Input-Aware Routing of Image-to-3D Models for Robotic Manipulation

Akash Anand[1], Aditya Agarwal[1], Leslie Pack Kaelbling[1]

***Abstract*— Robotic manipulation pipelines increasingly rely on single-image 3D reconstruction, but no single reconstruction method is reliable across the diverse inputs a robot encounters. Image-to-3D models depend heavily on the input viewpoint and inherit architectural and training biases, while multi-view reconstruction methods (which we term view-invariant) avoid hallucination but require additional sensing time. Fixing a single method therefore exposes the downstream pipeline to that method's failure modes, and querying all methods is computationally prohibitive. We present SCOUT, a routing framework that selects a reconstruction model per input under user-specified cost constraints. SCOUT decouples reconstruction scores into the relative performance of viewpoint-dependent models and the overall difficulty of the input image, which stabilizes training and lets view-invariant pipelines be added or reconfigured without retraining. We evaluate on the Google Scanned Objects, BigBIRD, and YCB datasets, demonstrating consistent improvements over routing baselines adapted from the LLM literature, and further validate the framework through simulated and real-world manipulation experiments. We release the code and additional results on our website.**

I. INTRODUCTION

Robotic manipulation increasingly relies on single-image 3D reconstruction because a robot often has only a single initial view and acquiring additional viewpoints is time-consuming or infeasible [1]. Recent Image-to-3D models [2], [3], [4], [5] produce meshes from a single image in seconds, and several recent manipulation pipelines [6] adopt them for perception. These models have complementary strengths: their relative performance varies across object categories and input viewpoints; thus, no single model is reliable across the diverse inputs encountered in real-world deployment. Fixing a single choice therefore exposes the downstream manipulation pipeline to that model's failure modes.

Reconstruction quality requirements also vary by task: dexterous manipulation demands fine-grained surface detail, whereas collision-free motion planning tolerates coarser representations. Beyond Image-to-3D models, multi-view reconstruction methods provide reconstructions whose quality depends on the object texture rather than on the robot's initial viewpoint, given sufficient additional views; however, they require additional sensing time. We refer to such methods as *view-invariant*, in contrast to the *viewpoint-dependent* Image-to-3D models mentioned above. The choice of reconstruction method, therefore, depends on the input viewpoint, task requirements, and available budget.

We address this problem via model routing: selecting a reconstruction model for each input prior to inference. Model routing has been studied extensively for LLMs [7], [8], [9], [10], [11], but the 3D reconstruction setting differs in important ways: robotics applications call for cost vectors over multiple dimensions (latency, memory, etc.) rather than just API price, and the model pool includes both viewpoint-dependent methods (*e.g.*, Image-to-3D models) and view-invariant methods that are reconfigurable post-deployment (*e.g.*, by changing the number of views captured).

We propose **SCOUT** (**S**core-**C**onditioned **O**ptimal **U**tility **T**argeting), a routing framework that decouples reconstruction scores into the relative performance of viewpoint-dependent models, captured by a learned probability distribution, and the overall difficulty of the input image, captured by a scalar partition function. The learned network predicts only over the viewpoint-dependent models, so view-invariant pipelines can be added or reconfigured without retraining, and arbitrary cost vectors are supported at inference time. Because routing selects per input, the failure modes of any single model no longer dominate the pipeline, improving *robustness* for integrated robot systems.

We evaluate SCOUT on the Google Scanned Objects [12], BigBIRD [13], and YCB [14] datasets under multiple mesh quality metrics, including Density-aware Chamfer Distance (DCD), IoU, and geometric metrics from Eval3D [15]. SCOUT consistently outperforms routing baselines adapted from the LLM literature, and we validate its practical utility through robotic manipulation tasks, including collision-free grasp proposal evaluation, dexterous manipulation in simulation, and real-world pick-and-place on a Franka Panda robot.

