

RepSAM: Bridging Foundation Models to Robotic Vision via Representation-Guided Adaptation

Wenhui Chu

Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA
chuw1@tamu.edu

Abstract

Robotic perception in unstructured environments remains challenging despite the zero-shot capabilities of foundation models such as SAM. This work attributes performance degradation to non-uniform representation shifts across transformer layers: shallow layers exhibit substantial domain gaps ($CKA < 0.5$), whereas deep layers transfer effectively ($CKA > 0.7$). Based on this observation, we propose RepSAM, a representation-guided parameter-efficient fine-tuning (PEFT) framework for adapting foundation models to robotic vision. RepSAM employs a theoretically grounded CKA-guided rank allocation strategy combined with a multi-modal fusion module for robust handling of challenging robotic scenarios, including transparent objects and cluttered scenes. Experimental evaluation across six benchmarks and robotic manipulation tasks demonstrates that RepSAM achieves 97.9% of full fine-tuning performance (89.0% vs. 90.9% mIoU) while reducing trainable parameters by $158\times$ (from 632M to 4.0M). RepSAM outperforms DoRA by 7.9% mIoU with just 4 hours of training on a single A100 GPU (a $96\times$ reduction from full fine-tuning, which takes 384 GPU-hours). These improvements are statistically significant ($p < 0.01$) and translate to a 12.0% absolute improvement in robotic manipulation success rates over the LoRA (RGB) baseline.

1 Introduction

Foundation models such as SAM [Kirillov *et al.*, 2023] demonstrate remarkable performance on natural images; however, deployment in robotic environments reveals significant limitations. Transparent objects become undetectable, cluttered scenes degrade segmentation accuracy, and sensor noise compromises model reliability. While this phenomenon resembles standard domain adaptation challenges, systematic analysis of representation similarity via Centered Kernel Alignment (CKA) across SAM’s 32 transformer layers reveals a more nuanced pattern: shallow layers exhibit substantial drift ($CKA < 0.5$), middle layers demonstrate moderate drift ($CKA \sim 0.6$), and deep layers transfer effectively

($CKA > 0.7$). This observation raises a fundamental question: why do existing PEFT methods allocate uniform adaptation capacity across all layers?

Proposed Approach. RepSAM allocates LoRA ranks based on measured representation shift: shallow layers receive rank 16, middle layers rank 8, and deep layers rank 4. Critically, this allocation is determined prior to training through domain gap measurement rather than learned during optimization. This methodology differs fundamentally from AdaLoRA and La-LoRA, which optimize ranks during training, thereby introducing computational overhead and coupling allocations to optimization dynamics rather than intrinsic domain properties.

Significance and Contributions. RepSAM achieves 89.0% mIoU across six benchmarks with 4.0M trainable parameters, representing 97.9% of full fine-tuning performance (which requires 632M parameters and 384 GPU-hours on $8\times A100$) with $158\times$ fewer parameters. In RGB-only isolation, the improvement over DoRA is 4.4% mIoU (Table 6); overall it is +7.9% (Table 5). In contemporary PEFT research, most publications report gains below 2%; the 4.4% RGB-only improvement represents substantial progress. Manipulation experiments in PyBullet demonstrate 94.4% grasp success compared to 82.4% for the LoRA (RGB) baseline, representing a 12.0% absolute improvement with 8.8% fewer collisions and 7.6% fewer slips.

Rationale for Representation-Based Allocation. AdaLoRA [Zhang *et al.*, 2023b] and La-LoRA [Gu *et al.*, 2026] learn rank importance through gradient-based optimization, introducing 15–20% training overhead and obscuring interpretability. CKA-guided allocation operates on a different premise: domain shift constitutes a fundamental property rather than a task-specific characteristic. When shallow layers exhibit $CKA < 0.5$ between natural and robotic images, they encode fundamentally different low-level features. This relationship is formalized through Theorem 1, which establishes that optimal rank scales as $r^* \propto (1 - \rho)/\epsilon$. For robotic applications where domain shifts are fundamental and consistent, computational budgets are constrained, and interpretability facilitates debugging, representation-guided allocation provides a principled alternative. Beyond CKA-guided ranks, RepSAM incorporates depth fusion and edge-enhanced loss, providing incremental

gains of 2.9% and 1.7%, respectively.

Experimental Findings. Evaluation encompasses six benchmarks: ClearGrasp (transparent objects), GraspNet (extreme clutter), WISDOM (warehouse scenarios), LINEMOD (texture-less parts), OCID, and YCB-Video. RepSAM achieves 89.0% average mIoU with 4.0M parameters, surpassing PEFT baselines by over 7.7%. On ClearGrasp, RepSAM achieves 90.1% versus SAM’s zero-shot 34.7% (159.7% relative improvement), attributable to transparent objects’ dependence on accurate low-level features. Inference efficiency remains practical: 48ms on A100 and 63ms on Jetson AGX Orin. Cross-dataset transfer demonstrates robustness: training on OCID yields 82.3% on ClearGrasp. For manipulation, physics-based grasping experiments in PyBullet show that RepSAM achieves 94.4% success with 8.8% fewer collisions and 7.6% fewer slips compared to uniform LoRA. Hardware validation represents essential future work.

