Rigid 3D Object Alignment: Optimization vs. Feed-Forward Prediction

Ostap Khomenko[1] and Maksym Shamrai[1]

MacPaw AI & Research

Abstract. We study rigid 3D asset alignment: placing an independently generated 3D asset onto a base object via a rigid transformation, without modifying geometry. We contrast two paradigms: (i) an optimization baseline based on differentiable rendering, which is accurate but requires reference views and minutes of compute per asset, and (ii) a feed-forward Vision–Language–Action (VLA) model that regresses the transformation in a single sub-second forward pass, optionally without any reference views. To enable this study, we curate a large-scale dataset of 905k asset–base pairs with ground-truth transformations derived from PartVerse-XL. Our analysis reveals a pronounced accuracy–efficiency trade-off: the feed-forward model matches the optimization baseline on translation while falling substantially short on rotation, and we analyze the causes of this gap. Our results position feed-forward prediction as a fast, reference-free initialization for downstream refinement, and the dataset as a foundation for future alignment research.

Keywords: 3D asset alignment · Vision–Language–Action models · Differentiable rendering

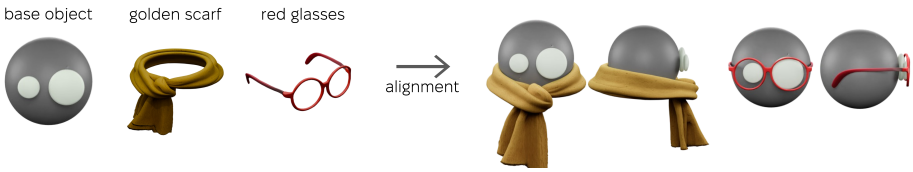

Fig. 1: 3D alignment of independently generated assets onto a common base object via rigid transformations, without modifying the geometry of either object.

1 Introduction

Asset integration is an important problem in 3D content creation, where the goal is to place independently generated 3D assets onto a base object in a geometrically and semantically consistent manner. This capability is critical for scalable content creation in games, film, AR/VR, and simulation. Existing approaches typically rely on template bodies [12], reconstruct full scenes jointly [15], or co-generate assets and base objects in a single pass. Such strategies entangle geometry and often distort the base mesh, limiting flexibility and generalization. We

instead formulate the asset integration task as *rigid alignment of independently generated assets*. By decoupling generation from alignment, this formulation preserves the original geometry of both objects and enables integration onto novel base objects without template-specific fine-tuning. The resulting aligned assets can be directly used for downstream tasks such as rigging and animation [11].

We conduct a study contrasting optimization-based and feed-forward paradigms for rigid asset alignment. On the feed-forward side, we adapt a Vision–Language–Action (VLA) model to regress rigid transformations in a single forward pass, enabling reference-free alignment in under one second[1]. We deliberately do *not* claim that the feed-forward model matches optimization accuracy—it does not, particularly for rotation–and we analyze why. Instead, we position feed-forward prediction as a fast, scalable component: a reference-free estimate that is competitive on translation and can serve as an initialization for downstream refinement.

Contributions.

1. **Large-scale dataset for rigid asset alignment.** Building on PartVerse-XL [6], we curate and release a dataset of approximately 905k asset–base object pairs with ground-truth rigid transformations, canonical multi-view renderings, depth, and visual prompts.
2. **A feed-forward VLA formulation of rigid alignment.** We adapt and fine-tuned a robotics-pretrained VLA model with decoupled translation and rotation heads to predict rigid transformations in a single sub-second forward pass, with or without reference views at inference time.
3. **Systematic analysis of the accuracy–efficiency trade-off.** We quantify the gap between optimization and feed-forward prediction, including a timing analysis, an investigation of why rotation estimation fails while translation succeeds, and an ablation of robotics-specific pretraining.

2 Related Work

Prior work on 3D alignment typically employs semantic-driven techniques or reference-guided approaches like REPARO [7], which leverages differentiable rendering to iteratively optimize transformations that match a target image. While accurate, these methods require high-quality reference images and incur significant computational overhead. In contrast, semantic-driven approaches [1, 13] predict alignment without explicit references by learning spatial and functional priors. However, they are often limited by narrow training distributions and coarse spatial reasoning. Other geometric methods [10] often restrict alignment to low-level surface matching. While CRAG [8] and Rectified Point Flow [14] could be adapted to our problem, our work investigates the adaptability of Vision-Language-Action (VLA) models—traditionally designed for robotics to the digital 3D asset alignment task. VLA models directly map visual observations and textual instructions to continuous spatial control actions. We hypothesize that this architecture can naturally generalize to 3D asset alignment: the

[1] `https://github.com/MacPaw/rigid-3d-alignment`

vision-language backbone offers the rich semantic reasoning needed to infer object orientation and spatial context, while the action head can be repurposed to execute the precise geometric transformations (translation and rotation) required for exact alignment.