The main contributions are: (1) the first model-routing formulation for 3D reconstruction, accommodating viewpoint-dependent and view-invariant methods under arbitrary cost vectors; (2) a partition function proxy that recovers absolute scores from the learned relative distribution with provably optimal weighting; and (3) a procedure that decouples image difficulty from relative model performance, enabling view-invariant methods to be added without retraining.

II. METHODS

A. Problem formulation

Let $\mathcal{M} = \{m_1, \cdots, m_k\}$ denote a set of k candidate 3D reconstruction models, each producing a mesh from an input image of the robot viewpoint, x. We seek a router that selects the optimal model for input x, determined by a score function $s(x, m)$ that measures reconstruction quality (higher is better), and an image-independent cost function $c(m)$ that captures user-specified model costs. Given a training dataset of input–score pairs $\left(x^{(i)}, \mathbf{s}^{(i)}\right)$, with $\mathbf{s}^{(i)} \in \mathbb{R}^k$, we learn a router h that takes an input x and cost vector $\mathbf{c} = [c(m_1), \cdots, c(m_k)] \in \mathbb{R}^k$, and outputs model selection

[1]Computer Science and Artificial Intelligence Laboratory, MIT {akash10, adityaag, lpk}@mit.edu

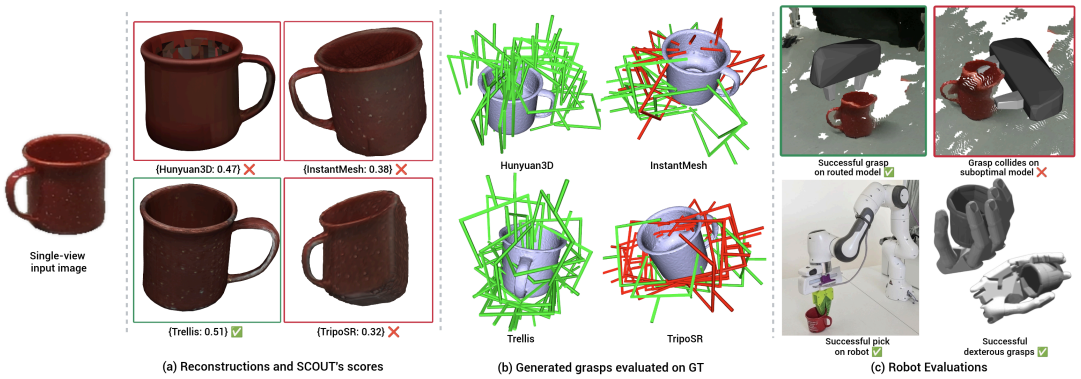

Fig. 1: Overview of SCOUT. Given an input image, (a) SCOUT routes to the best reconstruction model among candidates. (b) Grasp proposals on the reconstruction, evaluated against the ground-truth mesh (colliding grasps in red). (c) Utility of SCOUT's reconstruction-aware routing in downstream robust robot grasping and dexterous manipulation.

$\hat{m}$ maximizing $s(x,\hat{m})-c(\hat{m})$. We evaluate h by its expected regret relative to the oracle on a held-out set of n examples:

$$\mathcal{R}(h)=\frac{1}{n}\sum_{i=1}^{n}\left[\max_{m\in\mathcal{M}}\left(s(x^{(i)},m)-c(m)\right)-\left(s(x^{(i)},\hat{m}^{(i)})-c(\hat{m}^{(i)})\right)\right].$$

B. SCOUT

SCOUT is designed for two properties: (1) *scalability* - view-invariant methods can be added or reconfigured without retraining; and (2) *flexibility* - routing under arbitrary $\mathbf{c}\in\mathbb{R}^k$.

