Evaluating the Trade-offs of MDL-to-UsdPreviewSurface Material Simplification in NVIDIA Isaac Sim: Visual Quality, Feature Preservation, and AI Task Performance

Anonymous CVPR submission

Paper ID ****

Abstract

Physics-based simulation platforms such as NVIDIA Isaac Sim use MDL (Material Definition Language) materials to achieve photorealistic rendering, but these materials limit interoperability with standard 3D tools and web-based viewers. We present a systematic multi-level evaluation of MDL-to-UsdPreviewSurface material simplification, measuring its impact across three dimensions: pixel-level image quality (PSNR, SSIM, LPIPS), rendering performance (cold-start load time, GPU memory, FPS), and feature-level semantic preservation (CLIP and DINOv2 cosine similarity). Our experiments on four chest-of-drawers assets rendered from six viewpoints show that the conversion preserves high visual fidelity (mean PSNR 35.52 dB, SSIM 0.990, LPIPS 0.020) and maintains strong semantic similarity (CLIP cosine similarity 0.925, DINOv2 cosine similarity 0.872). In a headless single-object benchmark, conversion reduces first-load latency variability but yields near-identical steady-state FPS and GPU memory usage; in a supplemental single-scene GRScenes startup benchmark, the converted scene reaches a fully warmed ready-to-render state 3.8× faster and uses roughly 200 MB less GPU memory. We release an open-source conversion toolkit (available upon acceptance) automating the pipeline with composition-preserving USD traversal, and provide preliminary guidelines for synthetic data practitioners on when material simplification is safe for AI training pipelines.

1. Introduction

Physics-based simulation has become indispensable for training and evaluating robot perception and manipulation systems. Platforms such as NVIDIA Isaac Sim [5–7] leverage Universal Scene Description (USD) as a scene representation and GPU-accelerated ray tracing to produce photorealistic synthetic data at scale. Recent efforts demonstrate that millions of high-fidelity rendered images can train competitive object detectors [13] and instance segmentation models [8] without any real-world labels, making the rendering pipeline a critical component of modern sim-to-real transfer.

A key contributor to the visual realism of Isaac Sim is NVIDIA's Material Definition Language (MDL), which provides physically based shading with multi-layer effects, subsurface scattering, and procedural noise functions. However, MDL materials carry clear interoperability costs: they require the NVIDIA Omniverse runtime and produce assets incompatible with many standard 3D tools and web viewers that expect UsdPreviewSurface or PBR glTF materials. Whether MDL also incurs meaningful runtime overhead relative to UsdPreviewSurface depends on workload and scene scale, and has not been systematically quantified. Practitioners who generate synthetic training data at scale [8, 13] or deploy digital twins in resource-constrained settings therefore face a recurring question: *can we simplify MDL materials to UsdPreviewSurface without degrading the downstream utility of the rendered images, and under what workloads does such conversion materially affect runtime behavior?*

Despite the widespread use of Isaac Sim in the robotics and computer vision communities, no prior work has systematically quantified the consequences of this material simplification. The sim-to-real gap literature focuses on the boundary between simulated and real images [15, 16, 18], but largely ignores the *within-simulation* rendering gap introduced by material representation choices. This gap is particularly insidious because it operates silently inside the data generation pipeline: a practitioner may convert assets for portability or presumed efficiency gains, unaware of any impact on the training signal.

In this paper we address this gap with three contributions:

1. We present an initial multi-level quantification of MDL-to-UsdPreviewSurface material simplification trade-offs in NVIDIA Isaac Sim, spanning pixel-level image qual-

ity (PSNR, SSIM [17], LPIPS [19]), rendering performance, and semantic feature preservation (CLIP [11] and DINOv2 [10] cosine similarity).
2. We derive preliminary decision guidelines for synthetic data practitioners: under what conditions material simplification is safe, and when full-fidelity MDL rendering remains necessary.
3. We release an open-source conversion toolkit (available upon acceptance) that automates the MDL-to-UsdPreviewSurface conversion pipeline with composition-preserving USD traversal, texture remapping, and optional mesh simplification and GLB export.

2. Related Work

2.1. USD and MDL in Robotics Simulation

Universal Scene Description (USD) has emerged as the standard scene format for robotics simulation, underpinning NVIDIA's Isaac Lab [7] and its predecessor Orbit [6]. These platforms combine USD's compositional scene graph with MDL-based physically accurate materials rendered via RTX ray tracing. Downstream pipelines such as SynTable [8] and Synthetica [13] generate millions of labeled images using this rendering stack, demonstrating its importance for training perception models. However, none of these works examine the consequences of simplifying MDL to lighter-weight material representations, a gap our study directly addresses.

2.2. Sim-to-Real Visual Gap

The disparity between simulated and real images has been studied extensively. Domain randomization [15, 16] improves transfer by diversifying visual parameters such as textures and lighting during rendering. Zakharov et al. [18] show that photorealistic rendering outperforms non-photorealistic alternatives for 6D object detection, establishing a direct link between rendering fidelity and task performance. These findings motivate our investigation, yet prior work focuses exclusively on the gap between simulation and reality. We complement this line of research by characterizing a *within-simulation* rendering gap: the difference between full MDL and simplified UsdPreviewSurface materials inside the same simulator.