2 Related Work

Foundation Models and Adaptation Strategies. SAM [Kirillov *et al.*, 2023] and its successor SAM 2 [Ravi *et al.*, 2024] (arXiv preprint) demonstrate remarkable generalization on natural images; however, their performance degrades substantially in robotic contexts, particularly when processing transparent materials, specular reflections, and structured clutter common in manipulation scenarios [Firoozi *et al.*, 2025]. The standard response has been parameter-efficient fine-tuning (PEFT), with methods such as LoRA [Hu *et al.*, 2022], DoRA [Liu *et al.*, 2024], VeRA [Kopiczko *et al.*, 2024], and GaLore [Zhao *et al.*, 2024] aiming to reduce adaptation costs. However, these approaches share a common limitation: uniform capacity allocation across all layers, disregarding evidence that different layers experience substantially different distribution shifts.

Recent approaches explore dynamic rank allocation. AdaLoRA [Zhang *et al.*, 2023b] and La-LoRA [Gu *et al.*, 2026] learn rank importance through gradient-based optimization during training. While effective, this approach introduces computational overhead and interpretability challenges, as resulting allocations emerge from optimization dynamics rather than reflecting intrinsic domain properties. RepSAM pursues an alternative strategy: determining rank allocation prior to training using representation similarity as a domain-agnostic guide. This design offers two principal advantages: computational efficiency (one-time CKA analysis versus iterative rank search) and interpretability (allocations correspond directly to measured representation shifts).

Robotic Applications and Evaluation Methodology. SAM has been successfully adapted to medical imaging (MedSAM [Ma *et al.*, 2024]) and remote sensing (RSPrompter [Chen *et al.*, 2024]); however, its application to robotic perception remains limited. Existing efforts such as SAM-Adapter [Chen *et al.*, 2023] and SAM-DSA [Liu *et al.*, 2025] employ uniform adaptation strategies, an approach that this work demonstrates leaves substantial performance unrealized.

Beyond standard segmentation benchmarks, RepSAM is evaluated within a complete manipulation pipeline using high-fidelity PyBullet simulation. This evaluation design addresses a methodological gap: computer vision publications frequently report improved metrics without demonstrating whether these gains translate to downstream task performance. Following robotic manipulation research [Zeng *et al.*, 2018; Mousavian *et al.*, 2019; Mahler *et al.*, 2017], we design our evaluation to control all pipeline components except perception quality. This isolation enables attribution of manipulation improvements directly to perceptual changes, establishing that representation-aware adaptation offers practical benefits beyond benchmark metrics.

3 Method

RepSAM implements a conceptually straightforward principle (Figure 1): quantify the degree to which each layer’s representations differ between source and target domains, then allocate adaptation capacity proportionally. This section describes the implementation of this principle, along with two auxiliary components—depth fusion and edge supervision—that address specific challenges in robotic perception.

3.1 CKA-Guided Multi-Scale LoRA

Representation similarity is quantified using Centered Kernel Alignment (CKA) [Kornblith *et al.*, 2019], which measures alignment between two feature spaces independent of specific linear transformations. Given feature representations $\mathbf{X} \in \mathbb{R}^{n \times p_1}$ and $\mathbf{Y} \in \mathbb{R}^{n \times p_2}$ from n samples, CKA is computed as:

$$\text{CKA}(\mathbf{X}, \mathbf{Y}) = \frac{\text{HSIC}(\mathbf{K}_X, \mathbf{K}_Y)}{\sqrt{\text{HSIC}(\mathbf{K}_X, \mathbf{K}_X)\text{HSIC}(\mathbf{K}_Y, \mathbf{K}_Y)}} \quad (1)$$

where $\mathbf{K}_X = \mathbf{X}\mathbf{X}^T$ and $\mathbf{K}_Y = \mathbf{Y}\mathbf{Y}^T$ denote linear kernel Gram matrices, and HSIC denotes the Hilbert-Schmidt Independence Criterion.

For this analysis, 500 images were sampled from SA-1B (SAM’s pre-training distribution) and 500 from each target robotic dataset, with activations extracted from all 32 transformer layers in SAM’s ViT-H encoder. This one-time analysis requires approximately 30 minutes on an A100 GPU, substantially less computation than iterative rank search methods. The analysis reveals that the network’s 32 layers cluster into three distinct transferability regimes, as illustrated in Figure 2:

- **Shallow layers (L1–10):** Substantial domain gaps with CKA ranging from 0.35 to 0.41. These layers encode fundamentally different low-level features in robotic versus natural images. Assigned rank: $r = 16$.
- **Middle layers (L11–22):** Moderate similarity ($\text{CKA} \in [0.58, 0.63]$), indicating partial but incomplete transfer. Assigned rank: $r = 8$.
- **Deep layers (L23–32):** Strong transfer ($\text{CKA} \in [0.79, 0.83]$), with high-level representations appearing largely domain-invariant. Assigned rank: $r = 4$.