Preprocessing. We partition the score vector $\mathbf{s}^{(i)}=\left[\mathbf{s}^{(i)}_{\text{dep}},\mathbf{s}^{(i)}_{\text{inv}}\right]$ into k_1 viewpoint-dependent scores and the k_2 view-invariant scores. Motivated by score variance across object categories, we apply tag-based smoothing to $\mathbf{s}^{(i)}_{\text{dep}}$ following ZOOTER [7]. We group objects into semantic categories and blend each per-image score with its category mean, $\tilde{\mathbf{s}}^{(i)}_{\text{dep}}=\beta\mathbf{s}^{(i)}_{\text{dep}}+(1-\beta)\mathbf{s}_{\mathcal{T}(x^{(i)})}$, where $\mathbf{s}_{\mathcal{T}(x^{(i)})}$ averages $\mathbf{s}^{(i)}_{\text{dep}}$ over training examples sharing the tag of $x^{(i)}$.

Score decoupling. We train a neural network to predict $\hat{\mathbf{p}}\in\mathbb{R}^{k_1}$ from x under a KL divergence loss, with the target distribution given by the softmax of smoothed scores at temperature T: $p^{(i)}_j=\exp(\tilde{s}^{(i)}_{\text{dep},j}/T)/\sum_{m=1}^{k_1}\exp(\tilde{s}^{(i)}_{\text{dep},m}/T)$. We use KL because it isolates the relative performance of viewpoint-dependent models by normalizing out the image-difficulty offset shared by all of them, which MSE would absorb into its regression target. This shared-offset property holds only among the viewpoint-dependent models, motivating our decoupling from the view-invariant ones. View-invariant methods capture their own viewpoints through additional sensing, so their scores do not depend on the notion of hard (*e.g.*, top-down) versus easy (*e.g.*, corner) initial robot views.

Recovering the original scores from $\hat{\mathbf{p}}$ requires estimating the partition function $z^{(i)}=\sum_{m=1}^{k_1}\exp(s^{(i)}_{\text{dep},m}/T)$, so that $T\cdot\ln(\hat{p}^{(i)}_j\cdot z^{(i)})\approx s^{(i)}_{\text{dep},j}$. Because $z^{(i)}$ is a noisy per-example target, we instead train a second supervised network to predict $\hat{z}$ from x using a denoised regression target, $\tilde{z}^{(i)}$, obtained as an optimal weighted average of k_1 per-model proxies $\tilde{z}^{(i)}_j=\exp(s^{(i)}_{\text{dep},j}/T)/\hat{p}^{(i)}_j$:

$$\tilde{z}^{(i)}=\sum_{j=1}^{k_1}w_j\tilde{z}^{(i)}_j,\quad w_j\propto\left(\text{Var}[\hat{p}_j](1-R^2_j)\mathbb{E}_i\left[\frac{z^4}{\exp(2s_{\text{dep},j}/T)}\right]\right)^{-1}$$

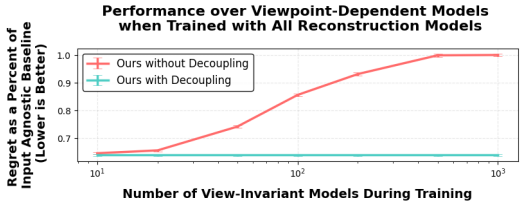

Fig. 2: Effect of the number of view-invariant methods on routing performance when evaluated on viewpoint-dependent methods. With decoupling, regret remains constant; without, regret grows with k_2.

where R^2_j is the coefficient of determination of $\hat{p}_j$ against p_j. (Appendix has derivation and t-tests against ground-truth z.)

Utility optimization. Given $\hat{\mathbf{p}}$ and $\hat{z}$, SCOUT selects $\hat{m}=\arg\max_{m_j\in\mathcal{M}}\hat{s}(x,m_j)-\mathbf{c}[j]$, where:

$$\hat{s}(x,m_j)=\begin{cases}T\cdot\ln\left(\hat{p}_j\cdot\hat{z}\right), & \text{if } j\le k_1,\\ s_{\text{inv}}(m_j), & \text{if } k_1<j\le k.\end{cases}$$

For view-invariant methods, $s_{\text{inv}}(m_j)$ may depend on object-level properties (*e.g.*, texture or reflectance) rather than the initial robot viewpoint. Because $\hat{\mathbf{p}}$ is predicted only over the k_1 viewpoint-dependent models, new view-invariant configurations can be added or reconfigured at inference time without retraining. Without decoupling, applying the softmax over the full score vector $\mathbf{s}^{(i)}$ couples the learned distribution to k_2 and degrades performance as the number of view-invariant methods increases, as demonstrated in Fig. 2.

