

Learned Suppression for 3D Keypoint Detection with a Graph-Transformer Backbone Supplementary Material

Batuhan Arda Bekar[✉], Can Sari[✉], Hüseyin Can Gülkan[✉], and Barış Özcan[✉]

Department of Computer Engineering, Bahçeşehir University, Istanbul, Türkiye
{batuhanarda.bekar, can.sari, huseyin.gulkan}@bahcesehir.edu.tr
baris.ozcan@bau.edu.tr

This supplement provides: **(A)** additional training settings; **(B)** the learned suppression module in detail; **(C)** the Building3D evaluation protocol; **(D)** KeypointNet results across distance thresholds; **(E)** additional qualitative results; **(F)** the component ablation across distance thresholds; **(G)** the baseline backbones; **(H)** sensitivity, runtime, and seed variance of the suppression module; **(I)** reproducibility.

A Implementation Details

Table A1 lists additional training configuration.

B Learned Suppression Module Details

We additionally detail the learned suppression module: how candidates are formed, the candidate feature, the sizes of its five networks (Tab. A2), and its algorithm details.

Candidate generation. Each mean shift pass moves every vote to the mean of the votes within radius 0.05 of its current position, weighted by their corner probabilities, and rebuilds these neighbourhoods from the updated positions before the next pass, so votes that agree on a location converge to it. Greedy NMS then processes the votes in order of decreasing corner probability: the vote

Table A1: Additional training configuration. The candidate feature is the input of the suppression module; its two feature parts are pooled from the 128 backbone feature channels of each candidate’s voters and of its 4 nearest candidates (Sec. B).

Setting	Building3D	KeypointNet
AdamW β_1, β_2	0.9, 0.999	0.9, 0.999
Weight decay	10^{-2}	10^{-2}
OneCycleLR warmup / final LR	5% / 3×10^{-7}	5% / 3×10^{-7}
Augmentation	flips (both vertical planes), rotation $[-\pi, \pi]$	flips (x and y), rotation $[-\pi, \pi]$, scale jitter 0.8 to 1.25
Suppression module optimiser	Adam, LR 10^{-3}	Adam, LR 10^{-3}
Candidate feature (266 dimensions)	10 vote statistics + 128 (voter mean) + 128 (mean of 4 nearest candidates)	
Keep threshold τ	tuned on the validation split for each trained module	

Table A2: Layers of the suppression module. MLP_{emb} and MLP_{keep} are shared across rounds; the other networks have independent weights in each of the $T=2$ rounds (Algorithm 1 of the main paper). Every arrow is one linear layer.

Network	Layers	Parameters
MLP_{emb} (shared)	$266 \rightarrow 128, 128 \rightarrow 128$	50,688
MLP_{rel} (per round)	projection $128 \rightarrow 64$; pair $129 \rightarrow 64 \rightarrow 1$	16,641
MLP_{upd} (per round)	$256 \rightarrow 128$	32,896
MLP_{mov} (per round)	$128 \rightarrow 3$	387
MLP_{keep} (shared)	$128 \rightarrow 1$	129
Total ($T=2$)		150,665

with the largest probability among those remaining survives as a candidate, every vote within radius 0.05 of it is absorbed and removed, and this repeats until no vote is left. The candidate takes the position of this surviving vote, and the absorbed votes form its voter set.

Candidate feature. The 266 dimension input of Tab. A1 is built per candidate from its voter set. The 10 statistics are: the logarithm of one plus the voter count; the mean, maximum, and standard deviation of the voter probabilities; the probability of the surviving vote; the 75th percentile of the voter probabilities; the mean, maximum, and standard deviation of the voter distances to the surviving vote; and the mean voter offset magnitude. The first 128 channel block is the mean of the voter backbone features, weighted by voter probability. The second is the mean of that block over the 4 nearest other candidates, and zero when the candidate is alone.

Network sizes. Table A2 lists every layer. Hidden activations are GELU, each followed by dropout 0.1 during training. MLP_{rel} first projects each 128 channel hidden state to 64 channels; its pair input is the two projections and the scalar distance, 129 values. The module totals 150,665 parameters.

Algorithm details. Both displacement heads are initialised to zero, weights and bias, so no candidate moves at initialisation. A candidate with no other candidate within ρ receives a zero message and zero suppression, so its keep probability is the output of MLP_{keep} on its hidden state alone. The keep loss and the displacement loss are summed with equal weight; the displacement term is smooth L_1 with transition point 0.01, computed only on matched candidates. Training candidates are precomputed once from the frozen backbone; each training scene contributes its candidate set under four random rotations about the vertical axis.