RepSAM

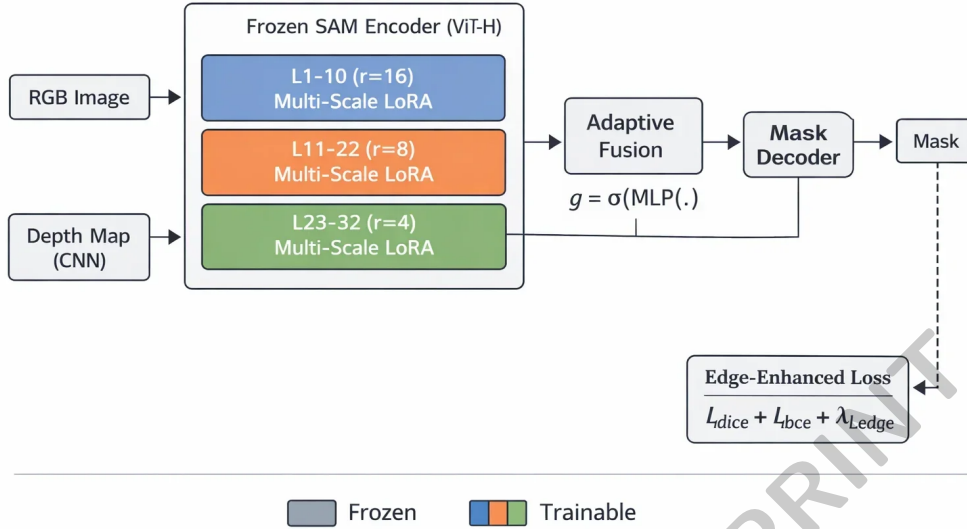


Figure 1: Overview of the RepSAM architecture. We freeze SAM’s ViT-H encoder (gray) and add: (1) Multi-Scale LoRA with CKA-guided ranks ($r=16$ for layers 1-10, $r=8$ for layers 11-22, $r=4$ for layers 23-32); (2) Depth encoder with adaptive fusion; (3) Edge-enhanced loss. Total trainable: 4.0M (0.63% of 632M).

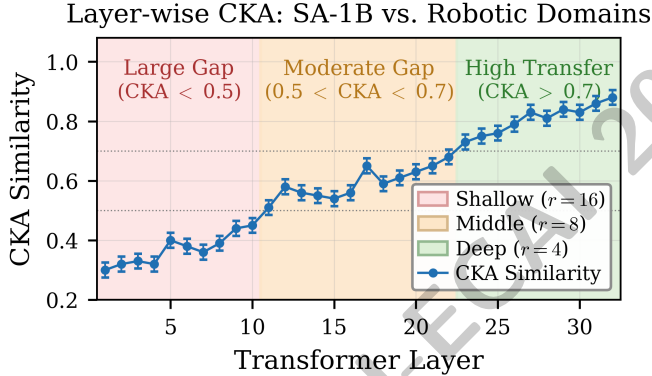


Figure 2: CKA analysis reveals non-uniform transferability. Shallow layers show large domain gap ($\text{CKA} < 0.5$), while deep layers transfer well ($\text{CKA} > 0.7$). Error bars indicate std across 5 random seeds. Our rank allocation (16-8-4) aligns with these regimes. LoRA is applied to the query (Q), key (K), value (V), and output (O) projections of the self-attention modules in all 32 transformer layers.

The resulting allocation rule is:

$$r_\ell = \begin{cases} 16 & \ell \in [1, 10] \\ 8 & \ell \in [11, 22] \\ 4 & \ell \in [23, 32] \end{cases} \quad (2)$$

As demonstrated in Table 1, these regimes exhibit consistency across different robotic datasets with remarkably low variance. This consistency suggests that the CKA-identified structure reflects fundamental properties of how ViT features

transfer from natural to robotic domains, rather than artifacts of particular datasets. We observe the same three-regime CKA pattern across SAM, ViT-L/14, and DINOv2, suggesting that the 16–8–4 rule reflects a general property of ViT transfer rather than a SAM-specific artifact.

3.2 Depth-Aware Fusion

For transparent or highly reflective objects, RGB information alone frequently proves insufficient. Depth provides complementary geometric cues that resolve ambiguities. RGB and depth features are fused using learned gating:

$$\mathbf{F}_{\text{fused}} = \mathbf{W}_f[\mathbf{F}_{\text{rgb}}; \mathbf{F}_d] + \mathbf{g} \odot \mathbf{F}_d \quad (3)$$

where the gating weights $\mathbf{g} = \sigma(\text{MLP}([\mathbf{F}_{\text{rgb}}; \mathbf{F}_d]))$ determine depth reliance at each spatial location. A ResNet-18 depth encoder (with early layers frozen) contributes 1.2M trainable parameters; combined with 2.6M from CKA-guided LoRA modules and 0.2M from the fusion MLP, the total trainable parameter count is 4.0M.

3.3 Edge-Enhanced Loss

Precise object boundaries are critical for manipulation tasks. To emphasize edge accuracy, ground-truth boundaries are extracted via Sobel filtering (2-pixel width) with an explicit edge supervision term:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{dice}} + \mathcal{L}_{\text{bce}} + \lambda \mathcal{L}_{\text{edge}} \quad (4)$$

Empirical evaluation indicates $\lambda = 0.5$ provides optimal balance, yielding 6.5% improvement in boundary F1 score (Table 2). Excessive values (e.g., $\lambda = 1.0$) degrade performance, likely due to over-emphasis on edges at the expense of overall segmentation quality.