3 Problem Statement and Dataset

3.1 Problem Statement

Given two objects, a source (*src*, the asset) and a target (*tgt*, the base object), we aim to find a transformation $\tau = \{\mathbf{t}, \mathbf{R}\}$, where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ denote rotation and translation, respectively. To ensure that rotation is invariant to the asset's global position, we define the transformation relative to the source object's centroid $\mathbf{c}_{\text{src}} \in \mathbb{R}^3$ and apply it only to the source object:

$$\Phi(\mathbf{x}; \tau) = \mathbf{R}(\mathbf{x} - \mathbf{c}_{\text{src}}) + \mathbf{c}_{\text{src}} + \mathbf{t}. \tag{1}$$

This formulation decouples the learned rotation from the global spatial displacement.

3.2 Dataset

Starting from the raw PartVerse-XL dataset [6] (40k scenes originally derived from Objaverse-XL [5], yielding 320k object pairs), we perform a preprocessing step to generate meaningful source–target training pairs. For each scene, we designate a single object as the source and treat the remaining scene objects as a composite target. To ensure that source and target are rendered at comparable scale, we enforce a relative volume constraint, filtering with an empirically chosen threshold $\lambda = 0.1$ such that only pairs satisfying $\min\left(\frac{V_{\text{src}}}{V_{\text{tgt}}}, \frac{V_{\text{tgt}}}{V_{\text{src}}}\right) > \lambda$ are preserved, where V_{src} and V_{tgt} denote the source and target volumes.

For each validated pair, we apply the inverse transformation $\Phi^{-1}(\mathbf{x}; \tau)$ and render three canonical views with fixed camera intrinsics—projections onto the ZY (right), ZX (rear), and XY (top) planes. To enhance the spatial reasoning of the VLA model, we augment these renderings with Set-of-Marks visual prompting [16]; a representative sample is shown in Fig. 2. We additionally incorporate depth images alongside the 2D orthographic renderings, and apply spatial augmentations so that the predicted transformation τ remains robust across diverse visibility conditions. Generating eight augmented variations per unique source-target configuration expands the dataset to 905k samples. For training, we select only the 452k samples that feature both rotations and translations, while reserving 2,360 samples for testing.

4 Methodology

We study two alignment paradigms under identical input assumptions. The differentiable-rendering method of Sec. 4.1 serves purely as an *optimization base-*

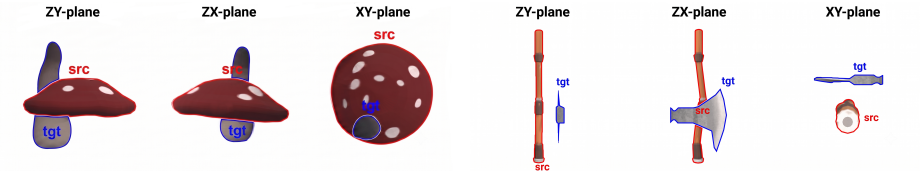

Fig. 2: Dataset samples. Representative canonical renderings with Set-of-Marks (SoM) visual prompting.

line. Our proposed method is the feed-forward VLA model of Sec. 4.2, which predicts the transformation for the source asset directly and requires no per-asset optimization for either object.

4.1 Baseline: Differentiable Rendering

We adapt REPARO [7] to estimate the rigid transformation τ of the source asset. This baseline employs differentiable rendering to optimize τ by minimizing the Sinkhorn distance [4] between *reference views* of the correctly assembled scene and renderings of the composed source–target scene under the current transformation estimate. While effective, this optimization has two drawbacks: it requires access to reference views of the assembled configuration, and it incurs high computational cost due to full-scene rendering at every optimization step.

To quantify this cost, we measure the per-step latency of the optimization, rendering at 256×256 resolution with nvdiffrast [9] on an NVIDIA RTX 4090 GPU, and report the average as a function of the number of reference viewpoints across several test scenes (see Fig. 3). While a single step appears inexpensive, the full optimization typically spans 250–600 steps, resulting in alignment times exceeding 10 minutes per asset.