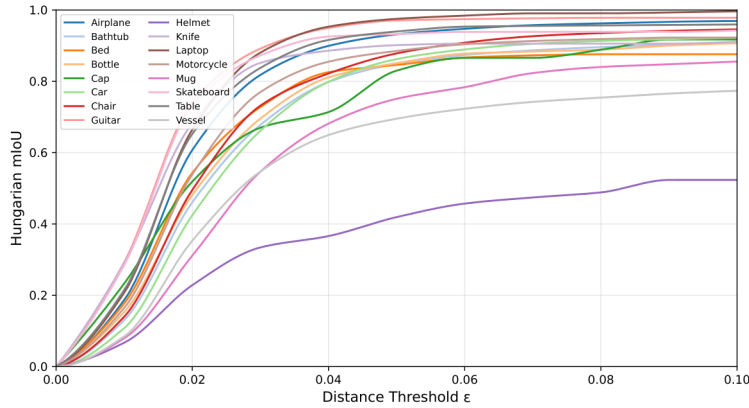


Fig. A1: Hungarian mIoU against the distance threshold ϵ for the 16 KeypointNet categories, from a single model trained on the whole KeypointNet dataset.

C Building3D Evaluation Protocol

Both test splits are released without ground truth (Entry-Level: 583 scenes; Tallinn: 3,472 scenes), so test metrics come from the official online evaluator of the benchmark [4]; no tuning uses the test set.

D KeypointNet Results Across Distance Thresholds

We additionally report the KeypointNet results at distance thresholds ϵ from 0.01 to 0.10 (Fig. A1, Tab. A3). The $\epsilon=0.10$ column equals the Ours row of Table 3 of the main paper. Helmet has the lowest mIoU at every threshold.

E Additional Qualitative Results

Figure A2 additionally presents one test shape from each of the twelve categories not shown in Fig. 1 of the main paper.

Figures A3 and A4 compare greedy NMS, DBSCAN, and learned suppression on identical outputs of our backbone, on KeypointNet test shapes and on Building3D Entry-Level validation scenes. Table 4 of the main paper compares the methods on the full test splits.

F Component Ablation Across Distance Thresholds

We additionally evaluate the KeypointNet component ablation at every distance threshold (Tab. A4). With geodesic neighbourhoods, mIoU at $\epsilon=0.01$ moves from 3.21 to 10.11. Selecting the DBSCAN threshold, radius, and minimum cluster

Table A3: KeypointNet results per category at distance thresholds $\epsilon \in \{0.01, \dots, 0.10\}$, from a single model trained on the whole KeypointNet dataset. N is the number of test shapes of the category. The $\epsilon=0.10$ column equals the Ours row of Table 3 of the main paper. CD does not depend on ϵ .

Category	N	Hungarian mIoU (%) at distance threshold ϵ										CD↓
		0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	
Airplane	205	18.69	60.75	81.81	89.95	93.19	94.74	95.78	96.27	96.72	96.93	0.038
Bathtub	99	13.04	46.10	67.80	79.82	84.94	87.39	88.71	89.62	90.26	90.83	0.049
Bed	30	17.73	54.36	72.81	82.58	84.57	86.68	87.30	87.61	87.61	87.61	0.056
Bottle	76	15.31	48.14	69.56	81.11	85.12	87.77	88.36	88.89	90.03	90.68	0.049
Cap	8	23.31	51.91	67.00	71.38	83.01	86.58	86.58	88.84	91.74	91.74	0.062
Car	201	10.81	42.46	66.14	79.89	86.24	88.96	90.54	91.36	91.82	92.25	0.051
Chair	200	14.01	49.54	73.19	82.22	87.84	90.86	92.67	93.55	94.17	94.58	0.051
Guitar	140	29.28	74.11	89.19	94.98	97.09	97.45	97.69	97.84	97.84	97.84	0.029
Helmet	18	6.70	22.85	33.43	36.62	41.93	45.73	47.45	48.86	52.36	52.36	0.407
Knife	54	29.10	68.24	85.18	88.60	90.17	90.56	90.56	90.56	90.56	91.12	0.036
Laptop	88	20.76	66.46	88.07	95.35	97.54	98.43	99.08	99.08	99.32	99.65	0.033
Motorcycle	60	16.46	53.91	77.00	85.51	88.53	90.40	91.33	91.87	92.28	92.28	0.047
Mug	38	7.94	31.14	54.67	68.16	75.11	78.37	82.33	84.04	84.80	85.55	0.096
Skateboard	29	28.48	73.08	87.28	92.56	93.25	93.87	93.87	93.87	93.87	94.25	0.037
Table	225	21.50	65.25	83.78	91.61	93.99	95.36	95.55	95.70	95.86	95.96	0.059
Vessel	182	8.48	35.32	54.72	64.95	69.53	72.32	74.25	75.43	76.57	77.35	0.079

