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 [*]Corresponding author
zishan.qin@unsw.edu.au edgong01@gmail.com

Abstract

Part-aware 3D generation requires not only globally plausible objects but also complete, well-separated semantic parts, yet high-quality part decompositions remain too scarce to support large-scale supervised fine-tuning. We study whether preference fine-tuning based on Direct Preference Optimization (DPO) can adapt a pretrained part-aware generator from limited part-annotated data without architectural changes. We rank candidates generated with different random seeds against ground-truth (GT) geometric metrics, and select the (object, target-part) examples with the largest across-seed quality gap. Each selected example stores a seed pair whose two flow trajectories share one clean GT compositional latent and differ only in Gaussian noise. We derive a reference-relative target-block preference loss from FlowDPO that compares the two trajectories against a frozen reference model. GT flow-matching losses on the target block and complete part set provide absolute supervision, while only the diffusion transformer (DiT) is updated. Human-ranked controls test an alternative ordering source, and synthetic cropping provides controlled loser targets that are incomplete by construction. Among the three training-set scales, the medium-scale setting gives the strongest result on held-out Objaverse and outperforms both the pretrained generator and the reported direct supervised fine-tuning baseline on the multi-part aggregate. Further analysis shows sensitivity to pair ordering, target-part coverage, and the balance between pairwise and GT supervision. Cross-dataset experiments show less consistent gains under distribution shift.

1. Introduction

Image- and text-conditioned 3D generators can now synthesize objects with convincing global geometry [22, 19, 8]. Many applications, however, manipulate objects at the part level. Editing a component, recomposing an object, or articulating a joint requires the relevant semantic parts to be complete, well separated, and independently usable [2, 14, 16, 18]. A plausible overall shape does not guarantee these properties. Generated parts may still be missing, fused, or fragmented. Such local failures may be obscured by object-level evaluation but remain important for downstream use, motivating supervision that explicitly targets part quality.

Supervised fine-tuning (SFT) on complete part decompositions is a direct solution, but fine-grained 3D part annotations are costly and remain limited [21, 28]. Making effective use of a limited number of annotated objects is therefore important. Preference optimization provides a complementary source of supervision. Direct Preference Optimization (DPO) learns from ordered comparisons relative to a frozen reference model without training a separate reward model [23]. Diffusion-DPO and FlowDPO extend this formulation to diffusion and flow-matching generators by replacing likelihood ratios with denoising- or flow-matching-error surrogates [27, 9]. We ask whether preference-guided fine-tuning can adapt a pretrained part-aware generator from limited annotated data without degrading its object-level generation ability.

Applying DPO to part-aware 3D generation is not straightforward. Existing 3D preference methods rank complete assets or rendered views [30, 12], while Mesh-RFT localizes feedback through quality-aware face masking [17]. These formulations do not directly address comparisons on one semantic part block in a generator that jointly models several parts. Existing part datasets likewise provide decompositions, not ordered target-part preferences [21, 28]. This leaves two practical questions: how to construct reliable target-part pairs from limited annotations, and how to improve the selected part without degrading the overall object quality.

We customize FlowDPO for target-part preference learning and fine-tune only PartCrafter's diffusion transformer (DiT) without architectural changes [14, 9]. PartCrafter represents each part with a separate latent-token block, allowing the pairwise error to be evaluated on a selected target block. For each target part, we rank candidates sampled from the frozen generator using GT-referenced geometric metrics and keep the examples with the largest across-seed

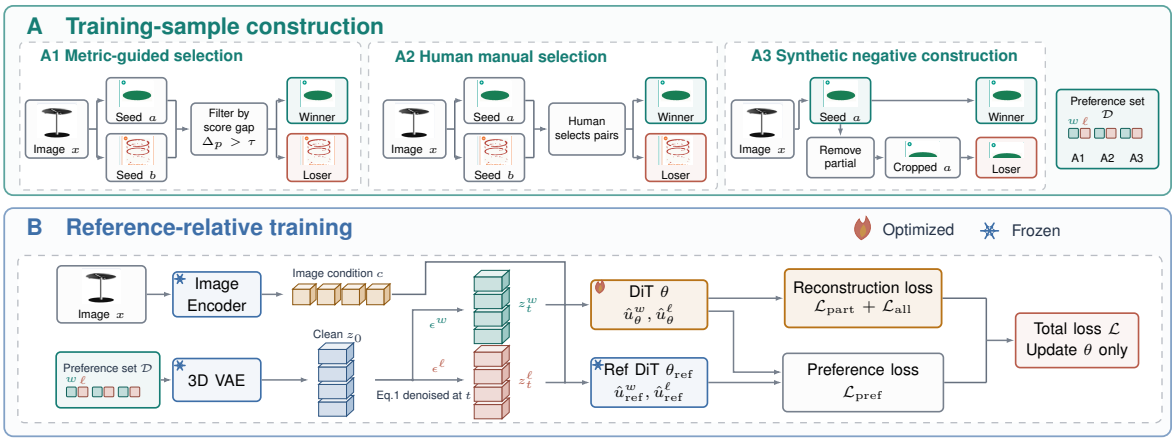

Figure 1. Overview of target-part preference fine-tuning. Superscripts w and ℓ denote winner and loser trajectories. (A) For target part p, GT-referenced metrics rank candidates and order their seeds, which instantiate different noise trajectories toward the same GT clean latent. Human controls rank candidates visually. Synthetic-crop controls remove loser target geometry before variational autoencoder (VAE) encoding. All three constructions write ordered (w, ℓ) pairs into one preference set $\mathcal{D}$. (B) $\mathcal{D}$ is the training input. In the primary construction, the 3D VAE encodes one clean latent z_0, which is noised at a shared timestep t with ϵ^w and ϵ^ℓ (Eq. 1). Only the PartCrafter Diffusion Transformer (DiT) θ is optimized with the total loss $\mathcal{L}$ in Sec. 4.3. The image encoder, 3D VAE, and stage-initial reference DiT remain frozen.

