SAMa: Material-Aware 3D Selection and Segmentation

Supplementary Material

In this supplemental document we provide additional details on training and implementation, as well as results that could not be included in the main text due to space restrictions. We strongly encourage the reader to view the videos in our supplemental HTML material for 3D selection visualizations, examples of our fine-tuning material dataset, and a video of our application GUI.

A. Implementation details

A.1. Fine-tuning

As mentioned in paper Sec. 3, we fine-tune parts of the SAM2 [10] model on material-specific video data. For all our experiments, we use the model in its "large" configuration, employing the Hiera [11] image encoder with ca. 212M params, which yielded the best results in our experiments.

As the original SA-V [10] dataset, we encode our video dataset as MP4 videos with 1024×1024 resolution and the annotations in CoCoRLE encoding for efficient storage.

Our video dataset sub-samples the video by skipping every other video frame to increase the intra-frame distance, and then randomly chooses sequences of six consecutive sub-sampled frames. For each material and each frame, we sample a click. We do not select a material if it is barely visible in the frames, *i.e.*, if it occupies less than 0.02% of the frame (150 pixels). We erode the material's ground-truth mask before using it as a sampling mask, ensuring that the sampled click is at least four pixels away from the material's border. We sample a positive click with 80% probability, and a negative click on a random other material with 20% and reverse the temporal order of the frame sequence with a chance of 50%. During the forward pass of the model, we use every other frame as a clicked frame and thus force the model to use its memory attention module to infer the selection for the intermediate, unclicked frames. Additionally, we make a random 50% choice between sampling the most salient material in the frame (with the highest number of annotated pixels) and any other material.

During training, we compute the per-frame loss on the model prediction and ground-truth annotation via the sum of two losses, a binary cross-entropy followed by a sigmoid (using the log-sum-exp [2] trick for numerical stability) and a sigmoid-normalized Dice loss [9] to account for the imbalance between (large) background and (smaller) material masks. We use the AdamW optimizer with weight decay 0.01 and learning rate 1×10^{-5}.

We additionally experiment with mixed video- and image-finetuning and find that the results perform roughly on-par with our video model when training on our video-dataset and 20% of the Materialistic [12] data set mixed in. For simplicity, all results in the main text therefore use solely our video-finetuned model.

A.2. kNN lookup

As explained in the main text, we perform k-nearest neighbour (kNN) lookup into our similarity point cloud to infer the material selection for new, unseen views. Here, we take advantage of modern, GPU-accelerated large-scale queries via the FAISS library [4, 6].

Specifically, we use the INDEXFLATL2 index for exact search w.r.t. the points' L_2 distance, encoded as an INDEXIVFFLAT for compactness, with 100 clusters, and push it to the GPU (a cluster is a representative subset of the data that can be traversed efficiently and narrows down the search region during later query operations). This index, as mentioned in the main text, must be re-constructed after each new click, since the initial camera from which the click was performed will add to, and therefore change, the similarity point cloud. This re-construction takes around 0.5 seconds (all timings, including those in the main text, are reported on a single NVIDIA 40GB A100).

Once the index is built, we visit five clusters during the search for the top-k nearest neighbors. We found this number of visited clusters to be a hyperparameter which, even with the lowest setting of a single cluster, does not significantly deteriorate performance since the point cloud is relatively dense. We show an example of a typical point cloud in Figure 9. Thanks to the point cloud density our selection handles sharp edges well.

A.3. Camera subsampling

To infer the 2D similarities which will later be projected to 3D, we need to sub-sample a set of cameras that cover the object well. Recall that we fine-tuned SAM2 with our material-centric video dataset using four different camera trajectories, all with smooth view progression, while now we additionally need to ensure maximum object coverage. For NeRFs and 3D Gaussians, we thus sub-sample 20% of the training views, for meshes we use spherical Fibonacci sampling with 30 sampled cameras. Once we have sub-sampled the cameras, we need to sort them into a coherent, smooth trajectory to enable our video model to keep temporal consistency between the frames. We use a greedy iterative search to achieve a smooth trajectory from the initial camera, as detailed in Algorithm 1.