size on the KeypointNet validation split (824 shapes) gives threshold 0.40 and radius 0.06, which read 80.76 mIoU and 0.089 CD on the test split (indented row of Tab. A4). With relocation disabled, learned suppression reads 90.69 mIoU and 0.071 CD.

G Baseline Backbones

The study of Table 4 of the main paper pairs the suppression module with two standard backbones. Each backbone outputs, for every point, a keypoint logit, an offset vector to the nearest keypoint, and a feature vector with 128 channels. Greedy NMS [2], DBSCAN [1], and the suppression module consume only these outputs, so the three methods run unchanged on every backbone. The channel count is fixed at 128 by the candidate feature of Tab. A1.

PointNet++. PointNet++ [3]: four set abstraction stages, four feature propagation stages, and one head per output. The input contains only the point coordinates.

DGCNN. DGCNN [5]: a dynamic graph over the k nearest neighbours, Edge-Conv blocks, a global feature, and a segmentation head, followed by the same output heads. The input contains only the point coordinates.

Table A4: KeypointNet component ablation across distance thresholds (mIoU %). Rows follow Table 2 of the main paper; each indented row varies only the row above it: DBSCAN settings selected on the KeypointNet validation split, and rescoring without relocation. Best per column in **bold**.

Configuration	Hungarian mIoU (%) at distance threshold ϵ										CD↓
	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	
Backbone, DBSCAN	3.21	23.13	47.47	59.55	65.63	69.74	72.47	74.70	76.31	77.37	0.091
Euclidean → geodesic graph	10.11	37.97	56.66	65.47	70.11	73.30	75.66	77.37	78.64	79.67	0.085
DBSCAN tuned on KeypointNet	11.38	39.70	58.10	66.25	71.40	74.46	77.00	78.67	79.88	80.76	0.089
DBSCAN → greedy NMS	9.00	35.44	54.85	65.58	72.08	76.62	79.45	81.43	82.87	84.25	0.080
learned supp. with relocation	16.99	53.92	74.35	83.50	87.45	89.44	90.54	91.14	91.66	92.01	0.055
learned supp. (rescore only)	2.65	20.13	51.45	74.93	83.86	87.10	88.71	89.63	90.24	90.69	0.071

Both backbones are trained for 120 epochs per dataset. The suppression module keeps its architecture and settings and is retrained on each frozen backbone.

H Sensitivity, Runtime, and Seed Variance

The suppression module has two inference parameters: keep threshold τ and suppression radius ρ . Table A5 varies each on every backbone’s trained checkpoint of Table 4 of the main paper, without retraining. The keep threshold is selected on the validation split (0.3 for the PointNet++ and DGCNN checkpoints, 0.4 for the full model; Tab. A1), while $\rho=0.08$ is the value used in training. The mIoU of every backbone changes by roughly a point across ρ from 0.04 to 0.12. The keep threshold is stable through a moderate band and falls at high τ , reaching 85.46, 63.55, and 67.67 at $\tau=0.8$ on Ours, PointNet++, and DGCNN. The round count is 2, fixed at training.

Table A6 reports the latency per shape for the three methods on each backbone, measured on 300 test shapes with a batch of 8 for the forward pass; the methods on one backbone share its forward pass. The accuracy of each method and backbone is in Table 4 of the main paper.

The suppression module trains on a frozen backbone, so the only source of variation between runs is the training seed. Table A7 retrains it under three different seeds on each backbone to measure that variance alone, without further tuning, and reports the mean and standard deviation. The small standard deviations, 0.14 to 0.34 mIoU, show that the module retrains stably.

I Reproducibility

Every experiment runs in a single Conda environment on one NVIDIA RTX 4070. KeypointNet metrics are computed locally; Building3D test metrics come from the online evaluator (Sec. C).