III. EXPERIMENTS

A. Experimental setup

Mesh reconstruction models. We evaluate SCOUT with four Image-to-3D models selected for architectural diversity: Hunyuan3D [5], InstantMesh [3], TripoSR [2], and TRELLIS [4]. The model pool also includes four view-invariant options covering Gaussian (2DGS [16]), NeRF (Nerf2Mesh [17]), and SDF (NeuS2 [18]) representations, plus a *skip* option that bypasses reconstruction with a fixed default score for time-sensitive settings.

Datasets. We evaluate on the Google Scanned Objects (GSO) [12] with 15,450 renders from 1,030 objects at non-

degenerate viewpoints (avoiding face-on views), and 2,106 images from 54 YCB [14] objects across viewpoints (often deliberately face-on) and image styles (real BigBIRD [13] photos, flash-lighting renders, surround-lighting renders). Each reconstructed mesh is registered and scored against the ground-truth mesh using standard reconstruction quality metrics (negated where needed so that higher is better).

Baselines. To the best of our knowledge, no prior routing methods exist for 3D reconstruction. We therefore implemented baselines based on LLM routing approaches: ZOOTER [7], RouterDC [8], an MLP from RouterBench [9], matrix factorization (MF) from RouteLLM [10], and kNN and linear regression (LR) which have been shown to outperform more complex neural architectures in LLM routing [11]. All baselines and SCOUT share the same pretrained image features. Among these, ZOOTER is closest to SCOUT. The key difference is SCOUT's decoupling, which enables arbitrary cost vectors and post-hoc view-invariant reconfiguration at inference time. Results are given relative to an input-agnostic baseline that selects $\arg\max_{m_j} \bar{s}(m_j) - \mathbf{c}[j]$ using each model's training-set mean score $\bar{s}(m_j)$.

Evaluations. Since reconstruction quality is continuous, we report average utility or regret over cost subspaces rather than accuracy. Similar to splitting $\mathbf{s}^{(i)}$ (Section II-B), we split the cost vector as $\mathbf{c} = [\mathbf{c}_{\text{dep}}, \mathbf{c}_{\text{inv}}]$. We analyze two zero-cost subspaces, $\{\mathbf{c}{=}\mathbf{0}\}$ (all models) and $\{\mathbf{c}_{\text{dep}}{=}\mathbf{0} \cap \mathbf{c}_{\text{inv}}{=}\infty\}$ (viewpoint-dependent models only), enabling comparison with ZOOTER and RouterDC, which do not support varying costs at inference time. The full cost subspace $\mathcal{C}$ keeps all models in comparable utility ranges, ensuring non-trivial routing decisions. For each m_j, the cost range for c_j is $[P_{75,j} - \min_j(\text{IQR}_j),\ P_{75,j}]$, where $P_{75,j}$ and IQR_j are the 75th percentile and interquartile range of m_j's scores.

B. Full dataset results

Table I reports regret on novel objects on GSO across cost subspaces (using DCD), and on BigBIRD + YCB across seven quality metrics. Different metrics induce different optimal routing policies—IoU is volume-based and penalizes hollow geometry, whereas DCD measures surface correspondences—and different manipulation tasks demand different criteria, so a router must generalize across metrics. View-invariant scores are predicted via one of two strategies: regressing on image embeddings, or the VLM-tagged category mean from Sec. II-B. The better-performing strategy is chosen per metric based on validation performance.

