

# Improved Convex Decomposition with Ensembling and Negative Primitives

## Supplementary Material

### 7. Optimizing the Inference Pipeline

Given the computational cost of ensembling, we seek to maximize throughput of our inference pipeline. We use `torch.jit` and pure BFloat16 for encoding the RGBD image and finetuning. We also get speedups from batching the test images instead of one at a time. Combined with our subsampling strategy, these improvements yield over an order of magnitude faster inference than prior work, making ensembling more practical (see Table 1).

Since our primitives are the blended union of half-spaces [9], they cannot be rasterized easily and raymarching the SDF is required. We note that rendering the primitives still requires FP32 precision to avoid unwanted artifacts. We accelerate our raymarcher by advancing the step size by  $0.8 \cdot \text{SDF}$  if it is greater than the minimum step size (we use 0.001 for large-scale metrics gathering, 0.0001 for beauty renders). We cannot advance by the full SDF because it is an approximation of how far the smoothed primitive boundary is. We apply interval halving at the intersection point to refine the estimate.

### 8. Negative Primitives Theory

Set differencing can result in very efficient representations. Qualitative evidence was shown in Fig 2b, in which we model a cube with a hole punched in it. Intuitively, one positive and one negative primitive are sufficient to model it perfectly (2 total primitives). Without CSG, approx. 5 primitives may be required, which is less parameter-efficient. Based on that, we can sketch a theoretical argument as to why having a vocabulary of mixed positive and negative primitives is expected to yield more accurate representations than the same number of positive-only primitives.

#### 8.1. CSG Representational Efficiency

It is known that, for a CSG model in 3D with  $n$  distinct faces in the CSG tree, the resulting object model can have  $O(n^3)$  faces [57]. Relatively little appears to be known about the effect of the number of negative primitives on the complexity. We show that, under the circumstances that apply here, there are geometries that admit short descriptions using negative primitives and have very much longer descriptions when only unions are allowed. The bound is obtained by reasoning about what is required to encode the area of an object.

For a shape  $S$ , let  $K_+(S)$  be the minimum description length using only positive primitives, and  $K_\pm(S)$  be the minimum description length using mixed primitives. We claim that, under the circumstances that apply here, there

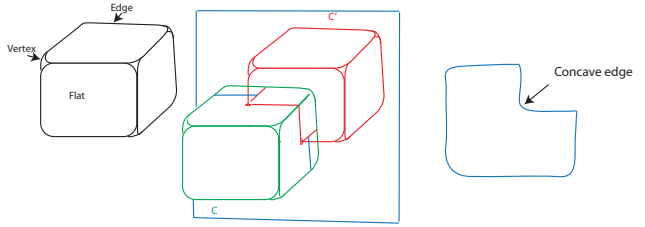


Figure 9. A shape which cannot be efficiently encoded as a union of convexes can be built out of a smoothed unit cube (on the left, showing flat regions, edge regions and vertex regions). Center sketches  $C$  (green) and  $C'$  (red), and a plane slicing the shape  $C - C'$  (blue). Right sketches the cross section cut by the plane: note the concave region, resulting from the smoothed convex edge region of  $C'$ . This requires at least  $O(1/\delta)$  convexes to approximate to precision  $\delta$ .

are shapes such that  $K_\pm(S) \ll K_+(S)$ . The construction is straightforward.

**Preamble:** Primitives are smoothed polytopes as described in [9]. There is some (very large) finite bound on the number of faces. These primitives are convex by construction (so gaussian curvature is non-negative), but are smooth. The surface of these primitives consists of flat regions (where both normal and gaussian curvature are very close to zero), edge regions, where gaussian curvature is close to zero but normal curvature may be large, and vertex regions, where there is considerable gaussian curvature. These regions correspond to 2-faces, 1-faces and 0-faces in the underlying polytope. The actual values of the curvature are not significant for our purposes.

**Theorem 2** (Restatement of Theorem 1 from main paper). *For the vocabulary of primitives described, there exist shapes  $S$  such that  $K_\pm(S) \ll K_+(S)$  when the representation of  $S$  is required to achieve sufficiently high precision.*

#### Sketch of Proof

We construct a shape with this property. Write  $C$  for an approximation of a unit cube, represented as a smoothed polytope, and  $C'$  for that cube translated by  $(0.5, 0.5, 0.5)$ . Now consider  $S = C - C'$  (Figure 9). Clearly,  $K_\pm(S)$  is  $O(1)$ . Consider  $K_+(S)$  as the resolution  $\delta$  of the representation increases. Note from Figure 9 that representing  $S$  requires representing three concave edge-like regions, and one concave vertex-like region. Since we can use only positive primitives which are convex, we must use the flat faces. It is straightforward that representing the concave edge-like regions to resolution  $\delta$  will require  $O(1/\delta)$  flats, and repre-

sending the concave vertex-like region to resolution  $\delta$  will require  $O(1/\delta^2)$  flats. It follows that

$$\lim_{\delta \rightarrow 0} \frac{K_{\pm}}{K_{+}} = 0$$

and so  $K_{\pm} \ll K_{+}$  in this case. Notice the underlying geometry of the effect is common – it takes a lot of convex shapes to represent concave cutouts. We expect that for “most shapes” the limit applies.

This theorem depends very delicately on the library of primitives and the transformations allowed to the primitives. In some cases, it is hopelessly *optimistic*. For example, if all primitives are cubes of two fixed sizes, where positives are large and negatives small, and transformations are purely Euclidean, it may not be possible to encode a set difference with positives only. We are not aware of bounds that take these effects into account.

## 8.2. Optimal Ratio of Negative Primitives in CSG Modeling

A simple model leads to an intriguing result. For a fixed budget of primitives, we aim to determine the optimal ratio of negative primitives ( $K^{-}$ ) to positive primitives ( $K^{+}$ ) that maximizes representational efficiency.

A CSG model can be described as:

$$\text{Object} = (P_1 \cup P_2 \cup \dots \cup P_{K^{+}}) - (N_1 \cup N_2 \cup \dots \cup N_{K^{-}}) \quad (2)$$

Where  $P_i$  are positive primitives and  $N_j$  are negative primitives, with  $K^{+} + K^{-} = K^{\text{total}}$ .

