Appendix

This document supplements the main paper as follows:

1. Dataset details (Section A).
2. More details about the training recipe and reproducibility (section B).
3. More visualizations and detailed tables (section C).

A. Additional Dataset Details

A.1. Fracture Simulation

(i) *Bone*. For elongated structures like limbs and ribs, we used Blender's skinning and subdivision surface techniques to create realistic cylindrical hollows, replicating bone morphology. We then applied the physics-based fracture method from Breaking Bad [42] to generate 2–20 fragments. The same approach was used for *os coxae* and vertebrae, forming the simulated subset of the bone category. (ii) *Eggshell*. Since scanned eggshells produce watertight solid ellipsoids, we removed 98% of the concentric volume to simulate thin shells. We then applied the same physics-based fracture method to generate realistic breakage patterns. (iii) *Ceramics*. Given that ceramic objects (e.g., bowls, pots, vases) closely resemble those in Breaking Bad's everyday category, we focused on scanning real fragments and did not include a simulated subset. (iv) *Lithics*. As an initial feasibility test, two generalized core morphologies were repeatedly virtually knapped with some randomized variation following methods described for the dataset in [36] to produce core and flake combinations with varying geometries.

A.2. FRACTURA Statistics

Table I presents detailed statistics for each category in FRACTURA. We continue to expand both the dataset's scale and diversity, aiming to establish a comprehensive cyberinfrastructure for the vision-for-science community.

Table I. Dataset Statistics of the FRACTURA Dataset.

Category	Fracture Type	# Assemblies	# Pieces
Bone	Real	17	37
	Synthetic	7056	39943
Eggshell	Real	3	12
	Synthetic	2268	12600
Ceramics	Real	9	51
	Synthetic	N/A	N/A
Lithics	Real	12	192
	Synthetic	403	807
Total	Real	41	292
	Synthetic	9727	53350

B. Additional Implementation Details

B.1. Data Preprocessing

We preprocess the BreakingBad dataset [42] to calculate the segmentation ground truth directly from meshes to reduce the computation overhead during training as described in Sec. 3.1, and there's no need for any hyperparameters. Unlike baseline methods (Global, LSTM, and DGL) provided by the dataset and PF++ [50], which samples $M = 1000$ points from the mesh per fragment, we used the same setting as in Jigsaw [30] to sample $M = 5000$ points per object, making all fragments have the same point density. With this sampling setting, we did not encounter any gradient explosion issues during training, as reported in FragmentDiff [56], which occur when sampling too many points for tiny pieces. Meanwhile, we employ the Poisson disk sampling method to ensure that the points are more uniformly distributed on the surface of the fragment. During training, standard data augmentation techniques are applied, including recentering, scaling, and random rotation.

B.2. Training Recipe

We modified a smaller version of Point Transformer V3 [54] as our backbone for the segmentation pretraining, as shown in Table II, which we found to be sufficient and more memory efficient. Since GARF uses a much larger training dataset, we reduce the training epochs to 150, other than the 400 epochs used in GARF-mini. Both pretrainings reach over 99.5% accuracy on the validation set. Samples of segmentation results are shown in Fig. I.

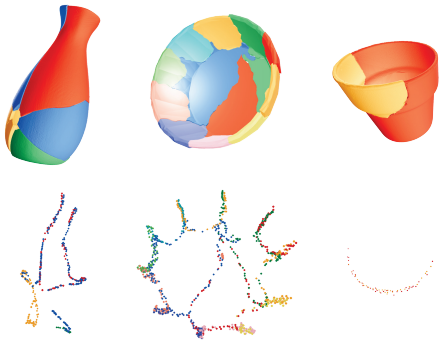

Figure I. Segmentation results on a real-world object (left), Breaking Bad [42] (center) and Fantastic Breaks [18] (right).

For FM training, we provide the hyperparameters in Table II for reproducibility. The settings are identical for both GARF and GARF-mini, as their only difference lies in the pretraining stage.

B.3. Preliminaries on Riemannian Flow Matching

Instead of simulating discrete noise addition steps, flow matching (FM) learns a probability density path p_t, which

Table II. Training Configurations.