quality gap; each kept example stores a seed pair. These seeds instantiate two noisy trajectories toward the same GT clean compositional latent. A reference-relative objective compares their target-block flow errors, while target-part and all-part GT losses provide absolute supervision. Among the tested scales, the medium-scale setting gives the strongest held-out Objaverse multi-part result, raising matched-part F-score by approximately 5% over pretrained PartCrafter and exceeding the reported direct-SFT baseline. We make three contributions.

- We devise a GT-referenced ranking procedure that converts generated candidates into winner–loser seed pairs with varying seed-conditioned trajectories. We derive a target-block preference objective from FlowDPO [9] and combine it with target-part and all-part GT flow-matching losses.
- We construct metric-ranked target-part training sets at three scales from PartObjaverse-Tiny [28], together with human-ranked and synthetic-crop controls.
- We analyze training-set scale, target-part coverage, ordering reliability, loser construction, objective components, and cross-dataset transfer to identify when target-part preference learning helps and where its gains do not transfer.

2. Related Work

2.1. 3D And Part-Aware Generation

Modern 3D generation includes score-distillation methods guided by pretrained image models [22, 13], sparse-view reconstruction [19], and feed-forward generators that predict 3D representations directly from images [8, 11]. These approaches substantially improve object-level fidelity, but a plausible object surface does not by itself provide explicit, separately usable components.

To generate better part components, part-aware methods generate or recover component-level structure through combining segmentation and reconstruction [2], compositional or contextual part latents [14, 6], autoregressive part discovery [3], semantic decoupling and structural cohesion [29], efficient multi-part packing [25], or retrieval-augmented generation and editing [10]. Explicit parts also support downstream articulation, where methods infer part graphs, joints, axes, or motion ranges [16, 18]. Our contribution is an alignment objective rather than a new 3D representation. We evaluate this objective on PartCrafter [14].

2.2. Preference Optimization For 3D Generation

Direct Preference Optimization (DPO) aligns a model from preference pairs relative to a fixed reference without training a separate reward model [23]. Diffusion-DPO and FlowDPO extend this formulation to diffusion and flow models by replacing intractable output likelihoods with denoising- or flow-matching-error surrogates [27, 9]. Because such objectives depend on reliable preference labels, robust-DPO work also studies learning under noisy feedback [24].

Preference alignment has recently been applied to 3D generation. DreamDPO ranks paired renderings with reward-model or multimodal-model feedback during text-to-3D optimization [30]. Circular-DPO transfers preference

information across stages of a 3D generation pipeline [12], while Mesh-RFT applies fine-grained reinforcement feedback to mesh generation [17]. DreamDPO and Circular-DPO primarily rank complete objects or rendered views, whereas Mesh-RFT provides fine-grained mesh feedback. Our study instead compares reference-relative flow errors on one target-part latent block. Section 4.1 defines the metric-ranked construction and output-ranked controls.

3. Preliminaries

We first summarize the PartCrafter backbone and the flow-matching notation used in this paper.

3.1. PartCrafter Backbone

PartCrafter jointly generates a requested number P of part meshes from one RGB image [14]. It inherits the TripoSG image encoder, variational autoencoder (VAE), and Diffusion Transformer (DiT) [11]. For input image x, the image encoder produces visual tokens $h_x = E_{\text{img}}(x)$, denoted by c in our loss.

Let $\mathcal{G} = \{g_q\}_{q=1}^{P}$ be the ground-truth part surfaces. The VAE represents each surface g_q by a block of N latent tokens with feature width d. We denote this block by $z_{0,q} \in \mathbb{R}^{N\times d}$ and the complete compositional latent by $z_0 = (z_{0,1}, \ldots, z_{0,P}) \in \mathbb{R}^{P\times N\times d}$. Learnable part identity embeddings distinguish the P token groups. The DiT alternates local attention within each part with global attention across all part tokens, and the shared VAE decoder maps each latent block to a part mesh. We use q for a generic part and reserve p for the part ranked and evaluated in one training example. Accordingly, $z_{0,p}$ denotes the latent token block associated with target part p. The index p guides offline ranking and selects the corresponding block in the loss. It is not a semantic class label or generator condition.