### Definitions:

1. *Primitive Interaction*: Overlapping volumes creating representational complexity
2. *PP Interaction*: Between two positive primitives
3. *PN Interaction*: Between a positive and negative primitive

### Assumptions:

1. Only positive volumes and their modifications by negative primitives are visible in the final result
2. The representational power comes primarily from *PP* and *PN* interactions
3. Primitives are distributed to maximize meaningful interactions
4. Optimal representation maximizes visible features per primitive used
5. Assume connected geometry

### Mathematical Model:

1. Number of *PP* Interactions:  $\binom{K^{+}}{2} = \frac{K^{+}(K^{+}-1)}{2}$
2. Number of *PN* Interactions:  $K^{+} \cdot K^{-}$

### Balancing *PP* and *PN* Interactions:

For optimal efficiency, *PP* and *PN* interactions should be balanced:

$$\frac{K^{+}(K^{+}-1)}{2} \approx K^{+} \cdot K^{-} \quad (3)$$

Substituting  $K^{+} = K^{\text{total}} - K^{-}$  and simplifying:

$$\frac{(K^{\text{total}} - K^{-})(K^{\text{total}} - K^{-} - 1)}{2} \approx (K^{\text{total}} - K^{-}) \cdot K^{-} \quad (4)$$

$$\frac{K^{\text{total}} - K^{-} - 1}{2} \approx K^{-} \quad (5)$$

For large  $K^{\text{total}}$ :

$$\frac{K^{\text{total}} - K^{-}}{2} \approx K^{-} \quad (6)$$

Solving:

$$K^{\text{total}} \approx 3K^{-} \quad (7)$$

$$K^{-} \approx \frac{K^{\text{total}}}{3} \quad (8)$$

Thus,

$$K^{+} \approx \frac{2K^{\text{total}}}{3} \quad (9)$$

**Verification:** With this ratio, both interaction types equal approximately  $\frac{2(K^{\text{total}})^2}{9}$ , confirming our balance criterion.

**Empirical Evidence and Practical Verification:** This result is intriguing, because it is consistent with our experimental observations. It should be noted that the assumptions may not be sound (the losses on primitives should tend to lead to fewer interactions between positive primitives than our model requires), but experimental results suggest quite strongly that the best results are obtained when  $K^{\text{total}}/K^{-}$  is about 3. We believe this is likely some form of formal geometric property, rather than a coincidence. In our experimentation, all ratios of  $K^{\text{total}}/K^{-}$  provide excellent primitive representations, showing that our underlying neural network, losses, and data pipeline are sound. But on average, the *best* representations were near  $K^{\text{total}}/K^{-} \approx 3 : 1$ . Observe on both LAION and NYUv2 how depth and segmentation metrics tend to be best when  $K^{\text{total}}/K^{-}$  are near 36/12, 24/8, and 12/4 (Tables 7, 8, 9).

**Limitations from Overlap Loss:** The theoretical model balances *PP* interactions, representing the formation of the base positive volume, with *PN* interactions, representing the carving of details. While the introduction of a loss encouraging positive primitives to spread out and avoid overlap alters the nature of *PP* interactions—shifting them from forming dense, complex unions to defining a more distributed positive scaffold—the fundamental requirement for balancing these two generative forces remains. The term  $\binom{K^{+}}{2}$  can thus be interpreted as the capacity of positive primitives to establish this initial, potentially dispersed, positive volume. Our quantitative evaluations, performed under these conditions, confirm that the optimal ratio of approximately  $K^{-} \approx K^{\text{total}}/3$  persists. This empirical result suggests

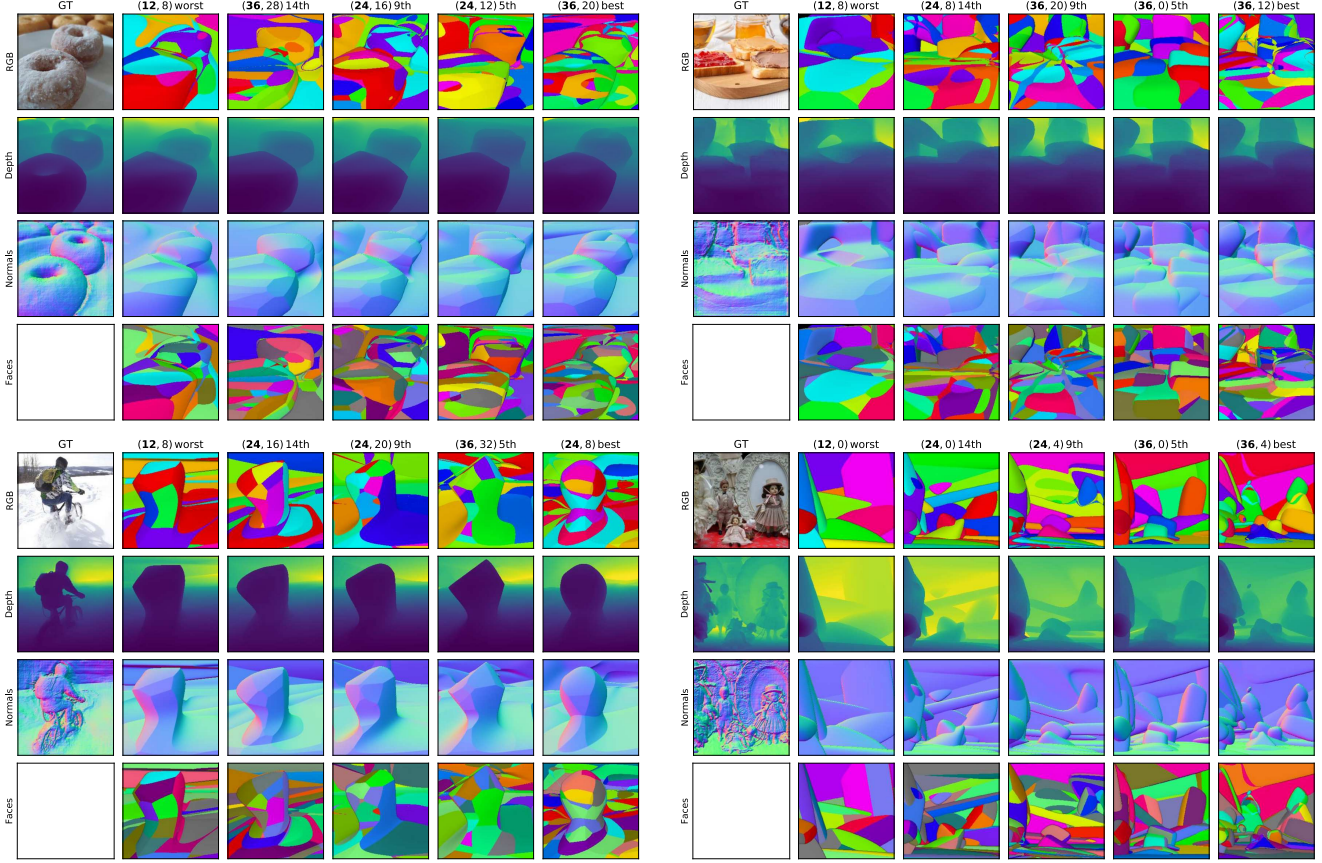


Figure 10. Additional qualitative examples shown.

that the derived balance point robustly maximizes representational efficiency by ensuring sufficient primitives for both establishing the foundational positive elements of the scene and for subsequently sculpting them with negative primitives, aligning with principles of diminishing returns and complementary information.

