LEARNING HOW MUCH AND WHERE FOR 3D VASCULAR RECONSTRUCTION FROM TWO BINARY PROJECTIONS

Anonymous authors
Paper under double-blind review

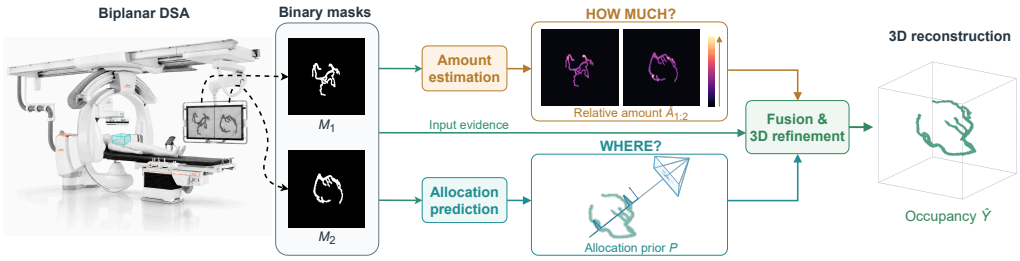

Figure 1: Our method decomposes 3D vascular reconstruction from two calibrated binary projections into amount estimation (*How much*) and occupancy-based allocation guidance (*Where*). Explicitly supervised view-normalized amount maps $\widehat{A}_{1:2}$ and a coarse allocation prior P are fused with mask and signed-distance evidence, then refined into 3D occupancy $\widehat{Y}$. Device photograph: illustrative acquisition setting, from Siemens Healthineers (2022).

ABSTRACT

Reconstructing sparse three-dimensional (3D) structures from two calibrated binary projections is a severely ill-posed inverse problem. In biplanar angiography, nonuniform contrast makes intensity unreliable for measuring vessel amount, motivating binary masks. These indicate ray–structure intersections but not occupied amounts or their depth distributions. We introduce Biplanar Amount–Allocation Reconstruction (BiAAR) around an explicit ray-wise decomposition: amount measures *how much* structure a pixel's ray contains; normalized allocation describes *where* it lies along that ray. Exact factors recover occupancy by multiplication, giving an information-preserving representation under complete calibrated coverage. From paired masks, signed distance fields, and cameras, BiAAR predicts view-normalized amounts and a coarse 3D occupancy prior supervised by the full target volume. This prior supplies allocation guidance; learned fusion and refinement combine both predictions with projection evidence rather than explicitly multiplying normalized factors. BiAAR achieves the highest mean Dice and clDice among evaluated methods on simulated ImageCAS and AVDNet-derived ASOCA projections. Each intermediate prediction improves reconstruction in ablations, with further gains when combined; fixed-capacity controls support their explicit supervision. These results support amount estimation and occupancy-based allocation guidance for two-view binary reconstruction. An anonymized repository containing the code is available at our project page.

1 INTRODUCTION

Reconstructing sparse three-dimensional (3D) structures from two calibrated binary projections is a severely ill-posed inverse problem. Few-view reconstruction nevertheless matters when additional observations incur acquisition time or sensing overhead (Kim et al., 2022). Binary projections indicate ray–structure intersections (Tulsiani et al., 2017), but multiple spatial configurations can produce the same masks even with calibrated cameras (Figure 2a).

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.

Vascular imaging illustrates the clinical value: 3D angiography detects small intracranial aneurysms missed by 2D angiography (van Rooij et al., 2008). Three-dimensional fusion guidance is associated with faster carotid cannulation during stenting (Swerdlow et al., 2019). Rotational X-ray acquisition takes several seconds (De Buck et al., 2013), with cardiac motion complicating coronary reconstruction (Hansis et al., 2008). Biplane acquisition supports time-resolved paired imaging (Wagner et al., 2016); synchronized exposures, assumed here, avoid inter-view motion registration and its uncertainty. Temporal alignment cannot resolve intensity ambiguity from nonuniform contrast-agent distribution (Chida et al., 2000; Ruedinger et al., 2021), motivating binary masks instead of intensity-derived vessel thickness or ray-wise amount.

Correspondence-based methods recover geometry from matched projections, but overlap and foreshortening make matching ambiguous (Bransby et al., 2023; Çimen et al., 2016). Projection-consistent optimization fits 3D representations to observations, yet two binary views admit many equally consistent structures (Fu et al., 2024; Tulsiani et al., 2017). Learned volumetric decoding incorporates structural priors (Zhu et al., 2025); unweighted backprojection replicates features along rays, whereas depth-aware lifting learns their assignment (Reading et al., 2021). The closest ray-wise approaches predict thickness, surface depths, or occupancy profiles (Shen et al., 2019; Gu et al., 2024; Kim & Lee, 2026). These representations can encode both extent and placement, yet binary observations constrain neither directly. The challenge is therefore to expose the missing 3D information as explicit, learnable quantities that jointly specify the full structure.

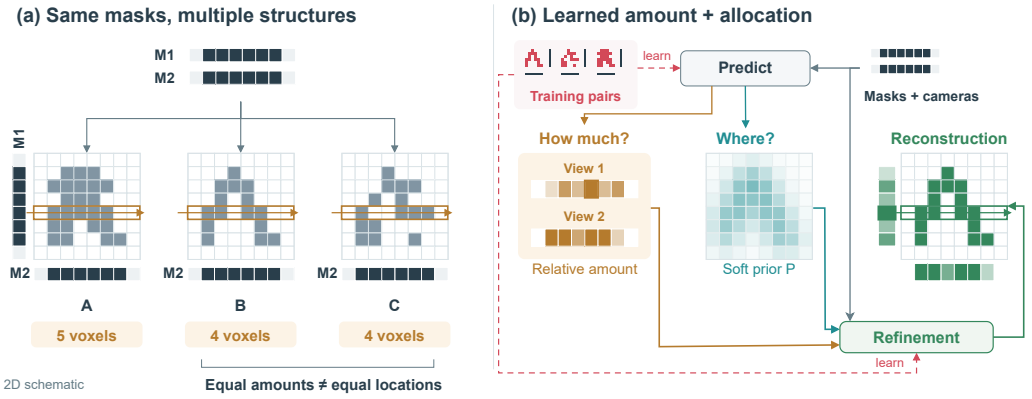

Figure 2: Amount–allocation decomposition in a 2D schematic. (a) Three structures share identical binary masks. Along the highlighted ray, A contains five voxels, whereas B and C contain four at different locations. (b) BiAAR learns view-normalized amount maps $\widehat{A}_{1:2}$ and a soft allocation prior P from training pairs, then fuses these predictions through refinement.

We factor each ray's occupancy profile into its total amount (*How much*) and normalized distribution along that ray (*Where*). Exact factors recover camera- and world-space structure under complete calibrated coverage (Appendix B). Biplanar Amount–Allocation Reconstruction (BiAAR; Figure 1) turns this representation into supervised intermediate cues. Paired masks and signed distance fields (SDFs) support view-normalized amount estimation and a 3D occupancy prior trained against the full volume that induces ray-wise allocations. BiAAR uses this prior to weight lifted features, then learns fusion and refinement rather than an explicit factor-product decoder.

Our contributions are summarized as follows:

- We formulate an information-preserving amount–allocation decomposition for two-view binary reconstruction, instantiated through supervised amount estimation and occupancy-based allocation guidance.
- We design a fully feed-forward model that fuses and refines both predictions. Ablations show that each improves reconstruction and their combination yields further gains. BiAAR achieves the highest mean Dice and clDice among evaluated methods on simulated ImageCAS and AVDNet-derived ASOCA projections, with limited degradation under mild calibration errors.

2 Related Work

We discuss four overlapping directions. Appendix G provides additional context.

Geometry- and correspondence-based reconstruction. Classical methods triangulate matched image points and fit vascular surfaces, point clouds, or meshes (Çimen et al., 2016; Galassi et al., 2018; Banerjee et al., 2019; Bransby et al., 2023). Automated matching and temporal cues aid correspondence estimation (Bappy et al., 2021; Frisken et al., 2025), but overlap and foreshortening remain difficult (Çimen et al., 2016). Visual hulls instead intersect silhouette constraints without explicit point correspondences (Laurentini, 1994).

Projection-consistent reconstruction. These methods constrain 3D representations through agreement between rendered and observed images. Neural angiographic methods use ray attention, projection losses, or cardiac-phase modeling (Cai et al., 2024; Maas et al., 2025). 3DGR-CAR and 4DRGS optimize radiative Gaussian primitives; the former also uses supervised center and depth prediction (Fu et al., 2024; Liu et al., 2025). Projection consistency can also supervise a predictor across training examples rather than fitting each scene independently (Tulsiani et al., 2017). When different structures render identical observations, their distinction depends on regularization or additional supervision. With fixed miscalibrated cameras, minimizing projection residuals can favor geometry that compensates for the incorrect imaging model.

Ray-wise shape and occupancy representations. Ray-wise representations expose thickness, surface locations, or occupancy profiles. Radiological-density shape recovery estimates thickness under restrictive attenuation and boundary assumptions (Poli & Valli, 1997); learned thickness maps also support Fourier-space reconstruction (Shen et al., 2019). X-ray In-Depth Decomposition assigns attenuation to fixed depth slabs (Albarqouni et al., 2017). 3DDX predicts front- and back-surface depths for individual bones (Gu et al., 2024). RayOcc models multiple occupied hypotheses through non-normalized Gaussian mixture intensity and interval-wise occupancy probabilities (Kim & Lee, 2026). These representations can encode both extent and placement. BiAAR separates amount and normalized allocation conceptually, implementing them through supervised amount maps and a full-occupancy prior that guides feature allocation.

Learning-based lifting and volumetric reconstruction. Image-to-occupancy methods decode aggregated image features into volumes (Choy et al., 2016). Calibrated lifting aligns features in 3D, but unweighted backprojection replicates them along rays. RGB-based depth-aware lifting instead weights feature assignment using learned depth probabilities (Philion & Fidler, 2020; Reading et al., 2021; Li et al., 2023). For biplanar vessels, DeepCA decodes binary backprojections, while Two Projections Suffice adds a connectivity prior and projection-consistent refinement (Wang et al., 2025; Cafaro et al., 2024). AutoCAR processes backprojected mask–SDF features with sparse convolutions (Zhu et al., 2025). XRA-diff conditions iterative diffusion refinement on two backprojections and their intersection (Lau et al., 2026). BiAAR uses its prior to guide feature contributions along each ray, then fuses this guidance with predicted amounts and mask–SDF evidence in one forward pass.

3 Method

3.1 Problem Formulation and Overview

Given two binary projection masks $M_v \in \{0,1\}^{H\times W}$, $v \in \{1,2\}$, BiAAR predicts occupancy $\widehat{Y} \in [0,1]^{N\times N\times N}$. In our vascular setting, Y denotes binary lumen occupancy. Simulated cameras have centers $\mathbf{c}_{1:2}$, share focal length f, and point toward the grid origin; $\mathcal{C} = (\mathbf{c}_{1:2}, f)$. Inputs $X_v = [M_v, D_v]$ concatenate masks and normalized SDFs $D_v \in [-1,1]^{H\times W}$. Calibrated foreground rays define binary 3D support S_v.

BiAAR separates each ray's occupancy into amount and allocation (Figure 2). In the ideal continuous-detector model, occupancy $y_v(\mathbf{p}, t)$ at distance t along detector coordinate $\mathbf{p}$'s ray factors as

$$y_v(\mathbf{p}, t) = \underbrace{\widetilde{A}_v^{\mathrm{ray}}(\mathbf{p})}_{\text{How much}} \underbrace{\alpha_v^{\mathrm{ray}}(t \mid \mathbf{p})}_{\text{Where}}. \tag{1}$$

Amount is total occupied length; allocation integrates to one on nonempty rays, with both zero on empty rays. Exact factors determine the covered structure (Appendix B.1), but this identity does not identify them uniquely from binary masks.

Our discrete model predicts view-normalized weighted voxel amounts $\widehat{A}_{1:2}$ and an occupancy-based allocation prior $P = \sigma(Z^P)$, with logits Z^P and sigmoid σ. The prior is supervised by Y and induces ray-wise allocations (Section 3.3). Exact normalized factors suffice for binary reconstruction under complete calibrated coverage (Appendix B.3); learned refinement instead fuses imperfect predictions with mask–SDF evidence. Inference uses only $M_{1:2}$ and $\mathcal{C}$ (Figure 1; Appendix A).

3.2 Geometry-Aware Amount Estimation

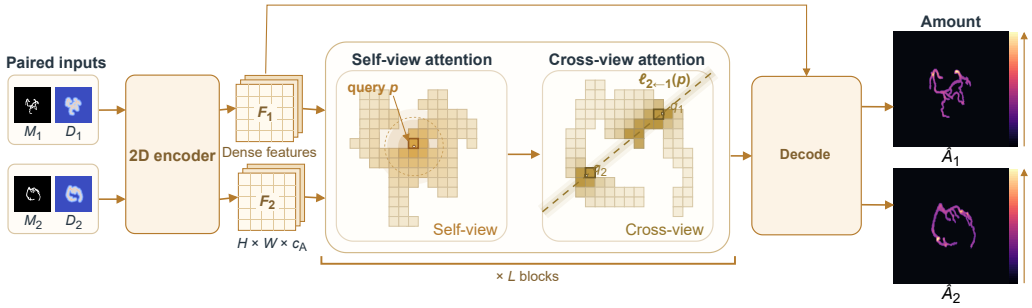

Figure 3: Geometry-aware amount estimation (*How much*). A shared 2D encoder extracts features from $X_v = [M_v, D_v]$ for mask-guided foreground tokenization. Each block applies self-view and cross-view attention, biased by pixel distance and point-to-epipolar-line distance. Scattered attention contexts update dense features before decoding $\widehat{A}_{1:2}$. In the diagram, F_v abbreviates $F_v^{A,0}$; both views undergo the illustrated updates.

On the reconstruction grid $\mathcal{G}$, nonnegative bilinear weights $w_v(\mathbf{p}, \mathbf{x})$ splat voxel occupancy onto detector pixels. The absolute weighted voxel amount and its supervised target are

$$\widetilde{A}_v(\mathbf{p}) = \sum_{\mathbf{x}\in\mathcal{G}} w_v(\mathbf{p}, \mathbf{x}) Y(\mathbf{x}), \qquad A_v^*(\mathbf{p}) = \frac{\widetilde{A}_v(\mathbf{p})}{s_v}, \qquad s_v = \max_{\mathbf{p}} \widetilde{A}_v(\mathbf{p}). \tag{2}$$

Empty views use $A_v^* = 0$. View normalization preserves support and ordering while expanding differences relative to normalization by a larger fixed scale (Appendix C.1).