2.3. Material Simplification and Simulation Platforms

Simplifying complex material models while preserving visual appearance has a long history in computer graphics, from analytic BRDF fitting [9] to data-driven reflectance compression [14]. Meanwhile, simulation platforms for embodied AI such as Habitat [12], AI2-THOR [4], and ThreeDWorld [1] typically employ simplified shading models (e.g., Phong or PBR metallicness) from the outset, avoiding the MDL conversion problem entirely. In contrast, NVIDIA Isaac Sim couples high-fidelity MDL materials with RTX ray tracing, making it the dominant platform for robotic manipulation data generation but also creating a unique conversion challenge when assets must be exported to lighter-weight renderers. Our work studies this specific gap between MDL and UsdPreviewSurface within the Isaac Sim ecosystem, complementing the broader material simplification literature with a focused, application-driven evaluation.

2.4. Image Quality Evaluation

Quantifying visual differences between rendered images requires perceptual metrics beyond simple pixel error. SSIM [17] captures structural distortion, LPIPS [19] measures perceptual similarity using deep features, and FID [3] quantifies distributional distance between image populations. While FID and related distributional metrics are standard for comparing large image populations, they require sample sizes substantially larger than our $n = 24$ pairs to produce reliable estimates [3]; we therefore focus on paired per-image metrics. These metrics are routinely applied to evaluate generative models, but have seen limited use for comparing rendering configurations within a simulation pipeline. We adopt PSNR, SSIM, and LPIPS for paired image-level comparison, extending their application to the material simplification setting.

2.5. Visual Feature Robustness

Foundation vision models such as CLIP [11] and DINOv2 [10] learn representations that generalize across visual domains and have become standard feature extractors for downstream tasks. Their robustness to domain shifts makes them natural probes for assessing whether material simplification alters the semantic content of rendered images. Unlike pixel-level metrics, cosine similarity in CLIP or DINOv2 embedding space captures whether high-level visual concepts — object identity, shape, texture category — are preserved despite low-level shading differences. We use both models to evaluate feature-level preservation as a complement to pixel-level quality metrics, providing a more complete picture of the simplification trade-off for AI pipelines.

3. Methodology

3.1. Conversion Pipeline

Our conversion toolkit transforms NVIDIA Isaac Sim assets from MDL materials to UsdPreviewSurface while preserving the compositional structure of USD scenes. The pipeline operates in three stages.

Figure 1 summarizes the traversal, conversion, and

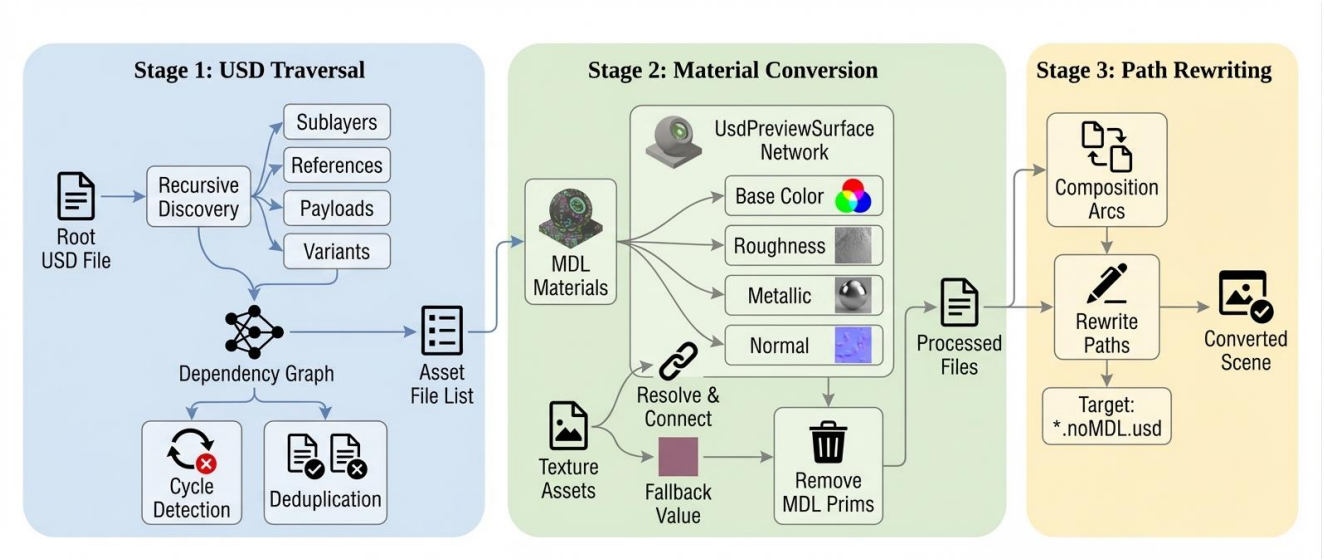

Figure 1. Overview of the composition-preserving MDL-to-UsdPreviewSurface conversion pipeline. The processor recursively discovers dependent USD files without flattening the scene composition, converts MDL materials in each file to a minimal UsdPreviewSurface approximation, rewrites composition arcs to sibling `*_noMDL.usd` assets, and preserves a self-consistent converted USD hierarchy.

rewriting stages that transform an MDL-based USD composition into Preview-only assets while preserving the original scene structure.