3.2. Flow Matching And FlowDPO

Following PartCrafter's rectified flow [14, 20], let $i \in \{w, \ell\}$ index the winner and loser trajectories of one stored preference pair. For either trajectory, $z_0^i, \epsilon^i, z_t^i, u^i \in \mathbb{R}^{P\times N\times d}$ denote the clean compositional latent, Gaussian noise, noised latent, and reverse-velocity target, respectively. The scheduler form is

$$z_t^i = (1-\sigma_t)z_0^i + \sigma_t\epsilon^i, \qquad u^i = z_0^i - \epsilon^i, \tag{1}$$

where $\epsilon^i \sim \mathcal{N}(0, I)$ has the same shape as z_0^i. In Circular-DPO notation [12], Eq. 1 follows the same straight path with endpoints $x_0^i = \epsilon^i$, $x_1^i = z_0^i$, and interpolation coordinate $\tau = 1 - \sigma_t$. Only the scheduler parameterization differs. Let $\xi^i = (z_0^i, \epsilon^i, c)$ collect one trajectory's clean latent, noise, and image conditioning. The mean-squared flow-matching error (MSE) over the $P \times N \times d$ compositional latent is

$$\text{MSE}_t(\xi^i;\theta) = \frac{1}{P\,N\,d}\left\|\hat{u}_\theta(z_t^i, t; c) - u^i\right\|_F^2. \tag{2}$$

Standard conditional flow matching minimizes $\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\xi,t}[\text{MSE}_t(\xi;\theta)]$ over clean latents, noise, and timesteps [15].

FlowDPO and Circular-DPO replace the output log-likelihood ratio with a flow-matching-error surrogate and apply a log-sigmoid to the winner-minus-loser difference of reference-relative MSE reductions [9, 12]. We retain this β-scaled structure and define the target-block error in Eq. 3. FlowDPO and Circular-DPO normally compare ordered clean samples. In our primary metric-ranked construction, both trajectories instead share one GT clean latent and differ only in seed-determined noise; offline candidate ranking fixes their winner/loser order. The frozen θ_{ref} is the DiT at the start of the current stage; both trajectories share c and t.

4. Method

Our goal is to adapt the pretrained part generator from a limited set of GT part decompositions so that it improves a selected part while preserving the remaining parts. PartCrafter represents each part as a separate latent-token block under a shared DiT, allowing the FlowDPO error surrogate to be evaluated on target block p. This localizes the training signal, although the parameter update still affects the shared DiT. Three ingredients realize it: (1) preference pairs built offline by ranking seed-generated candidates with GT part metrics, so the winner and loser trajectories share one GT clean latent and differ only in seeded noise; (2) a reference-relative loss on block p that favors the winner trajectory; and (3) target-part and all-part reconstruction on the winner that supply absolute GT supervision and keep the surrounding object faithful. Sections 4.1–4.2 detail the pair construction and its controls, and Sec. 4.3 presents the objective.

4.1. Metric-Supervised Trajectory Pairs

A training example is $(x, \mathcal{G}, p, s^w, s^\ell)$, containing the input image, GT part set, target-part index, and stored winner and loser random seeds. Before fine-tuning, the pair builder samples multiple candidates for the same image and requested part count. It evaluates each candidate's indexed part p against g_p with the automatic score in supplementary Sec. S2. For each retained pair, the higher- and lower-scoring candidates determine s^w and s^ℓ, which differ as seed-conditioned trajectories rather than as two encoded candidate meshes. The trainer consumes this stored order rather than recomputing candidate-quality scores. The GT part annotation supplies the clean geometry used by the flow-matching targets.

Encoding $\mathcal{G}$ gives compositional latent $z_0 \in \mathbb{R}^{P\times N\times d}$. The stored seeds deterministically instantiate noise tensors ϵ^w and ϵ^ℓ of the same shape, producing two noised latent states from the same GT compositional latent. The primary construction therefore compares seed-conditioned stochastic trajectories toward one target geometry rather than two encoded candidate meshes. Candidate ranking is used only offline to decide the winner/loser order; GT part geometry remains the clean training target. In the human-ranked controls, visual inspection labels the seeds associated with two generated candidates. Synthetic-crop controls instead crop the loser target surface and independently re-encode the complete loser part set. The non-target surfaces are unchanged, but their stochastic VAE latent samples need not be numerically identical across the two trajectories.

4.2. Pair Construction And Controls

The primary construction yields one winner/loser seed pair for each selected target part. Controls introduce visual ordering through human ranking, change the loser clean geometry through synthetic cropping, or perturb stored pairs through label reversal and random-pair augmentation. Section 5.1 and supplementary Sec. S2 give the dataset scales, intervention ratios, and configurations.

4.3. Target-Block Preference Objective

For each ordered pair, Eq. 1 is instantiated separately for $i \in \{w, \ell\}$. The generic error in Eq. 2 averages the squared error over all P part blocks, with N tokens and d features in each block. We restrict the preference comparison to the tokens and features of target block p.

$$\mathrm{MSE}_t^p(\xi^i;\theta) = \frac{1}{N\,d}\left\|\hat{u}_{\theta,p}(z_t^i,t;c) - u_p^i\right\|_F^2. \tag{3}$$