A shared 2D U-Net (Ronneberger et al., 2015) encodes $X_{1:2}$. Mask-guided foreground feature tokenization gathers feature vectors only where $M_v = 1$, retaining their detector coordinates. Each geometry-aware block applies self-view attention followed by cross-view attention. For target view v and source $u \in \{v, \bar{v}\}$, where $\bar{v} = 3 - v$, the usual scaled query–key logit subtracts

$$b_{v\leftarrow u}(\mathbf{p}, \mathbf{q}) = \begin{cases} \|\mathbf{p} - \mathbf{q}\|_2 / \sigma_{\text{self}}, & u = v, \\ d_{\text{epi}}(\mathbf{q}, \ell_{u\leftarrow v}(\mathbf{p})) / \sigma_{\text{cross}}, & u = \bar{v}. \end{cases} \tag{3}$$

Here, $\ell_{u\leftarrow v}(\mathbf{p})$ is the calibrated epipolar line, d_{epi} is point-to-line distance, and the positive σ parameters set distance scales in pixels. Thus attention combines local same-view context with epipolar-compatible cross-view evidence. Value-weighted contexts update the dense features before decoding $\widehat{A}_v = M_v \odot \sigma(Z_v^A)$, where $\odot$ denotes elementwise multiplication. Figure 3 illustrates the branch; Appendix A.2 gives tokenization, attention, and decoding details.

3.3 Camera-Aware Structural Allocation Prior

The allocation branch lifts mask-gated features $F_v^P = M_v \odot \Phi_P(X_v)$ through calibrated backprojection $\mathcal{B}_v$. A 3D U-Net $\mathcal{N}_P$ (Çiçek et al., 2016) decodes the shared-grid features:

$$Z^P = \mathcal{N}_P\left([\mathcal{B}_1(F_1^P), \mathcal{B}_2(F_2^P)]\right), \qquad P = \sigma(Z^P). \tag{4}$$

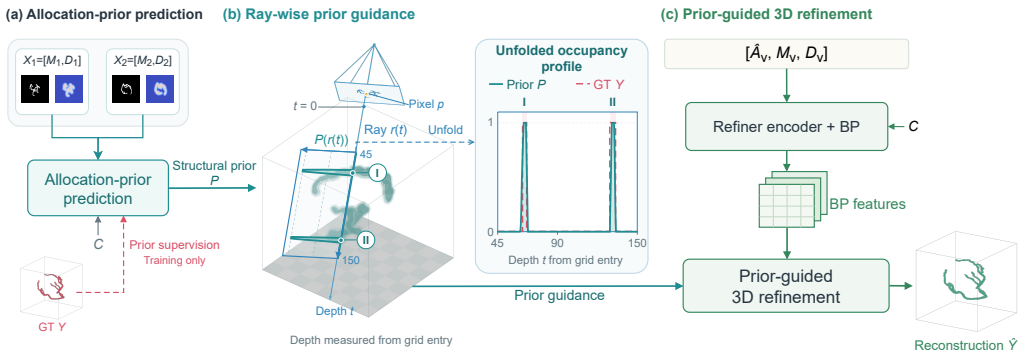

Figure 4: Allocation prior and refinement. (a) Paired mask–SDF features are backprojected to predict the occupancy prior P, supervised by Y. (b) A profile shows the prior's support along an individual pixel's ray. (c) Raw and prior-weighted features from $[\widehat{A}_v, M_v, D_v]$ are fused with prior cues to refine $\widehat{Y}$. The ray profile is explanatory; the refiner uses the full 3D prior.

The full-resolution occupancy prior induces $\alpha_v^P(\mathbf{x} \mid \mathbf{p}) = \frac{w_v(\mathbf{p},\mathbf{x})P(\mathbf{x})}{\sum_{\mathbf{x}' \in \mathcal{G}} w_v(\mathbf{p},\mathbf{x}')P(\mathbf{x}')}$, set to zero when the denominator vanishes (Figure 4(a,b)). Supervision by Y supplies allocation guidance while also constraining occupancy amplitude; it is stronger than supervising normalized allocations alone (Appendix C.2).

3.4 Prior-Guided 3D Refinement

The refiner encodes amount, mask, and SDF jointly, then backprojects both views:

$$F_v^R = M_v \odot \Phi_R([\widehat{A}_v, M_v, D_v]), \qquad V^R = [\mathcal{B}_1(F_1^R), \mathcal{B}_2(F_2^R)]. \tag{5}$$

The prior provides a soft gate $G = \gamma + (1-\gamma)P$, with $0 < \gamma \leq 1$, and an ambiguity cue $U = P \odot (1-P)$. Retaining raw V^R alongside its gated counterpart preserves evidence for correcting prior errors. With clipped prior logits $\bar{Z}^P = \mathrm{clip}(Z^P, -c_Z, c_Z)$, the 3D refiner predicts

$$Z = \mathcal{N}_R([V^R, V^R \odot G, P, U]) + \lambda_Z \bar{Z}^P, \qquad \widehat{Y} = \sigma(Z). \tag{6}$$

Prior gating implements allocation-weighted feature lifting (Appendix C.3). The refiner uses P directly, without ray normalization, explicit factor multiplication, or hard reprojection. Appendices A.4 and C.4 detail its implementation and bounds.

3.5 End-to-End Objective and Inference

We jointly supervise final occupancy, foreground amount, and the 3D allocation prior:

$$\mathcal{L} = \mathcal{L}_V(Z, Y) + \lambda_A \mathcal{L}_A(\widehat{A}_{1:2}, A^*_{1:2}) + \lambda_P \mathcal{L}_V(Z^P, Y). \tag{7}$$

Here, $\mathcal{L}_V$ combines binary cross-entropy, soft Dice, and soft clDice (Shit et al., 2021); $\mathcal{L}_A$ averages smooth-L_1 error over foreground pixels. All three branches are jointly optimized, allowing the final loss to update both intermediate predictions. Inference is fully feedforward, and $\widehat{Y}$ is thresholded at 0.5. Appendices A.5 and A.6 provide full losses and numerical settings.

4 Experiments

4.1 Experimental Setup

Datasets and projections. We evaluate on simulated projections from ImageCAS (Zeng et al., 2023) and the 40 public ASOCA training scans (Gharleghi et al., 2022). ASOCA references are released AVDNet artery/vein predictions, rather than the original manual annotations (Wang et al., 2024). ImageCAS uses 800/100/100 training/validation/test volumes. ASOCA is split by patient

(32/4/4), yielding 64/8/8 separate artery/vein volumes. Testing uses five camera pairs per ImageCAS volume and twenty per ASOCA volume: 500 and 160 paired-view samples, respectively. The simulator samples orthogonal viewing axes, with 256^2 masks and 128^3 outputs. Appendix D.1 details projection synthesis and statistical units.

Training and evaluation. From-scratch training uses 200 epochs; ImageCAS-to-ASOCA fine-tuning uses 50. Checkpoints are selected by validation Dice. We report Dice, soft clDice (Shit et al., 2021) (denoted clDice), and boundary distances in voxel units. Reported means and standard deviations summarize test samples for each selected checkpoint. Appendix D.2 provides optimization and metric details.

Efficiency. On an NVIDIA GeForce RTX 4090D GPU, BiAAR reconstructs one 128^3 volume in approximately **0.12** s using FP32 and batch size one. CUDA is synchronized before and after each forward pass; data loading and SDF/ray-support construction are excluded. 3DGR-CAR's GCP has 1,096,252 parameters and uses 5.966 GFLOPs per pair, excluding Gaussian optimization and voxelization; Appendix D.7 details the network-cost comparison.

4.2 Comparison with the State of the Art

We compare with Two Projections Suffice (Cafaro et al., 2024), AutoCAR (Zhu et al., 2025), DeepCA (Wang et al., 2025), and 3DGR-CAR (Fu et al., 2024). All methods share splits, paired masks, and cameras; additional inputs are mask-derived. Table 1 reports all six metrics; Appendix D.4 details implementations and adaptations.

Table 1: Complete single-dataset reconstruction benchmarks with each method trained from scratch on the corresponding dataset. Values are mean±standard deviation over 500 ImageCAS and 160 ASOCA paired-view test samples; distance metrics use voxel units. Arrows indicate the preferred direction; **bold** and underline mark the best and second-best means.

Method	Dice ↑	clDice ↑	Recall ↑	Precision ↑	HD95 ↓	ASSD ↓
ImageCAS						
Two Proj. Suffice	.672±.061	.756±.070	.754±.067	.608±.067	3.727±2.117	.807±.347
AutoCAR	.712±.064	.783±.069	.814±.072	.635±.063	2.881±1.492	.634±.270
DeepCA	.635±.061	.634±.072	.644±.064	.630±.072	4.727±5.243	.997±.857
3DGR-CAR	.239±.060	.230±.060	.202±.054	.295±.075	10.589±3.359	2.998±1.010
BiAAR (ours)	**.809±.054**	**.879±.054**	**.826±.065**	**.793±.050**	**1.806±1.552**	**.375±.234**
ASOCA						
Two Proj. Suffice	.613±.160	.569±.210	.602±.178	.628±.146	7.292±4.799	1.509±.922
AutoCAR	.570±.164	.486±.187	.592±.170	.554±.166	5.924±3.165	1.390±.667
DeepCA	.567±.152	.440±.150	.543±.153	.597±.159	**5.757±3.521**	**1.343±.713**
3DGR-CAR	.230±.143	.207±.124	.208±.113	.270±.190	10.553±4.349	3.241±1.390
BiAAR (ours)	**.636±.162**	**.586±.208**	**.606±.187**	**.681±.138**	8.379±6.576	1.703±1.320

BiAAR achieves the highest mean Dice and clDice on both datasets (Table 1), whereas DeepCA achieves the lowest HD95 and ASSD on ASOCA. Figure 5 shows improved continuity and local shape recovery, alongside residual missing branches in complex ASOCA structures. Appendix E provides additional examples, including 3DGR-CAR.

4.3 Ablation Studies

Component ablations test intermediate cues; loss-weight ablations isolate their direct supervision at fixed architecture and capacity.

Intermediate cues and refiner inputs. Adding the allocation prior (B1) or predicted amount (B5) improves B0; combining both (B6) further improves Dice/clDice (Table 2). Appendix D.5 reports all input toggles and amount oracles.

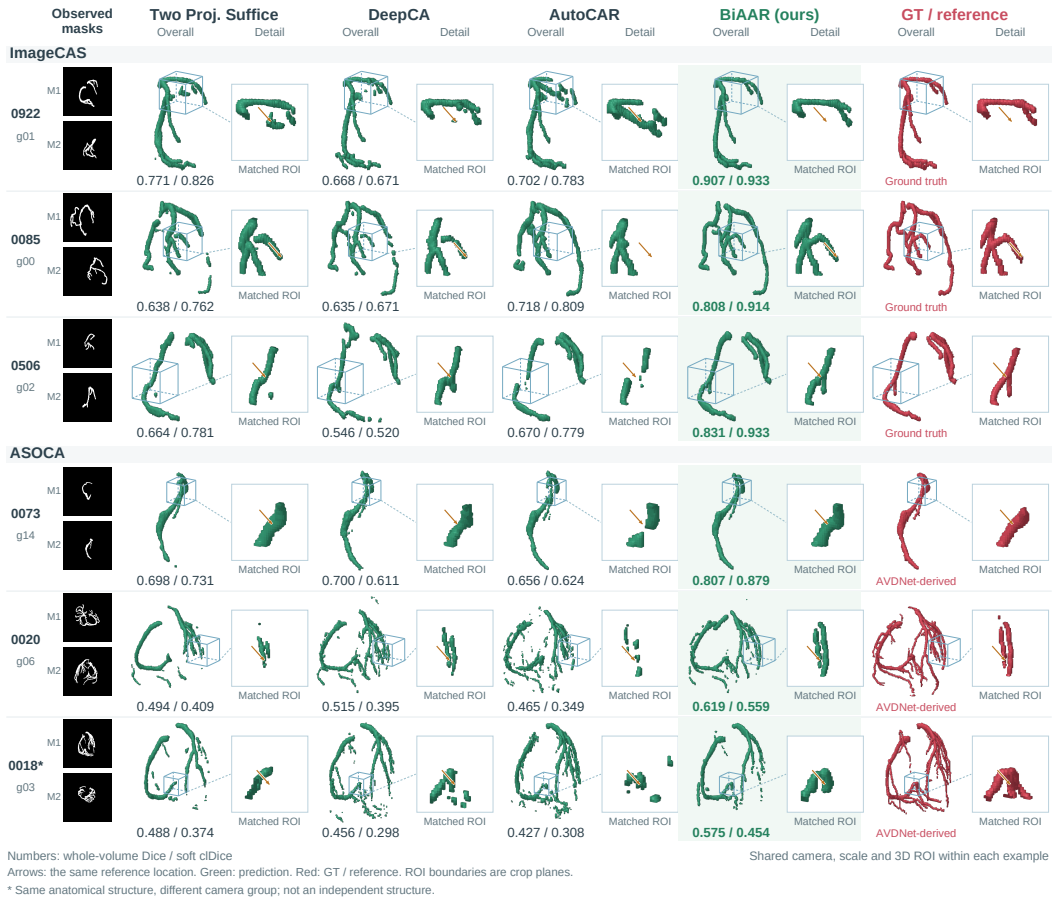

Figure 5: Single-dataset reconstruction on ImageCAS (top) and AVDNet-derived ASOCA (bottom), with all models trained from scratch. Rows show masks, reconstructions, and matched 3D regions of interest (ROIs). Predictions are green and references red; arrows mark corresponding locations. Scores are whole-volume Dice / soft clDice. Starred rows revisit a structure under another camera configuration.

Table 2: Key component ablations on ImageCAS (paired-view sample means). All rows retain mask–SDF refiner inputs. Pred/GT denote predicted/ground-truth amount maps; B11 is an amount oracle. Prior metrics evaluate P before refinement; dashes denote absent cues or outputs. Distance metrics use voxel units; **bold** marks the best non-oracle means among the displayed rows. Table 5 gives the complete sweep.

	Refiner inputs				Final reconstruction $\widehat{Y}$						Allocation prior P		
ID	Amount	M_v	D_v	P	Dice ↑	clDice ↑	Recall ↑	Prec. ↑	HD95 ↓	ASSD ↓	Dice ↑	Recall ↑	Prec. ↑
B0	–	✓	✓	×	.765	.835	.797	.738	2.245	.484	–	–	–
B1	–	✓	✓	✓	.800	.869	.818	.784	1.986	.419	.779	.800	.762
B5	Pred	✓	✓	×	.794	.867	.821	.770	1.828	.396	–	–	–
B6	Pred	✓	✓	✓	**.809**	**.879**	**.826**	**.793**	**1.806**	**.375**	**.789**	**.807**	**.774**
B11	GT	✓	✓	✓	.926	.950	.924	.929	.929	.120	.790	.812	.771

Auxiliary supervision at fixed capacity. We vary only (λ_A, λ_P) in Equation (7), retaining the B6 architecture, inputs, and training/test protocol. Both branches receive final reconstruction gradients in every setting (Table 3).

Table 3: Fixed-capacity supervision ablation (500 ImageCAS test-pair means). All B6 branches remain present; $(1, 1)$ is full supervision. MSE_A measures foreground normalized-amount error; Dice_P evaluates P at threshold 0.5 before refinement. Distances use voxel units; **bold** marks the best means.

Loss weights		Final reconstruction $\widehat{Y}$				Intermediate predictions	
λ_A	λ_P	Dice ↑	clDice ↑	HD95 ↓	ASSD ↓	MSE_A ↓	Dice_P ↑
0	0	.774	.851	2.026	.446	.043	.350
0	1	.794	.869	**1.753**	.386	.052	.773
1	0	.778	.855	1.999	.439	.012	.000
1	1	**.809**	**.879**	1.806	**.375**	**.011**	**.789**

Full supervision gives the best Dice/clDice, ASSD, and intermediate scores; $(0, 1)$ has lower HD95. At $(0, 0)$, amount MSE/prior Dice are 0.043/0.350, better than the respective outputs lacking direct supervision in $(0, 1)$ and $(1, 0)$. This suggests coupled intermediate learning; direct supervision improves target alignment and final overlap (Appendix D.5).