	Config	Value
Backbone	Encoder Depth	[2, 2, 6, 2]
	Encoder # Heads	[2, 4, 8, 16]
	Encoder Patch Size	[1024, 1024, 1024, 1024]
	Encoder Channels	[32, 64, 128, 256]
	Decoder Depth	[2, 2, 2]
	Decoder # Heads	[4, 8, 16]
	Decoder Patch Size	[1024, 1024, 1024]
	Decoder Channels	[256, 128, 64]
Pretraining	Global Batch Size	256
	Epochs	400 / 150
	Learning Rate	1e-4
	Scheduler	CosineAnnealingWarmRestarts
	Scheduler T_0	100 / 50
	# Trainable Params	12.7M
Training	Global Batch Size	128
	Epochs	1500
	Learning Rate	2e-4
	Scheduler	MultiStepLR
	Scheduler Milestones	[900, 1200]
	Scheduler γ	0.5
	# Trainable Params	43.5M

progressively transforms a noise distribution $p_{t=0}$ to the data distribution $p_{t=1}$, with a time variable $t \in [0, 1]$. As a simulation-free method aiming to learn continuous normalizing flow (CNF), FM models a probability density path p_t, which progressively transforms a noise distribution $p_{t=0}$ to the data distribution $p_{t=1}$, with a time variable $t \in [0, 1]$. Inspired by *learning assembly by breaking*, the rigid motion of the fragments corresponds to the geodesic on the *Lie group* $\mathrm{SE}(3)$, which is a differentiable Riemannian manifold. Inspired by previous works [4, 11, 57], FM can be extended to $\mathrm{SE}(3)$ manifold to learn the rigid assembly process.

On a manifold $\mathcal{M}$, the flow $\psi_t : \mathcal{M} \to \mathcal{M}$ is defined as the solution of an ordinary differential equation (ODE):

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_t(\boldsymbol{x}) = \boldsymbol{v}_t(\psi_t(\boldsymbol{x})), \quad \psi_0(\boldsymbol{x}) = \boldsymbol{x}, \tag{8}$$

where $\boldsymbol{v}_t(\boldsymbol{x}) \in \mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ is the time-dependent vector field, and $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ is the tangent space of the manifold at $\boldsymbol{x} \in \mathcal{M}$. In the context of $\mathrm{SE}(3)$, the tangent space is the *Lie algebra* $\mathfrak{se}(3)$, which is a six-dimensional vector space, presenting the velocity of the rigid motion of the fragments. Given the *conditional vector filed* $\boldsymbol{u}_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \in \mathcal{T}_{\boldsymbol{x}}\mathcal{M}$, which generates the conditional probability path $p_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$, the Riemannian flow matching objective can be defined as:

$$\mathcal{L}_{\text{CFM}} := \mathbb{E}_{t, p_1(\boldsymbol{x}_1), p_t(\boldsymbol{x}|\boldsymbol{x}_1)} \left[\| \boldsymbol{v}_t(\boldsymbol{x}, t) - \boldsymbol{u}_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \|_G^2 \right], \tag{9}$$

where $\| \cdot \|_G^2$ is the norm induced by the Riemannian metric G. Then the learned vector field $\boldsymbol{v}_t$ can be used to generate samples on the manifold at inference, which is $\mathrm{SE}(3)$ poses of the fragments. The rigid motion of fragments corresponds to the geodesic on the *Lie group* $\mathrm{SE}(3)$, a differentiable Riemannian manifold.

Table III. Results on Vanilla Breaking Bad [42] Dataset.

