A Image-Conditioned Evaluation

In the main paper, Native3DEditing is evaluated under a *text-only* setting where only the source asset and the natural-language instruction are provided, while VoxHammer additionally receives the projected edited-region masks from our annotations. Because every sample in Omni3DEdit contains a complete target 3D asset, we can further evaluate methods under an *image-conditioned* setting: one or more views rendered from the target serve as a visual reference for the expected output, substantially reducing the ambiguity of a pure text instruction.

Setup. For each test sample, we render the target asset from a canonical front view and supply this image as an additional input to methods that support image conditioning. For VoxHammer, this target view supplements the region mask already used in the main paper; for 3DEditVerse, the target view serves as the primary edit reference. All other settings (camera set, rendering pipeline, region masks where applicable) remain identical to the main-paper protocol. We additionally report reconstruction-oriented metrics (e.g., mLPIPS) between the edited prediction and the target renderings to directly assess visual fidelity.

Results. Table 5 reports text-only and image-conditioned results. For VoxHammer, adding a visual reference yields notable improvements in perceptual reconstruction (mLPIPS drops from 0.293 to 0.202) and edit locality (Region-F1 rises from 0.130 to 0.321), while faithfulness remains stable (CLIP-T 0.249→0.250). We additionally evaluate 3DEditVerse [24], an image-conditioned learned 3D editor. As a cross-method reference, we compare 3DEditVerse with the text-only Native3DEditing [3]. 3DEditVerse achieves higher semantic faithfulness (CLIP-T 0.336 vs. 0.253; DINO-I 0.667 vs. 0.622) and better geometric preservation (Preserve CD 0.807 vs. 0.970; mLPIPS 0.245 vs. 0.283), while maintaining comparable edit locality (Region-F1 0.176 vs. 0.165).

Analysis. The results above highlight two observations regarding image conditioning within Omni3DEdit. First, for the same method (VoxHammer), adding a reference image notably improves locality and reconstruction quality while keeping faithfulness stable; this controlled, same-method comparison provides direct evidence for the benefit of visual guidance. Second, across methods, the image-conditioned 3DEditVerse outperforms the text-only Native3DEditing on faithfulness and preservation. Because this cross-method comparison conflates differences in model architecture with differences in conditioning modality, the improvement cannot be solely attributed to image conditioning. Nonetheless, the consistent direction of improvement across both comparisons (one controlled, one cross-method) suggests that visual references reduce the ambiguity inherent in pure text instructions. Together, these findings confirm that Omni3DEdit supports meaningful evaluation of both text-guided and image-guided 3D editing pipelines within a single benchmark.

B Dataset Schema Details

We standardize all data subsets in Omni3DEdit into a common schema to facilitate unified data loading for benchmark evaluation. The comprehensive list of these annotation fields is provided in Table 6.

C Baseline Details

We provide detailed descriptions of each baseline evaluated in our benchmark.

Source Copy. This baseline directly outputs the source asset without performing any edit. It provides a useful lower bound for instruction faithfulness and a strong sanity check for preservation metrics: any plausible editing method should outperform Source Copy on faithfulness while approaching its preservation scores.

Native3DEditing [3]. This is a representative learned native 3D editing model that takes a source 3D asset and a natural-language instruction as input to produce the edited 3D asset. It is the most directly matched prior method to our benchmark setup: the model operates natively in 3D space with full attention across the entire asset, making it a natural upper-bound reference for instruction faithfulness at the potential cost of over-editing.

VoxHammer [14]. This is a representative training-free native 3D editing model that emphasizes local editing and preservation of unedited regions. It operates by constraining the editing scope through a localized edited-region mask, making it a particularly strong locality-aware baseline given that Omni3DEdit provides explicit edited-region annotations. However, VoxHammer requires a meaningfully localized mask to function: when supplied with a near-full mask (as would be necessary for the global pose changes in the Pose subset), the model alters the entire asset indiscriminately. We therefore exclude VoxHammer from the Pose subset.

3DEditVerse [24]. This is a learned image-conditioned 3D editing model that takes a source 3D asset together with one or more target-view reference images to produce the edited output. Because 3DEditVerse is designed exclusively for image-conditioned editing and does not support text-only input, we include it in the image-conditioned evaluation (Appendix A) as a cross-method reference alongside the image-conditioned variant of VoxHammer.

D Implementation Details

Rendering setup. All methods share a fixed multi-view camera set for standardized evaluation. Viewpoints are uniformly distributed on the upper hemisphere with a white background. Specifically, we render $V = 8$ viewpoints for each 3D asset. The cameras are positioned at a fixed distance from the object center, uniformly spaced at 45° intervals along the azimuth. To establish the elevation, we apply a fixed upward vertical offset equal to 0.3 times the object's bounding radius. The same rendering resolution of 512×512 pixels and a vertical field of view of 60° ($\pi/3$ radians) are used for both generating baseline outputs and computing all evaluation metrics.