Prior refinement and amount quality. Refinement raises B6's prior Dice from 0.789 to 0.809. GT amount (B11) raises final Dice/clDice to 0.926/0.950, indicating headroom for improved amount estimation. Across the complete non-oracle sweep, B6 leads five of six final-output metrics; B2 has the lowest HD95 (Table 5).

4.4 Data-Efficient Adaptation to ASOCA

AutoCAR and BiAAR are fine-tuned from their ImageCAS checkpoints for 50 epochs on nested subsets of the ASOCA training volumes. The 20/40/60/80/100% fractions use 13/26/38/51/64 volumes; 0% denotes zero-shot transfer. Appendix F provides the subset protocol and complete results.

On ASOCA, 13-volume fine-tuning reaches Dice 0.640, numerically close to 0.636 from scratch using all 64 volumes. Full fine-tuning reaches Dice/clDice 0.713/0.678 (Figure 6). These comparisons against full-data scratch training support transfer from ImageCAS; matched-subset scratch baselines remain untested.

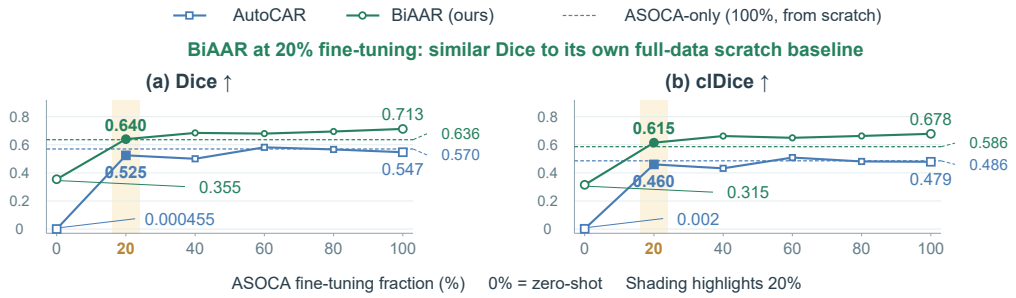

Figure 6: ImageCAS-to-ASOCA adaptation. Curves show mean Dice/clDice over 160 paired-view samples; labels mark 0/20/100%. Dashed lines indicate full-data ASOCA-only training from scratch. Figure 10 adds distance metrics; Table 8 gives all six metrics as mean±standard deviation.

Fine-tuning improves branch recovery in the illustrated examples (Figure 7).

4.5 Robustness to Calibration Errors

Under the independently applied bounds in Table 4, Dice/clDice decreased by at most 0.033/0.025, indicating tolerance to mild calibration errors. Larger errors substantially degraded reconstruction; Appendix D.6 provides the full results, sampling protocol, and clinical calibration context.



Figure 7: ASOCA-only training from scratch (64 volumes) versus ImageCAS pretraining with 0/20/100% fine-tuning. Predictions are green; AVDNet-derived references are red. Scores are whole-volume Dice/soft clDice; *Empty ROI* denotes no predicted occupancy in the crop.

Table 4: ImageCAS calibration robustness (500-pair means; full sweep: Table 6).

Error	Bound	BP IoU	Dice ↑	clDice ↑	HD95 ↓
Clean	–	1.000	.809	.879	1.806
Rotation	0.3°	.950	.799	.872	1.893
Translation	1 mm	.942	.792	.865	2.086
Focal length	1%	.929	.776	.854	2.148

5 Conclusion

BiAAR frames two-view binary reconstruction through each ray's total occupied amount (*How much*) and normalized depth distribution (*Where*). The model combines supervised amount maps with occupancy-based allocation guidance through learned fusion and refinement. ImageCAS ablations show that each prediction improves reconstruction, with further Dice/clDice gains when combined. Joint supervision also gives the highest Dice/clDice at fixed capacity. BiAAR achieves the highest mean Dice/clDice among evaluated methods on simulated ImageCAS and AVDNet-derived ASOCA projections. The substantial gains from GT amounts further support amount as an explicit intermediate representation and motivate improving its estimation. Clinical validation should cover real angiograms, mask errors, diverse biplane geometries, and motion compensation for rotational coronary scans.

AI USE STATEMENT

We used large language models (GPT-5.6 Sol and GPT-6 Astra, 2026 release) to assist with language polishing, literature discovery, conceptual illustration drawing, and checking mathematical proofs. These models were not involved in generating ideas or shaping the conceptual framework. The authors directed and revised the AI-assisted work and take responsibility for the final text, references, mathematical claims, and reported results.

REPRODUCIBILITY STATEMENT

Appendix A specifies the model operations, losses, and numerical settings; Appendices B and C provide the mathematical assumptions and proofs. Appendix D documents dataset sources and splits, projection synthesis, training, evaluation, and baseline implementations; Appendix F details the adaptation protocol. To ensure reproducibility, source code of the core components is available in the anonymous repository linked in the abstract and statement. Upon acceptance, all code and related resources will be released in a public GitHub repository.

REFERENCES

Shadi Albarqouni, Javad Fotouhi, and Nassir Navab. X-ray in-depth decomposition: Revealing the latent structures. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 444–452. Springer, 2017.

Abhirup Banerjee, Francesca Galassi, Ernesto Zacur, Giovanni Luigi De Maria, Robin P Choudhury, and Vicente Grau. Point-cloud method for automated 3d coronary tree reconstruction from multiple non-simultaneous angiographic projections. *IEEE transactions on medical imaging*, 39 (4):1278–1290, 2019.

DM Bappy, Ayoung Hong, Eunpyo Choi, Jong-Oh Park, and Chang-Sei Kim. Automated three-dimensional vessel reconstruction based on deep segmentation and bi-plane angiographic projections. *Computerized Medical Imaging and Graphics*, 92:101956, 2021.

Kit Mills Bransby, Vincenzo Tufaro, Murat Çap, Greg Slabaugh, Christos Bourantas, and Qianni Zhang. 3d coronary vessel reconstruction from bi-plane angiography using graph convolutional networks. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5. IEEE, 2023.

Alexandre Cafaro, Reuben Dorent, Nazim Haouchine, Vincent Lepetit, Nikos Paragios, William M Wells III, and Sarah Frisken. Two projections suffice for cerebral vascular reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 722–731. Springer, 2024.

Yuanhao Cai, Jiahao Wang, Alan Yuille, Zongwei Zhou, and Angtian Wang. Structure-aware sparse-view x-ray 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11174–11183. IEEE, 2024.

Koichi Chida, Masahiro Sai, Haruo Saito, Kei Takase, Masayuki Zuguchi, Masatoshi Sasaki, and Tokutaro Sato. Relationship between the pixel value in digital subtraction angiography and iodine concentration: study in high iodine concentration with original phantom. *The Tohoku Journal of Experimental Medicine*, 190(3):169–176, 2000.

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 3070–3079. IEEE, 2019.

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *European Conference on Computer Vision*, pp. 628–644. Springer, 2016.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pp. 424–432. Springer, 2016.

Serkan Çimen, Ali Gooya, Michael Grass, and Alejandro F Frangi. Reconstruction of coronary arteries from x-ray angiography: A review. *Medical image analysis*, 32:46–68, 2016.

Stijn De Buck, Becker SN Alzand, Jean-Yves Wielandts, Christophe Garweg, Thomas Phlips, Joris Ector, Dieter Nuyens, and Hein Heidbuchel. Cardiac three-dimensional rotational angiography can be performed with low radiation dose while preserving image quality. *Europace*, 15(12): 1718–1724, 2013.

Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 12872–12881. IEEE, 2022.

Sarah Frisken, Vivek Gopalakrishnan, David Dimitris Chlorogiannis, Nazim Haouchine, Alexandre Cafaro, Alexandra J Golby, William M Wells III, and Rose Du. Spatiotemporally constrained 3d reconstruction from biplanar digital subtraction angiography. *International journal of computer assisted radiology and surgery*, 20(8):1689–1701, 2025.

Xueming Fu, Yingtai Li, Fenghe Tang, Jun Li, Mingyue Zhao, Gao-Jun Teng, and S Kevin Zhou. 3DGR-CAR: Coronary artery reconstruction from ultra-sparse 2D X-ray views with a 3D gaussians representation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 14–24. Springer, 2024.

Francesca Galassi, Mohammad Alkhalil, Regent Lee, Philip Martindale, Rajesh K Kharbanda, Keith M Channon, Vicente Grau, and Robin P Choudhury. 3d reconstruction of coronary arteries from 2d angiographic projections using non-uniform rational basis splines (nurbs) for accurate modelling of coronary stenoses. *PloS one*, 13(1):e0190650, 2018.

Ramtin Gharleghi, Dona Adikari, Katy Ellenberger, Sze-Yuan Ooi, Chris Ellis, Chung-Ming Chen, Ruochen Gao, Yuting He, Raabid Hussain, Chia-Yen Lee, et al. Automated segmentation of normal and diseased coronary arteries–the asoca challenge. *Computerized Medical Imaging and Graphics*, 97:102049, 2022.

Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9224–9232. IEEE, 2018.

Yi Gu, Yoshito Otake, Keisuke Uemura, Masaki Takao, Mazen Soufi, Seiji Okada, Nobuhiko Sugano, Hugues Talbot, and Yoshinobu Sato. 3DDX: Bone surface reconstruction from a single standard-geometry radiograph via dual-face depth estimation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 3–13. Springer, 2024.

Eberhard Hansis, Dirk Schafer, Olaf Dossel, and Michael Grass. Evaluation of iterative sparse object reconstruction from few projections for 3-d rotational coronary angiography. *IEEE Transactions on Medical Imaging*, 27(11):1548–1555, 2008.

Avinash C Kak and Malcolm Slaney. *Principles of computerized tomographic imaging*. SIAM, 2001.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023.

Byeongjoon Kim, Hyunjung Shim, and Jongduk Baek. A streak artifact reduction algorithm in sparse-view ct using a self-supervised neural representation. *Medical physics*, 49(12):7497–7515, 2022.

Junho Kim and Seongwon Lee. Rayocc: Occlusion-aware ray occupancy estimation via gaussian mixture intensity. *arXiv preprint arXiv:2607.17660*, 2026.

Dominik Bernard Lau, Hubert Malinowski, Jerzy Szyjut, and Tomasz Dziubich. Xra-diff: Guided diffusion as a refinement for backprojected coronary angiography x-ray images. *Applied Soft Computing*, pp. 116236, 2026.

Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on pattern analysis and machine intelligence*, 16(2):150–162, 1994.

Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 1477–1485, 2023.

Zhentao Liu, Ruyi Zha, Huangxuan Zhao, Hongdong Li, and Zhiming Cui. 4drgs: 4d radiative gaussian splatting for efficient 3d vessel reconstruction from sparse-view dynamic dsa images. In *International Conference on Information Processing in Medical Imaging*, pp. 361–374. Springer, 2025.

Kirsten WH Maas, Danny Ruijters, Anna Vilanova, and Nicola Pezzotti. Nerf-ca: dynamic reconstruction of x-ray coronary angiography with extremely sparse-views. *IEEE Transactions on Visualization and Computer Graphics*, 31(10):8782–8795, 2025.

Marko Mihajlovic, Sergey Prokudin, Siyu Tang, Robert Maier, Federica Bogo, Tony Tung, and Edmond Boyer. Splatfields: Neural gaussian splats for sparse 3d and 4d reconstruction. In *European Conference on Computer Vision*, pp. 313–332. Springer, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pp. 194–210. Springer, 2020.

Riccardo Poli and Guido Valli. Shape from radiological density. *Computer Vision and Image Understanding*, 65(3):361–381, 1997.

Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8551–8560. IEEE, 2021.

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

KL Ruedinger, S Schafer, MA Speidel, and CM Strother. 4d-dsa: development and current neurovascular applications. *American Journal of Neuroradiology*, 42(2):214–220, 2021.

Weichao Shen, Yunde Jia, and Yuwei Wu. 3d shape reconstruction from images in the frequency domain. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4466–4474. IEEE, 2019.

Suprosanna Shit, Johannes C Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien PW Pluim, Ulrich Bauer, and Bjoern H Menze. clDice-a novel topology-preserving loss function for tubular structure segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16555–16564. IEEE, 2021.

Siemens Healthineers. ARTIS icono Biplane with detectors specially designed for cardiovascular care. `https://www.siemens-healthineers.com/press/releases/artis-icono-biplane`, 2022.

Nicholas J Swerdlow, Douglas W Jones, Alexander B Pothof, Thomas FX O'Donnell, Patric Liang, Chun Li, Mark C Wyers, and Marc L Schermerhorn. Three-dimensional image fusion is associated with lower radiation exposure and shorter time to carotid cannulation during carotid artery stenting. *Journal of Vascular Surgery*, 69(4):1111–1120, 2019.

Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2626–2634, 2017.

Willem Jan van Rooij, ME Sprengers, Anjob N de Gast, JPP Peluso, and Menno Sluzewski. 3d rotational angiography: the new gold standard in the detection of additional intracranial aneurysms. *American Journal of Neuroradiology*, 29(5):976–979, 2008.

Martin Wagner, Sebastian Schafer, Charles Strother, and Charles Mistretta. 4d interventional device reconstruction from biplane fluoroscopy. *Medical physics*, 43(3):1324–1334, 2016.

Wenji Wang, Qing Xia, Zhennan Yan, Zhiqiang Hu, Yinan Chen, Wen Zheng, Xiao Wang, Shaoping Nie, Dimitris Metaxas, and Shaoting Zhang. AVDNet: Joint coronary artery and vein segmentation with topological consistency. *Medical Image Analysis*, 91:102999, 2024.

Yiying Wang, Abhirup Banerjee, Robin P Choudhury, and Vicente Grau. DeepCA: Deep learning-based 3D coronary artery tree reconstruction from two 2D non-simultaneous X-ray angiography projections. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 337–346. IEEE, 2025.

Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. SurroundOcc: Multi-camera 3D occupancy prediction for autonomous driving. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21672–21683. IEEE, 2023.

Thomas Wolff, Andras Lasso, Markus Eblenkamp, Erich Wintermantel, and Gabor Fichtinger. C-arm angle measurement with accelerometer for brachytherapy: an accuracy study. *International journal of computer assisted radiology and surgery*, 9(1):137–144, 2014.

An Zeng, Chunbiao Wu, Guisen Lin, Wen Xie, Jin Hong, Meiping Huang, Jian Zhuang, Shanshan Bi, Dan Pan, Najeeb Ullah, et al. Imagecas: A large-scale dataset and benchmark for coronary artery segmentation based on computed tomography angiography images. *Computerized Medical Imaging and Graphics*, 109:102287, 2023.

Yinheng Zhu, Yong Wang, Chunxia Di, Hanghang Liu, Fangzhou Liao, and Shaohua Ma. Sparse and transferable three-dimensional dynamic vascular reconstruction for instantaneous diagnosis. *Nature Machine Intelligence*, 7(5):730–742, 2025.

