, and a dense field of isolated short segments. The resulting broken-line foreground contains $\beta_0(X) = 173.8 \pm 20.8$ connected components on average, substantially increasing perceptual clutter relative to the standard configuration. The corpus contains $16{,}000$ training scenes and fixed 900-scene validation and test sets. A blank border of 10 pixels permits the same dihedral transformations and integer translations used in the standard experiments without clipping the foreground.

We compare label-only training against relative $\Delta\beta_0$ supervision at $\lambda_\Delta = 0.5$, using the same randomly initialised plain stack ($w = 32$), data, and seeds 23, 7, and 42. Both conditions use AdamW with learning rate 10^{-3}, weight decay 10^{-4}, a five-epoch linear warmup followed by cosine decay to 10^{-6}, batch size 32, and a fixed 200-epoch budget.

Table 6: Matched dense-clutter comparison. "Selected" is the test accuracy at the checkpoint with highest validation same-path accuracy. Values are means ± sample standard deviations over three seeds.

Auxiliary objective	Params	Selected	Per-seed selected	Final	$\to 90\%$ val.	Time / seed
none	656,546	56.52±11.00	51.67, 48.78, 69.11	56.59±11.71	never	720.1±8.5 s
relative $\Delta\beta_0$	656,804	**95.15±0.39**	94.78, 95.56, 95.11	**95.41±0.28**	46.7±0.6	751.3±14.4 s

The validation-selected paired gain is 38.63 ± 11.09 percentage points. The relative-target runs all lie between 94.78% and 95.56%, with selected-checkpoint $F_1 = 0.9518 \pm 0.0037$, and each crosses 90% validation accuracy. Label-only training is seed-dependent: two runs remain near chance, the third reaches 69.11%, and none crosses 90% validation accuracy. Thus the relative target remains effective when the raster contains substantially more disconnected fragments, while the matched label-only objective does not learn the relation reliably.

D.5 Computational cost

Relative topological supervision changes training but does not alter the inference computation.

The relative targets are computed once when the corpus is constructed and stored with the training metadata. Training consequently requires no online component labelling or other topological computation. At inference, both auxiliary regression heads are discarded and the model uses only the same-path classification head.

Table 7 reports the recorded wall-clock measurements at the $16{,}000$-scene scale. The two configurations use the same training split, rendering, augmentations, seeds, batch size, numerical precision, and 200-epoch budget.

Table 7: Recorded wall-clock training times at the 16,000-scene scale. Both configurations use the same split, the dashed rendering, dihedral transformations and shifts, seeds $23, 7, 42$, bfloat16 precision, batch size 32, and a fixed 200-epoch budget. Values are mean $\pm$ sample standard deviation over the three seeds.

Model	Supervision	Total time	Time/epoch
Plain stack ($w = 32$)	Relative $\Delta\beta_0$	757.5 ± 48.5 s	3.79 ± 0.24 s
hGRU	Label-only	$16{,}108.7 \pm 237.8$ s	80.54 ± 1.19 s

The recorded mean wall-clock time for hGRU was approximately $21.3\times$ that of the plain stack. The models use their respective prescribed training configurations: relative supervision with AdamW and cosine decay for the plain stack, and label-only supervision with NAdam and a constant learning rate for hGRU. This comparison therefore reports the end-to-end cost of the two training configurations.

E Application to retinal vessel segmentation on DRIVE

We evaluate whether relative topological supervision transfers from synthetic same-path recognition to vessel segmentation in real fundus images. Retinal vessels form thin, branching structures in which small gaps can substantially alter connectivity while producing only minor changes in pixelwise overlap.

Dataset and split. DRIVE contains 40 colour fundus photographs with native resolution 565×584 and expert vessel annotations (Staal et al., 2004). We use the official 20-image training set, partitioned into 16 training and 4 validation images, and evaluate on all 20 images in the official test set. Images and vessel masks are resized to 512×512, and all evaluation metrics are computed inside the provided field-of-view mask.

The query metadata contains 50 vessel-location pairs for each image, giving 2,000 query instances across the corpus. These pairs are used to construct the training targets; the image, rather than an individual pair, remains the independent unit for evaluation and statistical analysis.