### 8.3. Face counts

In Sec. 3, we described how the number of faces scales with the number of negative primitives. When computing segmentation accuracy with negative primitives, we compute the triple  $(f_i, K_j^+, K_k^-)$  at each ray intersection point, where  $i$  is the face index,  $j$  is the index of the positive primitive we hit, and  $k$  is the index of the (potentially) negative primitive we hit. Each unique triple can get its own face label. Thus, given a fixed primitive budget  $K^{total}$ , replacing a pure positive primitive representation with a mixture of positives and negatives can yield more unique faces. For example,  $K^+/K^- = 12/0$  maxes out at  $12f$  unique faces;  $K^+/K^- = 6/6$  maxes out at  $42f$  faces. Note that  $f \times K^+ \times (1 + K^-)$  is the theoretical maximum of unique labels, as practical scenes do not involve every primitive

touching every other primitive.

## 9. Primitives by Descent Alone

We generate a large reservoir of 1M free-space (a.k.a. bbx samples) for each test image. We still generate  $H \times W$  “inside” surface samples and “outside” surface samples near the depth boundary respectively, with  $\epsilon = 0.02$  units separating these surface samples. We remind the reader that our point clouds are renormalized to approx. the unit cube during training to avoid scale issues. Then during finetuning, we subsample from all available samples at each step, providing a rich gradient analogous to the network training process (though here, we’re optimizing the parameters of primitives). We found subsampling 10% of available samples sufficient at each step.

Second, we find that vanilla SGD does not produce usable results; instead AdamW [35] was required. We set the initial LR to 0.01, and linearly warm up to it over the first 25% of iterations. We then halve the learning rate once at 50% of the steps and again at 75%.



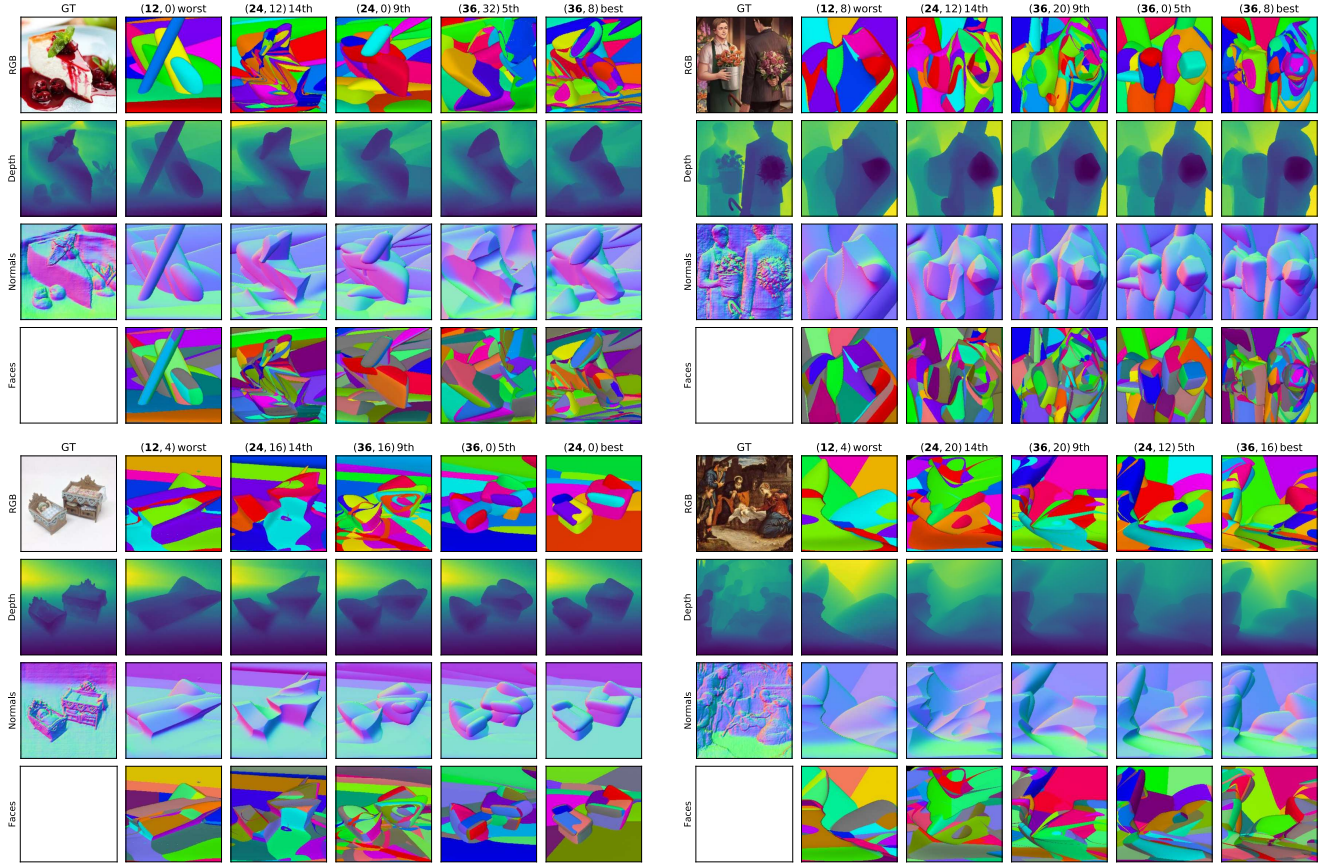
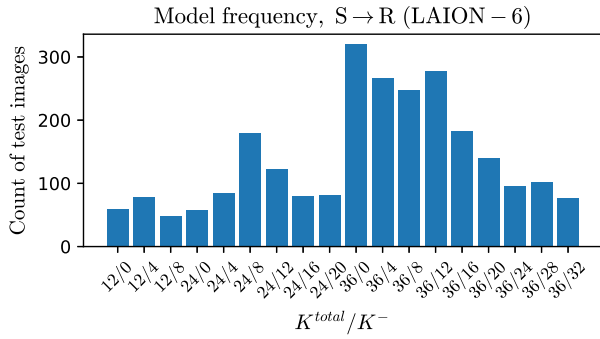
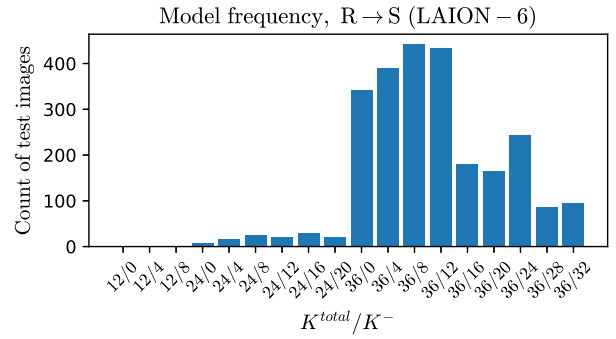