APPENDIX CONTENTS

A Method Implementation Details . . . 15
A.1 Inputs and target construction . . . 15
A.2 Foreground tokenization, attention, and decoding . . . 15
A.3 Camera geometry, backprojection, and the allocation prior . . . 16
A.4 Prior-guided fusion and refinement . . . 17
A.5 Training losses . . . 17
A.6 Numerical settings and inference . . . 18
B Mathematical Analysis: Amount–Allocation Factorization . . . 18
B.1 Exact factorization in camera and world coordinates . . . 19
B.2 Discrete factorization for the Method's splatting operator . . . 19
B.3 Sufficiency of normalized amount and the allocation prior . . . 20
C Mathematical Analysis: Design Properties . . . 21
C.1 What view-wise amount normalization preserves and expands . . . 22
C.2 Why occupancy supervision supplies an allocation target . . . 22
C.3 How prior gating implements allocation guidance . . . 23
C.4 Why prior guidance remains soft . . . 24
D Extended Experimental Protocols and Results . . . 24
D.1 Datasets, projection synthesis, and statistical units . . . 24
D.2 Training and evaluation . . . 25
D.3 Original task settings of the evaluated baselines . . . 25
D.4 Baseline implementations and comparison protocol . . . 26
D.5 Complete ablation results . . . 26
D.6 Calibration perturbations and complete robustness results . . . 27
D.7 Model size and computational cost . . . 28
E Single-Dataset Qualitative Comparisons . . . 29
F Extended ASOCA Transfer Results . . . 30
F.1 Complete adaptation curves and numerical results . . . 31
G Additional Related-Work Context . . . 32

The appendix separates implementation (Appendix A) from mathematical analysis (Appendices B–C). Experimental protocols and results appear in Appendices D–F, with additional related-work context in Appendix G.

A METHOD IMPLEMENTATION DETAILS

This section specifies BiAAR's inputs, branch operations, losses, and numerical settings. It expands the implementation summarized in Section 3; the factorization proofs and design-property analysis are presented separately in Appendices B and C. Encoder parameters are shared between views within a branch, while the three branches have independent parameters.

A.1 INPUTS AND TARGET CONSTRUCTION

World coordinates use voxel-grid units, with the N^3 grid $\mathcal{G}$ centered at the origin. The binary masks M_v and calibrated camera geometry $\mathcal{C} = (\mathbf{c}_{1:2}, f)$ determine the inputs. For each view, foreground detector rays define a binary support $S_v \in \{0,1\}^{N\times N\times N}$ in the reconstruction grid.

The SDF input is positive inside the mask and negative outside:

$$D_v = \frac{\mathrm{clip}(d_{\mathrm{in},v} - d_{\mathrm{out},v}, -\tau_D, \tau_D)}{\tau_D}. \tag{A.1}$$

Here, $d_{\mathrm{in},v}$ is the foreground distance to background, and $d_{\mathrm{out},v}$ is the background distance to foreground. The corresponding distance is zero outside its region. Empty masks use $D_v = -1$ and full masks use $D_v = 1$. The two-channel input $X_v = [M_v, D_v]$ supplies a binary boundary and its smooth distance encoding, not independent observations.

For supervision, the nonnegative projection weight $w_v(\mathbf{p}, \mathbf{x})$ bilinearly splats occupied voxels onto the detector. Equation (2) gives the resulting weighted amount $\widetilde{A}_v$ and view-normalized target A_v^*. Empty views use zero targets; in the idealized target construction, $M_v(\mathbf{p}) = \mathbf{1}[A_v^*(\mathbf{p}) > 0]$. The model predicts relative amounts, whereas s_v is used only to construct the supervision target. No absolute-amount target or reference volume is supplied at inference.

A.2 FOREGROUND TOKENIZATION, ATTENTION, AND DECODING

Figure 3 details the amount-estimation branch. A shared U-Net encoder (Ronneberger et al., 2015) maps $X_{1:2}$ to feature pyramids with spatial self-attention at the bottleneck. Let $F_v^{A,0} \in \mathbb{R}^{H\times W\times c_A}$ denote its full-resolution stem features, before any decoder upsampling, where c_A is the channel count. Mask-guided foreground feature tokenization gathers

$$\Omega_v = \{\mathbf{p} : M_v(\mathbf{p}) = 1\}, \qquad T_v^{(0)} = \mathrm{Gather}_{\Omega_v}(F_v^{A,0}) \in \mathbb{R}^{n_v\times c_A}, \qquad n_v = |\Omega_v|. \tag{A.2}$$

Each token retains its detector coordinate; attention operates on these n_v foreground vectors rather than all HW pixels.

Each block applies self-view attention followed by cross-view attention. For target view v, the source is $u \in \{v, \bar{v}\}$ with $\bar{v} = 3 - v$. Suppressing block indices, let T_v contain the tokens entering the current stage. For stage $k \in \{\mathrm{self}, \mathrm{cross}\}$ and head h, separate learned projections give

$$Q_{v,h}^k = T_v W_{Q,h}^k, \qquad K_{u,h}^k = T_u W_{K,h}^k, \qquad V_{u,h}^k = T_u W_{V,h}^k. \tag{A.3}$$

Here, $W_{Q,h}^k, W_{K,h}^k, W_{V,h}^k \in \mathbb{R}^{c_A\times d_h}$ and d_h is the head dimension. For $\mathbf{p} \in \Omega_v$ and $\mathbf{q} \in \Omega_u$, the attention logit is

$$e_{v\leftarrow u,h}(\mathbf{p}, \mathbf{q}) = \frac{\langle Q_{v,h}^k(\mathbf{p}), K_{u,h}^k(\mathbf{q})\rangle}{\sqrt{d_h}} - b_{v\leftarrow u}(\mathbf{p}, \mathbf{q}), \tag{A.4}$$

where Equation (3) defines the self-view or cross-view geometric penalty. Softmax over source foreground tokens produces attention weights and value-weighted contexts:

$$\begin{aligned} a_{v\leftarrow u,h}(\mathbf{p}, \mathbf{q}) &= \frac{\exp e_{v\leftarrow u,h}(\mathbf{p}, \mathbf{q})}{\sum_{\mathbf{q}'\in\Omega_u} \exp e_{v\leftarrow u,h}(\mathbf{p}, \mathbf{q}')}, \\ O_{v\leftarrow u,h}(\mathbf{p}) &= \sum_{\mathbf{q}\in\Omega_u} a_{v\leftarrow u,h}(\mathbf{p}, \mathbf{q}) V_{u,h}^k(\mathbf{q}). \end{aligned} \tag{A.5}$$

All source foreground tokens participate; the geometric penalty does not impose a distance threshold.

Let $F_v^{A,\ell-1}$ be the full-resolution features entering block $\ell \in \{1, \dots, L\}$. The operator $\mathrm{Attn}_{v\leftarrow u}^{k,\ell}$ applies tokenization and the attention above, then concatenates heads, zero-scatters contexts, and applies an output projection, returning dense feature maps. The self-view update is

$$F_v^{A,\ell,\mathrm{s}} = F_v^{A,\ell-1} + M_v \odot \mathrm{Attn}_{v\leftarrow v}^{\mathrm{self},\ell}(F_v^{A,\ell-1}, F_v^{A,\ell-1}). \tag{A.6}$$

Both views then use their self-updated features for cross-view attention:

$$C_v^{\ell} = \mathrm{Attn}_{v\leftarrow \bar{v}}^{\mathrm{cross},\ell}(F_v^{A,\ell,\mathrm{s}}, F_{\bar{v}}^{A,\ell,\mathrm{s}}). \tag{A.7}$$

The binary 2D validity map is $J_v^{\ell}(\mathbf{p}) = M_v(\mathbf{p})\mathbf{1}[n_{\bar{v}} > 0]$, indicating that source foreground tokens exist for the target query. It is not the 3D support S_v, an epipolar intersection mask, or a confidence estimate. A convolutional fusion block Ψ_ℓ gives

$$F_v^{A,\ell} = F_v^{A,\ell,\mathrm{s}} + M_v \odot \Psi_\ell([F_v^{A,\ell,\mathrm{s}}, C_v^{\ell}, M_v, J_v^{\ell}]). \tag{A.8}$$

Self- and cross-view QKV projections are separate, but each stage shares parameters between the two views. Different blocks have separate parameters. If either token set is empty, the corresponding pre-projection context and validity map are zero; output-projection biases may still be nonzero. Foreground masks gate both residual updates.

All foreground-attention blocks precede decoder upsampling. After the last block, a convolutional fusion of $[F_v^{A,L}, C_v^{L}, M_v, J_v^{L}]$ forms the finest-resolution skip. Starting from the encoder bottleneck, the decoder upsamples three times: the first two stages use the original coarser encoder skips, and the final stage uses the fused finest skip. A 1×1 convolution produces logits Z_v^A, followed by

$$\widehat{A}_v = M_v \odot \sigma(Z_v^A). \tag{A.9}$$

Scalar masks are broadcast across feature channels for elementwise multiplication.

Query chunking and complexity. Self- and cross-view attention evaluate queries in chunks of at most $b_q = 512$ tokens. Every chunk retains all source foreground keys and values, and softmax is taken over that complete key set. Chunking therefore evaluates the stated attention without a local window or key subsampling; it does not change its mathematical output. Although Equation (A.3) expresses projections on gathered tokens, the implementation applies equivalent dense 1×1 QKV projections before gathering.

For one paired-view sample, the two self-view calls have $n_1^2 + n_2^2$ query–key pairs and the two cross-view calls have $2n_1n_2$. Across L blocks, including dense projections and convolutional fusion, their computation scales as

$$\mathcal{O}\big(L\left[HWc_A^2 + c_A(n_1+n_2)^2\right]\big). \tag{A.10}$$

Foreground tokenization reduces pairwise work relative to attending over all image pixels, but the dependence on foreground count remains quadratic. For n_h heads, one directional query/key call needs an inference score/weight workspace of

$$\mathcal{O}(n_h \min(b_q, n_q) n_k), \tag{A.11}$$

where n_q and n_k denote its query and key counts. Thus chunking bounds the size of each attention matrix and helps avoid materializing it in full during inference; it does not reduce total pairwise arithmetic. These bounds exclude encoder/decoder features and the separate dense bottleneck attention. The latter has $m = (H/8)(W/8)$ spatial tokens and $c_B = 256$ channels, with pairwise computation $\mathcal{O}(c_B m^2)$ and attention storage $\mathcal{O}(n_h m^2)$ for image sizes divisible by eight. During training, ordinary automatic differentiation retains per-chunk attention probabilities for backpropagation, so the saved foreground-attention probability tensors still scale as $\mathcal{O}(Ln_h(n_1+n_2)^2)$. Equation (A.11) is not a bound on total training memory. Consequently, runtime and memory depend on mask occupancy as well as image resolution; chunking trades matrix size against loop overhead without guaranteeing a speedup.

A.3 Camera geometry, backprojection, and the allocation prior

The shared focal length and optical axes aimed at the grid origin simplify our synthetic acquisition model; they are not universal constraints on clinical biplane systems. Within this model, camera centers determine the extrinsic rotations as follows.

The look-at rotation uses world-up $(0,0,1)^\mathsf{T}$, switching to $(0,1,0)^\mathsf{T}$ when its absolute alignment with the optical axis exceeds 0.99. The camera-right axis is the normalized cross product of world-up and the forward axis. The camera-up axis completes the orthonormal frame, and these axes form the rows of $\mathbf{R}_v$. Camera coordinates satisfy $\mathbf{x}_v^c = \mathbf{R}_v(\mathbf{x} - \mathbf{c}_v)$. Writing $\mathbf{x}_v^c = (x_v^c, y_v^c, z_v^c)^\mathsf{T}$, the detector projection is

$$\pi_v(\mathbf{x}) = \begin{pmatrix} fx_v^c/z_v^c + (W-1)/2 \\ -fy_v^c/z_v^c + (H-1)/2 \end{pmatrix}, \qquad z_v^c > \varepsilon_z. \tag{A.12}$$

Detector coordinates are ordered as horizontal position followed by vertical position, whereas image arrays use row–column indexing. The vertical minus sign accounts for this convention.

For a target pixel $\mathbf{p}$, project two distinct points on its camera ray into source view u. Their homogeneous image coordinates $\widetilde{\mathbf{q}}_0, \widetilde{\mathbf{q}}_1$ define the epipolar line

$$\ell_{u \leftarrow v}(\mathbf{p}) = \widetilde{\mathbf{q}}_0 \times \widetilde{\mathbf{q}}_1 = (\ell_1, \ell_2, \ell_3)^\mathsf{T}, \qquad d_{\mathrm{epi}}(\mathbf{q}, \ell) = \frac{|\ell_1 q_x + \ell_2 q_y + \ell_3|}{\sqrt{\ell_1^2 + \ell_2^2}}. \tag{A.13}$$

This formula assumes distinct projected points and a nondegenerate line. The implementation selects two camera-axis depths spanning a sphere that encloses the reconstruction grid. It clamps the line-normal denominator to 10^{-6} for numerical safety.

The allocation-prior branch encodes each X_v with a shared 2D U-Net Φ_P:

$$F_v^P = M_v \odot \Phi_P(X_v). \tag{A.14}$$

The calibrated backprojection operator samples these mask-gated features at the projected voxel centers:

$$\mathcal{B}_v(F_v)(\mathbf{x}) = S_v(\mathbf{x})\, \mathbf{1}[(\mathbf{x}_v^c)_3 > \varepsilon_z]\ \mathrm{Bilin}(F_v, \pi_v(\mathbf{x})). \tag{A.15}$$

The bilinear sampler uses zero padding and aligned image corners; $\varepsilon_z > 0$ excludes invalid camera depths. The concatenated lifted features feed $\mathcal{N}_P$ in Equation (4). The resulting $P = \sigma(Z^P)$ has the same N^3 resolution as the final reconstruction and is supervised directly by Y.

Refinement uses the full 3D prior directly, without ray-wise normalization. Appendix B.3 defines the induced allocation for mathematical analysis, and Appendix C.2 establishes its connection to occupancy supervision.

A.4 Prior-guided fusion and refinement

The refiner's image-domain input has the fixed channel order $[\widehat{A}_v, M_v, D_v]$. Its 2D U-Net Φ_R shares parameters across views but not with the other branches. Equation (5) produces the concatenated backprojected features V^R. The prior defines

$$G = \gamma + (1-\gamma)P, \qquad U = P \odot (1-P), \qquad 0 < \gamma \le 1. \tag{A.16}$$

The scalar gate is broadcast across feature channels. The refiner receives $[V^R, V^R \odot G, P, U]$, retaining raw evidence alongside prior-weighted evidence. It adds a residual correction to $\lambda_Z \bar{Z}^P$, where $\bar{Z}^P = \mathrm{clip}(Z^P, -c_Z, c_Z)$, as specified in Equation (6). Neither the intermediate prior nor the final occupancy undergoes a hard reprojection-consistency step. The gate's lower bound and the clipped prior-logit bound are analyzed separately in Appendix C.4.