4.2 Feed-Forward VLA

Given the requirement for both semantically aware and geometrically grounded transformation prediction, we build our feed-forward predictor on a Vision–Language–Action model. We adapt the VLA model from Xiaomi-Robotics-0 [3] by introducing decoupled regression heads for translation and rotation. For the rotation head, we utilize a continuous 6D rotation representation [17] and predict the corresponding denoising velocity in $\mathbb{R}^6$ with a diffusion-transformer (DiT) action expert. The integrated pipeline is illustrated in Fig. 4. We fine-tune the pretrained model on our dataset, freezing the VLM backbone and training only the DiT heads.

5 Evaluation and Analysis

We compare the two alignment paradigms using rotation and translation errors (GE(R), RMSE(t)), geometric distances (Chamfer distance and ADD-S), and

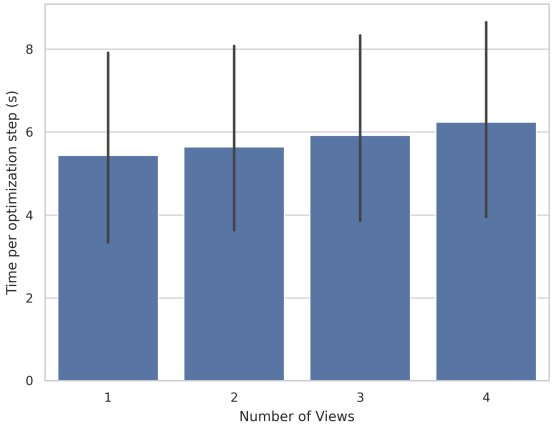

Fig. 3: Mean latency per optimization step (seconds) across test scenes, as a function of the number of reference viewpoints.

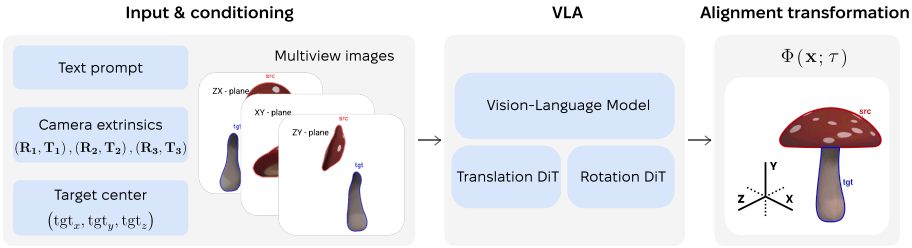

Fig. 4: Feed-forward asset alignment pipeline. Given an independently generated asset *(src)* and a base object *(tgt)*, the model predicts a rigid transformation τ in a single forward pass. From multi-view canonical projections, the corresponding camera extrinsics, and the target center in the 3D scene, decoupled translation and rotation parameters are estimated through a Vision–Language–Action (VLA) model.

accuracy, defined as the percentage of predictions with ADD-S below 10% of the object diameter (see Tab. 1). Differentiable rendering achieves the best performance across all metrics, driven by its iterative optimization, with an angular error of 52.92° and an alignment accuracy of 79.36%. This precision, however, comes with the computational bottleneck quantified in Sec. 4.1.

The feed-forward VLA infers the transformation in a single forward pass, reducing inference latency to under one second. It trades geometric accuracy for efficiency: translation error remains comparable to the baseline (0.18–0.22), but rotation error is substantially higher ($\approx$ 119°–124°), yielding ADD-S of $\approx$ 0.23–0.25 and accuracy of 37.6%–40.2%. We emphasize that a rotation error of this magnitude means the feed-forward model does not, on its own, deliver production-ready orientation; its value lies in speed, reference-free operation,

Table 1: Comparison of alignment methods on 2360 held-out asset–part pairs. "Ref. views" indicates whether the method receives reference views of the correctly assembled scene at inference time. Diff. rendering is an optimization baseline (Sec. 4.1); the VLA variants are single-pass feed-forward predictions.

Method	GE(R)↓	RMSE(t)↓	CD(R, t)↓	ADD-S↓	Acc.↑	Ref.
Diff. rendering	$\mathbf{52.92 \pm 51.73}$	$\mathbf{0.1523 \pm 0.1943}$	$\mathbf{0.1415 \pm 0.2430}$	$\mathbf{0.1438 \pm 0.2492}$	**79.36%**	Yes
VLA	124.02 ± 42.70	0.1882 ± 0.1471	0.2345 ± 0.1691	0.2360 ± 0.1727	37.55%	Yes
VLA	119.17 ± 51.96	0.2295 ± 0.1724	0.2548 ± 0.2080	0.2527 ± 0.2037	40.19%	No

competitive translation, and its potential as an initialization for optimization-based refinement (Sec. 6). Importantly, VLA without ground truth performs similarly to its supervised variant when estimating translations, demonstrating that alignment can be achieved without optimization or supervision. Furthermore, while optimization via differentiable rendering does not rely on semantic understanding, this method does. This highlights a substantial accuracy–efficiency trade-off, with VLA enabling orders-of-magnitude faster and scalable alignment.