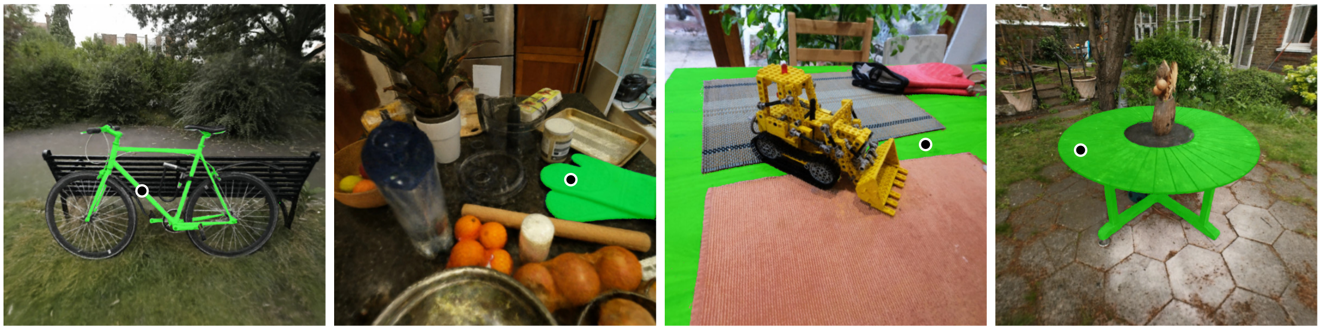


Figure 1. Selection results on real-world scenes from the MIPNeRF360 dataset [1].

A.4. Dataset details

To construct our fine-tuning dataset, we procedurally generate short multi-object videos using randomly picked objects from a subset of 9,082 textured Substance 3D objects. For each video, we randomly sample at least two objects and place them into a shared scene. Objects are randomly displaced by up to half of their bounding box extent to reduce large spatial overlap. Objects consisting of a single material are excluded, as they do not provide meaningful supervision for material selection.

We assign materials by sampling from a library of 29,472 Substance material maps, including multiple realizations of the same base material (*e.g*., different variants of the same type of wood). Materials are assigned to object parts such that they appear at least twice within a scene to ensure sufficient positive supervision and disambiguate from object selection. Material assignments remain consistent throughout a video but are resampled independently across videos. We generate dense per-pixel material annotations using integer indices, reserving the label 0 for the background.

Rendering is performed in Blender 4.3, using the Principled BSDF shader for all materials. To illuminate the scene, we randomly select an HDR environment map from a set of 420 HDRIs sourced from PolyHaven. In order to reduce the domain gap between our synthetic and real-world videos, we alpha-composite the rendered objects onto their envmap background.

We generate videos using four camera motion patterns: turntable, flyover, zoom-in, and zoom-out. Spherical, fixed-radius turntable trajectories are sampled with 33% probability since they represent the dominant camera trajectories (after sorting) that are used during SAMa's inference phase. Flyover trajectories are also sampled with 33% probability, as they resemble realistic camera motion and are most similar to those found in the SA-V [10] dataset. Zoom-in trajectories are sampled with 22% probability since during the zoom-in phase, more detail becomes visible, and zoom-out trajectories account for the remaining 12%. We leave experimenting with this percentages and more camera trajectories to future work.

B. Additional quantitative results

We here report a more detailed, per-scene evaluation of the metrics reported in the main text. The per-scene measurements for robustness and multiview-consistency are in Tab. 2 and Tab. 3, respectively.

Additionally, we report the per-scene selection accuracy as mean intersection over union (mIoU) and F1 scores. F1 is more robust than precision or recall alone, since either individual metric can easily be gamed by failure cases. Precision quantifies the relevance of the selected data (when the model says material A, is it really material A?), and can therefore easily be cheated by simply selecting a small amount of high-confidence elements (*e.g*., in our case, just the clicked pixel). Recall quantifies the amount of returned relevant data (when there is material A, how much of it does the model find?), and can easily be deceived by always selecting all the elements (*e.g*., in our case, a mask full of 1's). We show both mIoU and F1, computed on the NeRF-, MIPNeRF360- and our dataset, in Tab. 4, Tab. 5 and Tab. 1, respectively. We perform the evaluation on 3D Gaussians for rendering speed. For the real-world scenes from the

Algorithm 1 Camera trajectory sorting, starting from an initial camera. CalcNorms calculates the spatio-angular distances between a given camera and all other cameras.

```
Input: initial camera $i$, other cameras $o$
Output: sorted cameras
procedure SampleCameraTrajectory
    curr ← $i$                                  ▷ set current camera
    sorted ← [curr]               ▷ initialize sorted cameras list
    while len($o$) > 0 do
        norms ← CalcNorms(curr, $o$)
        cidx ← argmin (norms)              ▷ closest to current
        sorted.append($o$ [cidx])
        curr ← $o$ [cidx]
        $o$ [cidx].pop()
    end while
    return sorted
end procedure
```

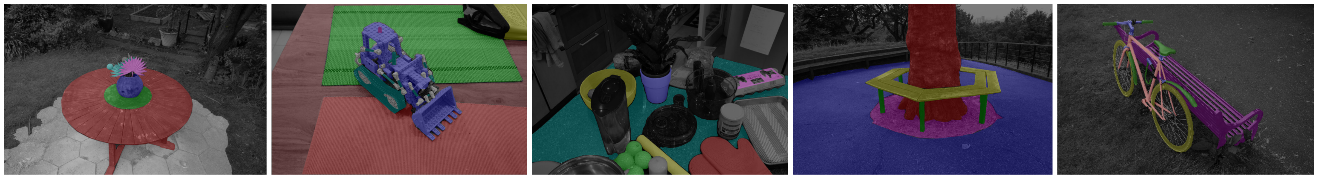


Figure 2. Exemplary visualizations of our annotated test frames from the MIPNeRF360 dataset [1].