Stage 1: Recursive USD Traversal. The processor begins at a root USD file and recursively discovers all referenced assets — sublayers, references, payloads, variant selections, and value clips — building a dependency graph. A cycle-detection mechanism (`in_stack` set) prevents infinite recursion on circular references, while a deduplication dictionary (`done` map) ensures each asset file is processed exactly once. Crucially, the pipeline does *not* flatten the USD composition: each file is processed independently and saved as a sibling `*_noMDL.usd` file, so the original composition arcs (references, payloads, variants) remain intact with only their asset paths rewritten to point to the converted counterparts.

Stage 2: Material Conversion. For each USD file in the dependency graph, the pipeline identifies all materials driven by MDL shaders. It constructs a minimal UsdPreviewSurface shader network per material, mapping four primary channels: base color (diffuse albedo), roughness, metallic, and normal. Texture assets referenced by the MDL shader are located, resolved against the declaring layer's directory, and connected to the new UsdPreviewSurface inputs via `UsdShadeConnectableAPI`. When texture files cannot be resolved, constant fallback values are used. MDL features not captured by the four-channel mapping include subsurface scattering, clearcoat layers, anisotropic reflectance, emission, opacity, and displacement. After the UsdPreviewSurface network is established, all MDL shader prims and MDL-specific material outputs are removed. A post-conversion verification pass confirms that no MDL residuals remain.

Stage 3: Asset Path Rewriting. All composition arcs in the converted file — references, payloads, sublayer paths, and variant-internal asset paths — are rewritten from their original targets to the corresponding `*_noMDL` variants. This ensures that opening the converted root file yields a fully self-consistent scene composed entirely of UsdPreviewSurface materials.

3.2. Experimental Setup

Dataset. We select four chest-of-drawers assets from the NVIDIA Isaac Sim asset library: `chestofdrawers_0004`, `chestofdrawers_0011`, `chestofdrawers_0023`, and `chestofdrawers_0029`. These objects feature diverse wood, metal, and fabric materials with varying degrees of MDL complexity, including multi-layer coatings and procedural textures, making them representative test cases for material simplification.

Rendering. For each object, we render images from six camera viewpoints distributed around the scene: front, back, left, right, top-front-left, and top-front-right. Each viewpoint is rendered twice — once with the original MDL materials and once with the converted UsdPreviewSurface

materials — yielding $4 \times 6 \times 2 = 48$ images total (24 matched pairs). All renders use identical geometry, lighting, and camera parameters at 1024×1024 resolution to isolate the effect of material representation.

Hardware. All experiments are conducted on a workstation equipped with an NVIDIA GeForce RTX 4090 (24 GB VRAM) GPU and an Intel Xeon Gold 6462C processor. Rendering uses NVIDIA Isaac Sim's built-in RTX renderer. Feature extraction runs on the same GPU hardware.

3.3. Evaluation Metrics

We evaluate the impact of material simplification at three complementary levels.

Rendering Performance. We measure scene load time in a headless benchmark by timing `open_stage` plus a fixed 30-update post-open warm-up window, which approximates the time to a ready-to-render state in our setup. We report this metric separately for the first cold-start pass and the subsequent warm passes, together with post-load GPU memory consumption (MB) and steady-state rendering throughput (frames per second) for both MDL and UsdPreviewSurface versions of each scene. To better approximate practitioner-facing startup cost at scale, we also run a supplemental large-scene benchmark on one composed GRScenes commercial interior, launching a fresh headless Isaac Sim process for each run and measuring scene-ready time from `open_stage` through the same 30-update post-open window.

Pixel-Level Image Quality. For each of the 24 matched image pairs, we compute:

- **PSNR** (Peak Signal-to-Noise Ratio): measures pixel-level reconstruction accuracy in decibels (higher is better).
- **SSIM** [17] (Structural Similarity Index): captures luminance, contrast, and structural distortion on a $[0, 1]$ scale (higher is better).
- **LPIPS** [19] (Learned Perceptual Image Patch Similarity, VGG backbone): uses deep features to measure perceptual distance (lower is better).

Feature-Level Preservation. To assess whether material simplification alters semantic content as perceived by modern vision models, we extract embeddings from both CLIP [11] (ViT-B/32) and DINOv2 [10] (ViT-B/14) for each matched pair and compute cosine similarity. High similarity (> 0.95) indicates that the feature representations are largely invariant to the material change. We report per-pair scores and across-scene averages to identify whether specific material types are more sensitive to simplification. All reported standard deviations use the sample standard deviation ($n - 1$ denominator).