Training-only relative target. Query locations are sampled from the expert vessel mask. The segment joining each pair is rasterised as an analytic capsule of radius 6 pixels at the native image resolution. Using foreground 8-connectivity and background 4-connectivity, the relative target is computed from the annotation mask as

$$\Delta\beta_0 = \beta_0(X) - \beta_0(X \cup A_q),$$

where X is the annotated vessel foreground and A_q is the query-induced capsule.

Query markers are rendered in the training input for all four experimental conditions so that query presentation is held fixed. The capsule itself is never supplied to the model. At inference, the auxiliary heads are discarded and segmentation is performed on clean, unmarked fundus images. The construction therefore uses no annotation beyond the vessel mask already provided for segmentation.

Architecture and training protocol. All conditions use an ImageNet-initialised ResNet-18 U-Net and the same segmentation loss,

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{Dice}}.$$

We compare segmentation-only training, query-independent global (β_0, β_1) supervision, permuted relative targets, and relative $\Delta\beta_0$ supervision. For each auxiliary condition, the auxiliary loss weight is $\lambda_\Delta = 0.05$; no connectivity-classification loss is used.

Models are trained for 60 epochs over three matched seeds $(23, 7, 42)$ with AdamW ($\text{lr} = 10^{-3}$, weight decay 10^{-4}), a five-epoch linear warmup followed by cosine annealing, batch size 16, and bfloat16 precision. Dihedral transformations and integer translations of at most 10 pixels are applied jointly to images, masks, and query locations.

Table 8: Complete results on the DRIVE test set using a ResNet-18 U-Net at 512×512. Values are mean $\pm$ sample standard deviation across three matched seeds. For statistical testing, seed results are first averaged within each of the 20 test images before applying a two-sided paired Wilcoxon test. The difference and raw-p rows compare relative $\Delta\beta_0$ supervision against segmentation-only training. For error metrics, a negative difference denotes improvement. Bold metric values indicate the best mean in each column; bold p-values denote raw $p < 0.05$.

Auxiliary condition	Dice ↑	IoU ↑	clDice ↑	APLS ↑	Betti-0 error ↓	Betti-1 error ↓
None	0.7995 ± 0.0023	0.6662 ± 0.0031	0.8097 ± 0.0010	0.4290 ± 0.0049	40.13 ± 3.73	18.27 ± 1.82
Global (β_0, β_1)	0.7996 ± 0.0026	0.6668 ± 0.0033	0.8075 ± 0.0048	0.4312 ± 0.0146	40.23 ± 6.30	18.08 ± 1.19
Permuted $\Delta\beta_0$	0.8005 ± 0.0012	0.6675 ± 0.0016	0.8134 ± 0.0006	$\mathbf{0.4344 \pm 0.0096}$	41.18 ± 4.31	17.07 ± 1.01
Relative $\Delta\beta_0$	$\mathbf{0.8015 \pm 0.0023}$	$\mathbf{0.6691 \pm 0.0032}$	$\mathbf{0.8167 \pm 0.0037}$	0.4305 ± 0.0080	$\mathbf{38.00 \pm 1.00}$	$\mathbf{15.75 \pm 1.17}$
Relative − None	$+0.0020$	$+0.0028$	$+0.0070$	$+0.0015$	-2.13	-2.52
Raw p	0.0620	0.0620	$\mathbf{0.000103}$	1.0000	0.6407	$\mathbf{0.000663}$

Table 9: Vessel-calibre and junction analysis on the DRIVE test set. Values are mean $\pm$ sample standard deviation across three matched seeds. Paired tests use one seed-averaged observation per test image ($n = 20$). The difference and raw-p rows compare relative $\Delta\beta_0$ supervision against segmentation-only training. The two pixel-recall columns use distance to background D; the centreline columns use the corresponding local-radius groups.