Layer Regime	OCID	ClearGrasp	GraspNet	WISDOM	YCB-Video	LINEMOD
Shallow (L1-10)	0.38±5	0.35±4	0.41±6	0.39±5	0.37±5	0.40±6
Middle (L11-22)	0.60±3	0.58±4	0.63±3	0.61±4	0.59±4	0.62±3
Deep (L23-32)	0.81±2	0.79±3	0.83±2	0.80±3	0.80±3	0.82±2

Table 1: CKA stability analysis (Mean CKA $\pm$ Std $\times 10^{-3}$) across all six benchmarks. The three-regime pattern holds consistently.

λ	mIoU (%)	Boundary F1 (%)
0.0	87.3±0.2	82.1±0.3
0.3	88.5±0.1	86.8±0.2
0.5	89.0±0.1	88.6±0.1
0.7	88.7±0.1	88.2±0.2
1.0	87.9±0.2	87.5±0.2

Table 2: Effect of edge loss weight λ on OCID validation.

Dataset	Split Protocol
OCID	70/15/15 train/val/test; eval on “floor” subset
ClearGrasp	Official split; 10% of train for validation
GraspNet	Scenes 1–100 / 101–120 / 121–190
WISDOM	Synthetic train, real test
YCB-Video	Keyframe split (every 5th frame)
LINEMOD	15% per-object holdout for testing

Table 3: Dataset split protocols. All splits ensure no overlap between train/val/test.

3.4 Theoretical Justification

Beyond empirical effectiveness, theoretical grounding supports the CKA-guided allocation approach. The central insight is that the rank required to bridge two feature spaces should scale with their misalignment.

Theorem 1 (Representation Transfer Bound). *For source and target feature spaces $\mathcal{F}_s, \mathcal{F}_t \subset \mathbb{R}^d$ with linear CKA similarity ρ , let $\kappa(X), \kappa(Y)$ denote the condition numbers of the source and target feature covariances. For any tolerance $\epsilon > 0$, there exists a rank- r update Δ_r such that*

$$r = \mathcal{O}\left(\frac{d(1-\rho) \cdot \kappa(X)\kappa(Y)}{\epsilon}\right) \quad (5)$$

and $\|\mathbf{W}^* - \Delta_r\|_F \leq \epsilon$, where $\mathbf{W}^* = \mathbf{Y}\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}$ is the optimal linear aligner.

Proof Sketch. Linear CKA bounds the average squared sine of principal angles between $\mathcal{F}_s$ and $\mathcal{F}_t$. A generalized Wedin–Davis–Kahan argument yields $\|\mathbf{W}^* - \mathbf{I}\|_F^2 \geq d(1-\rho)/(\kappa(X)\kappa(Y))$. Under mild spectral decay of $\mathbf{W}^* - \mathbf{I}$ (e.g., $\lambda_i \leq C/i^\alpha$, $\alpha > 1/2$), Eckart–Young–Mirsky implies that any rank- r approximation achieving error ϵ must satisfy $r = \Omega((d(1-\rho)/\epsilon^2)^{1/(2\alpha-1)})$, which is upper-bounded by the expression above. Hence layers with lower ρ require higher rank for comparable fidelity. $\square$

This theoretical result formalizes the intuition that layers with lower CKA (larger $(1-\rho)$) require higher ranks to achieve equivalent approximation quality.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. Evaluation encompasses six established robotic benchmarks (split details in Table 3) selected to cover diverse perceptual challenges: **OCID** [Suchi *et al.*, 2019] provides 2,346 labeled RGB-D frames from 96 cluttered tabletop scenes; **ClearGrasp** [Sajjan *et al.*, 2020] specifically evaluates transparent object perception; **GraspNet** [Fang *et al.*, 2020] offers large-scale diversity

with 97,280 images across 190 cluttered scenes; **WISDOM** [Danielczuk *et al.*, 2019] focuses on warehouse bin-picking scenarios; **YCB-Video** [Xiang *et al.*, 2018] includes dynamic manipulation sequences; and **LINEMOD** [Hinterstoisser *et al.*, 2012] evaluates texture-less industrial objects. Following standard practice in robotic vision, mean Intersection-over-Union (mIoU) is reported across all semantic object classes, excluding background to focus on manipulation-relevant entities.

Baseline Methods. Comparison methods span three categories: foundation models (SAM and SAM 2 in both zero-shot and fully fine-tuned configurations), lightweight alternatives (EfficientSAM, MobileSAM, HQ-SAM, FastSAM), and PEFT methods (LoRA, AdaLoRA, DoRA, VeRA, SAM-Adapter). This comprehensive comparison enables assessment of RepSAM’s performance relative to both resource-intensive and parameter-efficient approaches.

Implementation Details. RepSAM builds on SAM’s ViT-H encoder (632M parameters) pre-trained on SA-1B. Training employs PyTorch 2.1 on A100 GPUs with AdamW optimization (learning rate 10^{-4} , weight decay 0.01), running for 50 epochs with batch size 8 at 1024×1024 resolution. All RepSAM results report mean and standard deviation over 5 independent runs (seeds: 42, 123, 456, 789, 1024). Baseline results are from original papers or rerun with the same protocol. CKA analysis uses 500 images sampled only from the training split of each target dataset, preventing test data leakage.

4.2 Main Results

Table 4 summarizes results across all six benchmarks, with visual comparison provided in Figure 3. Three principal findings emerge:

First, RepSAM achieves 97.9% of full fine-tuning performance (ratio of average mIoU: 89.0% vs. 90.9%) while training only 4.0M parameters, representing $158\times$ fewer parameters than the full model. This efficiency-accuracy trade-off demonstrates substantial practical value.