Figure 11. Additional qualitative examples shown.



(a) Select then refine ensembling on LAION 6 faces.

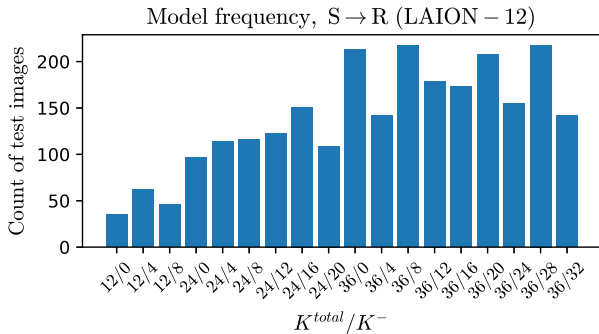


(b) Refine then select ensembling on LAION 6 faces

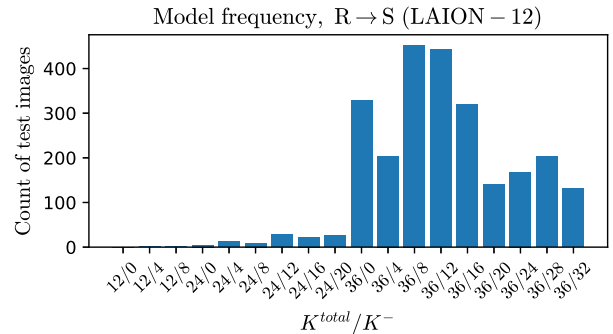
Figure 12. Distribution of models chosen on LAION (6 faces), 2500 image test set. Models with negative primitives are often chosen, especially after finetuning.

Ensemble	Refine	$K^{total}$	$K^-$	AUC@50 $\uparrow$	AUC@20 $\uparrow$	AUC@10 $\uparrow$	AUC@5 $\uparrow$	mean <sub>cm</sub> $\downarrow$	median <sub>cm</sub> $\downarrow$
No	Yes	<b>12</b>	0	0.91	0.827	0.725	0.572	0.186	0.0507
No	Yes	<b>12</b>	4	0.917	0.84	0.744	0.597	0.178	0.045
No	Yes	<b>12</b>	8	0.908	0.821	0.717	0.568	0.195	0.0523
<hr/>									
No	Yes	<b>24</b>	0	0.928	0.862	0.777	0.634	0.154	0.04
No	Yes	<b>24</b>	4	0.929	0.865	0.784	0.649	0.149	0.0395
No	Yes	<b>24</b>	8	0.932	0.872	0.795	0.663	0.144	0.0349
No	Yes	<b>24</b>	12	0.927	0.862	0.779	0.645	0.155	0.0414
No	Yes	<b>24</b>	16	0.927	0.859	0.774	0.637	0.154	0.0385
No	Yes	<b>24</b>	20	0.928	0.86	0.773	0.632	0.154	0.0392
<hr/>									
No	Yes	<b>36</b>	0	0.934	0.876	0.799	0.664	0.141	0.0358
No	Yes	<b>36</b>	4	0.935	0.878	0.806	0.677	0.138	0.0335
No	Yes	<b>36</b>	8	0.934	0.879	0.807	0.681	0.139	0.0351
No	Yes	<b>36</b>	12	0.936	0.882	0.812	<b>0.69</b>	0.134	<b>0.0314</b>
No	Yes	<b>36</b>	16	0.935	0.879	0.808	0.682	0.136	0.0324
No	Yes	<b>36</b>	20	0.934	0.876	0.802	0.676	0.138	0.0337
No	Yes	<b>36</b>	24	0.934	0.875	0.8	0.671	0.139	0.0337
No	Yes	<b>36</b>	28	0.934	0.875	0.8	0.672	0.14	0.0338
No	Yes	<b>36</b>	32	0.934	0.873	0.796	0.665	0.141	0.0346
<hr/>									
pos	$S \rightarrow R$	<b>26.1</b>	0	0.926	0.86	0.775	0.634	0.157	0.0407
pos + neg	$S \rightarrow R$	<b>29</b>	10.4	0.931	0.869	0.79	0.658	0.147	0.0363
pos	$R \rightarrow S$	<b>33.6</b>	0	0.934	0.875	0.797	0.661	0.14	0.0366
pos + neg	$R \rightarrow S$	<b>35</b>	14.7	<b>0.942</b>	<b>0.887</b>	<b>0.815</b>	0.689	<b>0.125</b>	0.0319
<hr/>									
No (Vavilala 2023)	Yes	<b>13.9</b>	0	0.869	0.725	0.565	0.382	0.266	0.101
No (Kluger 2021)	N/A	-	0	0.772	0.627	0.491	0.343	0.208	-

Table 4. **Baseline comparisons:** Ensembling strongly outperforms two recent SOTA methods, using the metrics reported by Kluger et al. [29], and using negative primitives in the ensemble produces further improvements. We show results with only positive primitives present pos, three networks,  $K^{total} \in \{12, 24, 36\}$ , as well as with positive and negative primitives pos + neg, 18 networks,  $K^- \in \{0, 4, \dots, K^{total} - 4\}$ . Our ensembles significantly outperform existing work. Further, we present results on the 18 methods we trained, where  $K^{total}/K^-$  is shown. Even without ensembling, any individual method we trained performs better than the baselines. Notice that negative primitives are helpful on average.



(a) Select then refine ensembling on LAION 12 faces.