Comparison to baselines. On GSO, SCOUT achieves statistically significant lower regret than all baselines in three of four cost subspaces, and is within the kNN's error bar in the fourth. On YCB, SCOUT achieves the lowest regret on all metrics except CD, where view-invariant methods produce meshes with interior artifacts and irregular surfaces that yield noisy CD scores affecting all learned routers. kNN and LR are SCOUT's closest baselines, but SCOUT outperforms both, achieving under half their regret in some experiments. SCOUT is also more efficient: end-to-end training and evaluation on GSO are $8.84\times$ faster than kNN (minutes vs. seconds) and $1.18\times$ faster than LR. This advantage increases as new Image-to-3D models emerge since SCOUT's runtime is independent of k_1, while kNN and LR scale linearly.

C. Robotics results

We evaluate SCOUT's downstream impact on robotic manipulation across three tasks following [6]. **Grasp proposal evaluation:** we generate two-fingered grasp proposals on each reconstructed mesh and report grasp collision rate and mesh-IoU against the ground truth across 10 YCB objects (Table II). SCOUT achieves the lowest mean collision rate and highest mean mesh-IoU, confirming that no single reconstruction model dominates across objects, and that per-input model selection translates to measurable gains in downstream grasp quality. **Dexterous manipulation:** since contact placement and force closure depend on local surface geometry, differences between reconstruction models are amplified relative to two-fingered grasping (Table III). **Real-world pick-and-place:** to test the full pipeline end-to-end, we reconstruct meshes from a single input image using each Image-to-3D method, generate antipodal grasp proposals, and execute 5 sampled grasps per object on a Franka Panda robot performing tabletop pick-and-place; success rates are in Table IV. The router's selections reflect input viewpoint difficulty: Hunyuan3D on challenging viewpoints (objects 1, 3, 4), where it produces the highest-fidelity reconstructions, and TRELLIS on easier viewpoints (objects 2, 5), which suffices and has lower latency. Real-world execution videos and reconstructed meshes are available on our website.

IV. CONCLUSION

We presented SCOUT, a routing framework that selects among 3D reconstruction models for a given input image under user-specified cost constraints, improving robustness to the diverse inputs a robot encounters. The key idea is to decouple reconstruction scores into relative model performance and image difficulty, which keeps the learned network independent of the view-invariant pool and supports arbitrary cost vectors without retraining. SCOUT consistently outperforms routing baselines adapted from the LLM literature across three datasets, multiple quality metrics, and diverse cost coefficient vector subspaces. Robotic manipulation experiments in simulation and the real world confirm routing improvements translate to downstream robotic performance.

Limitations & Future Work: We evaluate over eight models; scaling SCOUT to more models and classes of models (*e.g.*, point-cloud-based)—while pruning redundant models and accounting for switching overhead—is a natural next step. Closed-loop integration, where task feedback informs future routing decisions, is another promising direction.

REFERENCES

[1] S. S. Mohammadi, N. F. Duarte, D. Dimou, Y. Wang, M. Taiana, P. Morerio, A. Dehban, P. Moreno, A. Bernardino, A. Del Bue, *et al.*, "3dsgrasp: 3d shape-completion for robotic grasp," *arXiv preprint arXiv:2301.00866*, 2023.

[2] D. Tochilkin, D. Pankratz, Z. Liu, Z. Huang, A. Letts, Y. Li, D. Liang, C. Laforte, V. Jampani, and Y.-P. Cao, "Triposr: Fast 3d object reconstruction from a single image," 2024.

TABLE I: Regret (↓) as a proportion of the input-agnostic baseline regret on GSO (sweeping cost subspaces, measured by DCD) and YCB (sweeping quality metrics at $\mathbf{c} \in \mathcal{C}$). A detailed table with standard errors can be found in the appendix.

