INVALID DEPTH IS NOT MISSING DEPTH

Anonymous authors
Paper under double-blind review

ABSTRACT

A missing depth measurement is not a measurement of zero. Yet common RGB-D unseen object instance segmentation pipelines encode invalid pixels as $\mathrm{XYZ} = (0, 0, 0)$, placing them at the optical centre. On OCID and OSD, 11–24% of pixels lack depth, whereas the synthetic training data are complete. Controlled comparisons and within-model ablations identify invalidity encoding as a substantial confound in robustness evaluation: UCN without validity filtering drops to zero Object-F after just 0.5% additional i.i.d. pixel removal, while the tested image-depth baseline remains stable. We further show that i.i.d. removal poorly matches the spatial structure of observed sensor failures, and introduce a corruption generator calibrated to each sensor's missing-depth statistics. Our repair, PROXFILL, copies the XYZ triplet of the nearest valid pixel into an invalid pixel only when their image-space distance is at most τ. With τ fixed on a disjoint development split, this training-free rule improves the official UCN+ checkpoint by 2.8 Object-F and 6.9 Boundary-F points on OSD and transfers unchanged to MSMFormer. Under calibrated degradation, corrected UCN and MSMFormer remain within one Object-F point of their clean-input scores, while the published pipelines lose 14–16 points. Replacing the depth-validity filter with PROXFILL also reduces measured UCN processing latency by 25%, without retraining or additional learnable parameters. These findings show that a substantial part of the measured robustness gap between RGB-D methods can arise from how invalid depth is represented.

1 INTRODUCTION

Fig. 1 shows two boxes touching on a table. The ground truth separates them; the published RGB-D pipeline returns one mask covering both. Nothing is wrong with the RGB, the objects are not occluded, and the same checkpoint separates them correctly once a single line of preprocessing changes. What differs is how the pipeline wrote down the pixels where the depth sensor returned nothing.

Unseen object instance segmentation (UOIS) asks a model to segment every object in a cluttered scene without knowing the categories, and the dominant recipe trains on synthetic RGB-D and deploys on real sensors (Xie et al., 2021; Xiang et al., 2021; Lu et al., 2024). A real sensor returns nothing where the surface is specular, transparent, out of range, or unmatched across the stereo baseline. In an *organized point cloud* — the representation UCN and MSMFormer consume — there is no channel for "no data," so such a pixel is stored as $\mathrm{XYZ} = (0, 0, 0)$. That is not a sentinel the network can recognise. It is a valid 3D coordinate at the optical centre: the nearest point the camera could possibly observe. The PCD files shipped with both standard benchmarks mark these pixels correctly as `NaN`, and both loaders then run `pcloud[np.isnan(pcloud)] = 0`. OCID (Suchi et al., 2019) is 11.5% invalid and OSD (Richtsfeld et al., 2012) is 24.3%; the synthetic training sets are 0.0%.

The consequence is specific rather than diffuse. A pile of origin points between two touching objects is, to a network that reads depth as geometry, a surface bridging them — so the boundary the depth channel was supposed to supply disappears and the two objects merge. That is exactly the failure in Fig. 1, and it is why the effect concentrates where invalid pixels fall on the objects themselves rather than on the background (Section 6).

This paper makes three contributions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A diagnosis, with controlled experiments. Published comparisons report sensitivity of RGB-D UOIS methods to unreliable depth (Cao et al., 2024). We identify input *encoding* as a substantial contributor: tested point-cloud pipelines collapse while the image-depth baseline does not. We test two possible explanations. The image-encoded model also inpaints its depth, so we switch the inpainting off inside it with all else fixed — the effect is 0.2 points. And because a cross-codebase contrast leaves architecture confounded, we move the invalid points elsewhere along their own camera rays inside a single network, changing nothing else: the published origin convention is indeed the worst in-range choice on clean input, but merely relocating the points does not repair anything. Permuting ProxFill's own fill values across the invalid pixels — the right depths at the wrong pixels — costs 24.6 Object-F. What is needed is not plausible depth but *locally* correct depth (Section 3).

A calibrated measurement protocol. Our initial diagnostic removes i.i.d. pixels, which at a matched missing fraction produces $\approx$14,600 components where the real sensor produces $\approx$49 holes averaging $2{,}000$ px. It collapses tested point-cloud encodings at 0.5% severity while structured dropout at the same rate costs one point, so any cross-method ordering read off those columns reflects the encoding rather than the method. We replace it with a generator that resamples each sensor's own invalid-region shapes and verify the distribution match (Section 4).