A.5 Training losses

The volumetric objective combines binary cross-entropy with logits, soft Dice, and soft clDice:

$$\mathcal{L}_V(Z, Y) = \lambda_{\mathrm{bce}}\mathcal{L}_{\mathrm{bce}}(Z, Y) + \lambda_{\mathrm{dice}}\mathcal{L}_{\mathrm{dice}}(\sigma(Z), Y) + \lambda_{\mathrm{cl}}\mathcal{L}_{\mathrm{cl}}(\sigma(Z), Y). \tag{A.17}$$

For one volume, use $\widehat{Y} = \sigma(Z)$ and let $\varepsilon = 10^{-6}$ stabilize overlap ratios. Binary cross-entropy and soft Dice are

$$\mathcal{L}_{\mathrm{bce}}(Z, Y) = -\frac{1}{|\mathcal{G}|} \sum_{\mathbf{x} \in \mathcal{G}} \Big[Y(\mathbf{x}) \log \widehat{Y}(\mathbf{x}) + (1 - Y(\mathbf{x})) \log(1 - \widehat{Y}(\mathbf{x})) \Big], \tag{A.18}$$

$$\mathcal{L}_{\mathrm{dice}}(\widehat{Y}, Y) = 1 - \frac{2\sum_{\mathbf{x}} \widehat{Y}(\mathbf{x}) Y(\mathbf{x}) + \varepsilon}{\sum_{\mathbf{x}} \widehat{Y}(\mathbf{x}) + \sum_{\mathbf{x}} Y(\mathbf{x}) + \varepsilon}. \tag{A.19}$$

Binary cross-entropy is implemented directly on logits for numerical stability, with positive-class weight one. Let $\mathrm{Skel}(\widehat{Y})$ and $\mathrm{Skel}(Y)$ denote soft skeletons. Topology precision and sensitivity are

$$t_{\mathrm{prec}} = \frac{\sum_{\mathbf{x}} \mathrm{Skel}(\widehat{Y})(\mathbf{x}) Y(\mathbf{x})}{\max(\varepsilon, \sum_{\mathbf{x}} \mathrm{Skel}(\widehat{Y})(\mathbf{x}))}, \qquad t_{\mathrm{sens}} = \frac{\sum_{\mathbf{x}} \mathrm{Skel}(Y)(\mathbf{x}) \widehat{Y}(\mathbf{x})}{\max(\varepsilon, \sum_{\mathbf{x}} \mathrm{Skel}(Y)(\mathbf{x}))}. \tag{A.20}$$

Our implementation uses the following stabilized variant of the soft clDice loss (Shit et al., 2021):

$$\mathcal{L}_{\mathrm{cl}}(\widehat{Y}, Y) = 1 - \frac{2 t_{\mathrm{prec}} t_{\mathrm{sens}} + \varepsilon}{t_{\mathrm{prec}} + t_{\mathrm{sens}} + \varepsilon}. \tag{A.21}$$

For nonempty Y, an exactly zero probability volume $\widehat{Y} \equiv 0$ gives $t_{\mathrm{prec}} = t_{\mathrm{sens}} = 0$ and hence $\mathcal{L}_{\mathrm{cl}} = 0$. This degeneracy concerns the clDice term alone; binary cross-entropy still penalizes missed foreground voxels. Removing the numerator's ε would give unit clDice loss in this case, but would change the objective used for the reported experiments. The prior loss substitutes Z^P for Z and P for $\widehat{Y}$. Soft skeletonization uses differentiable min/max-pooling erosion and opening for three iterations. Volumetric losses are averaged across batch items.

The amount loss pools foreground pixels across both views:

$$\mathcal{L}_A(\widehat{A}_{1:2}, A^*_{1:2}) = \frac{\sum_{v=1}^{2} \sum_{\mathbf{p} \in \Omega_v} \rho_\beta(\widehat{A}_v(\mathbf{p}) - A^*_v(\mathbf{p}))}{\max(1, n_1 + n_2)}. \tag{A.22}$$

Here, $\rho_\beta(e) = e^2/(2\beta)$ for $|e| < \beta$, and $\rho_\beta(e) = |e| - \beta/2$ otherwise. Each foreground pixel receives equal weight; views are not weighted equally regardless of foreground size. For larger batches, the numerator and foreground count are pooled across all items before division.

The complete objective is Equation (7). In the main configuration, all three branches are randomly initialized and jointly optimized, so the final volumetric loss also updates the amount and allocation-prior branches.

A.6 Numerical settings and inference

The reconstruction grid is 128^3. The following dimensions are the defaults of the end-to-end structural-prior training entry point. The amount encoder has channel widths $(32, 64, 128, 256)$ at resolutions (H, W), $(H/2, W/2)$, $(H/4, W/4)$, and $(H/8, W/8)$. Foreground attention uses $c_A = 32$, $L = 3$ self–cross blocks, and $n_h = 4$ heads with $d_h = 8$. Bottleneck spatial self-attention instead uses 256 channels and four heads of width 64. The finest skip fusion maps $2c_A + 2 = 66$ channels to 32. Three stride-two transposed-convolution stages have output widths $(128, 64, 32)$, each followed by skip concatenation and a two-convolution block. A final 1×1 convolution maps 32 channels to one amount-logit channel.

Both Φ_P and Φ_R use a 2D U-Net with encoder widths $(8, 16, 32, 64)$, decoder widths $(32, 16, 8)$, and eight output channels per view. Their respective inputs have two and three channels. Both 3D networks use encoder widths $(16, 32, 64, 128, 256)$ and decoder widths $(128, 64, 32, 16)$, with one output-logit channel. The prior network $\mathcal{N}_P$ receives $8 + 8 = 16$ backprojected channels. The residual refiner $\mathcal{N}_R$ receives $16 + 16 + 1 + 1 = 34$ channels from $[V^R, V^R \odot G, P, U]$.

The geometric bias scales are $\sigma_{\mathrm{self}} = \sigma_{\mathrm{cross}} = 1$ pixel, and both foreground-attention stages use query chunks of 512. Camera-depth validity uses $\varepsilon_z = 10^{-6}$, and SDF clipping uses $\tau_D = 16$ pixels.

Residual fusion uses $\gamma = 0.1$, $\lambda_Z = 1$, and $c_Z = 8$, with prior temperature one and no prior dilation. Loss weights are $(\lambda_{\mathrm{bce}}, \lambda_{\mathrm{dice}}, \lambda_{\mathrm{cl}}) = (0.5, 0.5, 0.1)$ and $\lambda_A = \lambda_P = 1$; smooth-$L_1$ uses $\beta = 0.5$.

Training uses AdamW with cosine learning-rate decay and selects the checkpoint with the highest validation Dice. Dataset-specific schedules and settings are given in Section 4.1. At inference, the three branches run without target amounts or occupancy labels, and $\widehat{Y}$ is thresholded at 0.5 for binary reconstruction.

B Mathematical Analysis: Amount–Allocation Factorization

This appendix establishes the sufficiency of exact amount and allocation targets for binary 3D reconstruction, as introduced in Section 3.1. Sections B.1–B.2 establish continuous and discrete

factorization identities. Section B.3 connects the implemented outputs $\widehat{A}_v$, Z^P, and P to these factors and proves recovery from normalized amounts. We retain the Method's calibrated weights w_v and distinguish exact allocations α_v from prior-induced allocations α_v^P. Exact recovery assumes unquantized targets and calibrated coverage; implementation details and approximation bounds appear separately in Appendices A and C.

B.1 Exact factorization in camera and world coordinates

Consider a bounded reconstruction region entirely in front of a calibrated camera and within its field of view. In the ideal continuous-detector model, let $\mathbf{d}_v^c(\mathbf{p})$ be the unit camera-frame direction associated with detector coordinate $\mathbf{p}$. It is determined by f, the centered principal point, and the vertical-coordinate convention in Appendix A.3. Parameterizing distance along the ray by $t > 0$ gives

$$\mathbf{x}_v(\mathbf{p}, t) = \mathbf{c}_v + \mathbf{R}_v^{\mathsf{T}}\big(t\mathbf{d}_v^c(\mathbf{p})\big), \qquad y_v(\mathbf{p}, t) = Y\big(\mathbf{x}_v(\mathbf{p}, t)\big). \tag{B.1}$$

Here, Y denotes a measurable occupancy field, or a piecewise-constant extension of the voxel volume. Let $I_v(\mathbf{p})$ contain the ray distances inside the reconstruction region. Define its absolute amount and normalized allocation by

$$\widetilde{A}_v^{\mathrm{ray}}(\mathbf{p}) = \int_{I_v(\mathbf{p})} y_v(\mathbf{p}, t)\,\mathrm{d}t, \qquad \alpha_v^{\mathrm{ray}}(t \mid \mathbf{p}) = \frac{y_v(\mathbf{p}, t)}{\widetilde{A}_v^{\mathrm{ray}}(\mathbf{p})} \quad \text{if } \widetilde{A}_v^{\mathrm{ray}}(\mathbf{p}) > 0. \tag{B.2}$$

On a zero-amount ray, set $\alpha_v^{\mathrm{ray}} = 0$ by convention. On every positive-amount ray, $\alpha_v^{\mathrm{ray}} \geq 0$ and $\int_{I_v(\mathbf{p})} \alpha_v^{\mathrm{ray}}(t \mid \mathbf{p})\,\mathrm{d}t = 1$.

Proposition 1 (information equivalence). Under the stated coverage assumptions, exact absolute amounts and complete normalized allocations determine Y almost everywhere, and Y determines both factors. The equivalence is between occupancy fields and admissible factor pairs, with allocation fixed to zero on zero-amount rays.

Proof. Every point in the covered region has a unique detector coordinate $\mathbf{p} = \pi_v(\mathbf{x})$ and distance $t = \|\mathbf{x} - \mathbf{c}_v\|_2$. Equation (B.2) therefore gives

$$Y(\mathbf{x}) = \widetilde{A}_v^{\mathrm{ray}}\big(\pi_v(\mathbf{x})\big)\,\alpha_v^{\mathrm{ray}}\big(\|\mathbf{x} - \mathbf{c}_v\|_2 \mid \pi_v(\mathbf{x})\big). \tag{B.3}$$

Nonnegativity ensures that zero amount implies zero occupancy almost everywhere on that ray. Conversely, take a nonnegative amount and an allocation integrating to one on each positive-amount ray. If their product lies in $[0, 1]$, Equation (B.3) defines an admissible occupancy field. Integrating its restriction to each ray recovers the prescribed amount; dividing by that amount recovers its allocation. The calibrated coordinate transformation is invertible, completing the equivalence. □

This is an occupancy-field identity, not a globally normalized probability density over 3D locations. The amount has units of length, and allocation has units of inverse length; their product is dimensionless occupancy. Rotation and translation preserve this scalar field under coordinate change. A probability density expressed with respect to detector coordinates and ray distance would additionally require the corresponding volume Jacobian. Exact continuous ray profiles are also stronger information than finitely sampled detector pixels; the discrete counterpart follows below.

B.2 Discrete factorization for the Method's splatting operator

The simulator uses bilinear splatting, not the continuous line integral in Equation (B.2). Using precisely the weights from Equation (2), define

$$\widetilde{A}_v(\mathbf{p}) = \sum_{\mathbf{x} \in \mathcal{G}} w_v(\mathbf{p}, \mathbf{x}) Y(\mathbf{x}), \qquad \alpha_v(\mathbf{x} \mid \mathbf{p}) = \frac{w_v(\mathbf{p}, \mathbf{x}) Y(\mathbf{x})}{\widetilde{A}_v(\mathbf{p})} \quad \text{if } \widetilde{A}_v(\mathbf{p}) > 0. \tag{B.4}$$

Again, set $\alpha_v = 0$ on zero-amount pixels. Otherwise, $\sum_{\mathbf{x} \in \mathcal{G}} \alpha_v(\mathbf{x} \mid \mathbf{p}) = 1$. The allocation describes each voxel's share of a pixel's amount, including its interpolation weight. Binary ray-incidence weights would give a hit-voxel count; the Method's bilinear weights instead define a weighted voxel amount.

Proposition 2 (exact voxel recovery). Let $\kappa_v(\mathbf{x}) = \sum_{\mathbf{p}} w_v(\mathbf{p}, \mathbf{x})$, with the sum over detector pixels. For every voxel with $\kappa_v(\mathbf{x}) > 0$, the factors in Equation (B.4) recover its occupancy exactly:

$$Y(\mathbf{x}) = \frac{\sum_{\mathbf{p}} \widetilde{A}_v(\mathbf{p}) \alpha_v(\mathbf{x} \mid \mathbf{p})}{\kappa_v(\mathbf{x})}. \tag{B.5}$$

Proof. For each pixel, $\widetilde{A}_v(\mathbf{p})\alpha_v(\mathbf{x} \mid \mathbf{p}) = w_v(\mathbf{p}, \mathbf{x})Y(\mathbf{x})$, including zero-amount pixels by nonnegativity. Summing over pixels yields $\kappa_v(\mathbf{x})Y(\mathbf{x})$, proving Equation (B.5). Conversely, a given Y determines both factors through Equation (B.4). Thus the representation is lossless on covered voxels. □

The divisor accounts for boundary-truncated splatting weights that need not sum to one. Summing the numerator and denominator over v extends recovery to voxels covered by either view; uncovered voxels remain unconstrained. Arbitrary predictions need not describe a common volume: their products must agree with the same $w_v(\mathbf{p}, \mathbf{x})Y(\mathbf{x})$ across pixels and views.

B.3 Sufficiency of normalized amount and the allocation prior

Step 1: implemented outputs and supervised targets. The amount branch predicts mask-gated normalized amounts, while the allocation branch predicts full-resolution occupancy logits:

$$\widehat{A}_v = M_v \odot \sigma(Z_v^A), \qquad P = \sigma(Z^P), \qquad A_v^* = \frac{\widetilde{A}_v}{s_v}, \qquad P^* = Y. \tag{B.6}$$

Here, $s_v = \max_{\mathbf{p}} \widetilde{A}_v(\mathbf{p}) > 0$ for nonempty views; empty views use zero amount targets. The amount loss compares $\widehat{A}_v$ with A_v^*, and the prior's volumetric loss compares $\sigma(Z^P)$ with Y. No separate ray-normalized allocation target or absolute-amount prediction head is used.

The logits Z^P parameterize P rather than define an independent allocation. For finite logits, $Z^P = \log(P/(1-P))$ elementwise, so either representation determines the other. Finite sigmoid outputs lie strictly between zero and one. Thus $P = Y$ and the unit maximum of A_v^* describe ideal targets or limiting predictions, not exact outputs guaranteed by finite logits. The proof below concerns this exact-target limit.

Step 2: the prior induces allocation. The calibrated transformation gives $P_v^c(\mathbf{x}_v^c) = P(\mathbf{c}_v + \mathbf{R}_v^\mathsf{T}\mathbf{x}_v^c)$. Restricting this scalar occupancy field to a ray and normalizing its profile produces a camera-frame allocation. For the implemented splatting weights, the corresponding voxel contributions are

$$\alpha_v^P(\mathbf{x} \mid \mathbf{p}) = \frac{w_v(\mathbf{p}, \mathbf{x})P(\mathbf{x})}{\sum_{\mathbf{x}' \in \mathcal{G}} w_v(\mathbf{p}, \mathbf{x}')P(\mathbf{x}')}, \tag{B.7}$$

with zero assigned when the denominator vanishes. At $P = P^* = Y$, the denominator is $\widetilde{A}_v(\mathbf{p})$, giving $\alpha_v^P = \alpha_v$ on every positive-amount pixel. This also holds for $P = c_P Y$ with any constant $c_P > 0$ that keeps P in $[0, 1]$. Allocation therefore specifies relative contributions, not the prior's absolute amplitude. This normalization is an analysis operator; the implemented refiner receives P directly.