Why Does Rotation Fail? We hypothesize that the dominant cause is the inadequate conditioning signal available to the VLA model as a proprioceptive state for rotation estimation. For translation, the target center together with the camera extrinsics and the visual input is sufficient to estimate the displacement of the source object, whereas for rotation this is not the case. We verify the role of visual conditioning for translation with a shuffling test: randomly pairing each sample's input images with those of a different asset causes RMSE(t) to increase sharply, confirming that the model is genuinely conditioned on the images rather than on the numeric inputs alone. For rotation, however, the target center and camera extrinsics carry no information about orientation, and the visual input alone appears insufficient to compensate.

Does Robotics Pretraining Help? To analyze whether a robotics-pretrained VLM is a good fit for this task, we conduct a controlled study: we remove the DiT components and train the VLM with a cross-entropy loss to predict discretized action tokens, comparing the pretrained Xiaomi-Robotics-0 [3] checkpoint against its base model, Qwen3-VL-4B-Instruct [2]. The training curves (see Fig. 5) indicate that domain-specific robotics pretraining provides a superior initialization for asset alignment: the robotics-oriented checkpoint achieves consistently lower mean cross-entropy losses for both translation and rotation, suggesting a stronger spatial prior than the general-purpose vision–language baseline.

6 Conclusion and Future Work

We presented a study of rigid alignment of heterogeneous 3D objects without modifying geometry, contrasting optimization-based and feed-forward paradigms,

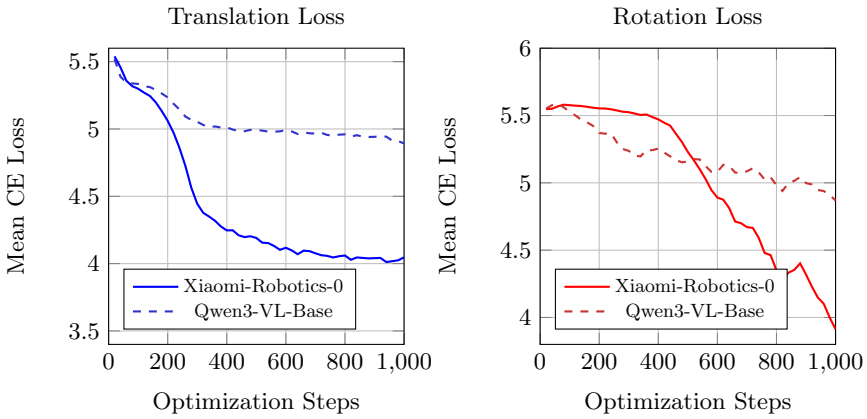

Fig. 5: Comparison of training convergence between the robotics-pretrained Xiaomi-Robotics-0 VLM (solid lines) and the base Qwen3-VL checkpoint (dashed lines).

and we processed a large-scale dataset of 905k asset–base object alignment samples with corresponding rigid transformations. Differentiable rendering achieves the highest accuracy through iterative refinement, while the proposed VLA-based approach enables single-pass, reference-free prediction that is roughly 600× faster and matches the baseline on translation, though not on rotation. These results indicate that translation-level alignment can be learned rather than optimized, while accurate orientation currently still requires optimization—a trade-off that motivates hybrid designs.

Several directions follow directly from our analysis. First, incorporating depth and orientation-bearing cues into the VLA conditioning may substantially improve rotation estimation. Second, iterative or refinement-based inference (*e.g.*, recurrent updates or self-conditioning) could bridge the gap to optimization while retaining efficiency. Third, enriching the dataset with diverse language instructions would enable instruction-conditioned disambiguation of placements. Finally, hybrid pipelines that use the VLA prediction as an initialization for differentiable-rendering refinement may combine feed-forward speed with optimization accuracy; given that our model already provides baseline-level translation in under a second, this is the most immediate practical application of our findings.

Limitations Our study has several limitations. First, we do not compare against fracture-assembly or scene-placement methods (see Sec. 2), as their input assumptions do not transfer to heterogeneous, independently generated asset pairs; constructing an adapted protocol that would make such a comparison meaningful is left for future work. Second, our language conditioning is currently limited: the dataset does not yet contain instruction diversity that would exercise ambiguity resolution, which we view as the natural setting where the language pathway of a VLA becomes essential.