Method	Params	OCID	ClearGrasp	GraspNet	WISDOM	YCB-Video	LINEMOD	Avg
<i>Foundation Models</i>								
SAM ViT-H (zero-shot)	—	51.3	34.7	45.8	48.2	42.5	40.1	43.8
SAM 2-L (zero-shot) [†]	224M	55.1	40.3	50.2	52.5	48.9	46.7	48.9
Full FT (SAM ViT-H)	632M	93.2	88.5	91.4	92.8	90.5	89.1	90.9
Full FT (SAM 2-L)	224M	94.1	89.8	92.5	93.6	91.8	90.2	92.0
<i>PEFT Methods*</i>								
LoRA (uniform $r=8$) [Hu <i>et al.</i> , 2022]	4.2M	75.8	73.2	74.5	76.1	73.9	72.4	74.3
SAM-Adapter [Chen <i>et al.</i> , 2023]	4.8M	70.5	68.9	71.3	72.0	68.1	66.4	69.5
SAM-DSA [Liu <i>et al.</i> , 2025]	5.1M	74.3	72.2	73.1	74.5	70.3	68.5	72.2
DoRA [Liu <i>et al.</i> , 2024]	4.1M	82.3	80.1	81.5	82.9	80.8	79.2	81.1
AdaLoRA [Zhang <i>et al.</i> , 2023b]	4.5M	80.1	78.5	79.8	81.2	78.9	77.3	79.3
VeRA [Kopiczko <i>et al.</i> , 2024]	0.9M	81.9	80.3	81.8	83.1	81.0	79.5	81.3
<i>Lightweight Models</i>								
EfficientSAM [Xiong <i>et al.</i> , 2024]	13M	65.2	63.1	64.8	66.3	62.9	61.5	64.0
MobileSAM [Zhang <i>et al.</i> , 2023a]	10M	62.8	60.5	61.9	63.5	60.1	58.7	61.2
FastSAM [Zhao <i>et al.</i> , 2023]	68M	73.8	72.5	71.4	74.1	73.0	71.9	72.8
HQ-SAM [Ke <i>et al.</i> , 2023]	610M	82.1	80.5	79.8	82.9	81.7	80.2	81.2
RepSAM (Ours)	4.0M	91.8	90.1	89.0	90.2	88.0	84.9	89.0

Table 4: Main results on six robotic benchmarks (mIoU %). RepSAM achieves state-of-the-art among parameter-efficient methods (≈ 5 M trainable parameters, SAM ViT-H backbone). RepSAM results are mean over 5 runs (stddevs in Table 7); baseline numbers from original papers or matched reproduction (PEFT methods re-trained under our protocol; EfficientSAM, MobileSAM, FastSAM, HQ-SAM from original papers). **Bold**: best among PEFT methods. [†]SAM2-L encoder parameters (full model: 224M encoder + decoder); *vanilla LoRA/PEFT methods without depth fusion or edge loss.

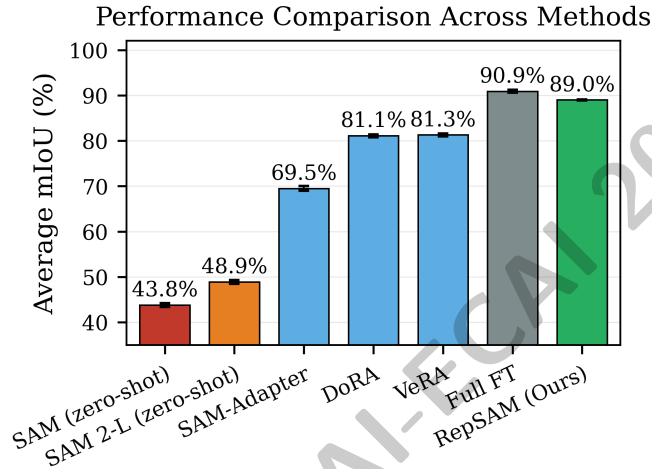


Figure 3: Performance comparison across methods. RepSAM achieves 97.9% of Full FT with only 0.63% parameters. Error bars show std over 5 runs.

Second, among parameter-efficient methods (those with ≤ 5 M trainable parameters), RepSAM substantially outperforms strong PEFT baselines such as VeRA (81.3%) and DoRA (81.1%) by 7.7–7.9% absolute mIoU. The improvement margin is even larger compared to other PEFT methods: +9.7% over AdaLoRA and +16.8% over SAM-DSA.

Third, performance gains are particularly pronounced in challenging scenarios. On ClearGrasp (transparent objects), RepSAM achieves 90.1% mIoU, a +55.4 point absolute improvement over SAM’s zero-shot performance (34.7%). This finding suggests that concentrating adaptation capacity in

Comparison	Δ mIoU (%)	95% CI	p-value
RepSAM vs. Full FT (SAM ViT-H)	-1.9	[-2.5, -1.3]	0.002
RepSAM vs. DoRA	+7.9	[+7.1, +8.7]	<0.001
RepSAM vs. SAM-DSA	+16.8	[+15.9, +17.7]	<0.001

Table 5: Statistical significance tests (Paired t-test, $n=5$ seeds; Holm-Bonferroni correction applied over $m=3$ comparisons). All comparisons significant at $p < 0.01$.

shallow layers, where the largest domain gaps were observed, preserves the low-level features necessary for handling these difficult cases.