Methods	RMSE(R) ↓ degree	RMSE(T) ↓ $\times 10^{-2}$	PA ↑ %	CD ↓ $\times 10^{-3}$
Tested on the **Everyday** Subset				
Global [22]	80.70	15.10	24.60	14.60
LSTM [52]	84.20	16.20	22.70	15.80
DGL [59]	79.40	15.00	31.00	14.30
SE(3)-Equiv [53]	79.30	16.90	8.41	28.50
DiffAssemble [41]	73.30	14.80	27.50	-
PHFormer [7]	26.10	9.30	50.70	9.60
Jigsaw [30]	42.30	10.70	57.30	13.30
PF++ [50]	38.10	8.04	70.60	6.03
GARF-mini	**10.41**	**1.91**	**92.77**	**0.45**
Tested on the **Artifact** Subset				
Jigsaw	52.40	22.20	45.60	14.30
PF++	52.10	13.90	49.60	14.50
GARF-mini	**11.91**	**2.74**	**89.42**	**1.05**

Table IV. **Ablation Study on Our Designs of FM.**

SE(3)	Multi-Anchor	One-Step	RMSE(R) ↓	RMSE(T) ↓	PA ↑
✗	✗	✗	10.24	1.95	89.08
✓	✗	✗	8.02	1.63	93.78
✓	✓	✗	7.63	1.60	94.02
✓	✓	✓	**6.68**	**1.34**	**94.77**

Table V. **Ablation Study on Sample Steps.**

Steps	RMSE(R) ↓	RMSE(T) ↓	PA ↑	CD ↓	Speed (ms)
1	12.52	3.18	86.88	2.14	38.26
One-Step + 1	9.79	2.46	91.31	1.42	45.76
5	8.25	1.92	93.70	0.53	57.32
One-Step + 5	7.15	1.66	94.43	0.46	76.23
20	7.63	1.60	94.02	0.35	185.05
One-step + 20	**6.68**	**1.34**	**94.77**	**0.25**	190.77
50	7.50	1.54	94.01	0.32	408.40

B.4. More attention on large fragments

GARF provides tailored designs to place more attention on large fragments. We observed that: (i) large fragments are easier to assemble; (ii) tiny fragments sometimes lead to unstable training. Driven by these insights, we apply weighted sampling based on the surface area of fragments and modify the self-attention module to allow more attention on large fragments.

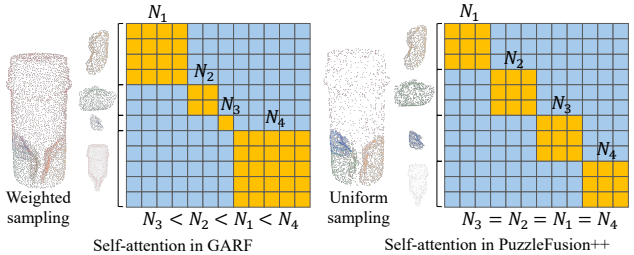

Figure II. Self-attention comparison between GARF (left) and PuzzleFusion++ [50] (right).

Table VI. **Ablation Study on the Different Anchor Initialization.**

Settings	RMSE(R) ↓	RMSE(T) ↓	PA ↑	CD ↓
Largest Anchor	6.10	1.22	95.33	0.22
Random Anchor	6.09	1.30	95.20	0.29
Anchor-Free	9.09	2.13	93.23	0.91

Table VII. **Comparison Between Diffusion and Our FM Models.**

Dataset	Methods	RMSE(R) ↓	RMSE(T) ↓	PA ↑
Everyday	Diffusion	7.45	1.47	94.30
	SE(3) Diffusion	N/A	N/A	N/A
	Diffusion w/ One-Step	7.51	1.47	94.27
	Vanilla FM	10.24	1.95	89.08
	GARF-mini	**6.68**	**1.34**	**94.77**
FRACTURA	Diffusion	32.38	7.90	71.73
	GARF-mini	**27.88**	**6.79**	**76.25**

C. Additional Results and Analyses

C.1. Ablation on Design Choices in Flow Matching

We conduct an ablation study to evaluate the impact of design choices in our FM module. As shown in Table IV, vanilla FM, trained with spherical linear interpolation (slerp) to approximate valid rotations in the forward process [12], achieves 89.08 PA, already surpassing previous methods [30, 50]. Incorporating the $\mathrm{SE}(3)$ representation further improves performance by pre-modeling the manifold distribution and better capturing distribution shifts during assembly. Multi-anchor training strategy further enhances results, while one-step pre-assembly significantly boosts performance by providing a more reasonable initial pose distribution, leading to the best overall outcomes.