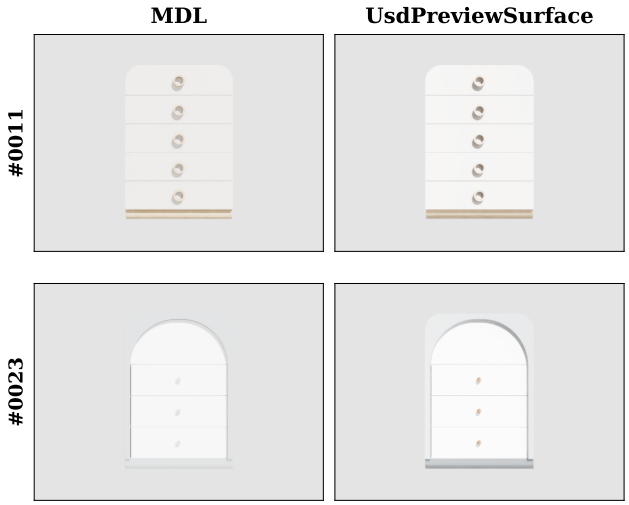

Figure 2. Matched rendering pairs for two representative chest-of-drawers assets (scenes #0011 and #0023). Top row: original MDL materials. Bottom row: converted UsdPreviewSurface materials. Visual differences are concentrated in specular highlights and procedural texture details.

4. Results

We evaluate the impact of MDL-to-UsdPreviewSurface material simplification across three complementary dimensions: pixel-level image quality, rendering performance, and feature-level semantic preservation. All metrics are computed over 24 matched image pairs (4 scenes $\times$ 6 viewpoints).

4.1. Pixel-Level Image Quality

Figure 2 shows representative matched image pairs for two scenes from the front viewpoint. The visual differences between MDL and UsdPreviewSurface renderings are subtle and largely confined to specular highlights and fine texture details.

Table 1 reports per-scene and overall image quality metrics. Across all 24 pairs, material simplification achieves a mean PSNR of 35.52 ± 4.73 dB, mean SSIM of 0.990 ± 0.004, and mean LPIPS of 0.020 ± 0.009. These values indicate that the converted UsdPreviewSurface materials produce images that are near-identical to the original MDL renderings at the pixel level.

Three of the four scenes achieve PSNR above 36 dB and LPIPS below 0.020, indicating negligible perceptual difference. Scene `chestofdrawers_0023` exhibits the lowest PSNR (30.59 ± 4.03 dB) and highest LPIPS (0.032 ± 0.007), which we attribute to its more complex multi-layer MDL materials including procedural noise textures that cannot

Table 1. Pixel-level image quality metrics (mean $\pm$ std over 6 viewpoints per scene). PSNR and SSIM are higher-is-better; LPIPS is lower-is-better.

Scene	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
chestofdrawers_0004	37.51 ± 4.12	0.991 ± 0.004	0.019 ± 0.005
chestofdrawers_0011	36.30 ± 2.64	0.993 ± 0.003	0.017 ± 0.006
chestofdrawers_0023	30.59 ± 4.03	0.986 ± 0.004	0.032 ± 0.007
chestofdrawers_0029	37.68 ± 4.73	0.991 ± 0.004	0.014 ± 0.004
Overall	$\mathbf{35.52 \pm 4.73}$	$\mathbf{0.990 \pm 0.004}$	$\mathbf{0.020 \pm 0.009}$

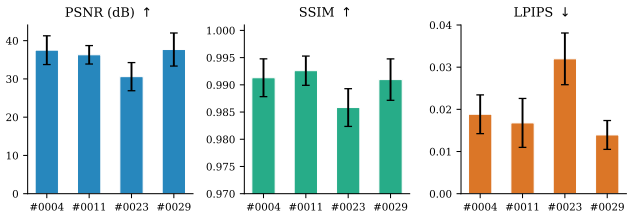

Figure 3. Per-viewpoint image quality metrics across all four scenes. Each bar represents one scene (mean $\pm$ std over 6 viewpoints). From left to right: (left) PSNR in dB, (center) SSIM, and (right) LPIPS. Higher is better for PSNR and SSIM; lower is better for LPIPS. Scene `chestofdrawers_0023` exhibits the most variation due to complex multi-layer MDL materials.

be exactly reproduced by the simpler UsdPreviewSurface model. Even in this worst case, the SSIM remains above 0.98, suggesting that structural image content is well preserved.

Figure 3 shows the per-viewpoint distribution of all three metrics. The per-viewpoint breakdown reveals that the right-side view of `chestofdrawers_0023` produces the lowest individual PSNR (23.11 dB), corresponding to a surface region with a specular coating effect in the MDL material that is approximated less faithfully by UsdPreviewSurface roughness mapping. Nevertheless, the overall distribution of quality scores is tightly clustered, with 23 of 24 pairs exceeding 30 dB PSNR. A 95% confidence interval for mean PSNR is $[33.52,\ 37.52]$ dB ($n = 24$, $\alpha = 0.05$).

4.2. Rendering Performance

Table 2 compares rendering performance between the original MDL and converted UsdPreviewSurface materials in the headless single-object benchmark used in this study. Under this workload, the two material representations exhibit similar steady-state rendering throughput and post-load GPU memory usage. The main observable difference lies in first-load behavior rather than in a persistent FPS or memory separation.