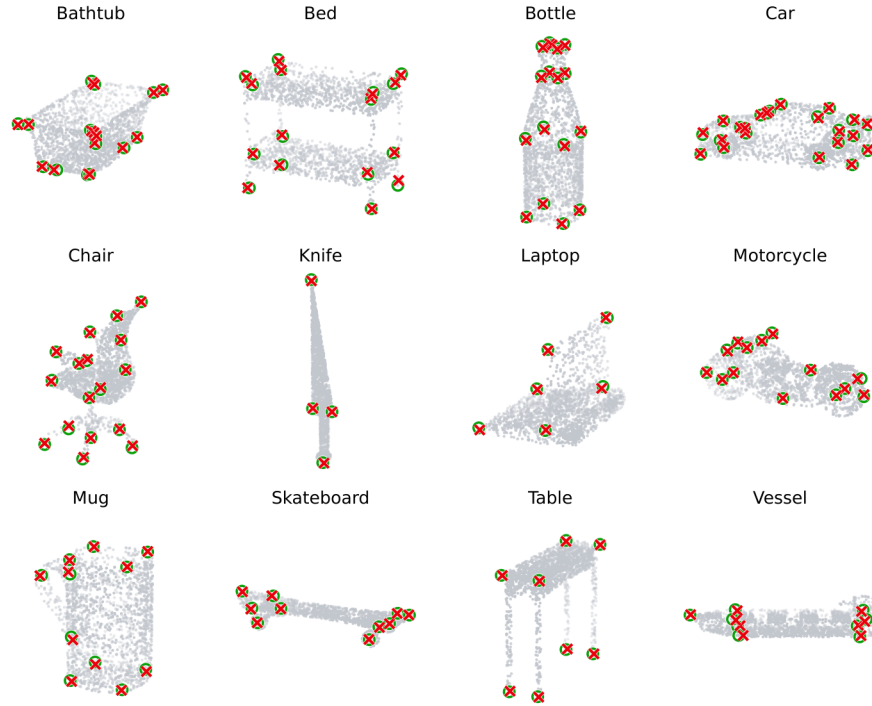


Fig. A2: Test shapes from the twelve KeypointNet categories not shown in Fig. 1 of the main paper. Red crosses: predictions. Green circles: ground truth. Single model trained on the whole KeypointNet dataset.

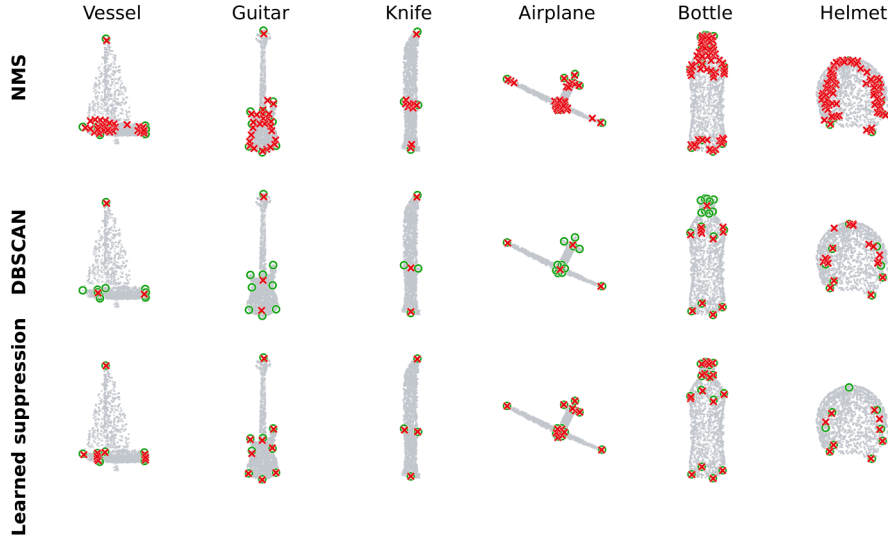


Fig. A3: Greedy NMS (top), DBSCAN (middle), and learned suppression (bottom) on identical outputs of our backbone, KeypointNet test shapes. Red crosses: kept keypoints. Green circles: ground truth.

Table A5: Sensitivity of the suppression module on the KeypointNet test split, at a fixed round count of 2. Each block varies one parameter while the other stays at the operating point ($\rho=0.08$ and the τ tuned on validation); the two are not varied jointly. * marks that operating point, the same run in both blocks. It is selected on the validation split without consulting the test set; a few settings score higher here, but adopting them would tune on the test set.