Proposition 3 (normalized-intermediate sufficiency). Let $\mathcal{G}_v = \{\mathbf{x} \in \mathcal{G} : \kappa_v(\mathbf{x}) > 0\}$, where $\kappa_v(\mathbf{x}) = \sum_{\mathbf{p}} w_v(\mathbf{p}, \mathbf{x})$. Suppose Y is binary, the weights are known and nonnegative, and $\widehat{A}_v = A_v^*$. On positive-amount pixels, suppose additionally that $\alpha_v^P = \alpha_v$. Then $(\widehat{A}_v, P, \mathcal{C})$ determines Y on $\mathcal{G}_v$ and determines $\widetilde{A}_v$. The complete volume is determined if $\mathcal{G}_1 \cup \mathcal{G}_2 = \mathcal{G}$ and these conditions hold for both views. BiAAR's exact supervision targets satisfy these sufficient conditions.

Proof, Step 3: recover occupancy up to a view-wise scale. For a nonempty view and each positive-amount pixel, the assumed factors obey

$$\widehat{A}_v(\mathbf{p})\alpha_v^P(\mathbf{x} \mid \mathbf{p}) = \frac{\widetilde{A}_v(\mathbf{p})}{s_v} \frac{w_v(\mathbf{p}, \mathbf{x})Y(\mathbf{x})}{\widetilde{A}_v(\mathbf{p})} = \frac{w_v(\mathbf{p}, \mathbf{x})Y(\mathbf{x})}{s_v}. \tag{B.8}$$

For zero-amount pixels, nonnegative weights imply $w_v(\mathbf{p}, \mathbf{x})Y(\mathbf{x}) = 0$, and $\widehat{A}_v(\mathbf{p}) = 0$. The identity therefore holds regardless of the prior-induced allocation on those pixels. Summing over pixels and dividing by coverage yields

$$\widetilde{Y}_v(\mathbf{x}) = \frac{\sum_{\mathbf{p}} \widehat{A}_v(\mathbf{p})\alpha_v^P(\mathbf{x} \mid \mathbf{p})}{\kappa_v(\mathbf{x})}, \qquad \mathbf{x} \in \mathcal{G}_v. \tag{B.9}$$

Equation (B.8) gives $\widetilde{Y}_v(\mathbf{x}) = Y(\mathbf{x})/s_v$. The divisor κ_v accounts for interpolation weights, including any truncation at detector boundaries.

Step 4: recover scale and combine covered regions. A nonempty view contains at least one covered occupied voxel, so binary occupancy has maximum one on $\mathcal{G}_v$. Consequently, $r_v = \max_{\mathbf{x}\in\mathcal{G}_v} \widetilde{Y}_v(\mathbf{x}) = 1/s_v$, and

$$Y(\mathbf{x}) = \frac{\widetilde{Y}_v(\mathbf{x})}{r_v}, \qquad s_v = \frac{1}{r_v}, \qquad \widetilde{A}_v(\mathbf{p}) = \frac{\widehat{A}_v(\mathbf{p})}{r_v}. \tag{B.10}$$

If a view is empty, nonnegative weights and zero amounts imply $Y = 0$ throughout $\mathcal{G}_v$; no division by $r_v = 0$ is performed. Applying the construction separately to both views gives identical occupancy on their overlap and recovers Y throughout their coverage union. Equation (B.9) already uses world-grid indices. Equivalently, camera-frame occupancy maps back by $\mathbf{x} = \mathbf{c}_v + \mathbf{R}_v^{\mathsf{T}}\mathbf{x}_v^c$, with no further inverse problem. This proves the stated sufficiency. □

Equivalence of the supervised representation. Let $\mathcal{E}_{\mathcal{C}}(Y) = (A^*_{1:2}(Y), \alpha_{1:2}(Y))$ denote the normalized amount–allocation factor pair. Under complete combined coverage, Proposition 3 constructs a decoder $\mathcal{R}_{\mathcal{C}}$ satisfying

$$\mathcal{R}_{\mathcal{C}}\big(\mathcal{E}_{\mathcal{C}}(Y)\big) = Y. \tag{B.11}$$

Identical factor pairs therefore imply identical binary volumes, and Y conversely determines both factors. Step 2 maps the supervised prior target $P^* = Y$ to the required allocation. Hence $(A^*_{1:2}, P^*, \mathcal{C})$ is also sufficient, establishing the information-level validity of the implemented intermediate targets.

Finite predictions and the target limit. For a fixed nonempty view, consider predictions with $\widehat{A}_v \to A_v^*$ and $P \to Y$ on the finite grids. Positive-amount denominators then stay bounded away from zero, so $\alpha_v^P \to \alpha_v$ there; Appendix C.2 quantifies this relationship. On zero-amount pixels, allocation need not converge, but its components remain in $[0, 1]$ and their products with $\widehat{A}_v \to 0$ vanish. Thus $\widetilde{Y}_v \to Y/s_v$ and $r_v \to 1/s_v > 0$, implying convergence of the constructive recovery to Y. This is consistency of the analytical decoder under convergent inputs, not a guarantee that training produces those inputs.

Connection to the learned refiner. The proof constructs a decoder without a separate absolute-amount input; it does not assert that BiAAR explicitly predicts s_v or executes this decoder. BiAAR instead learns fusion and residual refinement from the approximate intermediates and mask–SDF evidence. The clipped logits $\bar{Z}^P$ contribute only to residual fusion; the allocation argument uses the full prior P, not clipped logits. Coordinate transformation does not remove prediction errors, and intermediate supervision does not guarantee exact reconstruction.

Scale recovery uses the known unit maximum supplied by binary occupancy; without an amplitude convention, soft fields Y and cY have identical normalized factors. Sufficiency also does not imply minimality or that both branches are necessary: the ideal prior $P^* = Y$ already contains the complete structure. The decomposition's practical value is learning complementary, imperfect intermediates, as tested in Section 4.3. The proof does not make those intermediates uniquely identifiable from two observed masks.

C Mathematical Analysis: Design Properties

The following properties justify the representation and clarify its limitations. They do not prove that either branch is mathematically necessary or guarantee a performance improvement from a particular architecture.

C.1 What view-wise amount normalization preserves and expands

Consider a nonempty view with nonnegative absolute amounts $\widetilde{A}_v$ and maximum $s_v > 0$. Define $\mathrm{Norm}_v(\widetilde{A}_v) = \widetilde{A}_v / s_v$, as in Equation (2).

Proposition 4 (dynamic-range use and scale invariance). Normalization preserves zero support, ordering, and ratios between positive pixel values. Its maximum is one, and it is invariant to a common positive multiplicative factor. Compared with division by any fixed scale $K \geq s_v$, it expands pairwise absolute differences by K/s_v:

$$|A_v^*(\mathbf{p}) - A_v^*(\mathbf{q})| = \frac{K}{s_v} \left| \frac{\widetilde{A}_v(\mathbf{p})}{K} - \frac{\widetilde{A}_v(\mathbf{q})}{K} \right|. \tag{C.1}$$

Proof. Division by the same positive scalar preserves zeros, order, and positive-value ratios. The maximum becomes $s_v/s_v = 1$. For any $c > 0$, $\max_{\mathbf{p}} c\widetilde{A}_v(\mathbf{p}) = cs_v$, so

$$\mathrm{Norm}_v(c\widetilde{A}_v) = \frac{c\widetilde{A}_v}{cs_v} = \mathrm{Norm}_v(\widetilde{A}_v). \tag{C.2}$$

Equation (C.1) follows by factoring the difference of two normalized values. The expansion is strict when $K > s_v$ and the two values differ. □

Here, enhanced contrast means larger absolute differences within the numerical target range relative to fixed-scale normalization. It does not mean increased information, improved signal-to-noise ratio, or improved scale-invariant contrast measures. The foreground minimum need not be zero; the full image spans $[0, 1]$ when background pixels are present. Removing the absolute scale also prevents normalized amounts alone from specifying physical thickness or absolute voxel amount.

Sensitivity to perturbations. Let $\widetilde{A}'_v$ be nonnegative with $\|\widetilde{A}'_v - \widetilde{A}_v\|_\infty \leq \epsilon < s_v$, and write $s'_v = \max_{\mathbf{p}} \widetilde{A}'_v(\mathbf{p})$. Then $|s'_v - s_v| \leq \epsilon$ and

$$\left\| \frac{\widetilde{A}'_v}{s'_v} - \frac{\widetilde{A}_v}{s_v} \right\|_\infty \leq \frac{\epsilon}{s'_v} + \frac{|s'_v - s_v|}{s'_v} \leq \frac{2\epsilon}{s_v - \epsilon}. \tag{C.3}$$

The first inequality follows by adding and subtracting $\widetilde{A}_v/s'_v$ and using $\|\widetilde{A}_v\|_\infty = s_v$. Small maxima can therefore amplify perturbations; normalization is not a denoising operation. Proposition 3 recovers the discarded scale only with additional exact allocation information and the binary-occupancy constraint.

C.2 Why occupancy supervision supplies an allocation target

The world-space prior is a scalar occupancy field, rather than a probability density over 3D positions. Its camera-frame expression is

$$P_v^c(\mathbf{x}_v^c) = P(\mathbf{c}_v + \mathbf{R}_v^\mathsf{T} \mathbf{x}_v^c), \qquad P(\mathbf{x}) = P_v^c(\mathbf{R}_v(\mathbf{x} - \mathbf{c}_v)). \tag{C.4}$$

Since $\mathbf{R}_v^\mathsf{T}\mathbf{R}_v = I$, these transformations are inverses before discrete resampling. No density Jacobian is needed for this scalar-field identity. Along the calibrated ray from Equation (B.1), let $a_v^P(\mathbf{p}) = \int_{I_v(\mathbf{p})} P_v^c(t'\mathbf{d}_v^c(\mathbf{p}))\,\mathrm{d}t'$. Its normalized profile is

$$\alpha_v^{P,\mathrm{ray}}(t \mid \mathbf{p}) = \begin{cases} P_v^c(t\mathbf{d}_v^c(\mathbf{p}))/a_v^P(\mathbf{p}), & a_v^P(\mathbf{p}) > 0, \\ 0, & a_v^P(\mathbf{p}) = 0. \end{cases} \tag{C.5}$$

The discrete weighted counterpart is Equation (B.7).

At the occupancy target $P = Y$, both expressions produce the exact allocation. Thus world-space occupancy supervision is sufficient to specify the desired ray-wise allocation in every calibrated view. However, normalization is generally many-to-one: P and $c_P P$ induce identical allocations for any admissible $c_P > 0$. Coordinate transformation is invertible; removing each ray's amount through

normalization is not. Consequently, P carries allocation information but should not itself be identified with normalized α_v^P. In particular, the volumetric loss on P supervises both support and occupancy amplitude, whereas a loss on normalized allocations cannot distinguish these globally rescaled priors. The objectives are therefore not equivalent: occupancy supervision supplies a stronger target, whose exact solution already determines the full volume.

Proposition 5 (allocation error induced by an approximate prior). Fix a foreground pixel with true amount $a = \widetilde{A}_v(\mathbf{p}) > 0$ and prior amount $b = \sum_{\mathbf{x}} w_v(\mathbf{p}, \mathbf{x}) P(\mathbf{x}) > 0$. Define the weighted occupancy error

$$\delta_v(\mathbf{p}) = \sum_{\mathbf{x} \in \mathcal{G}} w_v(\mathbf{p}, \mathbf{x}) |P(\mathbf{x}) - Y(\mathbf{x})|. \tag{C.6}$$

Then the induced allocation error obeys

$$\sum_{\mathbf{x} \in \mathcal{G}} |\alpha_v^P(\mathbf{x} \mid \mathbf{p}) - \alpha_v(\mathbf{x} \mid \mathbf{p})| \le \min\left\{ 2, \frac{2\delta_v(\mathbf{p})}{\widetilde{A}_v(\mathbf{p})} \right\}. \tag{C.7}$$

Proof. For this fixed pixel, abbreviate $u_{\mathbf{x}} = w_v(\mathbf{p}, \mathbf{x}) P(\mathbf{x})$ and $t_{\mathbf{x}} = w_v(\mathbf{p}, \mathbf{x}) Y(\mathbf{x})$. Adding and subtracting $u_{\mathbf{x}}/a$ gives

$$\begin{aligned} \sum_{\mathbf{x}} \left| \frac{u_{\mathbf{x}}}{b} - \frac{t_{\mathbf{x}}}{a} \right| &\le \frac{\sum_{\mathbf{x}} |u_{\mathbf{x}} - t_{\mathbf{x}}|}{a} + \left| \frac{1}{b} - \frac{1}{a} \right| \sum_{\mathbf{x}} u_{\mathbf{x}} \\ &= \frac{\delta_v(\mathbf{p})}{a} + \frac{|a - b|}{a} \le \frac{2\delta_v(\mathbf{p})}{a}. \end{aligned} \tag{C.8}$$

The final inequality uses $|a - b| \le \delta_v(\mathbf{p})$. The alternative bound of two follows because both allocations are nonnegative and sum to one. □

This bound connects voxelwise prior accuracy to allocation accuracy without assuming exact binary predictions. It becomes weak on low-amount rays and does not cover zero-amount rays as normalized foreground distributions. A lower aggregate training loss alone does not guarantee lower error on every ray. Moreover, resampling onto a different finite grid introduces interpolation error, unlike the exact coordinate identity in Equation (C.4).

C.3 How prior gating implements allocation guidance

The preceding results concern information and supervision. Here we connect allocation to the actual feature operation in Equation (6), without identifying the refiner with the analytical decoder. Write the calibrated bilinear backprojection from Equation (A.15) as

$$\mathcal{B}_v(F)(\mathbf{x}) = m_v(\mathbf{x}) \sum_{\mathbf{p}} h_v(\mathbf{p}, \mathbf{x}) F(\mathbf{p}), \qquad m_v(\mathbf{x}) = S_v(\mathbf{x})\, \mathbf{1}[(\mathbf{x}_v^c)_3 > \varepsilon_z]. \tag{C.9}$$

Here $h_v \ge 0$ denotes the actual bilinear interpolation weights, including zero padding; the validity and support gate m_v remains explicit. We distinguish these gather weights from the forward splatting weights w_v. Define the prior-derived amount and allocation for this operator:

$$b_v^P(\mathbf{p}) = \sum_{\mathbf{x} \in \mathcal{G}} h_v(\mathbf{p}, \mathbf{x}) P(\mathbf{x}), \qquad \alpha_v^{P,\mathcal{B}}(\mathbf{x} \mid \mathbf{p}) = \frac{h_v(\mathbf{p}, \mathbf{x}) P(\mathbf{x})}{b_v^P(\mathbf{p})}. \tag{C.10}$$

Set the allocation to zero when $b_v^P = 0$. If $h_v = w_v$, it coincides with Equation (B.7); the following identity does not require that equality. Prior-weighted features satisfy

$$P(\mathbf{x}) \mathcal{B}_v(F)(\mathbf{x}) = m_v(\mathbf{x}) \sum_{\mathbf{p}} b_v^P(\mathbf{p}) \alpha_v^{P,\mathcal{B}}(\mathbf{x} \mid \mathbf{p}) F(\mathbf{p}). \tag{C.11}$$

For $b_v^P(\mathbf{p}) > 0$, this follows by substituting $b_v^P \alpha_v^{P,\mathcal{B}} = h_v P$. If $b_v^P(\mathbf{p}) = 0$, nonnegativity gives $h_v P = 0$ at every voxel, so the same identity holds under the zero-allocation convention. The equality applies componentwise to feature vectors. The soft-gated branch therefore decomposes as

$$G \odot \mathcal{B}_v(F) = \gamma \mathcal{B}_v(F) + (1 - \gamma) P \odot \mathcal{B}_v(F). \tag{C.12}$$

Thus the implemented gate blends unweighted lifting with an allocation-weighted assignment of image features, retaining each pixel's prior-derived amplitude b_v^P. This is an exact rewriting of the gate, not an additional normalization step. It explains how the occupancy prior provides spatial allocation guidance without requiring independent allocation predictions for every ray and view.