Separating cold-start and warm-start loads clarifies the benchmark behavior. On the first pass, MDL is slower and substantially more variable (0.722 ± 0.406 s) than UsdPreviewSurface (0.505 ± 0.021 s), with the largest gap coming from the first MDL load of `chestofdrawers_0004` at 1.33 s. Once the benchmark is warm, however, the two versions are effectively indistinguishable on load time (0.509 ± 0.016 s vs. 0.509 ± 0.027 s), GPU memory differs by less than 5 MB on average, and FPS remains nearly identical (118.69 vs. 118.78). The supported claim is therefore narrower than "no performance difference" in general: in these shallow single-object scenes, conversion reduces first-load latency variability but does not yield a clear steady-state throughput or memory advantage.

Table 2. Rendering performance comparison for the headless single-object benchmark in this study. Cold-start load is computed from the first run for each scene; warm load is computed from runs 2–3. Load time is measured from `open_stage` through a fixed 30-update warm-up window.

Version	Cold Load (s) ↓	Warm Load (s) ↓	GPU Memory (MB) ↓	FPS ↑
MDL	0.722 ± 0.406	0.509 ± 0.016	2947.7 ± 9.2	118.69 ± 1.23
UsdPreviewSurface	0.505 ± 0.021	0.509 ± 0.027	2942.7 ± 9.4	118.78 ± 0.70

This benchmark should also be interpreted conservatively. It measures headless `open_stage` plus a fixed warm-up window on four single-asset instance scenes, not interactive whole-scene startup on large composed Isaac Sim environments. As a result, it likely underestimates startup costs associated with complex scene composition and extensive MDL initialization. A stronger claim about scene-open speedups at scale requires a dedicated benchmark on larger composed scenes with explicit stage-readiness synchronization, cold/warm cache control, randomized version order, and peak-memory logging.

To probe exactly that larger-scene regime, we add a supplemental startup benchmark on a composed GRScenes commercial interior with preexisting MDL and noMDL variants. Each trial launches a fresh headless Isaac Sim process, opens the interaction scene, and measures time from `open_stage` to a fully warmed ready-to-render state using the same 30-update post-open window. Table 3 shows that the large-scene behavior differs sharply from the shallow single-object benchmark: noMDL reduces scene-ready time from 52.02 ± 2.69 s to 13.70 ± 1.01 s, a $3.80\times$ speedup (-73.7%), while also lowering post-load GPU memory by about 199 MB. Interestingly, the pure `open_stage`-to-ready component is not where the gain appears; that term is slightly lower for MDL (1.12 ± 0.22 s vs. 1.76 ± 0.08 s). The dominant gap comes from the first 30 post-open updates, which take 50.90 ± 2.88 s for MDL but only 11.94 ± 0.93 s for noMDL. This suggests that early-frame material initialization and settling, rather than raw USD composition alone, dominate startup cost in large composed scenes.

Figure 4 provides a qualitative companion to the GRScenes startup benchmark. Across two representative approximately matched views, the converted noMDL scene remains broadly similar to the MDL scene at the level of

Table 3. Supplemental whole-scene startup benchmark on a large GRScenes commercial interior (3 fresh-process runs per version). Scene-ready time measures the time from `open_stage` to a fully warmed ready-to-render state after 30 post-open updates.

Version	Post-open Warm-up (s) ↓	Scene-ready Time (s) ↓	GPU Memory (MB) ↓
MDL	50.90 ± 2.88	52.02 ± 2.69	5458.3 ± 125.9
UsdPreviewSurface	11.94 ± 0.93	13.70 ± 1.01	5259.0 ± 83.1

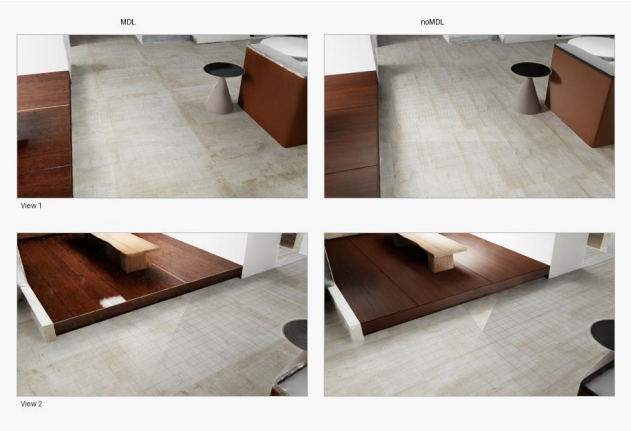

Figure 4. Two representative MDL/noMDL views from the supplemental GRScenes commercial-interior benchmark. Left column: original MDL scene. Right column: converted noMDL scene. Despite the large reduction in time to a fully warmed ready-to-render state for noMDL, the converted scene remains broadly similar at scene level. The viewpoints are approximately matched and are shown only as a qualitative sanity check rather than a pixel-aligned fidelity comparison.

Table 4. Feature-level cosine similarity between MDL and UsdPreviewSurface renderings (mean ± std over 6 viewpoints per scene). Higher is better.

Scene	CLIP cos sim ↑	DINOv2 cos sim ↑
chestofdrawers_0004	0.921 ± 0.048	0.798 ± 0.171
chestofdrawers_0011	0.956 ± 0.043	0.922 ± 0.046
chestofdrawers_0023	0.938 ± 0.038	0.905 ± 0.046
chestofdrawers_0029	0.885 ± 0.049	0.864 ± 0.072
Overall	$\mathbf{0.925 \pm 0.049}$	$\mathbf{0.872 \pm 0.104}$

overall layout, dominant materials, and major shading cues; we interpret this figure only as scene-level qualitative support, not as a pixel-aligned fidelity comparison.