C.2. Ablation on Sample Steps

Table V shows the effect of varying sampling steps in our framework. Surprisingly, even with just 5 steps, FM achieves 93.70% PA, highlighting its effectiveness in modeling global probabilistic paths. Additionally, our first-session initialization provides a more reasonable initial pose, further improving assembly quality while adding minimal computational overhead.

C.3. Ablation on Anchor Fragment

Similar to PF++ [50], we use the largest fragment as the anchor fragment at inference. We compare the performance of using the largest fragment, a randomly selected fragment, and no anchor fragment. As shown in Table VI, using a random fragment as the anchor fragment has almost no negative effect on the model. Only anchor-free initialization leads to a slight performance drop.

C.4. Comparison with Diffusion Models

Table VII compares our FM module with diffusion models. While diffusion, when paired with fracture-aware pretraining, achieves competitive performance, directly applying vanilla FM yields lower results (89.08 PA), emphasizing the importance of our subsequent design choices. A key limitation of diffusion models is their handling of $\mathrm{SO}(3)$ rotation, which cannot be naturally incorporated into the reverse process. Existing methods, such as score prediction [58], aim to maintain rotation validity but fall outside our current scope. Additionally, diffusion models rely on multi-step denoising without explicitly modeling the global probabilistic path, rendering one-step pre-assembly ineffective. Furthermore, on FRACTURA, diffusion models exhibit weaker generalization to unseen objects compared to GARF-mini.

C.5. Quantitative Results on Vanilla Breaking Bad

Given that all our previous experiments were conducted on the volume-constrained version of the Breaking Bad dataset [42], we here provide additional quantitative results on the non-volume-constrained version to align with the settings of previous methods. The results, shown in Table III, demonstrate that our GARF-mini model still significantly outperforms the previous state-of-the-art method, PF++ [50], by a large margin. This performance is consistent across both the everyday and artifact subsets, showcasing the model's robust generalization ability.

We also present the results of FragmentDiff [56] on their custom Breaking Bad dataset in Table VIII. FragmentDiff claims to remove tiny pieces, but it is unclear whether this applies only to their training setting or also to evaluation. Unfortunately, since they did not open source their code or provide their preprocessed data, we are unable to directly compare all other methods with FragmentDiff. Additionally, they did not adhere to the common settings used by other methods, which limit the number of pieces from 2 to 20, making direct comparisons on their provided metrics impossible. However, its significant performance drop from the Everyday subset to the Artifact subset suggests that GARF surpasses FragmentDiff in generalization capability.

C.6. Quantitative Results of Finetuning on the FRACTURA Synthetic Dataset

After finetuning GARF on the FRACTURA synthetic dataset, we report the per-category performance on the bone and eggshell categories, as shown in Table IX. The results demonstrate that finetuning the FM model in GARF significantly improves performance on these two unseen categories, showing the effectiveness of our finetuning techniques and the generalizability of our pretraining strategy.

Table VIII. FragmentDiff [56] Results on Their Custom Breaking Bad Dataset.

Methods	Subset	RMSE(R) ↓ degree	RMSE(T) ↓ $\times 10^{-2}$	PA ↑ %
FragmentDiff [56]	Everyday	13.68	7.41	90.20
	Artifact	18.18	8.12	82.30

Table IX. Quantitative Per-category Results on the FRACTURA (Synthetic Fracture).

Category	Method	RMSE(R) ↓ degree	RMSE(T) ↓ $\times 10^{-2}$	PA ↑ %	CD ↓ $\times 10^{-3}$
Bone	Jigsaw	66.44	20.54	27.24	91.70
	PF++	66.28	20.50	29.81	47.78
	GARF	17.70	3.80	85.18	5.11
	GARF$_{\text{LoRA}}$	**8.79**	**1.10**	**98.19**	**0.34**
Eggshell	Jigsaw	44.44	12.88	49.03	10.49
	PF++	54.81	13.81	61.36	1.50
	GARF	22.48	6.16	83.41	0.67
	GARF$_{\text{LoRA}}$	**7.10**	**1.95**	**95.68**	**0.26**

C.7. Additional Qualitative Comparison on the FRACTURA and Breaking Bad Dataset

Figures III, IV and V demonstrate more qualitative comparison on the FRACTURA and Breaking Bad Dataset, where our GARF shows superior performance than the other previous SOTA methods.