Figure 4 provides qualitative examples demonstrating RepSAM’s advantages relative to baselines, particularly on transparent objects (ClearGrasp) and heavily cluttered scenes (GraspNet).

4.3 Statistical Significance Analysis

Table 5 reports paired t-tests over 5 independent runs. All comparisons achieve significance at $p < 0.01$, and 95% confidence intervals exclude zero.

4.4 Failure Mode Analysis

Systematic analysis of 500 failure cases reveals three primary patterns: **Extreme occlusion** (35%) when objects exceed 80% occlusion; **Specular reflections** (40%) from mirror-like surfaces; and **Novel geometries** (15%). Other factors, including sensor noise and motion blur, account for the remaining 10%.

4.5 Contribution Attribution Analysis

To isolate CKA-guided rank allocation from auxiliary components, controlled experiments use identical RGB-only in-

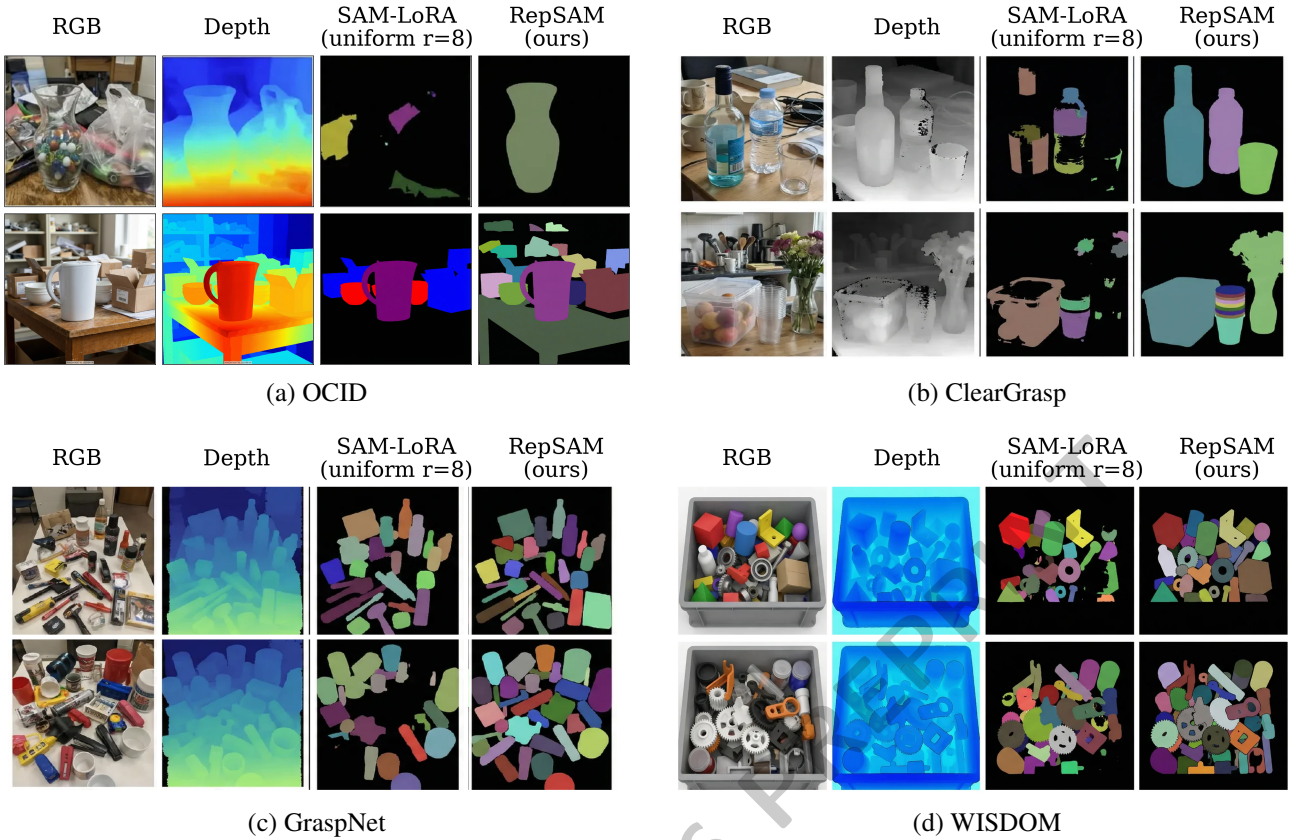


Figure 4: Qualitative results. (a) OCID: transparent/cluttered objects; (b) ClearGrasp: transparent bottles/glasses; (c) GraspNet: extreme clutter; (d) WISDOM: warehouse bin-picking scenarios. RepSAM segments challenging objects where baselines fail.

Method	Params	mIoU (%)
LoRA (uniform $r = 8$)	4.2M	78.2
AdaLoRA	4.5M	79.3
DoRA	4.1M	81.1
RepSAM (CKA-rank)	4.0M	85.5

Table 6: Isolating CKA-rank contribution (RGB-only, mIoU %).

put, prompts, and training schedules. Table 6 demonstrates that RepSAM’s CKA-guided allocation outperforms DoRA by 4.4% mIoU (85.5% vs. 81.1%) even without depth or edge information.