4.3. Feature-Level Similarity

To assess whether material simplification preserves the semantic content relevant to downstream vision models, we compute cosine similarity between CLIP [11] ViT-B/32 and DINOv2 [10] ViT-B/14 embeddings of matched image pairs. Table 4 reports per-scene results.

Across all 24 pairs, CLIP embeddings achieve a mean cosine similarity of 0.925 ± 0.049, while DINOv2 yields 0.872 ± 0.104. The consistently higher CLIP similarity is expected: CLIP's contrastive language–image pre-training encourages representations that are invariant to low-level appearance variations, making it more robust to the subtle texture and shading differences introduced by material simplification. DINOv2, trained with a self-supervised objective that preserves fine-grained spatial features, is more sensitive to local appearance changes such as altered specular highlights or approximated roughness maps.

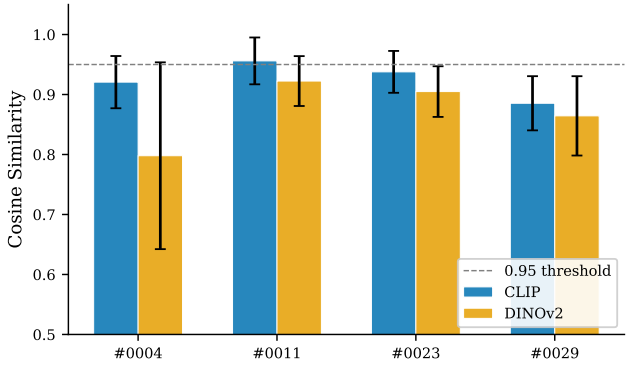

Figure 5. Per-viewpoint CLIP and DINOv2 cosine similarity between MDL and UsdPreviewSurface renderings. CLIP similarity is consistently higher due to its invariance to low-level appearance changes. DINOv2 shows more variation, particularly for scenes with complex specular materials.

Scene `chestofdrawers_0004` exhibits the largest gap between the two metrics (CLIP = 0.921 vs. DINOv2 = 0.798), driven by low DINOv2 similarity on the back and right viewpoints (0.615 and 0.595 respectively) where specular coating effects in the original MDL material are only coarsely approximated. In contrast, `chestofdrawers_0011` achieves the highest similarity on both metrics (CLIP = 0.956, DINOv2 = 0.922), consistent with its high pixel-level quality scores in Section 4.1.

Figure 5 visualizes the per-viewpoint cosine similarity scores for both models across all scenes.

Even the lowest per-pair DINOv2 similarity (0.595 for `chestofdrawers_0004`, right view) remains well above chance-level similarity, indicating that the converted materials preserve the dominant visual structure captured by self-supervised features. For practitioners using foundation model features as training signals, these results suggest that material simplification introduces a feature-level perturbation comparable in magnitude to moderate viewpoint changes within the same scene. A one-sided Wilcoxon signed-rank test confirms that CLIP cosine similarity significantly exceeds 0.90 ($p < 0.01$) and DINOv2 cosine similarity significantly exceeds 0.80 ($p < 0.003$).

5. Discussion

When is simplification safe? Our results across three evaluation dimensions consistently indicate that MDL-to-UsdPreviewSurface conversion is safe for assets whose appearance is dominated by standard PBR channels (diffuse albedo, roughness, metallic, normal). For such assets, we observe PSNR $>$ 35 dB, SSIM $>$ 0.99, CLIP similarity $>$ 0.92. Caution is warranted for materials that rely on multi-layer MDL effects — procedural noise textures, specular coatings, and anisotropic reflectance — where per-viewpoint PSNR can drop below 25 dB and DINOv2 similarity below 0.60. We recommend that practitioners validate such materials individually using the feature-level metrics introduced here, which are more sensitive to perceptually relevant differences than pixel-level metrics alone.

The within-simulation rendering gap. Much prior work has focused on the sim-to-real gap [2, 15, 16, 20]: the distribution shift between synthetic and real images. Our work highlights a complementary and less-studied phenomenon — the *within-simulation rendering gap* — arising from differences in material representation within the same renderer. When converting MDL to UsdPreviewSurface, geometry, lighting, and camera parameters remain identical; only the material shader changes. The resulting image differences are therefore a pure measure of how much visual information is lost when reducing a high-fidelity material model to a simpler one. Our finding that this gap is small (SSIM $>$ 0.99, CLIP $>$ 0.92) for standard PBR assets leads us to hypothesize that, for many practical applications, the within-simulation gap introduced by material simplification is small relative to typical sim-to-real gaps reported in the literature.