For any part $q \in \{1,\dots,P\}$, MSE_t^q is defined by replacing p with q in Eq. 3. For each trajectory, the trainable and reference models receive the same (z_t^i, c, t). Let r_i denote the reference-relative reduction in target-part error. A positive r_i means that the trainable DiT predicts trajectory i more accurately than the frozen reference at that timestep. In Eq. 4, $\sigma(\cdot)$ without a timestep subscript is the logistic sigmoid and is distinct from the scheduler level σ_t in Eq. 1. As in Circular-DPO, the per-example, per-timestep pair loss is

$$\begin{aligned} r_i &= \mathrm{MSE}_t^p(\xi^i;\theta_{\mathrm{ref}}) - \mathrm{MSE}_t^p(\xi^i;\theta), \qquad i \in \{w,\ell\},\\ \ell_{\mathrm{pair}} &= -\log\sigma(\beta\,[r_w - r_\ell])\,. \end{aligned} \tag{4}$$

Equation 4 keeps Circular-DPO's reference-relative winner-minus-loser structure but replaces its full-latent flow-matching error with target-block error. Minimizing it increases $r_w - r_\ell$. Relative to the frozen reference, the trainable DiT is encouraged to reduce target-part error more on the winner trajectory than on the loser trajectory. It does not independently require $\mathrm{MSE}_t^p(\xi^\ell;\theta) < \mathrm{MSE}_t^p(\xi^\ell;\theta_{\mathrm{ref}})$. Its gradient is $\nabla_\theta \ell_{\mathrm{pair}} = \beta\sigma[-\beta(r_w - r_\ell)]\,(\nabla_\theta\,\mathrm{MSE}_t^p(\xi^w;\theta) - \nabla_\theta\,\mathrm{MSE}_t^p(\xi^\ell;\theta))$, which is generally nonzero even when the primary pair shares z_0 because its two noises produce different inputs and velocity targets.

Two reconstruction terms on the winner trajectory retain absolute GT supervision. Let $\mathcal{L}_{\mathrm{pair}} = \mathbb{E}[\ell_{\mathrm{pair}}]$ and $\mathcal{L}_{\mathrm{target}} = \mathbb{E}[\mathrm{MSE}_t^p(\xi^w;\theta)]$ denote expectations over training examples and sampled timesteps. We compute $\mathcal{L}_{\mathrm{all}}$ by averaging $\mathrm{MSE}_t^q(\xi^w;\theta)$ over every part block q of every winner example. An example with P parts therefore contributes P blocks. In the reported primary configurations, each stored pair has unit example weight and pair margins are disabled. The implemented optimization schedule is

$$\mathcal{L}_s(\theta) = \begin{cases} \mathcal{L}_{\mathrm{target}} + \lambda_{\mathrm{all}}\mathcal{L}_{\mathrm{all}}, & s < S_{\mathrm{warm}},\\ \lambda_{\mathrm{pair}}\mathcal{L}_{\mathrm{pair}} + \lambda_{\mathrm{part}}\mathcal{L}_{\mathrm{target}} + \lambda_{\mathrm{all}}\mathcal{L}_{\mathrm{all}}, & s \geq S_{\mathrm{warm}}. \end{cases} \tag{5}$$

Here s without a superscript is the update index, distinct from the stored seeds s^w and s^ℓ, and S_{warm} is the reconstruction-only warm-up duration. Equation 5 switches once from reconstruction-only warm-up to the combined objective. Its coefficients are fixed within each phase. Before the switch, the pairwise term is absent and target reconstruction has coefficient one. Afterwards, the winner trajectory enters the pairwise term and both reconstruction terms, whereas the loser trajectory enters only the pairwise term. The reconstruction terms compare the trainable DiT with GT flow targets. They do not constrain its outputs or parameters to remain close to θ_{ref}. The frozen reference appears only in $\mathcal{L}_{\mathrm{pair}}$. Direct SFT instead uses standard flow matching without preference pairs. Supplementary Sec. S2 reports the configuration values.

5. Experiments

5.1. Experimental Setup

Training Data. We construct the main training set from PartObjaverse-Tiny [28]. After retaining objects with two to eight usable parts and removing parts whose face fraction is below 0.005, the set contains 140 objects and 796 target-part pair examples. For each example, the offline pair builder samples candidates and stores the winner/loser seeds, target-part index, and indexed GT geometry. During fine-tuning, the seeds generate independent noise for the same GT compositional latent, as defined in Sec. 4.1. Candidate-quality metrics are used only during offline pair construction and are not optimized as rewards. The automatic score is a weighted sum of object FS/CD (0.35/0.20), target-part FS/CD (0.20/0.10), normal consistency (0.10), and component quality (0.05). Examples are ranked by the across-seed spread of this score on the indexed part. Supplementary Sec. S2 gives the complete definition.

The training-set-size study spans three scales: small (56

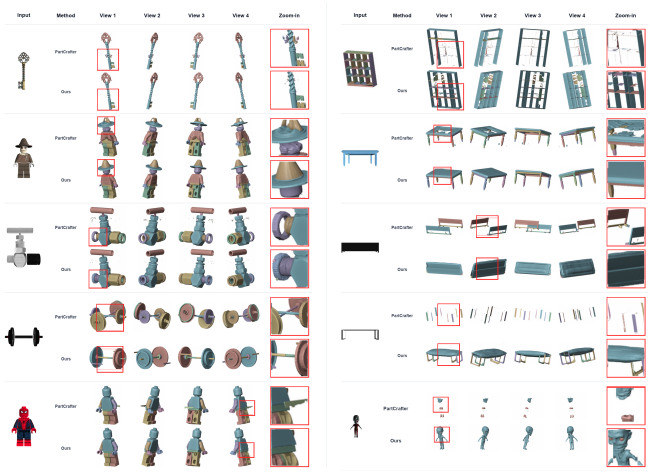

Figure 2. Qualitative effects of part-level alignment. Each comparison uses the same input image and requested part count for PartCrafter and Ours. Across articulated characters, furniture, and mechanical objects, Ours gives more complete target geometry and cleaner separation from neighboring parts; the zoomed regions highlight thin structures, contact boundaries, and small components.