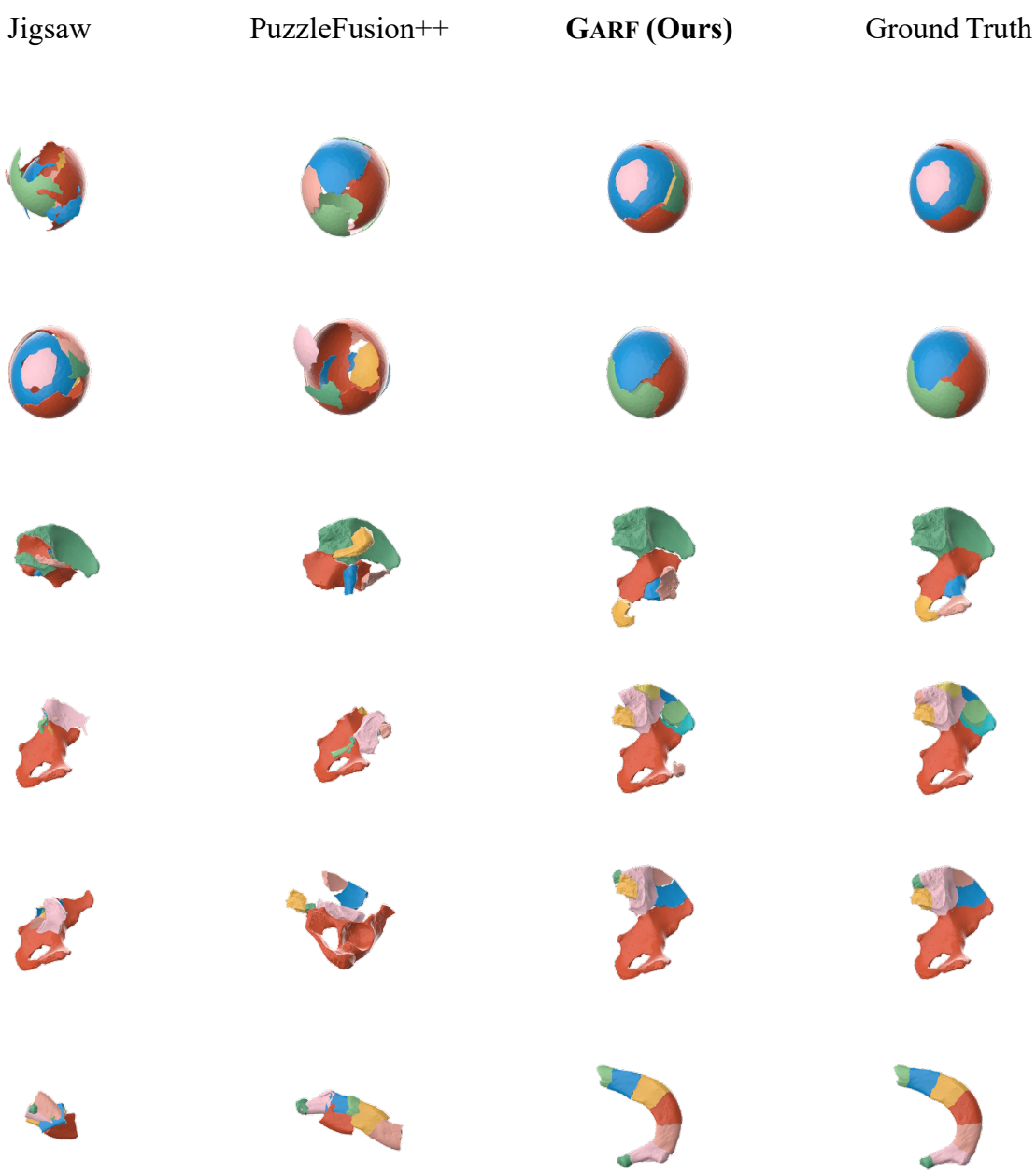

Figure III. Qualitative Results on the FRACTURA Synthetic Dataset.