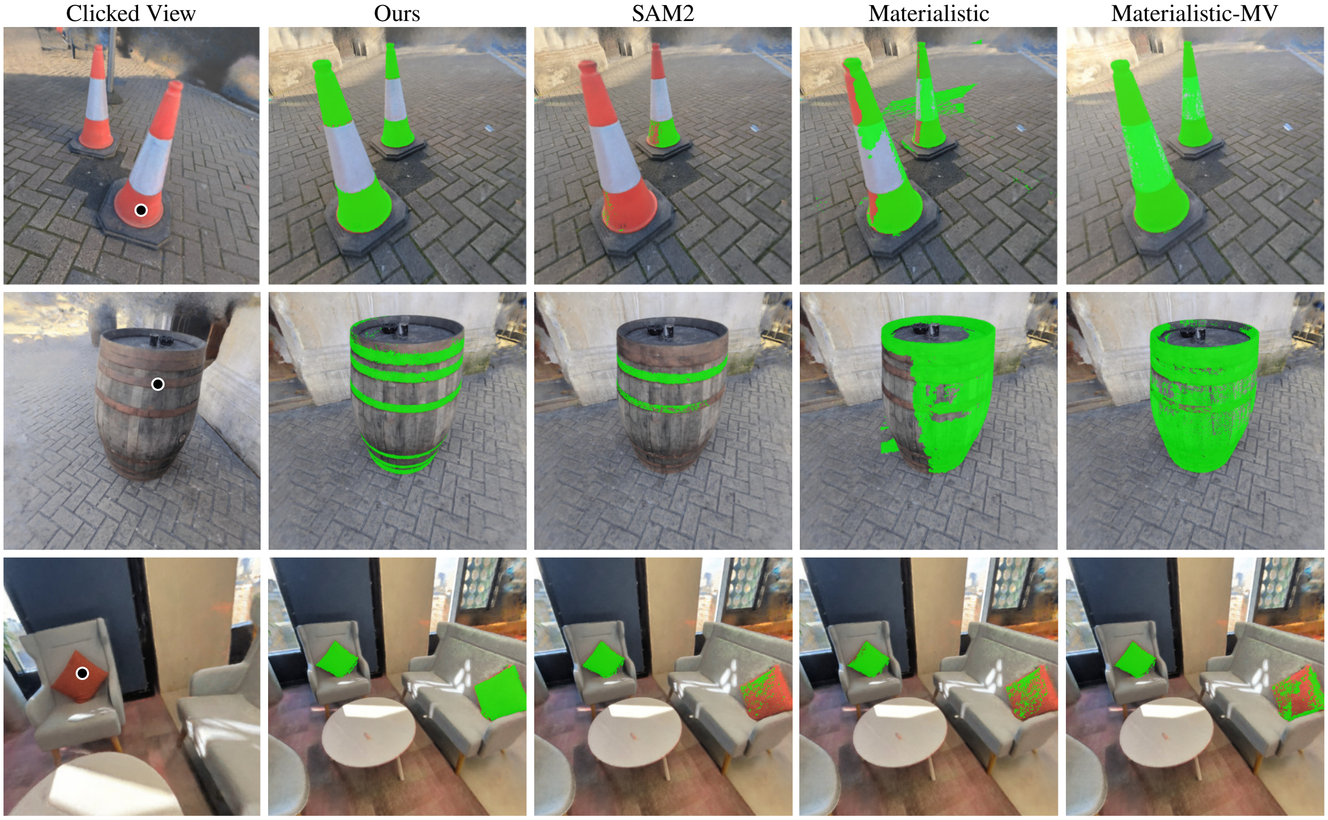


Figure 3. We test our method and competitors on 3DGS scenes we capture and find that it generalizes and performs considerably better than existing approaches. The first column shows the clicked view, the subsequent columns show the selection result from a novel, unseen view.

MIPNeRF dataset, we found the Gaussian's depth to not be sufficiently accurate and therefore use NeRFacto [13].