(b) Refine then select ensembling on LAION 12 faces

Figure 13. Distribution of models chosen on LAION (12 faces), 2500 image test set. Models with negative primitives are often chosen, especially after finetuning.

Ensemble	Refine	$K^{total}$	$K^-$	AUC@50 $\uparrow$	AUC@20 $\uparrow$	AUC@10 $\uparrow$	AUC@5 $\uparrow$	mean <sub>cm</sub> $\downarrow$	median <sub>cm</sub> $\downarrow$
No	Yes	<b>12</b>	<i>0</i>	0.953	0.904	0.841	0.75	0.128	0.0364
No	Yes	<b>12</b>	<i>4</i>	0.953	0.905	<i>0.844</i>	<i>0.755</i>	0.127	<i>0.0345</i>
No	Yes	<b>12</b>	<i>8</i>	<i>0.954</i>	<i>0.905</i>	0.842	0.75	<i>0.125</i>	0.037
No	Yes	<b>24</b>	<i>0</i>	0.963	0.924	0.87	0.79	0.104	0.0288
No	Yes	<b>24</b>	<i>4</i>	0.964	0.925	0.873	0.793	0.101	0.0276
No	Yes	<b>24</b>	<i>8</i>	0.964	<i>0.926</i>	<i>0.876</i>	<i>0.798</i>	0.101	<i>0.0267</i>
No	Yes	<b>24</b>	<i>12</i>	0.963	0.923	0.87	0.791	0.104	0.0302
No	Yes	<b>24</b>	<i>16</i>	<i>0.964</i>	0.923	0.87	0.788	<i>0.101</i>	0.0287
No	Yes	<b>24</b>	<i>20</i>	0.963	0.921	0.867	0.785	0.104	0.0293
No	Yes	<b>36</b>	<i>0</i>	0.967	0.932	0.883	0.807	0.0965	0.0257
No	Yes	<b>36</b>	<i>4</i>	0.967	0.933	0.886	0.811	0.0974	0.0272
No	Yes	<b>36</b>	<i>8</i>	<i>0.968</i>	<i>0.934</i>	<i>0.887</i>	<i>0.813</i>	<i>0.0921</i>	<b>0.024</b>
No	Yes	<b>36</b>	<i>12</i>	0.967	0.933	0.886	0.813	0.0929	0.0242
No	Yes	<b>36</b>	<i>16</i>	0.967	0.931	0.882	0.806	0.0932	0.0256
No	Yes	<b>36</b>	<i>20</i>	0.966	0.929	0.879	0.802	0.0956	0.0272
No	Yes	<b>36</b>	<i>24</i>	0.967	0.93	0.879	0.802	0.0928	0.0263
No	Yes	<b>36</b>	<i>28</i>	0.963	0.924	0.872	0.793	0.104	0.0365
No	Yes	<b>36</b>	<i>32</i>	0.962	0.923	0.871	0.793	0.106	0.0357
pos	$S \rightarrow R$	<b>30.3</b>	<i>0</i>	0.963	0.925	0.872	0.794	0.104	0.0286
pos + neg	$S \rightarrow R$	<b>31.3</b>	<i>10.6</i>	0.965	0.927	0.877	0.8	0.0985	0.0273
pos	$R \rightarrow S$	<b>34.4</b>	<i>0</i>	0.967	0.932	0.882	0.805	0.0949	0.0262
pos + neg	$R \rightarrow S$	<b>35.4</b>	<i>11.7</i>	<b>0.971</b>	<b>0.937</b>	<b>0.889</b>	<b>0.814</b>	<b>0.0827</b>	0.0244

Table 5. **Quantitative evaluation on LAION 6 face polytopes:** We train and ensemble models on a subset of LAION, with approx. 1.8M images in the training set and 2500 in the test set. We report error metrics defined in by Kluger et al. [29]. Negative primitives remain useful, noting the italicized error metrics in each block of  $K^{total}$  always has negative primitives. Ensembling produces further improvements similar to NYUv2. Our method scales very well to in-the-wild scenes, producing even better metrics than NYUv2 given the larger dataset.

Ensemble	Refine	$K^{total}$	$K^-$	AUC@50 $\uparrow$	AUC@20 $\uparrow$	AUC@10 $\uparrow$	AUC@5 $\uparrow$	mean <sub>cm</sub> $\downarrow$	median <sub>cm</sub> $\downarrow$
No	Yes	<b>12</b>	0	0.959	0.913	0.854	0.765	0.114	0.0339
No	Yes	<b>12</b>	4	0.96	0.918	0.863	0.779	0.108	0.0299
No	Yes	<b>12</b>	8	0.961	0.918	0.862	0.777	0.108	0.0301
No	Yes	<b>24</b>	0	0.967	0.931	0.881	0.804	0.0959	0.0268
No	Yes	<b>24</b>	4	0.968	0.934	0.885	0.81	0.0912	0.0248
No	Yes	<b>24</b>	8	0.968	0.933	0.885	0.811	0.0916	0.0246
No	Yes	<b>24</b>	12	0.969	0.935	0.887	0.812	0.0881	0.0243
No	Yes	<b>24</b>	16	0.968	0.934	0.886	0.812	0.0898	0.0243
No	Yes	<b>24</b>	20	0.967	0.928	0.877	0.801	0.0927	0.0274
No	Yes	<b>36</b>	0	0.97	0.939	0.893	0.821	0.0872	0.0236
No	Yes	<b>36</b>	4	0.971	0.94	0.895	0.823	0.0841	0.0227
No	Yes	<b>36</b>	8	0.971	0.941	0.897	0.827	0.0829	0.0218
No	Yes	<b>36</b>	12	0.972	0.942	0.898	0.828	0.0816	<b>0.0217</b>
No	Yes	<b>36</b>	16	0.971	0.94	0.896	0.825	0.0837	0.0221
No	Yes	<b>36</b>	20	0.971	0.939	0.892	0.82	0.0845	0.0231
No	Yes	<b>36</b>	24	0.971	0.939	0.894	0.822	0.0836	0.0227
No	Yes	<b>36</b>	28	0.971	0.939	0.892	0.819	0.084	0.0232
No	Yes	<b>36</b>	32	0.971	0.938	0.891	0.818	0.0851	0.0234
pos	$S \rightarrow R$	<b>29.8</b>	0	0.967	0.933	0.884	0.808	0.0943	0.0262
pos + neg	$S \rightarrow R$	<b>31.2</b>	13.5	0.969	0.935	0.888	0.815	0.0884	0.0245
pos	$R \rightarrow S$	<b>35.3</b>	0	0.97	0.938	0.892	0.82	0.0867	0.0241
pos + neg	$R \rightarrow S$	<b>35.5</b>	13.2	<b>0.974</b>	<b>0.944</b>	<b>0.899</b>	<b>0.829</b>	<b>0.0759</b>	0.022

Table 6. **Quantitative evaluation on LAION 12 face polytopes:** Most recent literature on primitive-fitting focuses on cuboids or parallelepipeds, but our model is capable of fitting polytopes of variable face count. All error metrics get better with more faces, which indicates more complex primitives yield more accurate representations.