A fix that costs nothing. ProxFill: fill an invalid point only if a real measurement lies within τ pixels. One scalar, selected on a disjoint scene split, no training and no added parameters. It improves the official checkpoints of two backbones, keeps them flat under degradation that costs the published pipelines 14 to 16 Object-F, and makes the hand-tuned depth-validity post-filter redundant — which leaves the corrected pipeline faster than the original (Sections 5 and 6).

2 Related work

RGB-D UOIS. UOIS-Net (Xie et al., 2021) separates depth-driven mask proposals from RGB refinement. UCN (Xiang et al., 2021) instead learns per-pixel RGB-D embeddings with a metric loss and groups them by mean shift, consuming depth as an organized point cloud; MSMFormer (Lu et al., 2024) builds on UCN's embeddings and replaces mean shift with transformer layers, so it inherits the same input convention. UOAIS (Back et al., 2022) adds amodal reasoning. All are trained on synthetic data and evaluated on OCID and OSD.

Responses to unreliable depth. Two strategies dominate. The first adapts the model: FTEA (Zhang et al., 2023) adapts the BatchNorm affine parameters of a simulation-trained UCN at test time, with the convolutional weights frozen. The second abandons depth: UOIS-SAM (Cao et al., 2024) segments from RGB using a promptable foundation model (Kirillov et al., 2023) and reports that RGB-D methods drop sharply on datasets rich in transparent and reflective objects (Wang et al., 2022; Jung et al., 2024), attributing this to "sensitivity to noisy depth data". Neither line asks whether the input encoding of invalidity is itself wrong.

Filling invalid depth. A large literature already fills the pixels a depth sensor does not return, including for the exact surfaces that cause the failure: Sajjan et al. (2020) detect transparent surfaces, *remove* their depth and reconstruct it by optimisation, and Fang et al. (2022) contribute 58K real frames for transparent-object completion. General-purpose completion runs from a single RGB-D image (Zhang & Funkhouser, 2018) through sparse-to-dense prediction (Ma & Karaman, 2018) to spatial-propagation networks (Park et al., 2020). What unifies it is the objective: it is trained and evaluated on the *accuracy* of the recovered depth, because its consumers are grasping and reconstruction. Our negative result is aimed at that premise, not at those methods. A completion front-end we trained reaches 35% lower masked depth error than copying the nearest valid measurement and still segments worse (Section K), and the control in Section 3 shows why: a fill with the right distribution of depths and the wrong assignment of them to pixels loses 24.6 Object-F. Fill accuracy is not the quantity that matters, which is why a threshold beats a regressor.

Corruption protocols and sensor noise. Severity-graded image corruptions are studied by Hendrycks & Dietterich (2019). We use i.i.d. depth removal as a diagnostic baseline, not as a protocol attributed to that work. It poorly matches the observed hole geometry of OSD and OCID (Section 4). Empirical depth-noise models also exist (Nguyen et al., 2012); our focus is the spatial

pattern of invalid regions rather than the measurement error of returned depths. We resample the sensor's own invalid regions.

Post-hoc mask correction. Boundary refiners such as BPR (Tang et al., 2021) improve mask quality after the fact. We show in Section K that post-hoc editing of the *instance set* — dropping or merging masks — has under one point of headroom on a strong backbone, so the gains reported here do not come from that channel.

3 Input encoding as a robustness confound

Two encodings of the same sensor output behave completely differently under degradation, and the difference is large enough to be read as an architectural result. This section tests input encoding as a confound in that comparison.