The quantitative evaluation confirms our qualitative findings: our method consistently performs well for the task of material selection, beating the other baselines in the majority of cases. In select cases, for instance the MIC scene from the NeRF dataset (see Tab. 4), SAM2 wins in terms of selection accuracy, since the materials of the object are visually indistinguishable from one another and applied to the object's subparts, which have a tendency to be selected by SAM2. Both Materialistic-based baselines under-perform in all experiments. This can be attributed in part to the fact that they are not multiview consistent, but, equally important, to the fact that the underlying model generally attends to coarser structures (due to the different ViT patchsizes, see Fig. 4) and is not sufficiently sensitive to object (sub-)parts.

C. Additional qualitative results

We show additional examples of recoloring NeRFs based on our material-selection in Fig. 7.

We show examples of our hand-annotated frames from the MIPNeRF dataset which we used for evaluation in Fig. 2. Additionally, we show examples of material selection on real-world scenes from these MIPNeRF360 scenes [1] in Fig. 1.

As claimed in the main text, our frame duplication strategy not only improves SAMa's predictions, but also helps to improve prediction confidence on the original SAM2 architecture, which we visualize in Fig. 6.

To add to our robustness evaluation, we show a qualitative example of how robust the methods are to different clicks on the same material in Fig. 5.

We also show the 2D material selection accuracy for

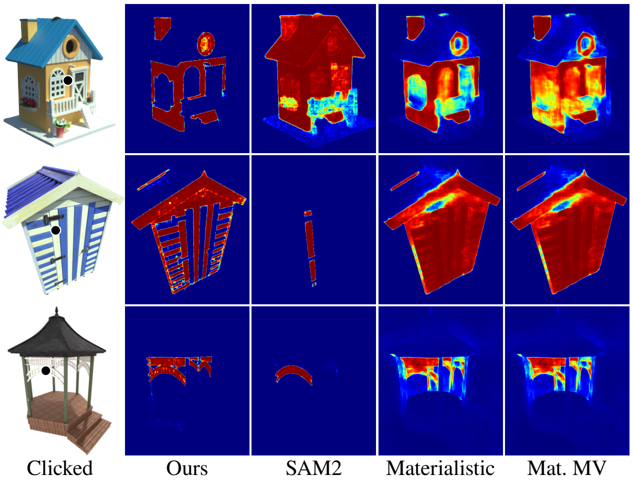


Figure 4. 2D selection results of the different methods for various models. We do not perform any point cloud lookup or novel view inference, the shown heatmap is obtained by directly feeding the clicked frame to the model.

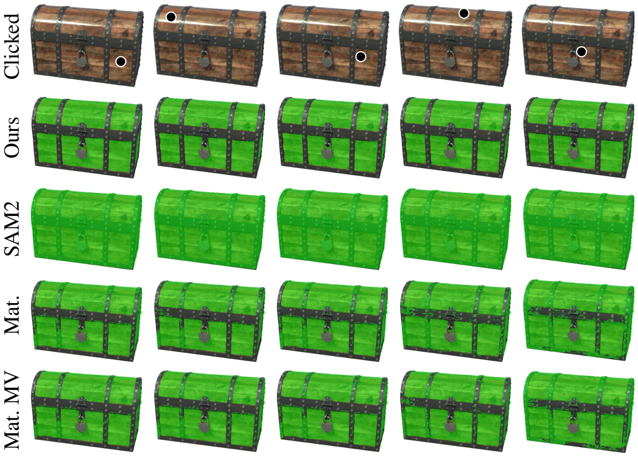


Figure 5. Robustness of the different approaches (rows) for clicks on different locations of the same material (columns).

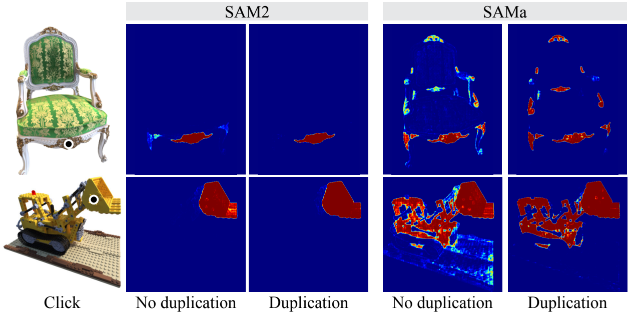


Figure 6. The effects of our frame duplication strategy translate from our SAMa model to the original SAM2 model.