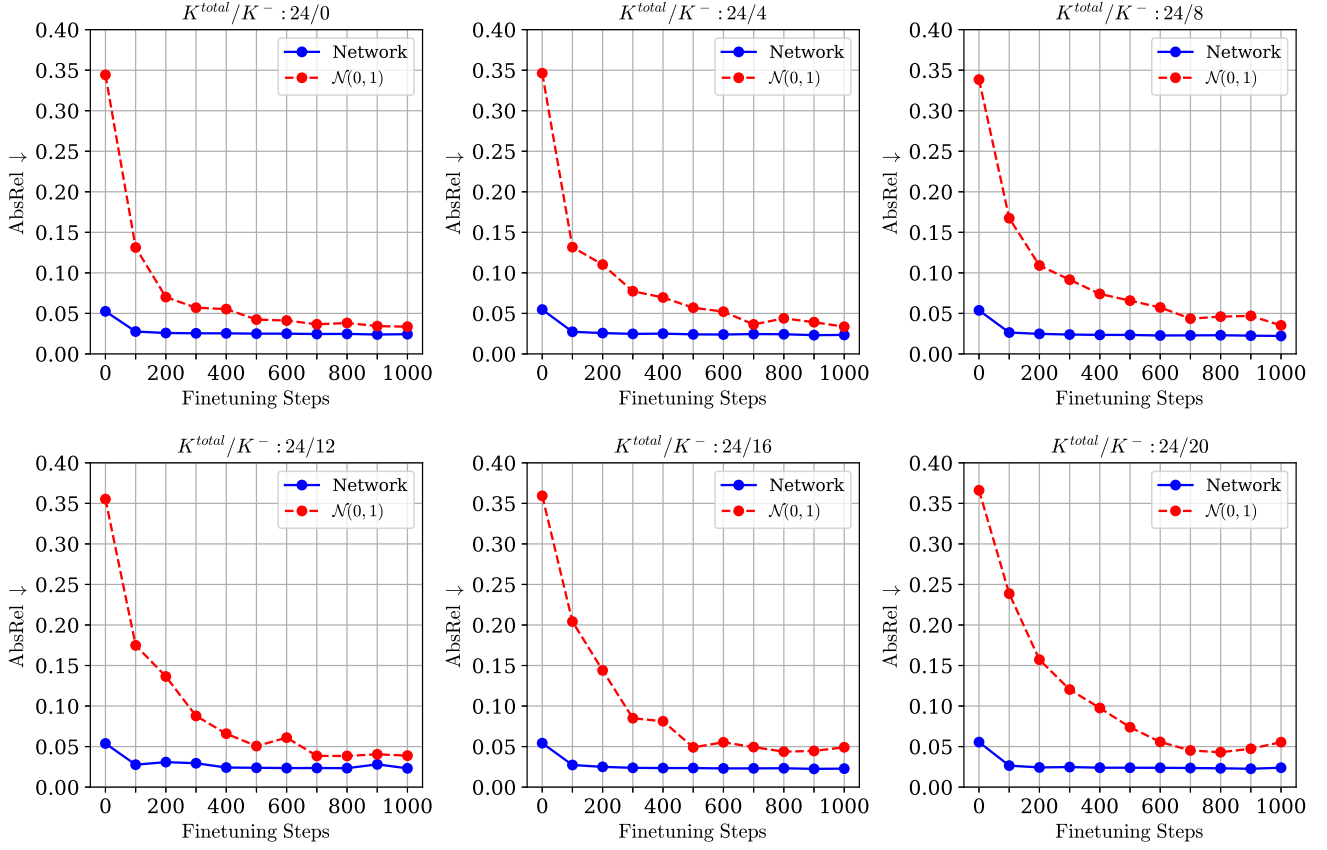


Figure 14. Additional examples on the value of network start, on 100 LAION test images,  $K^{total} = 24$ .



