Target-Part Preference Optimization for Part-Aware 3D Generation

Zishan Qin[1,2] Rui Li[3] Zeyu Zhang[1] Sevvandi Kandanaarachchi[2] Lina Yao[1] Dong Gong[1]

[1]UNSW Sydney [2]CSIRO [3]NTU Singapore zishan.qin@unsw.edu.au

PartDPO · DICTA 2026 Oral · Paper 67

Project page · 3D viewer

The task part-aware 3D generation

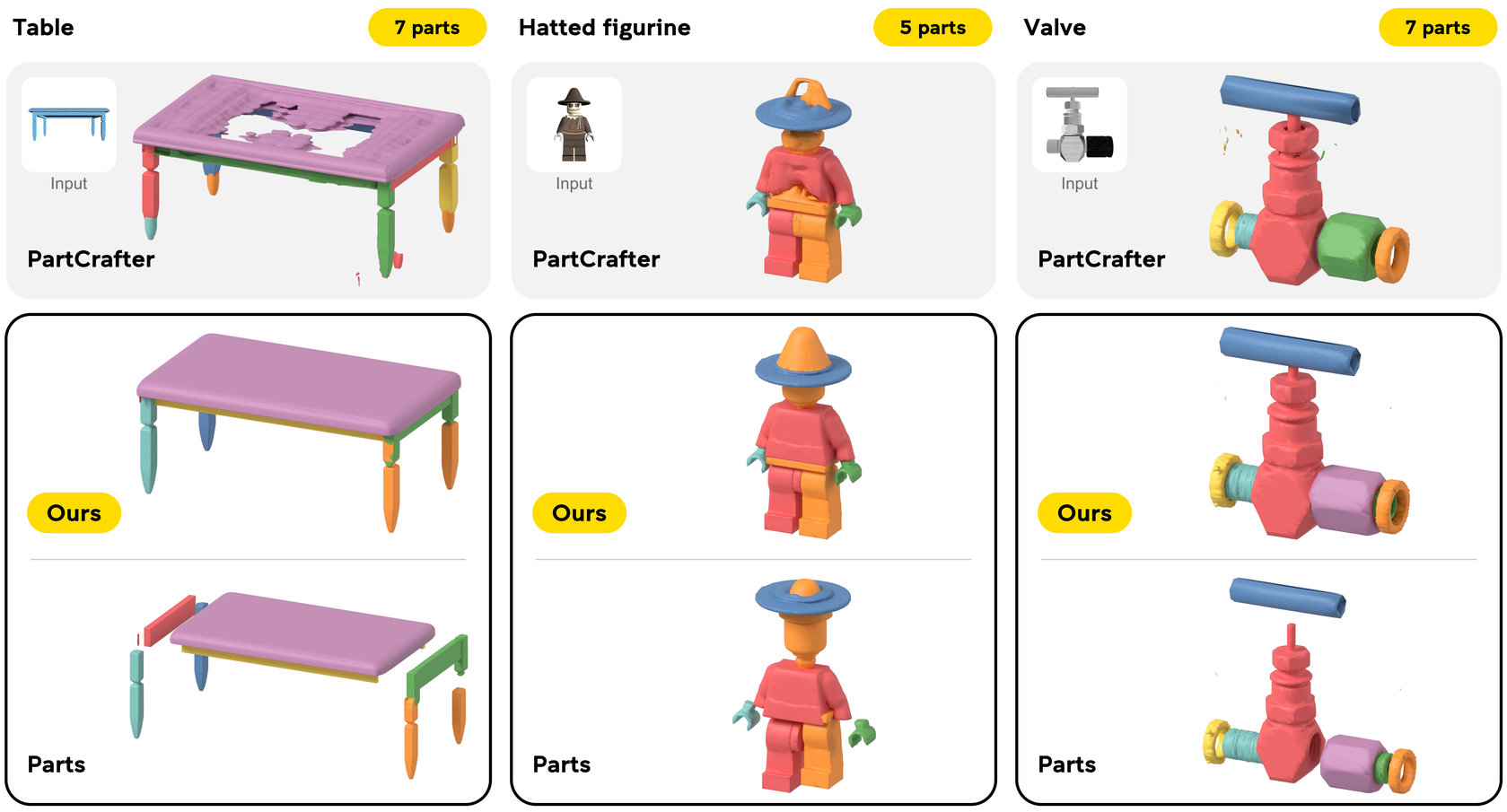
From one RGB image, jointly generate a requested number *P* of part meshes.

The problem pretrained PartCrafter

A plausible object is not a set of usable parts.