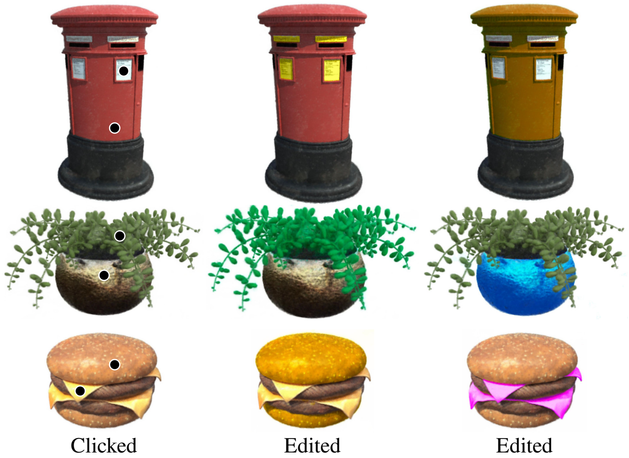


Figure 7. Additional examples of editing the NeRF's color based on the user's selected material.

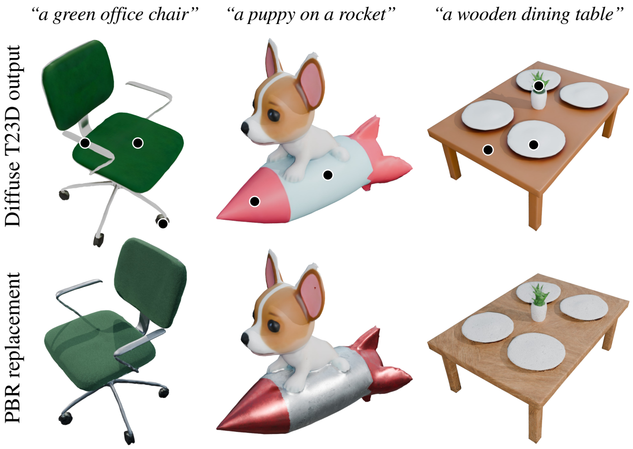


Figure 8. Our method can be used to select, and subsequently replace, the diffuse-only materials commonly found on text-to-3D [14] pipeline output meshes with PBR materials.

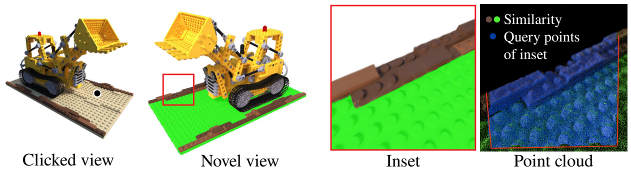


Figure 9. Visualization of the selection point cloud (pre-computed by SAMa, right subfigure, green & brown) and the query points obtained from a novel view (used for the look-up into the selection point cloud). The red inset's query points are visualized in blue in the right sub-figure. Best viewed zoomed-in.

all models in Fig. 4. From this figure, it becomes evident that the SAM-based methods benefit significantly from the smaller patchsize of the image encoder: Hiera, the encoder used by the SAM2 architecture (Ours, SAM2) uses a four-times smaller patchsize of 4×4, whereas Materialistic-based methods employ DINO features, which use a patchsize of 8×8, resulting in blurrier edges. We would like to emphasize that the input resolution is the same for all models, 512p. Moreover, we observe that our model deals well with perspective distortion (middle row in Fig. 4) and low-contrast input (bottom row in Fig. 4). Finally, we show thumbnail renderings of our synthetic dataset in Fig. 15.

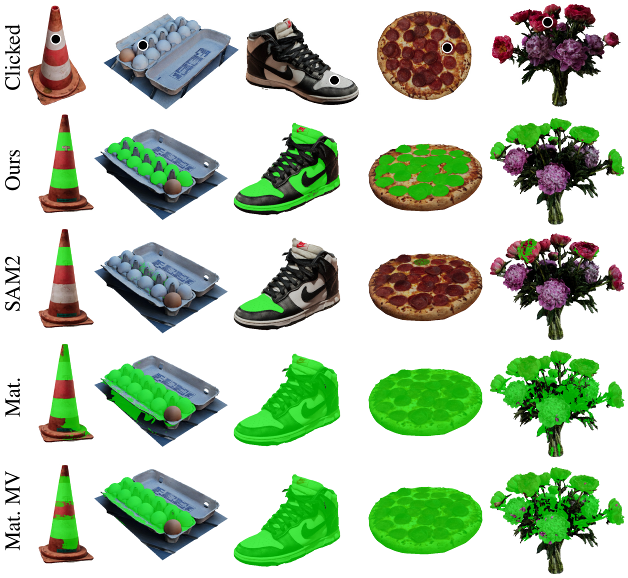
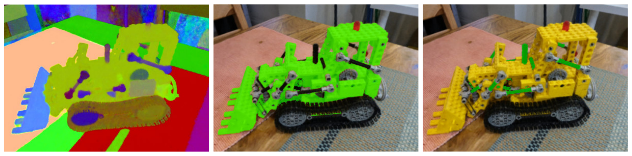


Figure 10. We compare our method and competitors on real mesh data that was obtained by using photogrammetry on real-world objects with Polycam. For animated version of these results, see the electronic supplemental.