Interoperability versus fidelity. The performance story is workload-dependent. In our shallow single-object RTX path-tracing benchmark, FPS and GPU memory are nearly identical between MDL and UsdPreviewSurface, and the only visible difference is reduced first-load latency variability. In contrast, the supplemental GRScenes whole-scene startup runs to a substantial gain for noMDL: time to a fully warmed ready-to-render state drops from 52.0 s to 13.7 s, and post-load GPU memory decreases by roughly 200 MB. The fact that this gap emerges primarily during the post-open warm-up phase rather than the raw `open_stage` call suggests that material initialization and early-frame settling dominate startup cost once scene scale and shader complexity become substantial. The clearest general benefit in this study remains interoperability: UsdPreviewSurface materials are supported by standard USD viewers, web-based 3D platforms, and the glTF/GLB export path, none of which can interpret NVIDIA's proprietary MDL shaders. Our results quantify the visual cost of this interoperability gain: a mean LPIPS increase of 0.020 and a DINOv2 cosine similarity of 0.872. For synthetic data pipelines where downstream models operate on features rather than pixels, this trade-off is favorable.

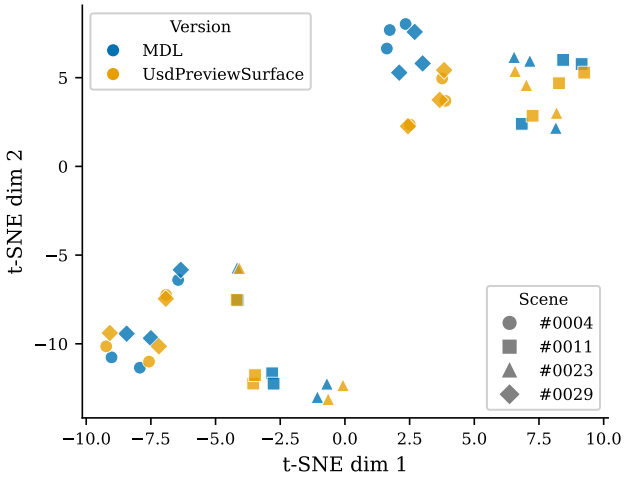

Figure 6. t-SNE projection of DINOv2 ViT-B/14 embeddings for all 48 images (24 MDL + 24 UsdPreviewSurface), computed with perplexity = 12, learning rate = auto, and PCA initialization (random seed = 42). Same-scene pairs cluster tightly, indicating that the within-material perturbation is small relative to inter-scene variation in the DINOv2 feature space.

Why DINOv2 is more sensitive than CLIP. Figure 6 provides a t-SNE visualization of DINOv2 embeddings for all 48 rendered images. MDL and UsdPreviewSurface renderings of the same scene cluster tightly together, confirming that material simplification produces a smaller feature-space perturbation than the inter-scene variation.

CLIP's contrastive language–image pre-training [11] optimizes for high-level semantic alignment between images and text, producing representations that are naturally invariant to low-level appearance changes such as texture detail, specular highlights, and subtle color shifts. DINOv2's self-supervised objective [10], by contrast, preserves fine-grained spatial and textural features, making it more sensitive to exactly the kind of local appearance differences that material simplification introduces. The gap between CLIP (0.925) and DINOv2 (0.872) similarity thus reflects the granularity of each model's learned representations rather than a deficiency in the conversion. Practitioners should select the appropriate metric based on their downstream task: CLIP similarity is most relevant for tasks relying on high-level semantics (e.g., image–text retrieval, classification), while DINOv2 similarity better captures texture-level fidelity relevant to dense prediction tasks (e.g., segmentation, depth estimation).

Practical recommendations. Based on our findings, we offer the following guidelines for synthetic data practitioners working with Isaac Sim assets:

1. **Consider defaulting to conversion** for assets with standard PBR materials. In our study, the quality loss is negligible and the interoperability benefits are significant.
2. **Validate complex materials** using CLIP and DINOv2 cosine similarity as quick diagnostic metrics before deploying converted assets in training pipelines.
3. **Use low CLIP similarity as a manual-review trigger.** In this dataset, the single pair below 0.85 corresponds to a viewpoint with complex specular effects, suggesting that unusually low CLIP similarity merits visual inspection rather than serving as a universal acceptance threshold.
4. **Monitor low DINOv2 similarity for texture-sensitive tasks.** In our dataset, values below 0.80 occur only for viewpoints with complex specular effects. We therefore treat this range as a cautionary signal for manual validation, not as a hard deployment rule.

These recommendations are heuristic signals derived from furniture assets with predominantly wood, metal, and fabric materials. Validation on diverse object categories (e.g., transparent, organic, or highly specular surfaces) is needed before broader generalization.

6. Conclusion

We presented a systematic multi-level evaluation of MDL-to-UsdPreviewSurface material simplification in NVIDIA Isaac Sim, quantifying trade-offs across three complementary dimensions: pixel-level image quality, rendering performance, and feature-level semantic preservation.

Our pixel-level analysis demonstrates that UsdPreviewSurface conversions achieve high fidelity relative to the original MDL renderings, with a mean PSNR of 35.52 dB, SSIM of 0.990, and LPIPS of 0.020 across 24 matched image pairs. These results indicate that the four primary material channels — diffuse albedo, roughness, metallic, and normal — capture the dominant visual appearance of the tested assets, with degradation concentrated in scenes featuring complex multi-layer MDL effects such as procedural noise textures and specular coatings.