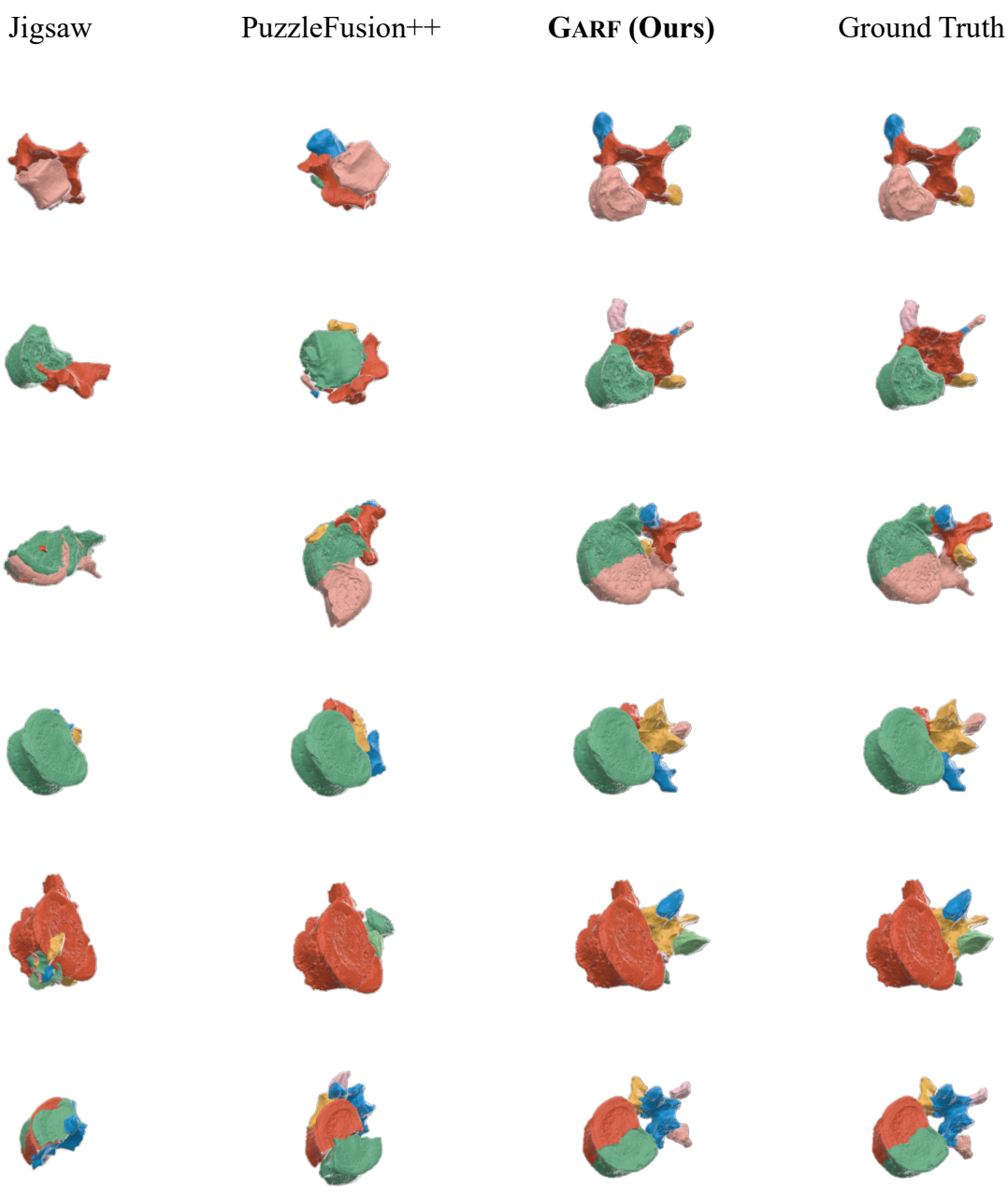

Figure IV. Qualitative Results on the FRACTURA Synthetic Dataset.