Baseline configurations. For all baselines, we use officially released code and checkpoints with default inference hyperparameters. Native3DEditing [3] takes the source GLB and the natural-language instruction as input. VoxHammer [14] additionally receives the projected edited-region mask from our annotations; it is excluded from the Pose subset because near-full masks degrade its locality-aware editing design. For the image-conditioned experiments (Appendix A), a single canonical front-view rendering of the target asset is supplied as the visual reference.

Table 5: Text-only vs. image-conditioned evaluation on the official test split. For the same method, image conditioning notably improves edit locality and reconstruction quality while faithfulness remains stable.

Method	Condition	CLIP-T ↑	DINO-I ↑	Preserve CD ↓	mLPIPS ↓	Region-F1 ↑
VoxHammer [14]	Text-only	0.249	0.700	0.224	0.293	0.130
VoxHammer [14]	Image-cond.	0.250	0.662	0.221	0.202	0.321
Native3DEditing [3]	Text-only	0.253	0.622	0.970	0.283	0.165
3DEditVerse [24]	Image-cond.	0.336	0.667	0.807	0.245	0.176

Table 6: Unified annotation fields in Omni3DEdit. Optional auxiliary tags depend on the subset, but edited-region supervision is the common benchmark signal.

Field	Description
`sample_id`	Unique sample identifier.
`source_asset`, `target_asset`	File paths or asset identifiers for source and target GLB models.
`instruction`	Natural-language edit instruction.
`edited_region`	Mesh-space edited region and projected view masks used for locality/preservation evaluation.
`edit_operation`	Edit operation type (e.g., add, remove, open, rotate, deform, recolor, restyle), from which the high-level edit family can be inferred.
`category`	Object category or asset group.
`provenance`	Source dataset, generation pipeline, redistribution status, and licensing metadata.

Evaluation protocol. All image-space metrics are computed over the full set of rendered views for each test sample. For CLIP-T, we use the ViT-B/32 CLIP encoder [18]. For DINO-I, we use the ViT-S/16 DINO model [4]. For LPIPS, we use the AlexNet backbone [28]. Preserve CD is computed by uniformly sampling points from mesh faces in the non-edited region. Detailed metric formulations are provided in Appendix F.

E Qualitative Comparison

Figure 3 presents representative examples across five major task families: part addition/deletion, articulation editing, pose editing, part-localized modification, and material editing. The visualizations confirm that the edited-region annotations correspond to semantically meaningful changes. Furthermore, they reveal distinct failure modes in existing baselines, which is consistent with the quantitative results reported in the main paper. Specifically, we observe two primary limitations in current methods.

First, existing baselines demonstrate a **failure to preserve unedited regions**. Because they lack local 3D awareness, localized edits often cause catastrophic global disruptions. For example, Native3DEditing suffers severe structural collapse in the pose editing task. Similarly, localized modifications frequently cause the unintended loss of non-edited parts—such as 3DEditVerse [24] deleting almost the entire chair body when asked to remove an armrest, or VoxHammer (image) losing the lamp stand during part-localized texture modification.

Second, baselines suffer from **failed edits and severe deformation**. They struggle with precise spatial instructions, resulting in ignored edits (e.g., VoxHammer (text) completely failing to remove the armrest) or severe geometric distortion. Part articulation proves especially challenging; attempting to open a cabinet door yields floating artifacts and edge deformations in Native3DEditing and 3DEditVerse, while both VoxHammer variants completely collapse the geometry into unrecognizable structures.

F Metric Definitions

We provide formal definitions of all evaluation metrics used in our benchmark. Let $\hat{\mathcal{A}}$ denote the edited prediction, $\mathcal{A}^{\text{tgt}}$ the ground-truth target, $\mathcal{A}^{\text{src}}$ the source asset, and T the editing instruction. All image-space metrics are averaged over a fixed set of V rendered views.

Instruction faithfulness. CLIP-T measures the semantic agreement between the editing instruction and the predicted result:

$$\text{CLIP-T} = \frac{1}{V}\sum_{v=1}^{V} \cos\big(\phi_{\text{text}}(T),\ \phi_{\text{img}}(I^v_{\hat{\mathcal{A}}})\big), \tag{2}$$

where ϕ_{text} and ϕ_{img} are the CLIP [18] text and image encoders, and $I^v_{\hat{\mathcal{A}}}$ is the v-th rendered view of the prediction. DINO-I measures the visual similarity between prediction and ground-truth target renderings:

$$\text{DINO-I} = \frac{1}{V}\sum_{v=1}^{V} \cos\big(\psi(I^v_{\hat{\mathcal{A}}}),\ \psi(I^v_{\mathcal{A}^{\text{tgt}}})\big), \tag{3}$$

where ψ denotes the DINO [4] `[CLS]` feature extractor.