Table 1. Comparison by input part count on held-out Objaverse [5]. We report object/matched-part Chamfer Distance (Obj-CD/Part-CD) and F-score (Obj-FS/Part-FS). The pretrained model is PartCrafter [14]. Ours is the large-scale setting; the medium setting appears as B2 in Table 2. SFT fine-tunes 140 GT objects with standard flow matching. The 2–8 average (avg.) covers valid multi-part examples. Supplementary Table S2 gives complete cross-dataset results.

Part Count	Method	Obj-CD ↓	Obj-FS ↑	Part-CD ↓	Part-FS ↑
1 part	Pretrained	0.2382	0.6366	0.2096	0.6949
	SFT	**0.2243**	**0.6441**	**0.1890**	**0.7273**
	Ours	**0.2184**	**0.6603**	**0.2044**	**0.7020**
2–4 parts	Pretrained	0.2631	0.5909	0.2574	0.5986
	SFT	**0.2568**	**0.5945**	0.2620	0.5955
	Ours	**0.2491**	**0.6121**	**0.2485**	**0.6237**
5–8 parts	Pretrained	0.2865	0.5332	0.3333	0.4646
	SFT	**0.2813**	**0.5463**	**0.3197**	**0.4846**
	Ours	**0.2824**	**0.5520**	**0.3062**	**0.4912**
avg.	Pretrained	0.2724	0.5680	0.2875	0.5453
	SFT	**0.2667**	**0.5751**	**0.2852**	**0.5509**
	Ours	**0.2625**	**0.5881**	**0.2716**	**0.5707**

pairs from 7 objects), medium (363 pairs from 50 objects), and large (796 pairs from 140 objects). The small and medium sets are selected from the ranked pool by usable part count, minimum part fraction, and category coverage. The small and medium checkpoints are first-stage runs, while the large set is a continuation checkpoint with refreshed stored pairs. Additional studies use 7 or 41 human-ranked pairs, crop the loser target surface before encoding, reverse the winner/loser labels, or keep the entire large set while adding 10% or 30% of randomly ordered alternatives. In particular, synthetic cropping changes the loser clean latent and is therefore not equivalent to reordering two noise realizations of the same GT latent.

Optimization. First-stage checkpoints initialize from released PartCrafter weights, while the continued large-set checkpoint initializes from the preceding first-stage model. Fine-tuning updates only the PartCrafter DiT θ. The image encoder, VAE encoder and decoder, and a stage-initial reference copy of the DiT remain frozen. The reported models use $\beta = 0.10$ and combine the reference-relative pairwise loss with target-part and all-part reconstruction. Matched reconstruction-only controls retain their corresponding training configurations but set $\lambda_{\text{pair}} = 0$. The primary control matches the medium-set first stage, and the secondary control matches the continued large-set setting. We use $(\lambda_{\text{pair}}, \lambda_{\text{part}}, \lambda_{\text{all}}) = (0.40, 0.30, 0.05)$, AdamW with learning rate 3×10^{-4}, batch size eight, fp16, EMA, and classifier-free-guidance dropout 0.1 on a single GPU. Supplementary Sec. S2 provides the complete settings.

Evaluation Protocol. For each of the three source datasets, Objaverse [5], Amazon Berkeley Objects (ABO) [4], and ShapeNet [1], we construct one fixed evaluation list shared by all methods. Before evaluation, each list is filtered against the union of object identities used by all fine-tuning configurations. No training input image, GT mesh, or target-part annotation is therefore reused for scoring. Because PartObjaverse-Tiny is derived from Objaverse, the held-out Objaverse list is the closest to the training distribution. ABO and ShapeNet test transfer across changes in source provenance, category composition, and part statistics. Multi-part inputs are the primary evaluation because they require both component generation and part separation. Single-part inputs are included as an auxiliary case.

Following PartCrafter's evaluation protocol [14], we report L_2 Chamfer Distance (CD) [7] and F-score (FS) [26] at the object level (Obj-CD/Obj-FS) and matched-part level (Part-CD/Part-FS). In this paper, *target part* denotes the part indexed by p during training, whereas *matched part* denotes a predicted part after assignment to a GT part during evaluation. Across the main-paper tables, bold metric values improve on the corresponding pretrained result, and underlines mark the best fine-tuned setting for each metric. Part metrics sample 40,960 surface points per mesh, build a GT-by-prediction L_2 CD cost matrix, and solve the rectangular one-to-one assignment exactly; an unmatched GT part takes penalty $\rho = 1.25 \max(1, \max C)$ and $\text{FS} = 0$. Supplementary Sec. S2 defines point-cloud normalization, part assignment, and the FS threshold, which is set to 0.1 times the normalized reference bounding-box diagonal.