D. Additional comparisons

We qualitatively compare our method against other 3D-aware selection and segmentation methods. Note that a full quantitative comparison against these baselines is not feasible due to their long per-asset optimization times.

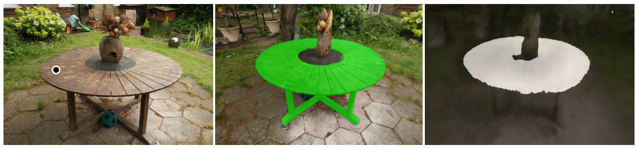
Selection. In Fig. 11, we show a comparison against Garfield [7], which requires asset-specific pre-training and does not target materials. The same holds for Feature-3DGS [15] (right side in Fig. 14). In contrast, our approach works with arbitrary assets and *without* asset-specific pre-training, as it merely needs to render the existing 3D asset to images and back-project the obtained similarity values. Our times from click-to-selection are therefore around three orders of magnitude lower. Finally, the comparison to SA3D [3] in Figure 12 shows that our approach correctly selects materials for NeRFs as well, at much lower runtimes than SA3D.



Garfield, ca. 35min Ours, ca. 2s Ours, ca. 2s

Figure 11. Comparison to Garfield [7], which cannot be run without asset-specific pre-training and does not target material selection.

Segmentation. Additionally, in Fig. 14, we show a comparison against SemanticGaussians [5] (left side), who inject semantic knowledge into a 3DGS capture via a separately trained network, but whose predictions are coarse, not material-aware and, by design, limited to 3DGS assets. Similarly, Fig. 13 shows an evaluation of MaterialSeg3D [8], a 3D segmentation method for materials that works on meshes. While this methods works well and is multiview-consistent, it classifies materials into 14 predefined semantic classes (metal, wood, plastic, ...) and thus is not able to distinguish between different materials within a category, such as the different types of wood on the beach hut. Our method, in contrast, selects and segments these materials correctly.



Clicked view SAMa, ca. 2 seconds SA3D, ca. 10 minutes

Figure 12. Comparison against SA3D. SAMa quickly selects the material, while SA3D, after lengthy pre-optimization, selects object (sub-) parts.

In summary, SAMa performs well on material-aware selection and segmentation in 3D, works on NeRFs, 3DGS and meshes, provides interactive click-to-selection and can infer selection results for novel views in real-time.

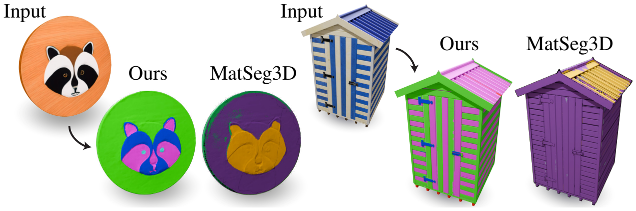
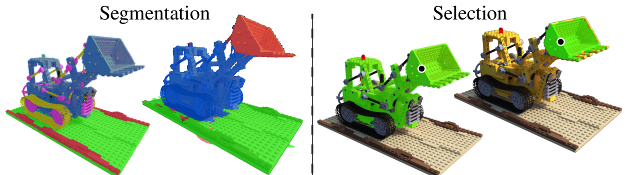


Figure 13. Comparison between the material-aware segmentation of SAMa and the semantic segmentation of MaterialSeg3D [8]



Ours ca. 15 seconds **SemanticGaussians** ca. 2.0 min **Ours** ca. 2 seconds **Feature-3DGS** ca. 15.5 min

Figure 14. Performance and timing of 2D-to-3D lifting methods. SemanticGaussians [5] does not target materials, while Feature-3DGS [15] requires a lengthy pre-optimization.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 2, 3

	WChair		Coffee		Perfume		Chest		Couch		Bike		Hut		Burger		Plant		Postbox		Car		Pooltable	
	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1
Ours	0.73	0.84	0.91	0.95	0.92	0.96	0.82	0.90	0.41	0.56	0.71	0.83	0.79	0.88	0.89	0.94	0.79	0.88	0.94	0.97	0.34	0.51	0.57	0.73
SAM2	0.44	0.60	0.47	0.64	0.91	0.96	0.41	0.56	0.09	0.16	0.22	0.36	0.17	0.25	0.57	0.72	0.26	0.41	0.69	0.81	0.06	0.11	0.12	0.21
Materialistic	0.51	0.67	0.51	0.68	0.81	0.89	0.46	0.61	0.18	0.30	0.63	0.77	0.44	0.61	0.25	0.40	0.74	0.85	0.91	0.95	0.11	0.19	0.15	0.26
Materialistic MV	0.61	0.75	0.48	0.64	0.89	0.94	0.51	0.65	0.17	0.29	0.57	0.73	0.52	0.69	0.53	0.67	0.75	0.86	0.94	0.97	0.10	0.18	0.25	0.40

Table 1. Per-scene metrics on our synthetic dataset for the different scenes (columns) and methods (rows). Higher is better.