Preservation. Preservation metrics quantify the integrity of the non-edited region. Let $\bar{R}$ denote the complement of the annotated edited region. Preserve CD is the Chamfer Distance computed on point clouds sampled exclusively from $\bar{R}$:

$$\text{Preserve CD} = \text{CD}\big(\mathcal{P}_{\bar{R}}(\hat{\mathcal{A}}),\ \mathcal{P}_{\bar{R}}(\mathcal{A}^{\text{tgt}})\big), \tag{4}$$

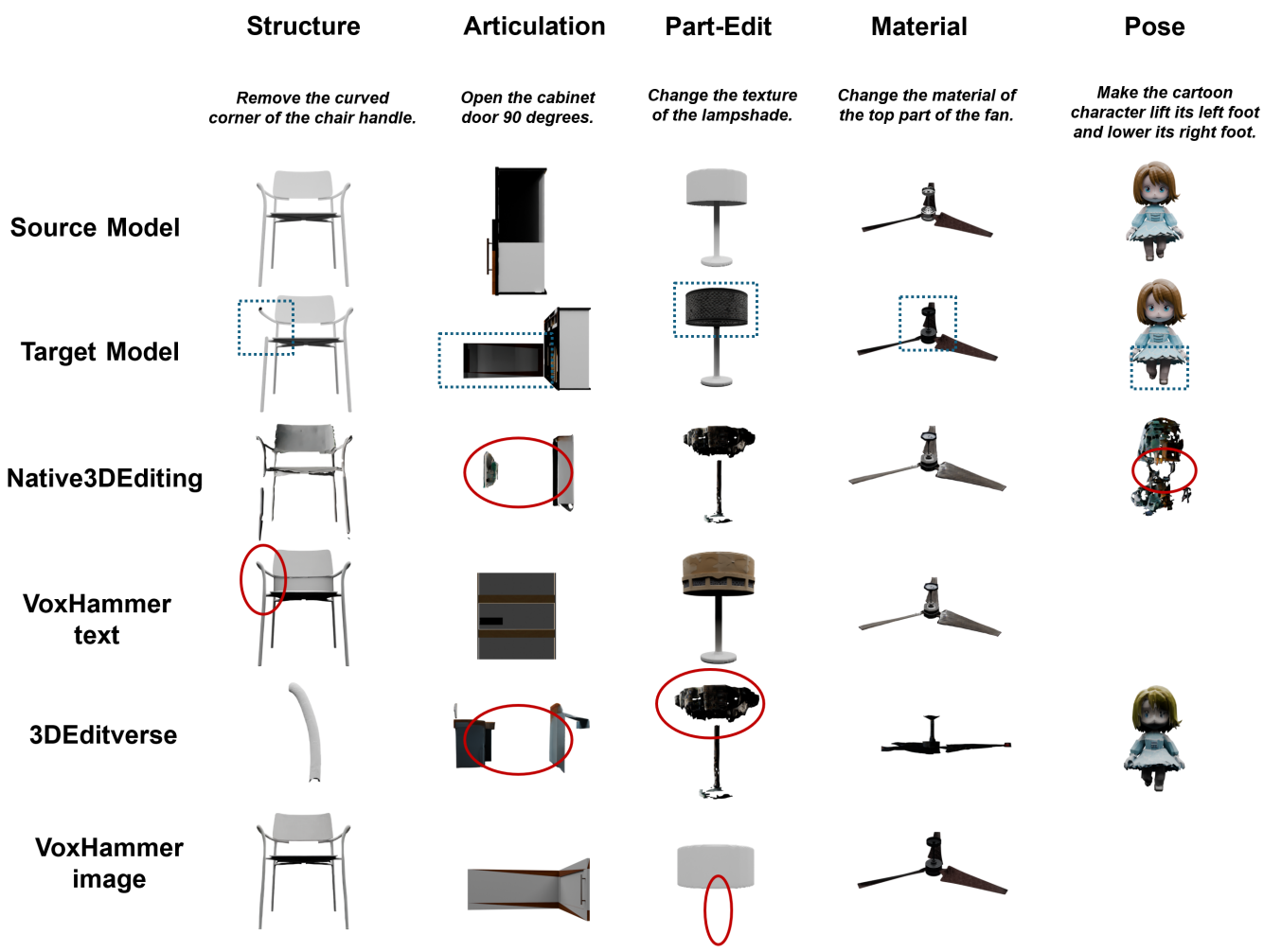

Figure 3: Qualitative comparison of 3D editing methods across the five task families of Omni3DEdit. The blue dashed boxes highlight the intended edited regions on the target models, while the red solid circles highlight representative failure modes of existing baselines. We observe that current methods frequently struggle with preservation (e.g., severe structural collapse or loss of unedited parts) and faithful execution of geometric manipulations (e.g., floating artifacts or ignored edits).