5.2. Part-Aware Generation From Limited GT

Performance Across Part Counts. Table 1 shows that absolute F-score declines as the requested decomposition grows from one part to 5–8 parts, indicating greater compositional difficulty. Ours improves both object and matched-

part F-score over pretrained PartCrafter in every group and gives the strongest matched-part result for both 2–4 and 5–8 parts. SFT gives the best one-part matched-part metrics, but this advantage disappears for multi-part inputs, where more components must be generated and separated. The 5–8-part result therefore indicates that target-part preference learning remains effective under more complex decompositions; on the 2–8-part aggregate, matched-part F-score improves by about 5% over the pretrained model alongside object-level gains. Figure 2 shows representative improvements in part completeness and separation.

DPO Versus SFT. Both DPO training and SFT use part-level GT and update the same DiT, but organize supervision differently. SFT directly reconstructs 140 GT object samples with PartCrafter's standard flow-matching objective. The medium setting adds three sources of supervision: a reference-relative comparison between two noised trajectories, target-part GT reconstruction, and all-part GT reconstruction. The resulting matched-part F-score is about 4% higher than the reported SFT result on the common held-out Objaverse list. This result compares the two complete training procedures. Because their data organization and objectives differ, it should not be interpreted as a one-factor comparison between DPO and SFT.

Cross-Dataset Transfer. The cross-dataset results reveal a limitation of our work. Global shape improves on ABO, but the matched-part advantage does not transfer consistently to ABO or ShapeNet. Freezing the image encoder, 3D VAE, and reference model keeps the representation close to pretrained PartCrafter, whereas the pairwise update targets block p using PartObjaverse-Tiny geometry and stored trajectory order. This source-domain supervision may preserve coarse object structure more readily than target-block behavior under distribution shift. The similar behavior of direct SFT suggests that this transfer limitation is not specific to the pairwise term. Supplementary Sec. S3 (Table S1) reports the complete ABO and ShapeNet pair-construction results, while Sec. S4 (Table S2) reports cross-dataset results by input part count.

5.3. Effect Of Training-Set Scale

Comparison Across Scales. Family B in Table 2 compares the small, medium, and large settings. Their results are non-monotonic. The small setting does not improve matched-part quality over the pretrained reference. The medium and large settings both improve over the pretrained reference, with the medium setting slightly ahead on matched-part metrics. Training-set size alone therefore does not explain the observed differences; geometric and category coverage may also contribute.

The cross-dataset results expose a related coverage trade-off. The medium set transfers local part quality more effectively to ShapeNet, whereas the large set gives better global-shape metrics on ABO. Because the large-set model is a continuation checkpoint with refreshed seeds and the small and medium models are first-stage checkpoints, these results show a non-monotonic trend among the reported checkpoints rather than a universal optimum at the medium scale.

Table 2. Pair-construction, order-perturbation, and augmentation results on held-out Objaverse. The table reports 2–8-part averages for families A (pretrained reference), B (training-set scale), C (alternative construction), and D (pair perturbation). Supplementary Table S1 gives complete cross-dataset results.

Family	Setting	Obj-CD ↓	Obj-FS ↑	Part-CD ↓	Part-FS ↑
A	**A1** Pretrained	0.2724	0.5680	0.2875	0.5453
B	**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**
C	**C1** Human 7	**0.2603**	**0.5854**	**0.2805**	**0.5573**
	C2 Human 41	0.5127	0.1745	0.5616	0.1285
	C3 Crop 30%	0.3529	0.4127	0.3821	0.3398
	C4 Crop 50%	0.3636	0.3721	0.4222	0.2836
D	**D1** Reversed 10%	0.3848	0.3238	0.4336	0.2690
	D2 Reversed 30%	0.3628	0.3709	0.4267	0.2858
	D3 Reversed 50%	0.3738	0.3487	0.4480	0.2579
	D4 Random add. (+80)	**0.2461**	0.5330	0.3122	0.4319
	D5 Random add. (+239)	0.3542	0.3554	0.4398	0.2364

Non-Target Parts. Because the seed pair perturbs every part block, improving the target part could cost the others. Supplementary Sec. S6 tests this on 59 held-out multi-part ShapeNet objects: the medium setting raises the pretrained weakest-part F-score from 0.182 to 0.427 and the all-parts mean by 0.03, while its remaining-parts drop of 0.046 is not significant (bootstrap 95% CI $[-0.110, 0.018]$); only the small setting degrades the remaining parts markedly.

5.4. Alternative Pair Construction

Human-Ranked Pairs. Within family C, C1 and C2 replace metric ordering with visual comparison between generated candidates. The small, screened C1 set improves upon A1, whereas the larger C2 set degrades sharply. The objective consumes only the comparison direction, so ambiguous differences or inconsistent target-part judgments can weaken the signal. C1, which combines part-level metric checks with multiview inspection, suggests that comparison clarity can matter more than count.

Synthetic-Crop Pairs. Settings C3 and C4 alter loser construction rather than the ordering source. Synthetic cropping removes target-part surface points and re-encodes the loser part set, changing its clean latent. In this setup, neither cropped-target control improves on the pretrained baseline or the metric-ranked small setting. Increasing the crop severity also provides no consistent benefit.