	GSO — cost subspaces (DCD)				**YCB — quality metrics ($\mathbf{c} \in \mathcal{C}$)**						
Method	$\mathcal{C}$	$\mathcal{C} \cap \mathbf{c}_{\text{inv}}{=}\infty$	$\mathbf{c}{=}\mathbf{0}$	$\mathbf{c}_{\text{dep}}{=}\mathbf{0} \cap \mathbf{c}_{\text{inv}}{=}\infty$	DCD	CD L2	CD L1	IoU	MMD-EMD	E3D-geo	E3D-struct
ZOOTER	N/A	N/A	N/A	0.6769	N/A	N/A	N/A	N/A	N/A	N/A	N/A
RouterDC	N/A	N/A	N/A	0.8013	N/A	N/A	N/A	N/A	N/A	N/A	N/A
MLP	0.7993	0.5819	0.6626	0.7319	1.0303	15.2150	1.7054	0.7681	1.0942	0.8845	0.8905
MF	0.6870	0.5667	0.7356	0.7584	1.2078	13.1332	2.2107	0.9076	1.9530	2.7425	1.3766
kNN	0.4917	0.4992	**0.6261**	0.6489	0.9282	1.7599	1.0074	0.7474	0.9788	1.1768	0.8522
LR	0.4909	0.5046	0.6361	0.6497	0.8919	7.8104	1.4418	0.7330	1.0073	0.7511	0.8258
Ours (no decoupling)	0.5249	0.5146	0.6580	0.6769	0.9258	10.8840	1.3402	0.6700	0.9678	0.6893	0.8576
Ours	**0.4734**	**0.4810**	0.6340	**0.6282**	**0.8417**	6.6110	1.3728	**0.6248**	**0.9146**	**0.3547**	**0.8243**
Input-agnostic	1.0000	1.0000	1.0000	1.0000	1.0000	**1.0000**	**1.0000**	1.0000	1.0000	1.0000	1.0000

TABLE II: Grasp collision rate (↓) / mesh-IoU (↑) across 10 YCB objects, using a two-fingered gripper. SCOUT is evaluated under two settings: routing over novel views of known objects, and over novel views and novel objects.

Object	**Hunyuan3D**	**InstantMesh**	**TRELLIS**	**TripoSR**	**Ours (novel obj.)**	**Ours (novel view)**
Tomato Soup Can	0.7174 / 0.0685	0.0227 / 0.9164	0.3864 / 0.2071	0.6585 / 0.1807	InstantMesh	InstantMesh
Cup (vC)	0.0 / 0.4019	0.2273 / 0.1381	0.0 / 0.4372	0.1500 / 0.1487	Hunyuan3D	Hunyuan3D
Marble	0.0 / 0.9580	0.0 / 0.7423	0.0 / 0.1464	0.0 / 0.9537	TripoSR	TripoSR
Chef Can	0.0 / 0.8543	0.0 / 0.8763	0.0250 / 0.9406	0.1190 / 0.7730	InstantMesh	TRELLIS
Potted Meat	0.0 / 0.6589	0.6098 / 0.4604	0.0682 / 0.7573	0.3571 / 0.4632	InstantMesh	TRELLIS
Cup (vB)	0.0 / 0.2960	0.0238 / 0.1423	0.1087 / 0.0338	0.3659 / 0.1080	Hunyuan3D	Hunyuan3D
Tennis Ball	0.0 / 0.9717	0.0 / 0.9663	0.0 / 0.1525	0.0 / 0.9645	InstantMesh	InstantMesh
Brick	0.0652 / 0.2421	0.0652 / 0.2547	0.6136 / 0.0949	0.0 / 0.8334	TripoSR	TripoSR
Power Drill	0.0714 / 0.6230	0.0500 / 0.5662	0.2889 / 0.1399	0.3636 / 0.4793	InstantMesh	InstantMesh
Spatula	0.0851 / 0.2218	0.0465 / 0.0076	0.0851 / 0.1014	0.0 / 0.0392	InstantMesh	Hunyuan3D
Mean	0.0939 / 0.5296	0.1045 / 0.5071	0.1576 / 0.3011	0.2014 / 0.4944	0.0729 / 0.6278	0.0251 / 0.6854

TABLE III: Dexterous manipulation success rate (↑) in simulation for each reconstruction model across 5 objects (HY3D: Hunyuan3D, IM: InstantMesh, TRL: TRELLIS, TSR: TripoSR). Ours reports the model selected by SCOUT.