where $\mathcal{P}_{\bar{R}}(\cdot)$ uniformly samples points from mesh faces outside the edited region. For the Pose subset, which lacks localized edited-region annotations, $\bar{R}$ defaults to the full mesh.

In image space, we compute masked PSNR, SSIM [21], and LPIPS [28] (denoted mPSNR, mSSIM, mLPIPS) between prediction and target renderings, restricted to the projected non-edited mask $M^v_{\bar{R}}$:

$$\text{mLPIPS} = \frac{1}{V}\sum_{v=1}^{V} \text{LPIPS}\big(I^v_{\hat{\mathcal{A}}} \odot M^v_{\bar{R}},\ I^v_{\mathcal{A}^{\text{tgt}}} \odot M^v_{\bar{R}}\big). \tag{5}$$

mPSNR and mSSIM are computed analogously.

Edit locality. We derive a per-view binary change map C^v by thresholding the per-pixel ℓ_1 difference between source and prediction renderings:

$$C^v(p) = \mathbf{1}\Big[\frac{1}{3}\|I^v_{\hat{\mathcal{A}}}(p) - I^v_{\mathcal{A}^{\text{src}}}(p)\|_1 > \tau\Big], \tag{6}$$

and compare it with the projected ground-truth edited-region mask M^v_R via pixel-level precision, recall, and F1:

$$\text{Region-F1} = \frac{1}{V}\sum_{v=1}^{V} \text{F1}(C^v,\ M^v_R). \tag{7}$$

Region-F1 is our primary locality metric. High precision indicates that changes are confined to the annotated region (no over-editing); high recall indicates that the annotated region is indeed modified

(no under-editing). To extract the predicted change mask C^v, we calculate the mean absolute difference across RGB channels between the rendered source and edited images, binarizing it at a threshold of $\tau = 0.05$.

G Discussion

The geometry–appearance gap. Our results reveal a consistent difficulty gap between appearance-only edits and geometry-altering edits. On the Material subset, VoxHammer achieves a Region-F1 of 0.475, demonstrating that mask-constrained editing can effectively localize appearance changes. In contrast, both baselines produce near-zero Region-F1 on the Structure subset and incur high perceptual error on Articulation and Part-Edit. We hypothesize that this gap stems from a fundamental asymmetry: appearance edits modify texture or material properties while leaving the underlying mesh untouched, whereas geometric edits require adding, removing, or deforming mesh elements—operations that current native 3D representations handle less reliably, especially when the edit must remain spatially confined. Bridging this gap likely requires advances in 3D representations that support fine-grained, region-aware geometric manipulation.

Region annotations as supervision for future methods. The explicit edited-region annotations in Omni3DEdit serve not only as evaluation signals but also as potential training supervision. Current methods either ignore locality entirely (Native3DEditing) or rely on user-provided masks at inference time (VoxHammer). A promising middle ground would be to train models that *learn* to predict the edit region from the instruction alone, using annotated regions as ground truth. Such region-aware training could improve both locality (by confining changes to the predicted region) and preservation (by explicitly protecting the complement), and our benchmark provides the large-scale paired data needed to explore this direction.

Extensibility. Omni3DEdit currently covers five edit families, but the unified schema and annotation pipeline are designed to be extensible. For example, topology changes (e.g., splitting or merging parts), physical property edits (e.g., modifying mass or friction), and scene-level compositional edits can all be integrated by defining the corresponding programmatic transformation and edited-region extraction. We hope the standardized infrastructure encourages the community to progressively broaden the benchmark's coverage as 3D editing methods continue to diversify.

H Limitations

The source–target pairs are largely constructed through programmatic transformations or foundation-model-based generation, so the instruction distribution may not fully reflect real user requests, and evaluation difficulty varies across subsets. The Pose subset does not provide localized edited-region annotations because pose transformations are inherently global, and the edited-region annotations do not encode richer semantic or kinematic structures. The benchmark may also inherit category imbalance and language artifacts from its upstream sources; we mitigate this through manual curation of the official test set.