	Lego	Hotdog	Ship	Ficus	Mic	Drums	Materials	Chair	Garden	Kitchen	Counter	Treehill	Bicycle
Ours	0.21	1.01	1.68	0.45	2.89	0.85	1.47	0.25	0.38	0.19	1.25	2.79	1.22
SAM2	0.43	1.04	2.55	0.46	1.75	0.43	0.33	3.03	0.76	4.41	0.22	1.51	7.68
Materialistic	4.33	2.69	2.62	2.40	3.10	2.48	3.69	3.88	10.24	6.16	13.51	4.83	5.90
Materialistic MV	7.37	3.17	2.33	2.99	4.38	2.51	3.67	4.82	3.27	2.35	5.04	4.47	2.52

	WChair	Coffee	Perfume	Chest	Couch	Bike	Hut	Burger	Plant	Postbox	Car	Pooltable
Ours	0.06	0.09	0.01	0.12	0.60	0.95	0.87	0.04	0.61	0.15	0.43	0.15
SAM2	0.10	0.01	0.51	0.02	0.34	0.45	2.25	0.60	0.02	3.31	1.17	0.05
Materialistic	0.42	0.73	0.50	1.28	4.80	2.53	2.63	1.16	0.60	0.71	1.86	4.88
Materialistic MV	0.19	0.85	0.61	2.30	4.83	2.34	5.46	2.97	2.35	0.68	1.25	1.95

Table 2. Per-scene (columns) breakdown of our robustness evaluation metric for all methods (rows) from the main text. Lower is better. The NeRF- and MIPNeRF360-scenes are in the top sub-table, our custom scenes in the bottom sub-table. This only evaluates the robustness and not whether the selection is correct.

	Lego	Hotdog	Ship	Ficus	Mic	Drums	Materials	Chair	Garden	Kitchen	Counter	Treehill	Bicycle
Ours	0.91	1.20	2.20	0.30	5.64	0.62	5.77	0.85	0.18	0.72	0.87	1.32	3.71
SAM2	2.99	1.44	3.03	0.37	1.46	0.94	1.54	6.13	0.28	1.04	0.40	1.43	2.77
Materialistic	5.18	4.57	7.98	0.62	4.59	1.77	12.59	6.56	2.59	4.40	2.29	9.06	5.92
Materialistic MV	2.79	4.87	2.97	0.45	4.26	1.00	8.58	6.32	2.15	2.11	0.89	9.00	6.36

	WChair	Coffee	Perfume	Chest	Couch	Bike	Hut	Burger	Plant	Postbox	Car	Pooltable
Ours	0.89	0.31	0.26	1.16	1.05	1.17	3.46	0.32	0.61	0.69	1.61	9.03
SAM2	1.60	1.61	0.47	4.03	1.98	5.58	16.63	1.54	0.68	2.69	2.88	20.95
Materialistic	3.12	2.72	0.73	8.12	3.61	2.92	13.27	7.32	2.33	0.93	13.79	10.66
Materialistic MV	1.78	3.11	0.30	6.79	2.08	2.32	11.04	4.07	1.90	0.61	16.89	5.40

Table 3. Per-scene (columns) breakdown of our multiview-consistency evaluation metric for all methods (rows) from the main text. Lower is better. The NeRF- and MIPNeRF360-scenes are in the top sub-table, our custom scenes in the bottom sub-table. This only evaluates the multiview-consistency and not whether the selection is correct.

	Lego		Hotdog		Ship		Ficus		Mic		Drums		Materials		Chair	
	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1
Ours	0.78	0.87	0.87	0.93	0.06	0.12	0.68	0.81	0.24	0.39	0.25	0.39	0.16	0.27	0.76	0.87
SAM2	0.05	0.09	0.77	0.87	0.10	0.18	0.68	0.81	0.51	0.68	0.07	0.14	0.10	0.18	0.35	0.52
Materialistic	0.22	0.36	0.17	0.29	0.10	0.17	0.63	0.77	0.19	0.32	0.18	0.31	0.12	0.21	0.30	0.46
Materialistic MV	0.42	0.36	0.23	0.37	0.08	0.15	0.64	0.78	0.17	0.29	0.19	0.13	0.14	0.22	0.32	0.29

Table 4. Per-scene metrics on the NeRF datasets for the different scenes (columns) and methods (rows). Higher is better.