Editing, recomposition and articulation need parts that are complete and well separated.





Inspiration why preference optimization

Labels are scarce. Seeds are free.

Fine-tuning runs out of labels.

Fine-grained 3D part annotation is costly, so the annotated set is small. Direct supervised fine-tuning on all of it changes matched-part F-score only marginally.

Seeds already disagree.

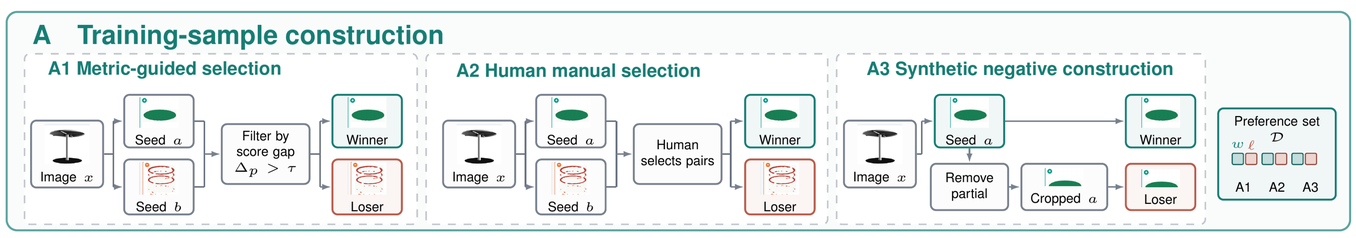
For one image, different seeds of the pretrained generator return a better or a worse version of the same part. Ranking them against the GT part gives ordered pairs with no new annotation.

DPO learns from the order.

It compares a winner and a loser against a frozen reference: no reward model, no architectural change. GT flow-matching losses keep absolute geometry supervision.

Step 1 build the preference set, offline

Rank seeds on one part. Keep the big gaps.



1 **Score one target part.**

Sample several seeds from the frozen generator and score part *p* of each candidate against its GT part g_p.

2 **Keep the largest gaps. Store the seeds.**

Keep examples whose across-seed gap Δ_p exceeds τ. Store the seeds s^w, s^ℓ in winner/loser order. Metrics are never a reward.

3 **Cover enough objects.**

The small set (B1) falls below pretrained. The medium set (B2) gives the best matched-part scores.

4 **Keep the order clean.**

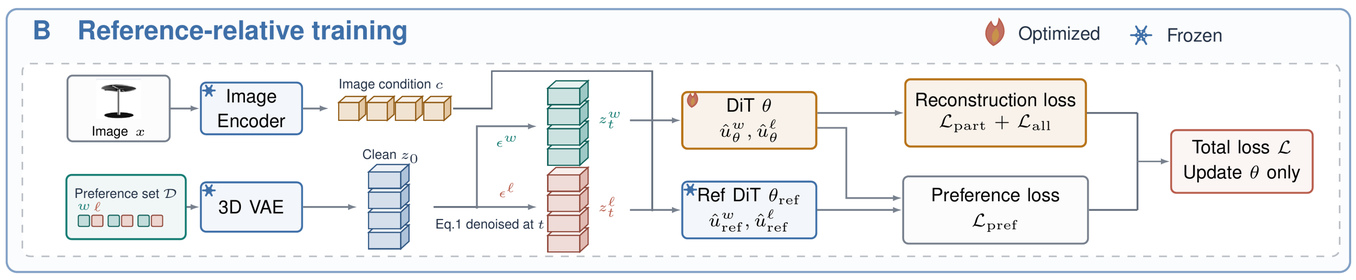
Reversing labels (D1) or adding randomly ordered pairs (D5) drops part scores below pretrained. Human ranking helps only when screened (C1, C2).

What the pairs do to held-out quality.

Objaverse, 2–8 parts	Obj-CD ↓	Obj-FS ↑	Part-CD ↓	Part-FS ↑
A1 Pretrained	0.2724	0.5680	0.2875	0.5453
B1 Small	0.2626	0.5328	0.3157	0.4545
B2 Medium	**0.1986**	**0.6958**	**0.2542**	**0.5725**
B3 Large	**0.2625**	**0.5881**	**0.2716**	**0.5707**
C1 Human-ranked, screened	0.2603	0.5854	0.2805	0.5573
C2 Human-ranked, larger	0.5127	0.1745	0.5616	0.1285
C3 Synthetic crop	0.3529	0.4127	0.3821	0.3398
D1 Labels reversed	0.3848	0.3238	0.4336	0.2690
D5 Random pairs added	0.3542	0.3554	0.4398	0.2364

Step 2 reference-relative training

One clean latent. Two noise trajectories. Favor the winner.