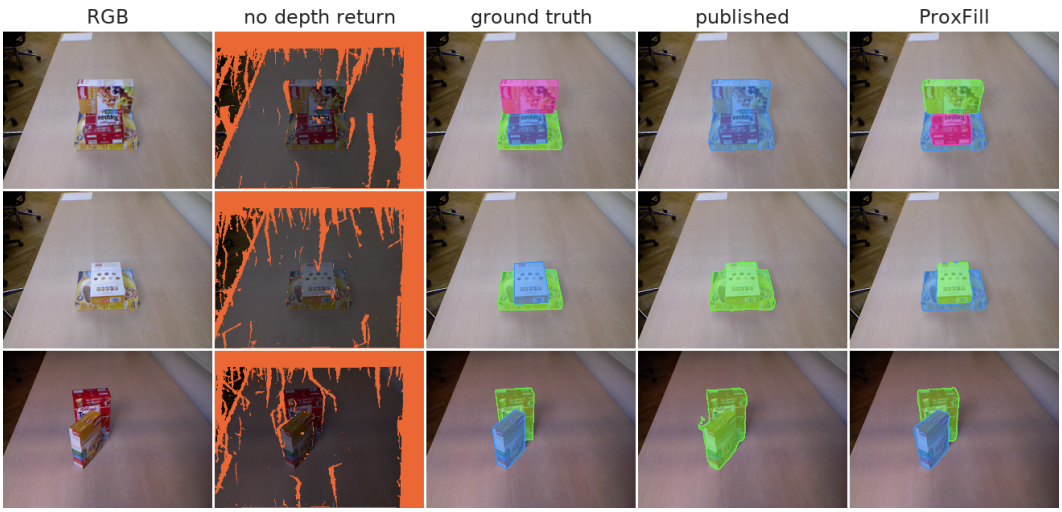


Figure 1: What the flaw does. Left to right: RGB; the pixels where the sensor returned no depth (orange); ground truth; the published pipeline; ProxFill. The published pipeline merges touching objects into a single mask wherever depth is absent — without a valid depth return there is no geometric boundary between them — and a single distance threshold on filling recovers the separation. OSD, $+10\%$ added missing depth. Object-F per row: $45.11 \rightarrow 95.55$, $50.86 \rightarrow 96.26$, $52.89 \rightarrow 95.64$.

What the two encodings do with an invalid pixel. In an organized point cloud the three channels are metric coordinates, so writing $\mathrm{XYZ} = (0, 0, 0)$ places the pixel at the camera centre. An invalid region becomes coincident 3D points at the origin, assigned to different image locations in the organized tensor. Where that region falls between two touching objects, it erases the depth discontinuity that separates them, and the network returns one mask covering both (Fig. 1). The failure is therefore not a loss of information but the insertion of confident false geometry, and it scales with the size of the invalid region rather than with the number of invalid pixels. An image encoding writes the same pixel as an intensity, where the channel carries no metric commitment: an invalid pixel does not directly specify a 3D coordinate. The gap this opens is large enough to be mistaken for an architectural one: under our i.i.d. diagnostic, the point-cloud pipeline emits zero instances once a further 0.5% of pixels lose their return, while the image-encoded model is unmoved across the whole range. That particular number is an result of a protocol that poorly matches observed sensor failures (Section 4); under a calibrated one the collapse becomes a 14–16 point decline rather than a cliff — but the ordering it induces remains similar. Such comparisons need to control input handling before attributing the gap to architecture.

It is not the origin, and it is not the values: it is locality. The argument so far contrasts two codebases, which leaves architecture as a confound. We therefore test the mechanism inside a single network with no training and no weight change, by moving the invalid points elsewhere along their

Table 1: Inpainting is not the cause. DA-Fusion with its depth inpainting switched off, everything else fixed, under the calibrated corruption of Section 4. Object-F.

	OSD-111		OCID-200	
Severity	inpaint on	inpaint off	inpaint on	inpaint off
clean	88.82	89.03	85.25	85.16
+5%	88.96	88.71	85.41	85.14
+10%	88.00	87.78	85.25	84.95
+20%	80.45	79.42	85.15	84.51

own camera rays — intrinsics recovered per frame by least squares from the valid points, which is exact here (0.0 px reprojection error). The invalid pixel set is identical in every variant; only the 3D position assigned to it differs. The depth-validity post-filter is disabled throughout, because it keys on $z \leq 0$ and would otherwise be active for the published encoding and inert for every relocated one (Table 8).

Two things follow. The published convention is the worst in-range choice on clean input: moving these points to *any* depth inside the frame's own valid range, even an obviously wrong one, recovers 4.4 Object-F, so $\mathrm{XYZ} = (0, 0, 0)$ is not a neutral way to spell "no data". But relocating them is not the repair either. The SHUFFLED row permutes the depth values PROXFILL would have written across the invalid pixels: the multiset of depths is exactly right and only the assignment of value to pixel is wrong, and it loses 24.6 Object-F under degradation. What the backbone needs is not depth that is plausible in aggregate but depth that is correct *locally* — and a real measurement within τ pixels is precisely the condition under which a neighbour's depth is locally correct. That is the whole method, reached by elimination rather than assertion.

Encoding, not inpainting. DA-Fusion (Park et al., 2025) encodes depth as a normalized single-channel image, where an invalid pixel becomes an intensity extreme rather than a 3D outlier, and additionally Telea-inpaints (Telea, 2004) its depth before inference. Under missing depth it is unaffected while the point-cloud methods collapse, which admits two explanations: the encoding, or the inpainting. We disambiguate by disabling the inpainting inside DA-Fusion with all else held fixed (Table 1). Removing it changes clean Object-F by +0.21 on OSD and costs at most 1.03 points at the highest severity. The inpainting is not what protects the model; the encoding is.