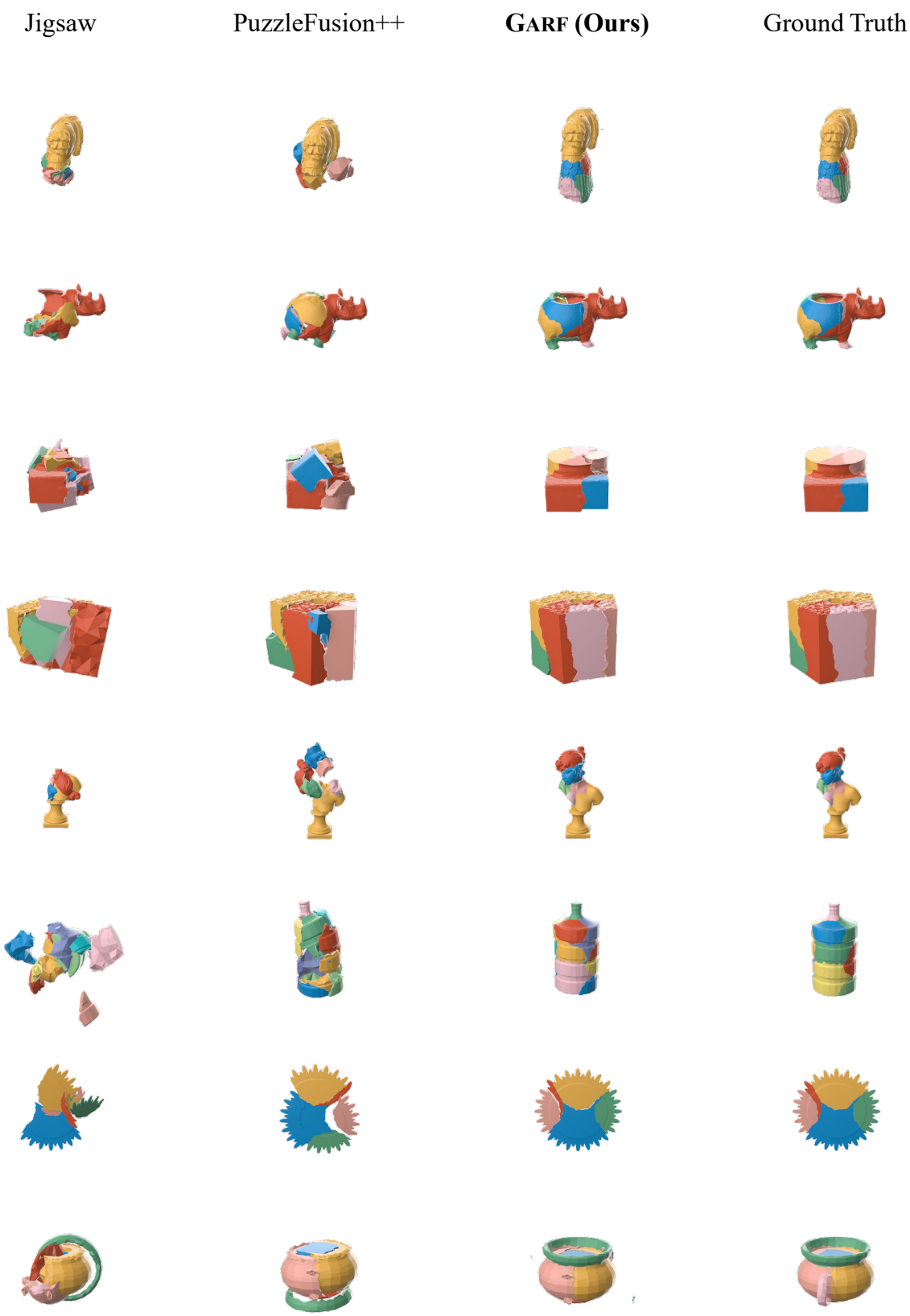

Figure V. Qualitative Results on the Breaking Bad Dataset Artifact Subset.