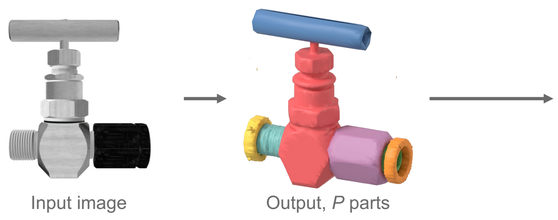
Seeds s^w, s^ℓ fix the noises ε^w, ε^ℓ. The GT parts are encoded once: one clean latent z_0, one block per part. Both trajectories share timestep *t* and image condition *c*. Only the DiT is updated; image encoder, 3D VAE and reference stay frozen.

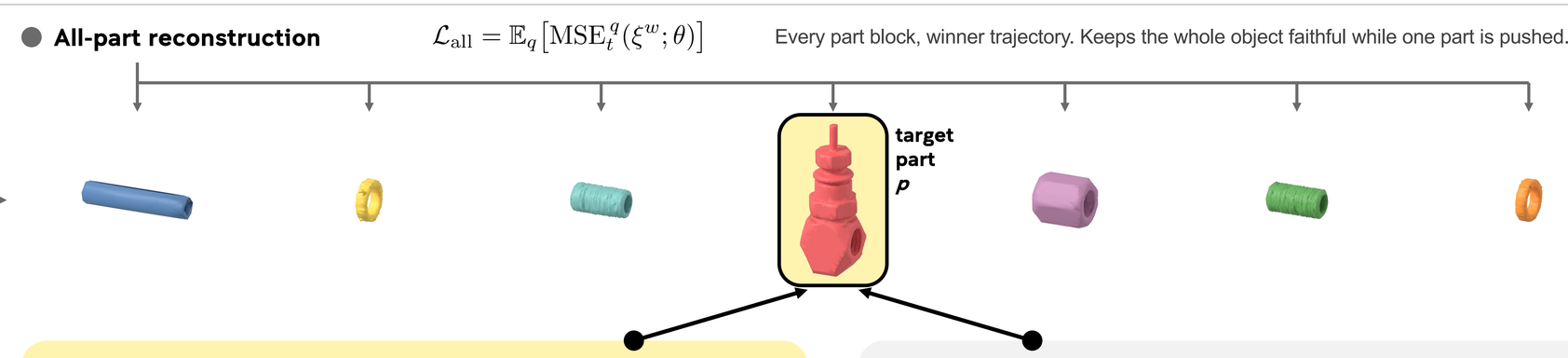
The pairwise term does the work.

Setting	Obj-CD ↓	Obj-FS ↑	Part-CD ↓	Part-FS ↑
F1 Pretrained	0.2724	0.5680	0.2875	0.5453
F2 Medium, recon. only	0.2650	0.5179	0.3001	0.4552
F3 Medium, full	**0.1941**	**0.7036**	**0.2624**	**0.5488**
F4 Large, recon. only	0.2819	0.4878	0.3328	0.4061
F5 Large, full	**0.2625**	**0.5881**	**0.2716**	**0.5707**
F3 vs. F2 gain, 95% CI	[0.0577, 0.0846]	[0.1561, 0.2156]	[0.0290, 0.0465]	[0.0735, 0.1140]

What each loss acts on.

Shown on one of our outputs, one colour per part.





Total objective, after reconstruction-only warm-up

$$\mathcal{L} = \lambda_{\text{pair}}\,\mathcal{L}_{\text{pair}} + \lambda_{\text{part}}\,\mathcal{L}_{\text{target}} + \lambda_{\text{all}}\,\mathcal{L}_{\text{all}}$$

In the diagram above: $L_{pref} = L_{pair}$, $L_{part} = L_{target}$.

Numbers held-out Objaverse

Better parts. Better objects.

2–8 parts	Obj-CD ↓	Obj-FS ↑	Part-CD ↓	Part-FS ↑
Pretrained PartCrafter	0.2724	0.5680	0.2875	0.5453
Direct SFT	0.2667	0.5751	0.2852	0.5509
Ours, medium	**0.1986**	**0.6958**	**0.2542**	**0.5725**
Ours, large	0.2625	0.5881	0.2716	0.5707

Holds as the part count grows.

Matched-part F-score ↑	1 part	2–4 parts	5–8 parts	2–8 avg.
Pretrained PartCrafter	0.6949	0.5986	0.4646	0.5453
Direct SFT	**0.7273**	0.5955	0.4846	0.5509
Ours, large	0.7020	**0.6237**	**0.4912**	**0.5707**