The amplitude b_v^P comes from P, not the independently estimated $\widehat{A}_v$. Moreover, F_v^R encodes $[\widehat{A}_v, M_v, D_v]$ and is not a scalar amount. The complete refiner also receives raw features, P, U, and clipped prior logits. Consequently, Equation (C.11) establishes the allocation interpretation of feature gating, not equality between the learned reconstruction and an amount–allocation product. Rescaling P preserves its normalized profiles but can change the gate and logits; those amplitudes are intentionally retained by the implementation.

In summary, the factorization gives an information-preserving representation, occupancy supervision supplies an allocation-bearing intermediate, and prior-guided fusion is the learned reconstruction mechanism. Neither the factorization nor the gate identity guarantees recovery of these intermediates from binary masks. Their practical complementarity and the effect of direct supervision are evaluated separately in Section 4.3; Appendix D.5 specifies what those comparisons establish.

C.4 Why prior guidance remains soft

For $0 \le P \le 1$ and $0 < \gamma \le 1$, Equation (A.16) implies

$$\gamma \le G \le 1, \qquad 0 \le U = P(1-P) \le \tfrac{1}{4}, \qquad |\lambda_Z \bar{Z}^P| \le |\lambda_Z| c_Z. \tag{C.13}$$

The first bound follows from the affine gate; the second follows from $P(1-P) = 1/4 - (P-1/2)^2$. The third follows directly from clipping. For fixed P, multiplication by G is invertible because G is everywhere positive. More directly, concatenation preserves V^R as an unchanged input block, irrespective of the gated branch.

These facts prevent the prior-guidance stage from irreversibly deleting information already contained in V^R. They do not recover information lost earlier through masks, feature extraction, or support gating. The cue U is Bernoulli variance associated with the predicted occupancy probability, not a calibrated measure of model uncertainty. Bounded prior logits also permit, but do not ensure, corrective residuals. Robustness or accuracy gains therefore require empirical evaluation rather than following from these bounds alone.

D Extended Experimental Protocols and Results

This section records experimental procedures and complete results. Network operations appear in Appendix A; mathematical arguments appear in Appendices B and C.

D.1 Datasets, projection synthesis, and statistical units

We use ImageCAS (Zeng et al., 2023) and the 40 ASOCA training scans (Gharleghi et al., 2022). For ASOCA, we use released AVDNet predictions with separate artery and vein trees and finer arterial detail than the original annotations (Wang et al., 2024). ImageCAS contains 1,000 annotated coronary computed tomography angiography (CCTA) volumes, split into 800/100/100 for training, validation, and testing. ASOCA is split at the patient level into 32/4/4; treating its artery and vein trees as separate volumes gives 64/8/8 volumes without patient leakage.

Both datasets use the same simulator. Each pair has randomly sampled orthogonal optical axes, camera centers 160 voxel-grid units from the origin, and focal length 200 pixels. The poses render paired view-normalized ray-wise vessel-amount maps $A^*_{1:2}$ from each 3D vessel; thresholding gives $M_{1:2}$ and the SDFs $D_{1:2}$ follow from the masks. Only in the diagnostic amount oracles is $\widehat{A}$ replaced by A^*. For testing, we randomly sample 5 paired-camera configurations per ImageCAS volume and 20 per ASOCA volume. The 100 ImageCAS and 8 ASOCA test volumes yield 500 and 160 paired-view samples, respectively. Views are 256×256 and outputs are 128^3.

The ASOCA test set therefore contains four independent patients, not 160 independent patients. Standard deviations describe variation across paired-view test samples, not across independently trained models.

D.2 Training and evaluation

We train BiAAR end-to-end for 200 epochs with batch size one using AdamW and cosine learning-rate decay. The learning rate is 5×10^{-4} per branch, weight decay is 10^{-5}, and checkpoints are selected by validation Dice. Single-dataset models use random initialization and seed 42; transfer experiments instead start from ImageCAS checkpoints (Section 4.4). The predicted occupancy probability is thresholded at 0.5. We report Dice, soft clDice (Shit et al., 2021) (denoted clDice throughout), recall, precision, HD95, and ASSD; distance metrics use voxel units.

Empty-volume distance metrics. The released BiAAR evaluation code assigns HD95 and ASSD zero when both binary volumes are empty and $+\infty$ when exactly one is empty. For nonempty volumes, HD95 is the 95th percentile of the pooled bidirectional surface distances and ASSD their mean. Aggregate distance statistics in this code use finite entries only: infinite values remain in per-sample records but are omitted from the aggregate, which is undefined if no finite entries remain. These conventions describe the evaluation implementation, not evidence that any particular test case was excluded.

D.3 Original task settings of the evaluated baselines

The four baselines differ in their original observations and temporal objectives. We first summarize those settings, then specify the shared binary-mask comparison protocol in Appendix D.4.

Two Projections Suffice: static reconstruction with accumulated-intensity observations. This method reconstructs cerebral vasculature from two orthogonal projections synthesized from binary 3D vessel segmentations (Cafaro et al., 2024). Its mean-intensity ray projections retain accumulated vessel/path-length information rather than only binary support. A 3D U-Net disambiguates the backprojection, followed by maximum-a-posteriori refinement with projection consistency and connectivity regularization. The original experiments use static structures derived from magnetic resonance angiography. Reference segmentations generate the synthetic intensities during data construction; inference uses the projections, not the reference volume itself. Our adaptation instead supplies binary masks without those intensity cues.

AutoCAR: reconstruction and temporal registration. AutoCAR lifts features from two segmented angiographic views through sparse backward projection and predicts 3D coronary structure (Zhu et al., 2025). Its full framework additionally addresses pose-domain differences and dynamic reconstruction through vascular-graph matching and non-rigid registration across time. Reconstruction accuracy is therefore one component of a broader system for clinical angiographic sequences, including temporally misaligned views. Our synchronized-pair benchmark evaluates its reconstruction component rather than the performance of its temporal registration pipeline.

DeepCA: binary projections with inter-view motion. DeepCA reconstructs the right coronary artery from two binary projections by combining their backprojections and learning implicit motion compensation (Wang et al., 2025). Its training simulations perturb the vessel between projections to model non-simultaneous acquisition, and its original evaluation includes eight clinical cases. It therefore shares our binary observation type while addressing a different temporal setting. The synchronized projections used here isolate reconstruction without introducing inter-view motion.

3DGR-CAR: static attenuation reconstruction with Gaussian optimization. 3DGR-CAR initializes Gaussian centers with a learned predictor and optimizes Gaussian parameters against sparse grayscale X-ray projections (Fu et al., 2024). Its cone-beam digitally reconstructed radiographs encode attenuation, and its original experiments evaluate static ImageCAS and ASOCA structures without an explicit temporal model. The binary-mask protocol here removes the intensity information used in that original formulation; the reported results therefore characterize our evaluated binary-input implementation, not its performance with attenuation-valued observations.

3DGR-CAR was originally evaluated with two views on ImageCAS and ASOCA, with additional comparisons using 2, 4, 8, and 16 views on ImageCAS (Fu et al., 2024).

D.4 Baseline implementations and comparison protocol

We reproduce Two Projections Suffice from its paper because source code is unavailable. For AutoCAR, we use the public code and supply its missing backprojection component with the same implementation used by BiAAR: calibrated projection associates 3D voxel locations with 2D features, consistent with the paper's sparse backward-projection formulation (Zhu et al., 2025). This shared implementation controls the choice of geometric feature lifting while retaining each model's distinct backbone. DeepCA and 3DGR-CAR use their public implementations. Apart from the common observation protocol and the stated implementation choices, we retain each method's published reconstruction design. Each method is trained independently from scratch on each dataset, using the same splits and paired-view banks. Test outputs are mapped to the same 128^3 grid before evaluation.

All compared methods receive the same observation information: two binary vessel masks and calibrated cameras. Additional inputs, such as SDFs or backprojected features, are derived from those masks and geometry; no baseline receives grayscale attenuation measurements or ground-truth amount maps. The diagnostic amount oracles in the ablation study are explicitly separate. Tensor formats and processing remain method-specific, but the available evidence is shared. In particular, evaluating Two Projections Suffice and 3DGR-CAR on binary masks changes the intensity-based observation setting of their original studies. Appendix D.3 describes these original settings; our benchmark assesses reconstruction under the shared binary protocol, not every temporal or registration capability of each original system. Table 1 preserves all six metrics, including recall and precision.

D.5 Complete ablation results

Table 5 reports all twelve component configurations; Table 2 shows B0, B1, B5, B6, and B11. Mask/SDF toggles affect only refiner inputs. Active branches retain $X_{1:2}$ and camera geometry; the prior also uses binary ray support. B7–B11 replace predicted amounts with ground-truth maps.

Table 5: Complete component and refiner-input ablations on ImageCAS (500 test-pair means). Pred/GT denote predicted/ground-truth amount; B7–B11 are amount oracles. Prior metrics evaluate P before refinement. MSE_A evaluates predicted amount only; dashes denote absent outputs or oracle amount inputs. Distances use voxel units; **bold** marks the best non-oracle means within each output group.

Refiner inputs					Final reconstruction $\widehat{Y}$						Allocation prior P			$\widehat{A}$
ID	Amount	M_v	D_v	P	Dice ↑	clDice ↑	Recall ↑	Prec. ↑	HD95 ↓	ASSD ↓	Dice ↑	Recall ↑	Prec. ↑	MSE_A ↓
B0	–	✓	✓	×	.765	.835	.797	.738	2.245	.484	–	–	–	–
B1	–	✓	✓	✓	.800	.869	.818	.784	1.986	.419	.779	.800	.762	–
B2	Pred	×	×	×	.792	.866	.811	.776	**1.741**	.388	–	–	–	0.013
B3	Pred	×	✓	×	.777	.850	.794	.763	2.155	.460	–	–	–	0.012
B4	Pred	✓	×	×	.791	.865	.818	.767	1.871	.404	–	–	–	0.012
B5	Pred	✓	✓	×	.794	.867	.821	.770	1.828	.396	–	–	–	0.015
B6	Pred	✓	✓	✓	**.809**	**.879**	**.826**	**.793**	1.806	**.375**	**.789**	**.807**	**.774**	**0.011**
B7	GT	×	×	×	.908	.935	.906	.911	1.122	.158	–	–	–	–
B8	GT	×	✓	×	.907	.935	.903	.912	1.043	.149	–	–	–	–
B9	GT	✓	×	×	.924	.949	.927	.923	.984	.120	–	–	–	–
B10	GT	✓	✓	×	.924	.950	.922	.927	.940	.116	–	–	–	–
B11	GT	✓	✓	✓	.926	.950	.924	.929	.929	.120	.790	.812	.771	–

Intermediate prediction metrics. For each paired-view test sample, amount error is measured against the view-normalized targets over foreground pixels from both views:

$$\mathrm{MSE}_A = \frac{\sum_{v=1}^{2} \sum_{\mathbf{p}} M_v(\mathbf{p}) \big(\widehat{A}_v(\mathbf{p}) - A_v^*(\mathbf{p})\big)^2}{\sum_{v=1}^{2} \sum_{\mathbf{p}} M_v(\mathbf{p})}. \tag{D.1}$$

Entries average this error over 500 paired-view samples; empty foreground sets contribute zero. Foreground evaluation excludes background dilution and differs from the smooth-L_1 training objective. Prior Dice thresholds P at 0.5 and compares it with Y before refinement. Intermediate metrics remain defined when their auxiliary losses are disabled; oracle amounts are not scored as predictions.

Complete-sweep interpretation. Mask/SDF toggles (B2–B5) are nonmonotonic. B6 leads five of six non-oracle reconstruction metrics; B2 has the lowest HD95. Amount MSE alone does not rank reconstruction quality: B3 improves MSE over B5 but has lower Dice. The B11 oracle improves Dice/clDice over B6 by 0.117/0.071. With GT amounts, adding the prior changes Dice by only 0.002 (B10–B11), consistent with partly redundant cues.

Indirect supervision and intermediate semantics. With both auxiliary weights zero, the reconstruction loss still supervises both branches (Table 3). The asymmetric intermediate scores suggest co-adaptation; adding each direct loss improves its corresponding target metric. Zeroing a loss retains a branch; $(0, 1)$ and $(1, 0)$ have lower final Dice than their branch-removal counterparts B1 and B5, respectively.

What the controls establish. Component ablations show the utility of the added cues, but also change model capacity. The loss-weight sweep holds architecture, inputs, and parameter count fixed, supporting the benefit of the prescribed intermediate supervision beyond added capacity alone. Because P is supervised as occupancy, these comparisons do not isolate normalized allocation learning from generic occupancy deep supervision, nor compare learned refinement with explicit factor-product decoding. Prior Dice measures occupancy agreement, not ray-normalized allocation error. The evidence therefore supports the implemented combination of amount estimation and occupancy-based allocation guidance; it does not establish that two independently learned factors are necessary or that this is the only effective decomposition.

D.6 Calibration perturbations and complete robustness results

We perturb rotation, translation, and focal length separately on ImageCAS to simulate calibration errors. Per view, rotation and focal errors are uniform within the signed bound; translation has random direction and magnitude in $[0, t]$. Ray support is regenerated by the controlled simulator under the perturbed geometry. Each of 500 samples receives one seed-42 draw per view and condition; one resampled grid unit denotes 1 mm.

Table 6 reports the complete sweep; Table 4 retains the clean baseline and three selected settings in the main text. BP IoU measures the overlap between perturbed and clean binary ray supports, with lower values indicating greater geometric misalignment.

Table 6: Complete robustness sweep on ImageCAS (means over 500 paired-view test samples). Each error type is perturbed independently within the stated bound; Clean uses unperturbed cameras. BP IoU measures binary ray-support overlap with clean geometry; HD95 uses voxel units.