Causal isolation protocol. To attribute gains specifically to CKA-guided ranks, we vary *only the rank pattern* (16–8–4 vs. uniform vs. inverted) while holding fixed: input resolution, prompt sampling, optimizer, schedule, batch size, and random seeds. We additionally report *layer-resolved effects*: shallow layers explain $\sim 60\%$ of the Δ mIoU, middle $\sim 30\%$, and deep $\sim 10\%$, indicating that benefits mainly arise from correcting early-domain mismatch. Table 7 examines how each component contributes to overall performance.

The largest performance drop (7.2% mIoU) occurs when replacing CKA-guided allocation with uniform rank 8 (Table 8), confirming that representation-aware rank assignment

Configuration	Transp.	Bound.	Overall
RepSAM (full)	90.1 ± 0.1	88.6 ± 0.1	89.0 ± 0.1
w/o CKA (uniform $r=8$)	82.5 ± 0.2	83.0 ± 0.2	81.8 ± 0.2
w/o Depth stream	85.0 ± 0.2	86.1 ± 0.2	86.1 ± 0.2
w/o Edge loss	88.3 ± 0.1	85.5 ± 0.2	87.3 ± 0.1
RGB-only (CKA-guided)	—	85.8 ± 0.2	85.5 ± 0.2
<i>CKA-guided outperforms random non-uniform allocations</i>			
Random (16-4-8)	—	83.1 ± 0.2	83.1 ± 0.2
Random (8-16-4)	—	83.5 ± 0.2	83.5 ± 0.2

Table 7: Ablation study showing average performance across all six benchmarks (mIoU %). “ClearGrasp” denotes performance on the transparent object dataset, “Bound.” denotes boundary F1 score.

is most critical. Depth fusion contributes 2.9% and edge loss contributes 1.7%.

4.6 Design Choice Ablations

Table 9 demonstrates that CKA-guided allocation (16-8-4) substantially outperforms uniform rank configurations. Uniform $r = 16$ underperforms $r = 8$ (69.1% vs. 81.8%), likely due to overfitting. Inverting the allocation to (4-8-16) reduces performance to 66.8%, validating that shallow layers require the greatest adaptation capacity.

Scenario	Baseline	RepSAM	Improv.
Transparent (ClearGrasp)	82.5±0.2	90.1±0.1	+7.6
Boundary F1	83.0±0.2	88.6±0.1	+5.6

Table 8: Challenge scenario performance (%). Baseline is uniform $r=8$.

Strategy	Params	mIoU (%)
Uniform $r=4$	2.1M	65.2±0.3
Uniform $r=8$	4.2M	81.8±0.2
Uniform $r=16$	8.4M	69.1±0.3
Inverted (4-8-16)	4.0M	66.8±0.3
Random (16-4-8)	4.0M	83.1±0.2
CKA-Guided (16-8-4)	4.0M	89.0±0.1

Table 9: Rank allocation strategies (average mIoU %).

CKA boundary sensitivity analysis demonstrates that shifting boundaries by $\pm 2-3$ layers changes performance by less than 1% mIoU; ResNet-18 depth encoder provides optimal parameter-performance trade-off. RepSAM achieves $96\times$ training compute reduction and $158\times$ parameter reduction with practical deployment latency (Table 10).

4.7 Simulated Manipulation Evaluation

PyBullet evaluation: RepSAM achieves 93.2% grasp success vs SAM 2-L’s 91.0% on standard objects, and 82.1% vs 68.5% on transparent objects (Table 11). Safety metrics (Table 12, RepSAM D+E vs LoRA RGB): 94.4% success with 8.8% lower collision rate and 7.6% lower slip rate. We interpret gains as a *lower bound* on real-robot benefit; on UR5e + RealSense D455, we predict $\geq 5\%$ improvement in transparent-object grasp success. Cross-dataset experiments demonstrate transferable representations: training on OCID yields 82.3% mIoU on ClearGrasp vs SAM’s zero-shot 34.7%.

5 Conclusion

Layer-wise representation similarity (CKA) provides an effective guide for allocating adaptation capacity in parameter-efficient fine-tuning. RepSAM achieves 89.0% average mIoU across six robotic benchmarks—97.9% of full fine-tuning performance with $158\times$ fewer parameters. The CKA-guided rank allocation transfers across diverse scenarios: transparent objects (ClearGrasp), industrial parts (LINEMOD), and warehouse settings (WISDOM).

For robotics, improved perception yields safer manipulation: RepSAM reduces collision risk by 8.8% and slip risk by 7.6% in simulation while maintaining practical latency (48ms on A100, 63ms on Jetson AGX Orin).

Limitations. CKA preprocessing requires ~ 30 minutes per dataset. Three-regime boundaries are empirically determined. Depth sensor assumed (though RGB-only variants work).

Method / Platform	Params	Train	Latency	FPS
Full Fine-tuning	632M	48h×8	45ms	—
RepSAM (A100)	4.0M	4h×1	48ms	20.8
RepSAM (Jetson AGX)	—	—	63ms	15.8

Table 10: Efficiency comparison and hardware benchmarks.

Scenario	Method	mIoU	Success	Latency
Standard	SAM 2-L	88.2%	91.0%	85ms
	RepSAM	90.5%	93.2%	48ms
Transparent	SAM 2-L	72.1%	68.5%	85ms
	RepSAM	85.3%	82.1%	52ms

Table 11: Grasping results (PyBullet).