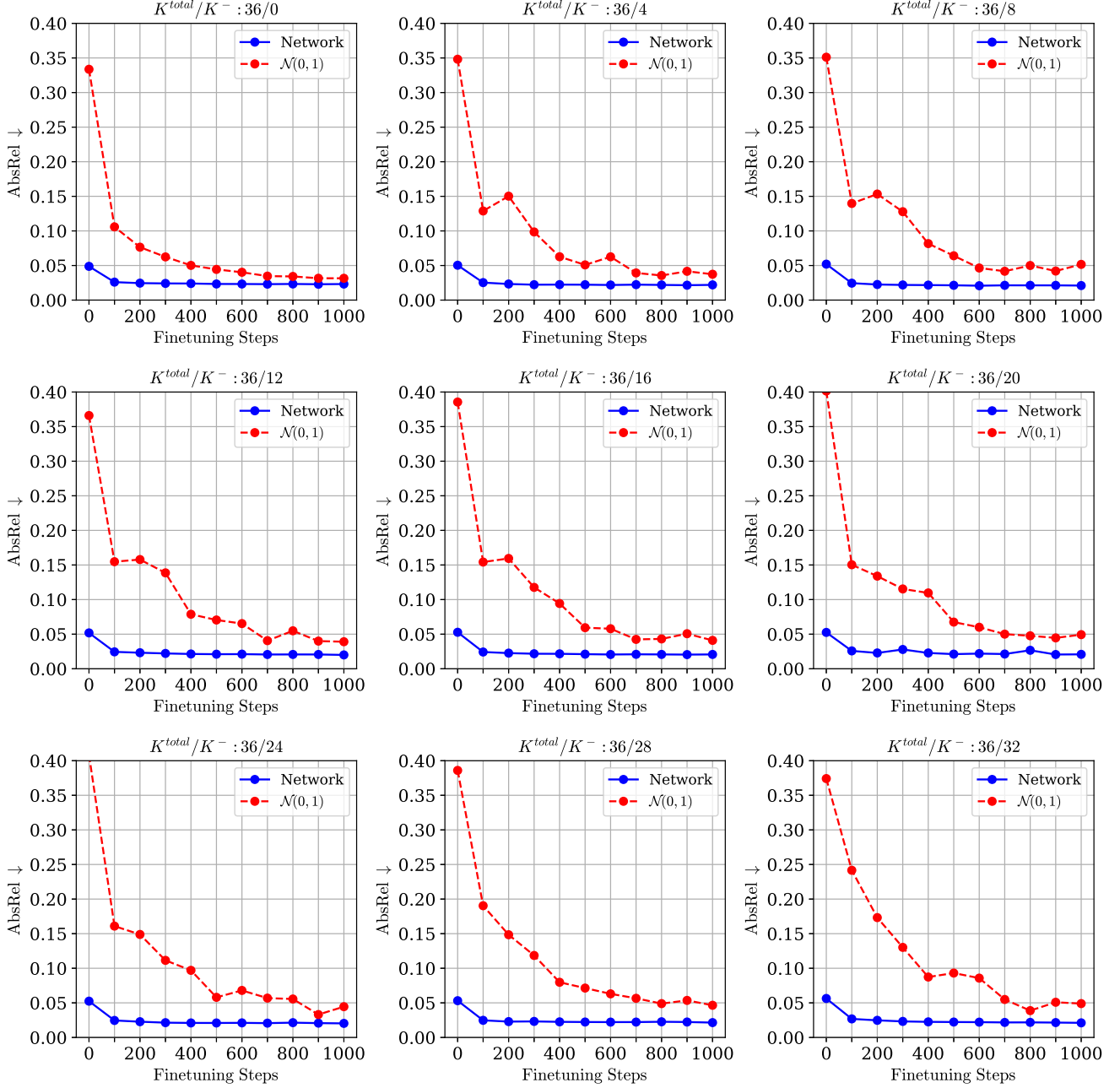


Figure 15. Additional examples on the value of network start, on 100 LAION test images,  $K^{total} = 36$ .

Ensemble	$K^{total}$	$K^-$	AbsRel↓	Normals Mean↓	Normals Median↓	SegAcc↑
No	<b>12</b>	0	0.0622	34.4	26.3	0.651
No	<b>12</b>	4	0.0597	34.5	26.3	0.68
No	<b>12</b>	8	0.064	35.7	27.2	0.666
No	<b>24</b>	0	0.052	33	25	0.7
No	<b>24</b>	4	0.0504	33	24.9	0.726
No	<b>24</b>	8	0.0486	32.7	24.6	0.742
No	<b>24</b>	12	0.0514	33.4	25.3	0.724
No	<b>24</b>	16	0.0506	33.8	25.7	0.724
No	<b>24</b>	20	0.0508	33.9	25.6	0.709
No	<b>36</b>	0	0.0484	32.3	24.4	0.723
No	<b>36</b>	4	0.0467	32.3	24.3	0.752
No	<b>36</b>	8	0.0469	32.2	24.2	0.757
No	<b>36</b>	12	0.0452	32.1	24	<b>0.765</b>
No	<b>36</b>	16	0.0462	32.3	24.3	0.756
No	<b>36</b>	20	0.0463	32.6	24.5	0.756
No	<b>36</b>	24	0.0464	32.8	24.6	0.748
No	<b>36</b>	28	0.0466	32.8	24.7	0.745
No	<b>36</b>	32	0.047	33	24.8	0.735
pos $S \rightarrow R$	<b>26.1</b>	0	0.0527	33	24.9	0.7
pos + neg $S \rightarrow R$	<b>29</b>	10.4	0.0492	32.9	24.7	0.733
pos $R \rightarrow S$	<b>33.6</b>	0	0.0476	32.3	24.4	0.72
pos + neg $R \rightarrow S$	<b>35</b>	14.7	<b>0.0417</b>	<b>31.9</b>	<b>24</b>	0.756
Vavilala & Forsyth [62]	<b>13.9</b>	0	0.098	37.4	32.4	0.618

Table 7. Detailed error metrics on NYUv2.

Ensemble	$K^{total}$	$K^-$	AbsRel↓	Normals Mean↓	Normals Median↓	$Neg\_per\_pos$
No	<b>12</b>	0	0.0297	35.6	29.3	0
No	<b>12</b>	4	0.0295	35.6	29.1	1.42
No	<b>12</b>	8	0.029	35.8	29.2	4.13
No	<b>24</b>	0	0.0244	33.9	27.5	0
No	<b>24</b>	4	0.0238	33.9	27.4	0.719
No	<b>24</b>	8	0.0235	33.7	27	1.54
No	<b>24</b>	12	0.0242	34	27.4	2.39
No	<b>24</b>	16	0.0235	34	27.4	3.94
No	<b>24</b>	20	0.0242	34.2	27.5	8.04
No	<b>36</b>	0	0.0225	33	26.7	0
No	<b>36</b>	4	0.0223	32.9	26.4	0.548
No	<b>36</b>	8	0.0217	32.9	26.3	1.01
No	<b>36</b>	12	0.0218	32.9	26.3	1.5
No	<b>36</b>	16	0.0217	33.2	26.6	1.9
No	<b>36</b>	20	0.0221	33.4	26.8	2.55
No	<b>36</b>	24	0.0215	33.3	26.7	3.81
No	<b>36</b>	28	0.0236	33.8	27.1	5.49
No	<b>36</b>	32	0.0243	33.9	27.3	10.8
pos $S \rightarrow R$	<b>30.3</b>	0	0.0242	33.6	27.2	0
pos + neg $S \rightarrow R$	<b>31.3</b>	10.6	0.0229	33.4	26.8	2.04
pos $R \rightarrow S$	<b>34.4</b>	0	0.0221	33.1	26.7	0
pos + neg $R \rightarrow S$	<b>35.4</b>	11.7	<b>0.0193</b>	<b>32.7</b>	<b>26.2</b>	1.94

Table 8. Additional error metrics on LAION, 6 faces. The final column,  $Neg\_per\_pos$ , evaluates the average number of negative primitives touching each positive primitive, quantitatively showing negative primitives active in the geometric abstraction.