Auxiliary condition	Pixel recall $D \le 1.5\,\mathrm{px}$	Pixel recall $D > 1.5\,\mathrm{px}$	Thin-centreline recall	Thick-centreline recall	Junction precision	Junction recall	Junction F_1
None	0.7167 ± 0.0140	0.9811 ± 0.0026	0.6213 ± 0.0143	0.9834 ± 0.0014	$\mathbf{0.8097 \pm 0.0105}$	0.5120 ± 0.0268	0.6195 ± 0.0209
Global (β_0, β_1)	0.7097 ± 0.0093	0.9781 ± 0.0023	0.6111 ± 0.0029	0.9808 ± 0.0011	0.7928 ± 0.0167	0.5308 ± 0.0111	0.6266 ± 0.0131
Permuted $\Delta\beta_0$	0.7124 ± 0.0177	0.9795 ± 0.0012	0.6277 ± 0.0368	0.9839 ± 0.0014	0.8071 ± 0.0144	0.5205 ± 0.0415	0.6240 ± 0.0275
Relative $\Delta\beta_0$	$\mathbf{0.7248 \pm 0.0183}$	$\mathbf{0.9814 \pm 0.0021}$	$\mathbf{0.6413 \pm 0.0313}$	$\mathbf{0.9853 \pm 0.0022}$	0.7999 ± 0.0058	$\mathbf{0.5408 \pm 0.0306}$	$\mathbf{0.6371 \pm 0.0213}$
Relative − None	$+0.0081$	$+0.0003$	$+0.0200$	$+0.0018$	-0.0098	$+0.0288$	$+0.0176$
Raw p	$\mathbf{0.0276}$	0.1169	$\mathbf{0.000681}$	$\mathbf{0.00427}$	0.1560	$\mathbf{0.000293}$	$\mathbf{0.00171}$

Evaluation and statistical analysis. We select the checkpoint with the highest validation Dice. Predictions are binarised at probability > 0.5 and evaluated inside the field of view. We report Dice and IoU for volumetric overlap, clDice for centreline continuity (Shit et al., 2021), APLS for graph-routing fidelity (Van Etten et al., 2018), and absolute Betti-0 and Betti-1 errors for fragmentation and spurious cycles. For each seed, metrics are averaged over the 20 test images; tables report the mean and sample standard deviation of these image-averaged scores across the three matched seeds.

For paired statistical testing, each metric is first averaged across the three seeds within each test image. A two-sided paired Wilcoxon signed-rank test is then applied across the 20 independent test images. All reported p-values are unadjusted.

Complete segmentation and topology results. Table 8 reports the complete four-condition comparison.

Relative supervision achieves the highest mean Dice, IoU, and clDice and the lowest mean Betti-0 and Betti-1 errors. Relative to segmentation-only training, the clearest image-level effects are the 0.70-percentage-point clDice improvement ($p = 0.000103$) and the reduction of 2.52 Betti-1 errors per image ($p = 0.000663$). Dice and IoU improve in mean but do not reach raw $p < 0.05$ ($p = 0.0620$ for both), and APLS does not show evidence of improvement. The permuted condition attains the highest mean APLS but does not reproduce the joint improvement in centreline continuity and Betti-number errors.

Vessel-calibre and junction analysis. Let $D(x)$ denote the Euclidean distance from an annotated vessel pixel x to the background at the evaluation resolution of 512×512. We report pixel recall separately for the distance bands $D(x) \le 1.5\,\mathrm{px}$ and $D(x) > 1.5\,\mathrm{px}$. On the annotated vessel skeleton, $D(x)$ estimates the local vessel radius; the same threshold defines the thin- and thick-centreline groups. Junctions are defined on the annotated vessel skeleton and matched to predicted junctions within a 3-pixel tolerance.

The structural differences concentrate in thin vessels and branch junctions. Relative supervision increases thin-centreline recall by 2.00 percentage points and junction recall by 2.88 points, producing a 1.76-point increase in junction F_1. Pixel recall in the interior-distance group $D > 1.5\,\mathrm{px}$ is already above 98% under every condition, leaving little room for improvement. Junction precision decreases slightly, but the paired difference is not supported by the raw image-level test ($p = 0.1560$).