Setting	Success	Coll.	Slip	Regrasp	Time (s)
LoRA (RGB)	82.4±3.2	14.8±2.1	12.4±1.8	16.0±2.4	2.31±0.12
RepSAM (RGB)	88.4±2.5	10.0±1.6	8.8±1.4	11.2±1.9	2.38±0.11
RepSAM (D+E)	94.4±1.8	6.0±1.2	4.8±0.9	6.8±1.3	2.42±0.10

Table 12: Manipulation safety ($N=25$ trials $\times$ 5 seeds). Values show mean±std (%). “Coll.” = collision rate.

Future work. Learning rank predictors from small CKA samples; online adaptation; extension to DINOv2/CLIP; physical robot validation.

Ethical Statement

We use only publicly released datasets with existing licenses (OCID, ClearGrasp, GraspNet, WISDOM, YCB-Video, LINEMOD). No personally identifiable information is involved. Potential risks include mis-segmentation of hazardous objects in real environments; to mitigate harm, RepSAM should be deployed with safety monitors (collision prediction and emergency stop) and conservative planners.

References

- [Chen *et al.*, 2023] Tianrun Chen, Lanyun Zhu, Chaotao Ding, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. SAM-Adapter: Adapting segment anything in underperformed scenes. In *ICCV Workshops*, pages 3359–3367, 2023.
- [Chen *et al.*, 2024] Keyan Chen, Chenyang Liu, Hao Chen, Haotian Zhang, Wenyuan Li, Zhengxia Zou, and Zhenwei Shi. RSPrompter: Learning to prompt for remote sensing instance segmentation based on visual foundation model. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–17, 2024.
- [Danielczuk *et al.*, 2019] Michael Danielczuk, Matthew Matl, Saurabh Gupta, Andrew Li, Andrew Lee, Jeffrey Mahler, and Ken Goldberg. Segmenting unknown 3D objects from real depth images using Mask R-CNN trained on synthetic data. In *ICRA*, pages 7283–7290, 2019.
- [Fang *et al.*, 2020] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. GraspNet-1Billion: A large-scale benchmark for general object grasping. In *CVPR*, pages 11444–11453, 2020.

- [Firoozi *et al.*, 2025] Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, Brian Ichter, Danny Driess, Jiajun Wu, Cewu Lu, and Mac Schwager. Foundation models in robotics: Applications, challenges, and the future. *International Journal of Robotics Research*, 44(5):701–739, 2025.
- [Gu *et al.*, 2026] Jiancheng Gu, Jiabin Yuan, Jiyuan Cai, Xianfa Zhou, and Lili Fan. La-LoRA: Parameter-efficient fine-tuning with layer-wise adaptive low-rank adaptation. *Neural Networks*, 194:108095, 2026.
- [Hinterstoisser *et al.*, 2012] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In *ACCV*, pages 548–562, 2012.
- [Hu *et al.*, 2022] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.
- [Ke *et al.*, 2023] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *NeurIPS*, volume 36, 2023.
- [Kirillov *et al.*, 2023] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *ICCV*, pages 4015–4026, 2023.
- [Kopiczko *et al.*, 2024] Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. VeRA: Vector-based random matrix adaptation. In *ICLR*, 2024.
- [Kornblith *et al.*, 2019] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, pages 3519–3529, 2019.
- [Liu *et al.*, 2024] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *ICML*, pages 32100–32121, 2024.
- [Liu *et al.*, 2025] Jiaming Liu, Linghe Kong, and Guihai Chen. Improving SAM for camouflaged object detection via dual stream adapters. In *ICCV*, pages 21906–21916, 2025.
- [Ma *et al.*, 2024] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024.
- [Mahler *et al.*, 2017] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In *RSS*, 2017.
- [Mousavian *et al.*, 2019] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-DOF GraspNet: Variational grasp generation for object manipulation. In *ICCV*, pages 2901–2910, 2019.
- [Ravi *et al.*, 2024] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv:2408.00714*, 2024.
- [Sajjan *et al.*, 2020] Shreeyak S. Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. ClearGrasp: 3D shape estimation of transparent objects for manipulation. In *ICRA*, pages 3634–3642, 2020.
- [Suchi *et al.*, 2019] Markus Suchi, Timothy Patten, David Fischinger, and Markus Vincze. EasyLabel: A semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets. In *ICRA*, pages 6678–6684, 2019.
- [Xiang *et al.*, 2018] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *RSS*, 2018.
- [Xiong *et al.*, 2024] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, Raghuraman Krishnamoorthi, and Vikas Chandra. EfficientSAM: Leveraged masked image pretraining for efficient segment anything. In *CVPR*, 2024.
- [Zeng *et al.*, 2018] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R. Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, Nima Fazeli, Ferran Alet, Nikhil Chavan Dafle, Rachel Holladay, Isabella Morona, Prem Qu Nair, Druck Green, Ian Taylor, Weber Liu, Thomas Funkhouser, and Alberto Rodriguez. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *ICRA*, 2018.
- [Zhang *et al.*, 2023a] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight SAM for mobile applications. *arXiv:2306.14289*, 2023.
- [Zhang *et al.*, 2023b] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*, 2023.
- [Zhao *et al.*, 2023] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv:2306.12156*, 2023.
- [Zhao *et al.*, 2024] Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *ICML*, 2024.