Ensemble	$K^{total}$	$K^-$	AbsRel↓	Normals Mean↓	Normals Median↓	$Neg\_per\_pos$
No	<b>12</b>	0	0.0265	34.8	28.4	0
No	<b>12</b>	4	0.0253	34.3	27.8	1.35
No	<b>12</b>	8	0.0252	34.4	27.8	3.89
No	<b>24</b>	0	0.0223	33.1	26.6	0
No	<b>24</b>	4	0.0215	32.9	26.2	0.832
No	<b>24</b>	8	0.0216	32.8	26.2	1.49
No	<b>24</b>	12	0.0208	32.7	26.1	2.48
No	<b>24</b>	16	0.021	32.8	26.1	4.04
No	<b>24</b>	20	0.0222	33.3	26.6	7.64
No	<b>36</b>	0	0.0203	32.2	25.7	0
No	<b>36</b>	4	0.0199	32.2	25.6	0.648
No	<b>36</b>	8	0.0196	32	25.3	1.1
No	<b>36</b>	12	0.0193	32	25.3	1.59
No	<b>36</b>	16	0.0198	32.1	25.4	2.11
No	<b>36</b>	20	0.0199	32.3	25.6	2.76
No	<b>36</b>	24	0.0197	32.2	25.6	3.88
No	<b>36</b>	28	0.0198	32.3	25.6	6.17
No	<b>36</b>	32	0.0201	32.4	25.7	10.9
pos $S \rightarrow R$	<b>29.8</b>	0	0.0218	32.7	26.3	0
pos + neg $S \rightarrow R$	<b>31.2</b>	13.5	0.0207	32.5	25.9	2.93
pos $R \rightarrow S$	<b>35.3</b>	0	0.0201	32.2	25.8	0
pos + neg $R \rightarrow S$	<b>35.5</b>	13.2	<b>0.0178</b>	<b>31.8</b>	<b>25.2</b>	2.53

Table 9. Additional error metrics on LAION, 12 faces. The final column,  $Neg\_per\_pos$ , evaluates the average number of negative primitives touching each positive primitive, quantitatively showing negative primitives active in the geometric abstraction.

## 10. Common Questions

### Q1. How is the method different from CvxNet?

**A1.** While our approach draws inspiration from CvxNet, the technical setting is fundamentally different and introduces non-trivial challenges. CvxNet fits convex primitives to clean, segmented meshes or point clouds, often with full object geometry available. In contrast, our method fits primitives directly to *in-the-wild* RGB-D scenes, where:

- Geometry is partial (due to occlusion and unknown backs of objects).
- Scene segmentation may not be available.
- Lighting, noise, and clutter introduce significant fitting ambiguity.

To address this, we:

1. Integrate **set-differencing (negative primitives)**—a first in primitive-fitting for real RGB scenes—allowing representation of concavities and occluded voids.
2. Design a **test-time ensembling pipeline** to select the optimal primitive count per scene using only geometric consistency metrics.
3. Introduce **improved optimization, sampling, and hyperparameters** so that even our smallest models (12 primitives, no negatives) outperform prior work [29, 62] in accuracy, segmentation, and latency.

### Q2. Why use negative primitives and difference operations for real scenes? Aren't convex-only shapes sufficient?

**A2.** Positive-only convex shapes cannot represent important concave features without excessive oversegmentation. Negative primitives allow parsimonious modeling of real-world voids and cavities (e.g., under desks, chair legs, hollow shelves) without increasing primitive count. Quantitatively:

- **Segmentation Accuracy** improves from 0.723 (positive-only) to 0.765 (with negatives).
- **Depth AbsRel** improves significantly for the same primitive count when letting some primitives be negative (Table 7).

While some surfaces may appear “carved,” the improved geometry aligns better with ground-truth depth/normals. Moreover, the ensemble retains multiple reconstructions—users can choose purely convex outputs if visually preferred.

### Q3. The ensembling strategy is a simple heuristic. Why is it a central contribution?

**A3.** Fixed-count primitive-fitting fails to adapt to scene complexity variation. Our setting is unique because we can **quantitatively score reconstructions against input depth maps without ground truth primitives**—allowing a principled selection of primitive count at test time. This:

- Consistently improves over the best single fixed-count model (Table 1).
- Produces results faster than prior SOTA [62], even in the full ensemble mode (29.9s vs. 40s).

- Enables *variable abstraction levels*, which prior works do not offer.

While ensembling is simple in concept, its *problem-specific application* here solves a long-standing gap in scene decomposition—choosing the right model complexity without retraining.

### Q4. Why convex polytopes instead of cuboids or superquadrics?

**A4.** Convex polytopes offer:

- **Flexibility:** By varying face count, they generalize cuboids/parallelepipeds while remaining convex.
- **Better optimization:** Represented as intersections of halfspaces, they are differentiable via logsumexp, unlike superquadrics, which often produce singularities, non-convexities, or self-intersections in certain parameter regimes.
- **Fair comparison:** Using the same primitive vocabulary as [29, 62] ensures direct metric comparability.

Empirically, prior work [29] found cuboids outperform superquadrics in scene fitting; our parallelepipeds further improve accuracy (Table 4).

### Q5. The visual results sometimes appear overfragmented or structurally implausible. How does this affect applications?

**A5.** There is an inherent trade-off between **parsimonious, human-intuitive parses** and **high-fidelity geometric accuracy**.

- For tasks like *robotics grasp planning, simulation, or depth-to-image editing*, precise geometry may be preferred, even if primitives are more fragmented.
- For *artist-friendly modeling*, fewer, larger convex shapes may be desired—our method supports this by training and exposing multiple abstraction levels (12–36 parts) and convex-only variants. Observe in Figs. 3 and 4 how simple abstractions can be created with a few positive-only primitives, while complex (but more detailed) scene decompositions can be generated with a mixture of many primitives.

Prior works did not explore this trade-off or provide user-selectable abstraction.

### Q6. How does the method compare in segmentation stability and runtime?

**A6.** **Stability:** Despite increased geometric interactions from negative primitives, segmentation accuracy improves (SegAcc 0.756 vs. 0.618 in [62]). This suggests greater alignment with real object boundaries.

**Runtime:** Even the largest ensemble is **faster than prior SOTA** [62] and individual models run significantly faster (Table 1).

### Q7. What about scalability and limitations?

**A7.**



- **Training scalability:** Current approach requires training separate models per primitive count; future work will explore unified variable-count architectures (e.g., transformer-based). It is non-trivial to avoid mode collapse with transformers because we do not know the GT positions of the primitives (unlike, say object detection).
- **Primitive operations:** Only union and subtraction are explored; intersection is technically feasible but left for future work.
- **Failure cases:** Performance may degrade in scenes with extreme clutter, thin structures, or non-rigid shapes; examples are shown in Figures 3 and 4 (worst ensemble members).
- **Resource constraints:** Ensembling increases inference cost, though individual models still surpass prior work in both speed and accuracy.