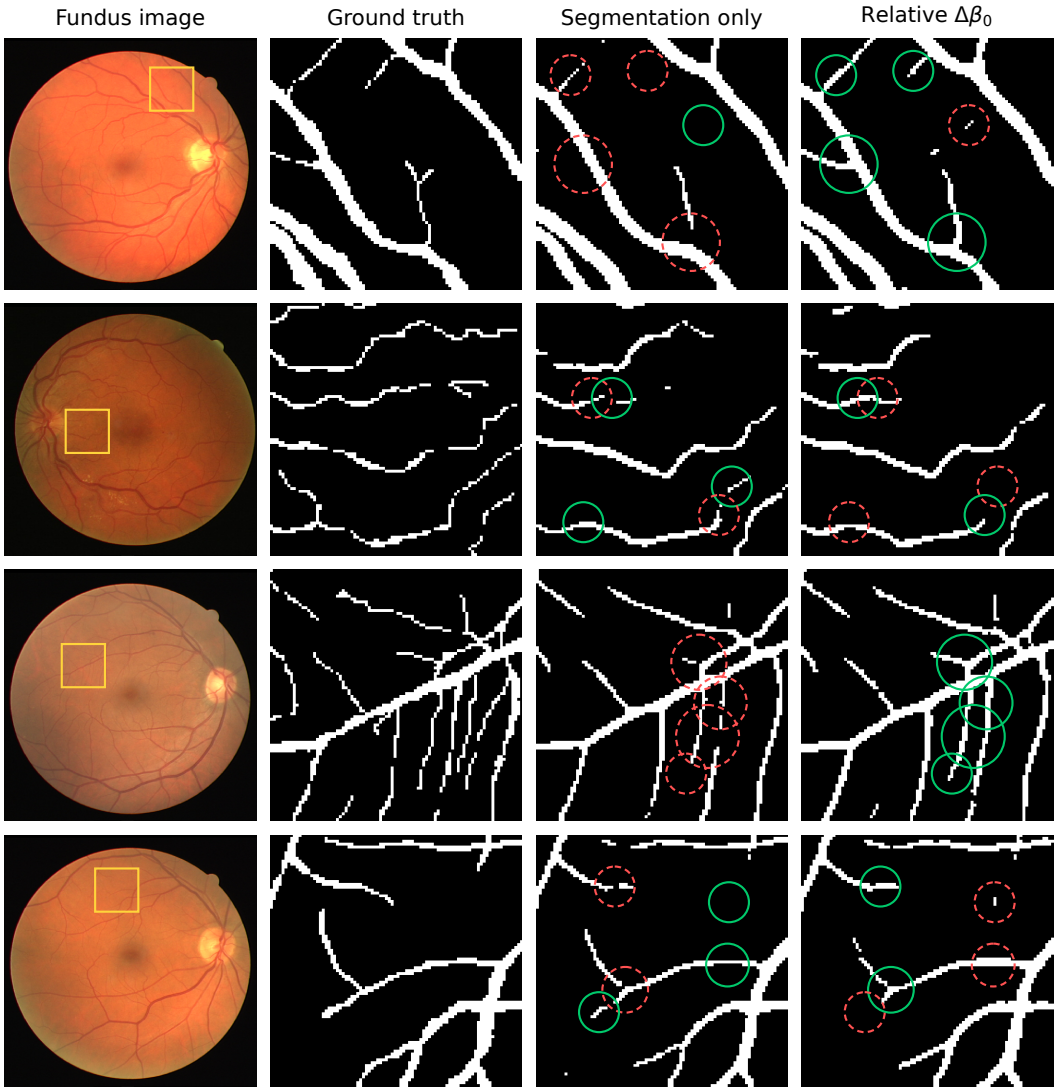


Figure 5: Qualitative comparison of vessel segmentation on DRIVE. Rows show four selected test images from seed 23. Yellow boxes locate the regions shown in the ground-truth and prediction columns. Green solid and red dashed circles mark local improvements and degradations in agreement with the ground truth relative to the other prediction, respectively. Corresponding circles indicate the same locations across the two prediction columns. Aggregate results are reported in Tables 8 and 9.

Qualitative examples. Figure 5 shows four selected local comparisons from one matched training seed. The examples illustrate local improvements and degradations; aggregate results are reported in Tables 8 and 9.

Overall, relative topological supervision improves centreline continuity, reduces cycle error, and better preserves thin-vessel and junction structure on DRIVE without changing the inference architecture or requiring additional annotations. These results provide real-image evidence that the benefit of the proposed training signal is not restricted to the synthetic Pathfinder generator.