Error	Bound	BP IoU	Dice ↑	clDice ↑	HD95 ↓
Clean	–	1.000	.809	.879	1.806
Rotation	0.1°	.981	.808	.879	1.812
Rotation	0.3°	.950	.799	.872	1.893
Rotation	0.5°	.923	.785	.860	2.040
Translation	1 mm	.942	.792	.865	2.086
Translation	2 mm	.897	.758	.835	2.612
Translation	5 mm	.790	.645	.724	4.249
Focal length	1%	.929	.776	.854	2.148
Focal length	2%	.872	.710	.794	3.266
Focal length	5%	.738	.539	.609	5.862

Sensitivity and interpretation. BiAAR retains Dice/clDice of 0.785/0.860 at 0.5° rotation error, compared with 0.809/0.879 for clean cameras, despite BP IoU decreasing to 0.923. Larger errors substantially degrade reconstruction: Dice falls to 0.645 at 5 mm translation and 0.539 at 5% focal-length error. This tolerance is consistent with avoiding exact output-reprojection constraints, although the experiment does not establish a causal benefit.

Clinical calibration context. Wolff et al. evaluated static poses of a calibrated GE OEC 9800 C-arm against optical tracking. Its encoder errors were $-0.1 \pm 0.2^\circ$ and $0.1 \pm 0.2^\circ$ for the primary

and secondary angles, respectively (mean±standard deviation) (Wolff et al., 2014). These measurements provide a device-based reference for our rotation tests; the 0.5° bound additionally probes a moderate stress condition. Robustness at this sub-degree scale supports the potential of BiAAR for reconstruction with imperfect calibration, including clinical imaging settings. These simulated perturbations do not replace validation on real angiograms.

D.7 Model size and computational cost

Table 7 reports network parameter counts, floating-point operation (FLOP) estimates, and mean end-to-end runtimes on ImageCAS for the benchmark implementations. Network parameter counts agree across the two datasets; case-specific Gaussian parameters are reported separately below.

Table 7: Network parameters, counted forward operations, and mean end-to-end time per paired-view sample on ImageCAS. GFLOPs average the two dataset-specific profiling records; runtimes are ImageCAS means in milliseconds. Case-specific counts denote scalar parameters, not Gaussian primitives. Dashes indicate unavailable or inapplicable entries.

Method / component	Parameters	Mean GFLOPs / pair	Mean end-to-end time ImageCAS (ms)
Two Projections Suffice	31,189,377	954.331	88.127
AutoCAR	26,153,816	79.948	61.764
DeepCA	41,844,930	1200.166	30.557
3DGR-CAR†	1,096,252	5.966	8908.355
BiAAR (ours)	13,883,723	654.760	129.322
Case-specific Gaussian parameters (mean)			
3DGR-CAR (ImageCAS)	2,135	–	–
3DGR-CAR (ASOCA)	1,985	–	–

† 3DGR-CAR parameters and GFLOPs cover only GCP with two 128×128 masks, excluding Gaussian optimization and voxelization. Its end-to-end runtime includes 1,000 Gaussian optimization iterations and voxelization.

End-to-end runtime is reported separately from the forward-only timing in Section 4.1.

Counting scope. For the four non-GCP networks, one multiplication and one addition count as two FLOPs. Counted operators include convolutions, transposed convolutions, linear layers, and attention matrix multiplications. Normalization, activation, data/SDF preparation, backprojection geometry, grid sampling, evaluation metrics, and post-processing are excluded. These estimates describe counted neural operations under method-specific inputs, not exhaustive end-to-end computation or wall-clock latency. For BiAAR and our Two Projections Suffice implementation, each estimate profiles the first test sample at batch size one, rather than averaging the test set. BiAAR's estimate depends on the number of foreground tokens used in attention.

3DGR-CAR profiling scope. The ImageCAS and ASOCA checkpoints have identical GCP architectures. CPU profiling with two 128×128 masks gives 5.966004 GFLOPs for one GCP forward pass in either dataset. We use 1,000 Gaussian optimization iterations per case; this optimization and subsequent voxelization are outside the GCP FLOP count. Each Gaussian carries 11 additional optimizable scalar parameters; the dataset-specific means in Table 7 already include this factor. These case-specific parameters are separate from the shared GCP weights, not additional network parameters.

E SINGLE-DATASET QUALITATIVE COMPARISONS



Figure 8: Additional single-dataset reconstruction comparisons on ImageCAS (top) and AVDNet-derived ASOCA (bottom), including 3DGR-CAR. Training and visualization conventions follow Figure 5; numbers report whole-volume Dice / soft clDice. Starred rows show repeated structures under different camera configurations.

F Extended ASOCA Transfer Results

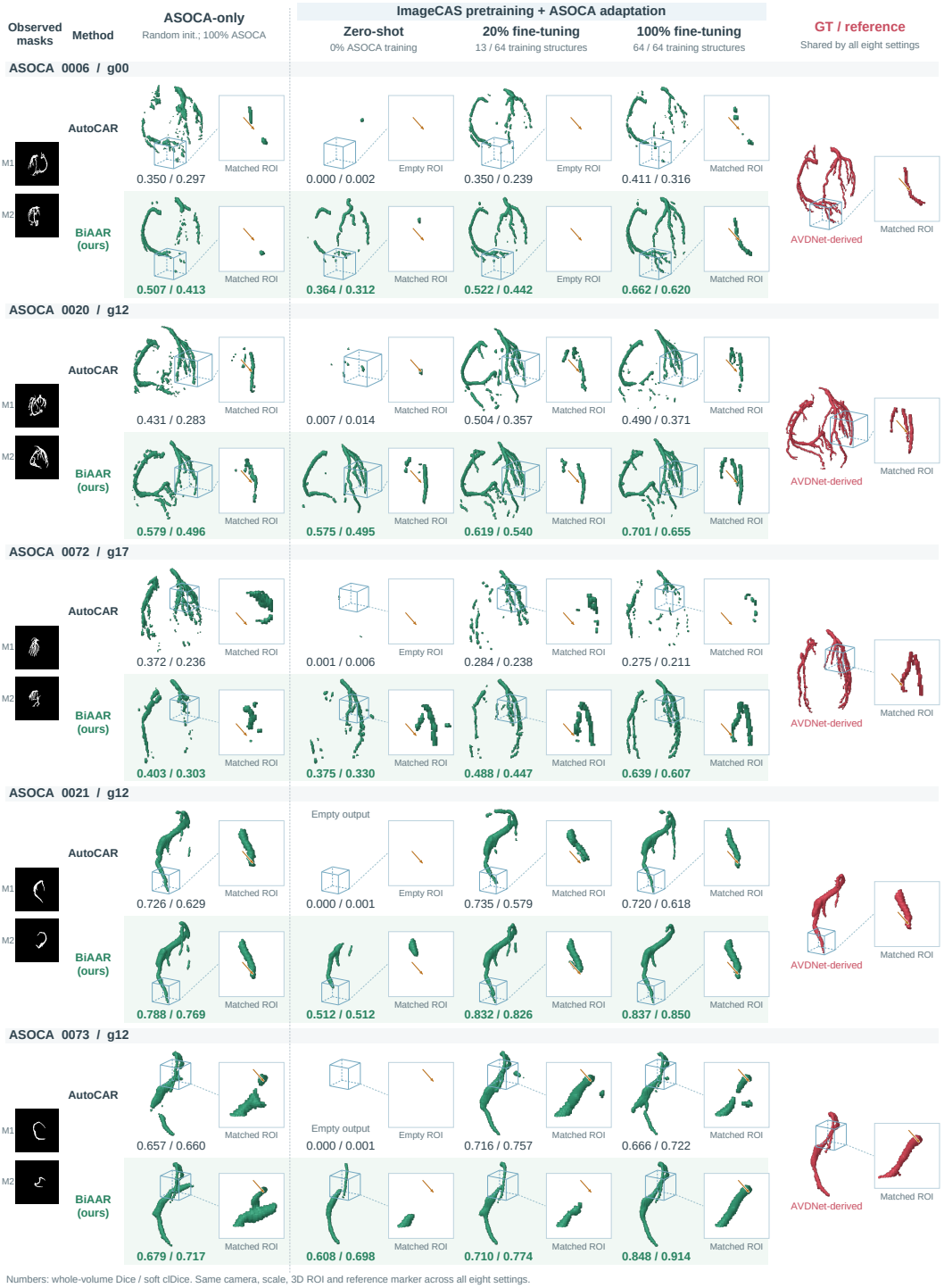

Figure 9: Additional ASOCA transfer examples under the four settings in Figure 7. Anatomical structures may reappear under different camera configurations; scores report whole-volume Dice / soft clDice. Empty regions of interest and empty full-volume outputs are distinguished explicitly.

Subset and training protocol. AutoCAR and BiAAR start from their respective ImageCAS checkpoints. Both use nested prefixes of the same seed-42 shuffled ASOCA training list. Fractions of 20/40/60/80/100% contain 13/26/38/51/64 label-specific volumes; zero-shot evaluation uses no ASOCA training data. Fine-tuning uses 50 epochs, with other settings unchanged from Appendix D.2. ASOCA-only baselines use random initialization and all 64 volumes for 200 epochs.

F.1 Complete adaptation curves and numerical results

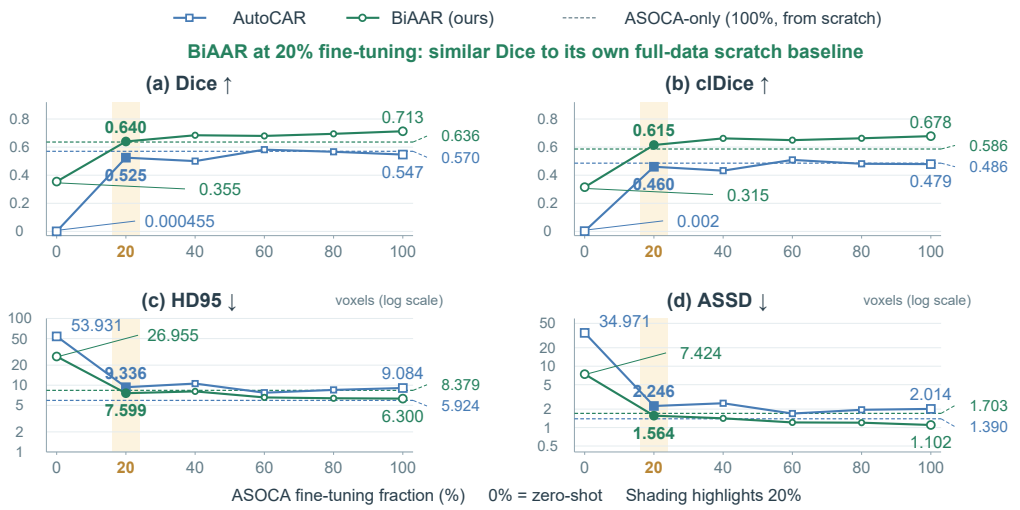

Figure 10: Complete ImageCAS-to-ASOCA adaptation curves. The upper panels also appear in Figure 6; lower panels report HD95 and ASSD in voxel units on logarithmic axes. Curves show test-sample means, not error bars. Labels highlight 0/20/100%; the full mean±standard deviation results are in Table 8.

Aggregate results. Table 8 gives all six metrics for both methods at every tested fraction, plus the ASOCA-only baselines. Figure 10 plots four metrics; its Dice/clDice panels appear in the main text. Recall and precision are included in the table. Statistics summarize 160 paired-view test samples, comprising 20 camera configurations for each of eight label-specific volumes from four held-out patients. Standard deviations describe test-sample variability rather than variation across training runs. Fine-tuning uses 50 epochs and ASOCA-only training uses 200 epochs, following Sections 4.1 and 4.4.

Full fine-tuning improves all six mean metrics for BiAAR relative to its ASOCA-only baseline. AutoCAR gains precision but loses recall and worsens on the four plotted metrics. Improvements are not monotonic across fractions: BiAAR Dice and clDice decrease slightly between 40% and 60% before increasing again. These mean comparisons support transfer for BiAAR under this protocol, without implying monotonic improvement as more target data are added.

Local and sample-level variation. Figure 9 retains local failures alongside improved whole-volume scores. For 0006/g00, the 20% fine-tuned BiAAR output has an empty highlighted region despite a whole-volume Dice of 0.522. Mean rankings also need not hold for every example. At 20% fine-tuning on 0073/g12, AutoCAR has higher Dice than BiAAR (0.716 versus 0.710), whereas BiAAR has higher soft clDice (0.774 versus 0.757). These examples distinguish volumetric overlap, local branch recovery, and aggregate performance.

Table 8: Complete ASOCA adaptation results (mean±standard deviation over 160 paired-view test samples). Scratch denotes ASOCA-only training from random initialization; percentage rows use ImageCAS pretraining, with 0% denoting zero-shot evaluation. Parentheses give the number of ASOCA training volumes. HD95 and ASSD use voxel units. Values are rounded to three decimals, except near-zero AutoCAR Dice and recall, which retain six.

Training	Dice ↑	clDice ↑	Recall ↑	Precision ↑	HD95 ↓	ASSD ↓
AutoCAR						
Scratch (64)	.570±.164	.486±.187	.592±.170	.554±.166	5.924±3.165	1.390±.667
0% (0)	.000455±.001838	.002±.004	.000231±.000933	.566±.483	53.931±10.237	34.971±9.039
20% (13)	.525±.186	.460±.206	.521±.232	.551±.136	9.336±5.172	2.246±1.301
40% (26)	.500±.190	.432±.208	.493±.222	.569±.170	10.590±6.565	2.491±1.567
60% (38)	.582±.172	.509±.198	.557±.217	.633±.117	7.726±4.582	1.692±.972
80% (51)	.567±.189	.481±.208	.525±.218	.643±.154	8.508±5.253	1.947±1.311
100% (64)	.547±.176	.479±.192	.480±.193	.678±.132	9.084±5.502	2.014±1.345
BiAAR (ours)						
Scratch (64)	.636±.162	.586±.208	.606±.187	.681±.138	8.379±6.576	1.703±1.320
0% (0)	.355±.055	.315±.070	.259±.056	.672±.095	26.955±5.584	7.424±1.741
20% (13)	.640±.153	.615±.200	.602±.173	.693±.126	7.599±4.594	1.564±.985
40% (26)	.684±.142	.662±.188	.703±.165	.671±.125	8.084±5.863	1.414±.904
60% (38)	.680±.103	.650±.147	.636±.114	.735±.104	6.565±3.916	1.212±.581
80% (51)	.695±.121	.663±.161	.663±.132	.735±.116	6.377±4.954	1.201±.769
100% (64)	.713±.107	.678±.145	.677±.116	.758±.111	6.300±4.964	1.102±.667

G Additional Related-Work Context

Tomography and multiview regularization. Tomography obtains complementary line-integral constraints through angular sampling (Kak & Slaney, 2001; Hansis et al., 2008). Canonical NeRF and 3D Gaussian Splatting use multiview captures (Mildenhall et al., 2021; Kerbl et al., 2023); sparse-view variants add depth or learned regularization (Deng et al., 2022; Mihajlovic et al., 2024). These techniques contextualize the projection-consistent direction in Section 2, without implying that all such methods use identical supervision or view counts.

Volumetric feature processing. SurroundOcc uses spatial 2D–3D attention followed by 3D convolutions and multilevel occupancy supervision for RGB-based scene perception (Wei et al., 2023). Sparse-convolution frameworks provide computational tools for volumetric processing (Graham et al., 2018; Choy et al., 2019), as used by AutoCAR for backprojected features (Zhu et al., 2025). These are antecedents of feature lifting and volumetric inference; BiAAR combines explicit amount estimation with occupancy-based allocation guidance.