Setting	Value	Ours		PointNet++		DGCNN	
		mIoU@0.10	CD↓	mIoU@0.10	CD↓	mIoU@0.10	CD↓
keep threshold τ	0.3	90.77	0.050	74.99*	0.115	77.87*	0.099
	0.4	92.01*	0.055	74.53	0.131	78.05	0.108
	0.5	91.51	0.055	73.14	0.148	76.88	0.121
	0.6	90.67	0.060	71.15	0.158	75.02	0.134
	0.7	89.30	0.069	68.18	0.177	72.18	0.157
	0.8	85.46	0.084	63.55	0.207	67.67	0.185
suppression radius ρ	0.04	90.88	0.053	74.93	0.115	77.44	0.099
	0.06	90.89	0.053	75.28	0.116	77.61	0.099
	0.08	92.01*	0.055	74.99*	0.115	77.87*	0.099
	0.10	91.70	0.054	75.12	0.116	78.05	0.098
	0.12	91.43	0.053	74.79	0.118	78.05	0.098

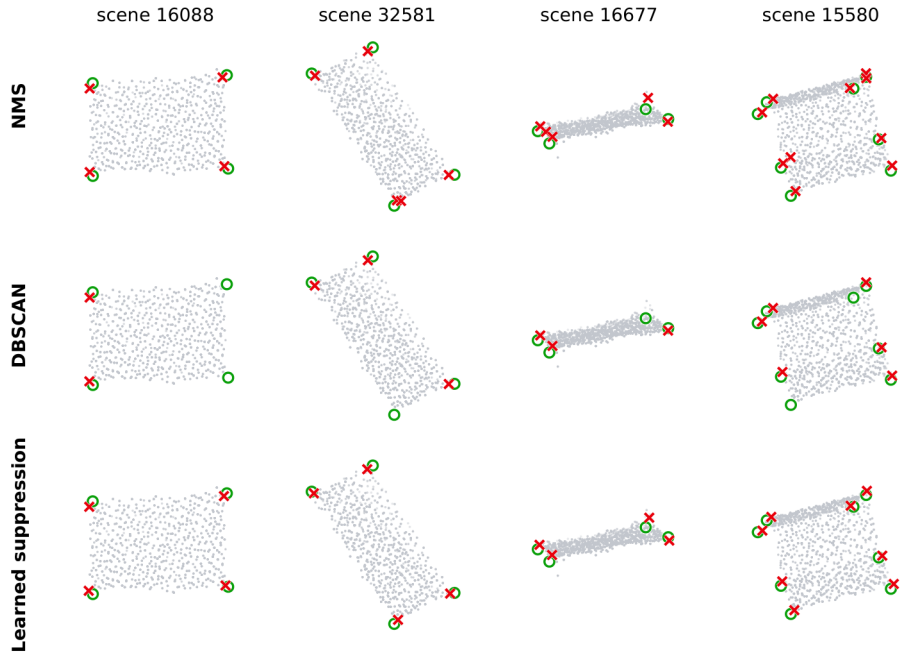


Fig. A4: Greedy NMS (top), DBSCAN (middle), and learned suppression (bottom) on identical outputs of our backbone, Building3D Entry-Level scenes held out from the training data with ground truth available. Red crosses: kept corners. Green circles: ground truth corners.

Table A6: Latency on KeypointNet (ms per shape, 300 shape subset). Total includes the backbone forward pass; the methods on one backbone share that forward pass. Accuracy per backbone and method is in Table 4 of the main paper.

Method	Ours		PointNet++		DGCNN	
	Total	After backbone	Total	After backbone	Total	After backbone
Backbone only	16.60	–	4.75	–	3.40	–
Greedy NMS	59.85	43.25	14.59	9.84	11.24	7.84
DBSCAN	30.19	13.58	18.78	14.03	16.38	12.98
Learned suppression	68.12	51.52	21.22	16.47	15.11	11.70

Table A7: Seed variance of the suppression module on KeypointNet (three training seeds per backbone, frozen backbone; only the suppression module is retrained). Mean $\pm$ standard deviation.

Backbone	mIoU@0.10	CD↓
Ours	91.56 ± 0.28	0.056 ± 0.002
PointNet++	74.52 ± 0.14	0.119 ± 0.001
DGCNN	77.81 ± 0.34	0.104 ± 0.005

References

1. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. pp. 226–231. KDD’96, AAAI Press (1996)
2. Neubeck, A., Van Gool, L.: Efficient non-maximum suppression. In: 18th international conference on pattern recognition (ICPR’06). vol. 3, pp. 850–855. Ieee (2006)
3. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017)
4. Wang, R., Huang, S., Yang, H.: Building3d: A urban-scale dataset and benchmarks for learning roof structures from point clouds. In: ICCV. pp. 20076–20086 (2023)
5. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM TOG **38**(5), 1–12 (2019)