5.5. Pair Perturbations

Pair Reversal. Family D perturbs or augments the stored metric-ranked pairs. D1–D3 swap winner and loser labels

Table 3. Effect of β on ShapeNet multi-part generation. Family E compares pretrained PartCrafter (E1) with fine-tuned settings E2–E5 on the same fixed list.

Setting	Obj-CD ↓	Obj-FS ↑	Part-CD ↓	Part-FS ↑
E1 Pretrained	0.1814	0.7130	0.3262	0.4457
E2 $\beta = 0.05$	0.3003	0.4293	0.3872	0.2931
E3 $\beta = 0.10$	**0.1776**	**0.7237**	0.3288	**0.4513**
E4 $\beta = 0.20$	0.2249	0.6194	**0.3049**	**0.4564**
E5 $\beta = 0.40$	0.2755	0.5033	0.3580	0.3368

for increasing fractions of the primary pairs. With the reference model fixed, each swap reverses the target-part comparison. All three levels reduce quality, so label flipping is not a useful augmentation for this objective. The decline also shows that stored pair order materially affects optimization.

Random-Pair Augmentation. Settings D4 and D5 keep all 796 rows and append rows with randomly drawn seed pairs, enlarging the set to 876 and 1035 examples. Unlike reversal, this changes trajectories without deliberately flipping an existing comparison. D4 gives the best object CD among family D, but its object F-score and matched-part metrics remain below pretrained PartCrafter. The benefit disappears at the larger augmentation. These controls provide no evidence that random-pair augmentation adds useful supervision.

5.6. Ablation Studies

Effect Of β. The parameter β scales the reference-relative trajectory gap in Eq. 4 and does not weight the reconstruction losses. Table 3 reports the ShapeNet β ablation (family E), evaluated on one fixed list. Because this checkpoint series and evaluation set differ from the Objaverse loss controls, absolute values should not be compared across the two tables.

The result is non-monotonic. E2 falls below E1 on all four metrics. E3 gives the best object metrics and improves Part-FS, whereas E4 gives the best matched-part metrics at the cost of object quality. E5 degrades both metric families. The reported configurations therefore set $\beta = 0.10$, the most balanced value in this ablation rather than a universal optimum.

Pairwise-Term Control. Table 4 reports family F on the held-out Objaverse set. F2/F3 and F4/F5 are matched pairs. Setting $\lambda_{\text{pair}} = 0$ retains winner-trajectory target-part and all-part reconstruction, along with the corresponding data, initialization stage, optimizer, budget, and evaluation, but removes the reference-relative comparison.

Adding the pairwise term improves every metric in both matched comparisons. In the primary medium-set control, it reduces object CD by about 27% relative to reconstruction-only training. Paired bootstrap intervals over the 200 shared inputs also exclude zero in the favorable direction for every metric (Supplementary Sec. S5). The continued large-set control shows the same directional pattern. These matched controls isolate the contribution of the pairwise term within the reported objective.

Table 4. Effect of the pairwise term on held-out Objaverse. Family F evaluates 2–8-part inputs with pretrained F1 as reference. F2/F3 and F4/F5 respectively share the medium- and large-scale configurations. "Recon." sets $\lambda_{\text{pair}} = 0$, while "full" includes the pairwise term. F3 is evaluated in a separate run from B2 in Table 2.

Setting	Obj-CD ↓	Obj-FS ↑	Part-CD ↓	Part-FS ↑
F1 Pretrained	0.2724	0.5680	0.2875	0.5453
F2 Medium, recon.	**0.2650**	0.5179	0.3001	0.4552
F3 Medium, full	**0.1941**	**0.7036**	**0.2624**	**0.5488**
F4 Large, recon.	0.2819	0.4878	0.3328	0.4061
F5 Large, full	**0.2625**	**0.5881**	**0.2716**	**0.5707**

6. Limitations And Future Work

Our gains are strongest on Objaverse and transfer less consistently to ABO and ShapeNet, whose category and decomposition statistics differ from PartObjaverse-Tiny. Broader pair collection across categories, part counts, thin structures, and target-domain samples may improve robustness. The method still requires part-level GT geometry as the reconstruction target, and offline pair ordering makes performance sensitive to candidate coverage, ordering reliability, and the selection criterion. Closing the loop by regenerating and rescoring candidates from the adapted model, while retaining a smaller GT subset for reconstruction, is a natural next step. Finally, because the preference loss acts on block p but updates a DiT shared by all blocks, stronger local alignment can shift whole-object geometry. The β sweep shows that the best local and global settings differ. All-part reconstruction constrains but does not eliminate this coupling; part-adaptive weights or lightweight adapters could address it.

7. Conclusion

We adapt a pretrained part-aware generator with metric-ranked target-part seed pairs and a reference-relative target-block preference loss. Target-part and all-part GT flow-matching losses provide absolute supervision, while only the DiT is updated. Among the tested training-set scales, the medium-scale setting gives the strongest held-out Objaverse result, raising matched-part F-score by approximately 5% over pretrained PartCrafter and exceeding the reported direct-SFT baseline on the multi-part aggregate. Matched controls at the medium and large scales show that adding the pairwise term improves upon reconstruction-only training. Results across the three scales are non-monotonic, while the remaining controls show sensitivity to pair ordering, target-part coverage, and objective balance. Because gains transfer less consistently to ABO and ShapeNet, we do not claim broad cross-dataset improvement.