References

1. Abdelreheem, A., Aleotti, F., Watson, J., Qureshi, Z., Eldesokey, A., Wonka, P., Brostow, G., Vicente, S., Garcia-Hernando, G.: Placeit3d: Language-guided object placement in real 3d scenes (2025), https://arxiv.org/abs/2505.05288
2. Bai, S., Cai, Y., Chen, R., et al.: Qwen3-vl technical report (2025), https://arxiv.org/abs/2511.21631
3. Cai, R., Guo, J., He, X., Jin, P., Li, J., Lin, B., Liu, F., Liu, W., Ma, F., Ma, K., Qiu, F., Qu, H., Su, Y., Sun, Q., Wang, D., Wang, D., Wang, Y., Wu, R., Xiang, D., Yang, Y., Ye, H., Zhang, Y., Zhou, Q.: Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution (2026), https://arxiv.org/abs/2602.12684
4. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) Advances in Neural Information Processing Systems. vol. 26. Curran Associates, Inc. (2013), https://proceedings.neurips.cc/paper_files/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf
5. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., VanderBilt, E., Kembhavi, A., Vondrick, C., Gkioxari, G., Ehsani, K., Schmidt, L., Farhadi, A.: Objaverse-xl: A universe of 10m+ 3d objects. arXiv preprint arXiv:2307.05663 (2023), https://arxiv.org/abs/2307.05663
6. Ding, L., Dong, S., Li, Y., Gao, C., Chen, X., Han, R., Kuang, Y., Zhang, H., Huang, B., Huang, Z., Wang, Z., Xu, D., Xue, T.: Fullpart: Generating each 3d part at full resolution (2025), https://arxiv.org/abs/2510.26140
7. Han, H., Yang, R., Liao, H., Xing, J., Xu, Z., Yu, X., Zha, J., Li, X., Li, W.: Reparo: Compositional 3d assets generation with differentiable 3d layout alignment (2024), https://arxiv.org/abs/2405.18525
8. Jiang, Z., Li, S., Tan, S., Xu, C., Zhang, J., Galway-Witham, J., Wang, X., Williams, S.A., Iovita, R., Feng, C., Zhang, J.: Crag: Can 3d generative models help 3d assembly? In: International Conference on Machine Learning (ICML) (2026), https://arxiv.org/abs/2602.22629
9. Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., Aila, T.: Modular primitives for high-performance differentiable rendering. ACM Transactions on Graphics **39**(6) (2020). https://doi.org/10.1145/3414685.3417861
10. Li, S., Jiang, Z., Chen, G., Xu, C., Tan, S., Wang, X., Fang, I., Zyskowski, K., McPherron, S.P., Iovita, R., Feng, C., Zhang, J.: Garf: Learning generalizable 3d reassembly for real-world fractures. In: International Conference on Computer Vision (ICCV) (2025), https://arxiv.org/abs/2504.05400
11. Liu, I., Xu, Z., Yifan, W., Tan, H., Xu, Z., Wang, X., Su, H., Shi, Z.: Riganything: Template-free autoregressive rigging for diverse 3d assets (2025), https://arxiv.org/abs/2502.09615
12. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 851–866 (2023). https://doi.org/10.1145/3596711.3596800
13. Qi, Y., Ju, Y., Wei, T., Chu, C., Wong, L.L., Xu, H.: Two by two: Learning multi-task pairwise objects assembly for generalizable robot manipulation. CVPR 2025 (2025), https://arxiv.org/abs/2504.06961
14. Sun, T., Zhu, L., Huang, S., Song, S., Armeni, I.: Rectified point flow: Generic point cloud pose estimation. In: Advances in Neural Information Processing Systems (NeurIPS) (2025), https://arxiv.org/abs/2506.05282

15. Team, S.D., Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., Lin, A., Liu, J., Ma, Z., Sagar, A., Song, B., Wang, X., Yang, J., Zhang, B., Dollár, P., Gkioxari, G., Feiszli, M., Malik, J.: Sam 3d: 3dfy anything in images (2025), https://arxiv.org/abs/2511.16624
16. Yang, J., Zhang, H., Li, F., Zou, X., Li, C., Gao, J.: Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. arXiv preprint arXiv:2310.11441 (2023), https://arxiv.org/abs/2310.11441
17. Zhou, Y., Barnes, C., Jingwan, L., Jimei, Y., Hao, L.: On the continuity of rotation representations in neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019), https://arxiv.org/abs/1812.07035