	Garden		Kitchen		Counter		Treehill		Bicycle	
	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1	mIoU	F1
Ours	0.85	0.92	0.85	0.92	0.74	0.85	0.30	0.46	0.27	0.43
SAM2	0.70	0.82	0.62	0.76	0.65	0.79	0.34	0.50	0.22	0.36
Materialistic	0.34	0.49	0.65	0.79	0.27	0.43	0.16	0.28	0.13	0.23
Materialistic MV	0.13	0.28	0.75	0.86	0.34	0.56	0.25	0.37	0.15	0.25

Table 5. Per-scene metrics on our hand-annotated images from the MIPNeRF360 dataset for the different scenes (columns) and methods (rows). For both metrics, higher is better.

[2] Christopher M Bishop. Pattern recognition and machine learning. *Springer google schola*, 2:1122–1128, 2006. 1

[3] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. In *NeurIPS*, 2023. 5

[4] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv*, 2024. 1

[5] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting. *arXiv preprint arXiv:2403.15624*, 2024. 5

[6] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on*

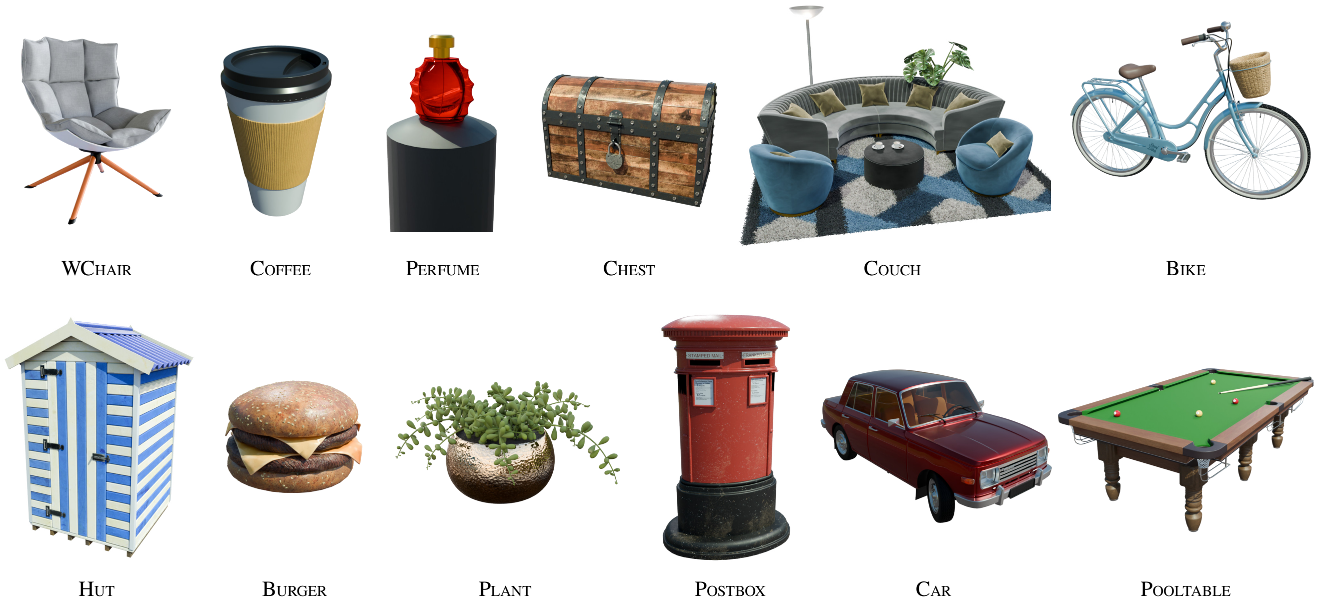


Figure 15. Our dataset of synthetic objects. Each object has dense material annotations.

Big Data, 7(3):535–547, 2019. 1

[7] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. GARField: Group Anything with Radiance Fields, 2024. arXiv:2401.09419 [cs]. 5

[8] Zeyu Li, Ruitong Gan, Chuanchen Luo, Yuxi Wang, Jiaheng Liu, Ziwei Zhu, Qing Li, Xucheng Yin, Man Zhang, Zhaoxiang Zhang, et al. Materialseg3d: Segmenting dense materials from 2d priors for 3d assets. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 370–379, 2024. 5

[9] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016. 1

[10] 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 preprint arXiv:2408.00714*, 2024. 1, 2

[11] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *International Conference on Machine Learning*, pages 29441–29454. PMLR, 2023. 1

[12] Prafull Sharma, Julien Philip, Michaël Gharbi, Bill Freeman, Fredo Durand, and Valentin Deschaintre. Materialistic: Selecting similar materials in images. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023. 1

[13] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 3

[14] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrm: Large reconstruction model for high-quality mesh. *arXiv preprint arXiv:2404.12385*, 2024. 4

[15] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 5