References

[1] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[2] M. Chen, R. Shapovalov, I. Laina, T. Monnier, J. Wang, D. Novotny, and A. Vedaldi. PartGen: Part-level 3D generation and reconstruction with multi-view diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5881–5892, 2025.

[3] M. Chen, J. Wang, R. Shapovalov, T. Monnier, H. Jung, D. Wang, R. Ranjan, I. Laina, and A. Vedaldi. AutoPartGen: Autoregressive 3D part generation and discovery. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[4] J. Collins, S. Goel, K. Deng, A. Luthra, L. Xu, E. Gundogdu, X. Zhang, T. F. Y. Vicente, T. Dideriksen, H. Arora, M. Guillaumin, and J. Malik. ABO: Dataset and benchmarks for real-world 3D object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21126–21136, 2022.

[5] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3D objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[6] S. Dong, L. Ding, X. Chen, Y. Li, Y. Wang, Y. Wang, W. Qi, J. Kim, C. Gao, Z. Huang, Z. Wang, T. Xue, and D. Xu. From one to more: Contextual part latents for 3D generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8230–8240, 2025.

[7] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 605–613, 2017.

[8] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan. LRM: Large reconstruction model for single image to 3D. In *The Twelfth International Conference on Learning Representations*, 2024.

[9] R. Jiao, X. Kong, W. Huang, and Y. Liu. 3D structure prediction of atomic systems with flow-based direct preference optimization. In *Advances in Neural Information Processing Systems*, volume 37, pages 110197–110217, 2024.

[10] P. Li, Z. Zhang, and H. Tang. PartRAG: Retrieval-augmented part-level 3D generation and editing. *arXiv preprint arXiv:2602.17033*, 2026.

[11] Y. Li, Z.-X. Zou, Z. Liu, D. Wang, Y. Liang, Z. Yu, X. Liu, Y.-C. Guo, D. Liang, W. Ouyang, and Y.-P. Cao. TripoSG: High-fidelity 3D shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2025.

[12] Z. Li, J. Ma, H. Xu, W. Zheng, Y. Zhu, C. Meng, P. Chen, L. Yang, Z. Yang, C. Yang, G. Yang, I. Koh, and L. Sun. Circular-DPO: Aligning multi-stage 3D generative models via preference feedback loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 32592–32601, 2026.

[13] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin. Magic3D: High-resolution text-to-3D content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.

[14] Y. Lin, C. Lin, P. Pan, H. Yan, Y. Feng, Y. Mu, and K. Fragkiadaki. PartCrafter: Structured 3D mesh generation via compositional latent diffusion transformers. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[15] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[16] J. Liu, D. Iliash, A. X. Chang, M. Savva, and A. Mahdavi-Amiri. SINGAPO: Single image controlled generation of articulated parts in objects. In *The Thirteenth International Conference on Learning Representations*, 2025.

[17] J. Liu, J. Xu, S. Guo, J. Li, J. Guo, J. Yu, H. Weng, B. Lei, X. Yang, Z. Chen, F. Zhu, T. Han, and C. Guo. Mesh-RFT: Enhancing mesh generation via fine-grained reinforcement fine-tuning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[18] Q. Liu, X. Yao, S. Zhang, Y. Deng, G. Liu, Z. Liu, and K. Jia. PAct: Part-decomposed single-view articulated object generation. *arXiv preprint arXiv:2602.14965*, 2026.

[19] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023.

[20] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

[21] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019.

[22] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.

[23] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741, 2023.

[24] S. Ray Chowdhury, A. Kini, and N. Natarajan. Provably robust DPO: Aligning language models with noisy feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 42258–42274. PMLR, 2024.

[25] J. Tang, R. Lu, M. Li, Z. Hao, X. Li, F. Wei, S. Song, G. Zeng, M.-Y. Liu, and T.-Y. Lin. Efficient part-level 3D object generation via dual volume packing. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[26] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox. What do single-view 3D reconstruction networks learn? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019.

[27] B. Wallace, M. Dang, R. Rafailov, L. Zhou, A. Lou, S. Purushwalkam, S. Ermon, C. Xiong, S. Joty, and N. Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

[28] Y. Yang, Y. Huang, Y.-C. Guo, L. Lu, X. Wu, E. Y. Lam, Y.-P. Cao, and X. Liu. SAMPart3D: Segment any part in 3D objects. *arXiv preprint arXiv:2411.07184*, 2024.

[29] Y. Yang, Y. Zhou, Y.-C. Guo, Z.-X. Zou, Y. Huang, Y.-T. Liu, H. Xu, D. Liang, Y.-P. Cao, and X. Liu. OmniPart: Part-aware 3D generation with semantic decoupling and structural cohesion. In *ACM SIGGRAPH Asia 2025 Conference Papers*, pages 1–12, 2025.

[30] Z. Zhou, X. Xia, F. Ma, H. Fan, Y. Yang, and T.-S. Chua. DreamDPO: Aligning text-to-3D generation with human preferences via direct preference optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 79414–79435. PMLR, 2025.