Method	**Can**	**Spatula**	**Drill**	**Brick**	**Cup**	**Mean**
HY3D	13.6	15.2	38.1	61.8	49.4	35.6
IM	29.9	0	41.7	52.9	44.5	33.8
TRL	61.8	0	4.5	50.4	45.8	32.5
TSR	21.9	0	17.6	39.7	45.0	24.8
Ours	TRL	HY3D	IM	TSR	HY3D	41.6

TABLE IV: Real-world pick-and-place success rate (↑) on a Franka Panda robot across 5 objects. For each reconstructed object, 5 antipodal grasps are sampled and executed.

Method	**Drill**	**Ball**	**Can**	**Cup**	**Mug**	**Mean**
HY3D	3/5	4/5	5/5	5/5	5/5	4.4/5
IM	1/5	5/5	1/5	2/5	4/5	2.6/5
TRL	0/5	5/5	3/5	0/5	5/5	2.6/5
TSR	1/5	0/5	2/5	0/5	0/5	0.6/5
Ours	HY3D	TRL	HY3D	HY3D	TRL	4.6/5

[3] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan, "Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models," 2024.

[4] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang, "Structured 3d latents for scalable and versatile 3d generation," *arXiv preprint arXiv:2412.01506*, 2024.

[5] T. H. Team, "Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation," 2025.

[6] A. Agarwal, G. Singh, B. Sen, T. Lozano-Pérez, and L. P. Kaelbling, "Scenecomplete: Open-world 3d scene completion in cluttered real world environments for robot manipulation," *IEEE Robotics and Automation Letters*, vol. 11, no. 1, pp. 482–489, 2026.

[7] K. Lu, H. Yuan, R. Lin, J. Lin, Z. Yuan, C. Zhou, and J. Zhou, "Routing to the expert: Efficient reward-guided ensemble of large language models," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1964–1974, 2024.

[8] S. Chen, W. Jiang, B. Lin, J. Kwok, and Y. Zhang, "Routerdc: Query-based router by dual contrastive learning for assembling large language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 66305–66328, 2024.

[9] Q. J. Hu, J. Bieker, X. Li, N. Jiang, B. Keigwin, G. Ranganath, K. Keutzer, and S. K. Upadhyay, "Routerbench: A benchmark for multi-llm routing system," *arXiv preprint arXiv:2403.12031*, 2024.

[10] I. Ong, A. Almahairi, V. Wu, W.-L. Chiang, T. Wu, J. E. Gonzalez, M. W. Kadous, and I. Stoica, "Routellm: Learning to route llms with preference data," *arXiv preprint arXiv:2406.18665*, 2024.

[11] Y. Li, "Rethinking predictive modeling for llm routing: When simple knn beats complex learned routers," *arXiv preprint arXiv:2505.12601*, 2025.

[12] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560, Ieee, 2022.

[13] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," in *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 509–516, IEEE, 2014.

[14] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*, pp. 510–517, IEEE, 2015.

[15] S. Duggal, Y. Hu, O. Michel, A. Kembhavi, W. T. Freeman, N. A. Smith, R. Krishna, A. Torralba, A. Farhadi, and W.-C. Ma, "Eval3d: Interpretable and fine-grained evaluation for 3d generation," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13326–13336, 2025.

[16] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2d gaussian splatting for geometrically accurate radiance fields," in *ACM SIGGRAPH 2024 conference papers*, pp. 1–11, 2024.

[17] J. Tang, H. Zhou, X. Chen, T. Hu, E. Ding, J. Wang, and G. Zeng, "Delicate textured mesh recovery from nerf via adaptive surface refinement," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17739–17749, 2023.

[18] Y. Wang, Q. Han, M. Habermann, K. Daniilidis, C. Theobalt, and L. Liu, "Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3295–3306, 2023.