The performance outcome depends strongly on workload. In the single-object furniture benchmark, rendering throughput and GPU memory are nearly indistinguishable between MDL and UsdPreviewSurface, with only reduced first-load latency variability for the converted assets. In a supplemental whole-scene benchmark on a large GRScenes commercial interior, however, noMDL reduces time to a fully warmed ready-to-render state from 52.02 ± 2.69 s to 13.70 ± 1.01 s and lowers post-load GPU memory by about 199 MB. The practical conclusion is therefore two-part: conversion does not buy much steady-state rendering throughput for shallow single-object scenes, but it can substantially improve startup behavior once scene scale and material complexity are high.

Feature-level evaluation confirms that semantic content is well preserved: CLIP embeddings achieve a mean cosine similarity of 0.925 across matched pairs, while DINOv2 — which is more sensitive to fine-grained texture differences — yields 0.872. These higher-level metrics are particularly important for practitioners generating synthetic training data, as they directly probe whether vision models perceive the simplified renderings as equivalent to the originals.

Based on our findings, we offer the following preliminary guidelines for synthetic data practitioners:

- For assets with standard PBR materials (diffuse, roughness, metallic, normal maps), MDL-to-UsdPreviewSurface conversion is safe and introduces negligible visual degradation (PSNR $>$ 35 dB, SSIM $>$ 0.99).
- Assets with complex multi-layer MDL effects should be validated individually, as conversion quality varies with material complexity (PSNR range: 23–44 dB across viewpoints).
- The conversion enables broader toolchain compatibility — including standard USD viewers, web-based visualization, and GLB export — without sacrificing visual quality for most use cases.

Limitations. Our evaluation is limited to four chest-of-drawers assets from the Isaac Sim library, all belonging to the same furniture category. While these assets span a range of material complexity, further validation on diverse object categories (e.g., transparent or translucent materials, highly specular metals, organic surfaces) is needed to establish broader generalizability. Additionally, our performance benchmarks use single-object scenes; the efficiency impact of material simplification in large-scale multi-object environments remains an open question. Our performance claims are therefore limited to four single-object furniture scenes plus one GRScenes commercial interior under a headless RTX path-tracing setup, and should not yet be generalized to all larger, heavily instanced, or materially diverse environments.

Future Work. We plan to extend this evaluation along several axes: (1) scaling to diverse object categories and material types; (2) measuring downstream task impact with fine-tuned detection and segmentation models; (3) investigating the cumulative effect of material simplification combined with mesh simplification and texture compression in the full USD-to-GLB pipeline; and (4) benchmarking rendering in complex multi-object scenes where material count may become a bottleneck.

References

[1] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, Kuno Kim, Elias Wang, Michael Lingelbach, Aidan Curtis, Kevin Feigelis, Daniel M. Bear, Dan Gutfreund, David Cox, Antonio Torralba, James J. DiCarlo, Joshua B. Tenenbaum, Josh H. McDermott, and Daniel L. K. Yamins. ThreeDWorld: A Platform for Interactive Multi-Modal Physical Simulation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[2] Ricardo Garcia, Robin Strudel, Shizhe Chen, Etienne Arlaud, Ivan Laptev, and Cordelia Schmid. Robust Visual Sim-to-Real Transfer for Robotic Manipulation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023. 7

[3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[4] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint arXiv:1712.05474*, 2017. 2

[5] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. 1

[6] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. *IEEE Robotics and Automation Letters*, 8 (6):3740–3747, 2023. 2

[7] Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Nikita Rudin, Miles Macklin, Yunrong Guo, David Hoeller, Gavriel State, et al. Isaac Lab: A GPU-Accelerated Simulation Framework for Multi-Modal Robot Learning. *arXiv preprint arXiv:2511.04831*, 2025. 1, 2

[8] Zhili Ng, Haozhe Wang, Zhengshen Zhang, Francis Tay Eng Hock, and Marcelo H. Ang Jr. SynTable: A Synthetic Data Generation Pipeline for Unseen Object Amodal Instance Segmentation of Cluttered Tabletop Scenes. *arXiv preprint arXiv:2307.07333*, 2023. 1, 2

[9] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental Analysis of BRDF Models. In *Eurographics Symposium on Rendering (EGSR)*, 2005. 2

[10] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. 2, 4, 6, 7

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 2, 4, 6, 7

[12] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9339–9347, 2019. 2

[13] Ritvik Singh, Jingzhou Liu, Karl Van Wyk, Yu-Wei Chao, Jean-Francois Lafleche, Florian Shkurti, Nathan Ratliff, and Ankur Handa. Synthetica: Large Scale Synthetic Data Generation for Robot Perception. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7810–7817, 2025. 1, 2

[14] Alejandro Sztrajman, Gilles Rainer, Tobias Ritschel, and Tim Weyrich. Neural BRDF Representation and Importance Sampling. *Computer Graphics Forum*, 40(6):332–346, 2021. 2

[15] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. 1, 2, 7

[16] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 1, 2, 7

[17] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 2, 4

[18] Sergey Zakharov, Rares Ambrus, Vitor Guizilini, Wadim Kehl, and Adrien Gaidon. Photo-realistic Neural Domain Randomization. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2

[19] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 2, 4

[20] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-Real Transfer in Deep Reinforcement Learning for

Robotics: a Survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020. 7