4 A SENSOR-CALIBRATED CORRUPTION PROTOCOL

We first diagnose robustness by removing pixels i.i.d. at a target rate. This does not match the observed sensors. We measured the invalid regions of both benchmarks directly: OSD has 49 connected invalid components per frame with a mean area of 2,017 px (median 4, heavy-tailed), concentrated at the frame border by 2.82×; OCID has 10 components of mean area 3,642 px, border-concentrated by 7.10×. Removing i.i.d. pixels at a matched fraction instead produces ≈14,600 components of mean area 7 px — two to three orders of magnitude off in both statistics (Table 2).

The consequence is not cosmetic. Table 3 shows the dose response of UCN with the post-filter disabled. Under i.i.d. removal the model emits *zero* instances at 0.5% severity — a cliff, not a curve — while structured removal at the same rate costs one point and degrades smoothly to 51.70 at 25%. Since OSD is already 24% invalid natively and UCN handles that, the collapse cannot be a response to missing data per se; it is a response to scattered origin points. Any cross-method robustness ordering read off i.i.d. columns is therefore an artifact of how each method encodes depth.

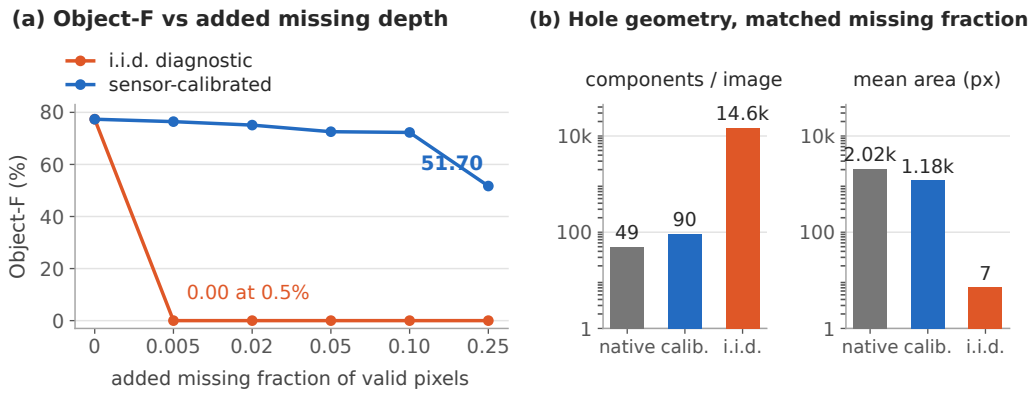


Figure 2: Independent pixel removal differs from observed sensor failures. **(a)** UCN on OSD with the post-filter off: i.i.d. pixel removal produces a cliff — zero instances at 0.5% added missing depth — while sensor-calibrated removal degrades smoothly. **(b)** At a matched total missing fraction the two models differ by two to three orders of magnitude in both the number of invalid regions and their size; the calibrated generator stays within an order of magnitude of the native sensor statistics.

Table 2: Hole geometry at matched total missing fraction (OSD, $+10\%$ severity).

Model	missing	holes/img	mean px
native sensor	24.3%	49	2,017
calibrated (this work)	32.0%	**90**	**1,176**
i.i.d. diagnostic	31.8%	14,582	7

Table 3: Dose response, UCN on OSD, post-filter off. Object-F.

Added missing	i.i.d.	structured
0	77.37	77.37
0.005	**0.00**	76.44
0.02	0.00	75.11
0.05	0.00	72.57
0.10	0.00	72.29
0.25	0.00	51.70

A calibrated replacement. We avoid inventing a noise model. Our generator harvests the connected components of a dataset's *own* native invalid mask into a shape bank and pastes sampled shapes at locations drawn from that dataset's empirical invalid-pixel distribution; shapes are never harvested from the frame being corrupted, and severity is expressed against valid pixels so it is comparable across sensors with different native rates. Components spanning more than a quarter of the frame are excluded from the bank: those are structural field-of-view artifacts rather than failures a sensor produces more of as conditions worsen. Table 2 verifies the match: at an identical total missing fraction the calibrated generator stays within an order of magnitude of the native statistics on both axes, where i.i.d. is off by 160–$540\times$.

5 One Scalar

Fig. 3 shows in cross-section what a repair has to achieve. An origin point is harmful because it asserts a surface where there is none, so the fix is to stop asserting it, and the obvious way — copy the nearest valid measurement into the hole — is right for the reason the flaw is wrong: inside one surface a neighbour's depth is a good estimate, and because the nearest neighbour differs on the two sides of a gap, the boundary comes back rather than being smoothed away.

The same operation fails once the hole is large, when the nearest valid neighbour may belong to a different surface and copying it manufactures a plane spanning the gap — the same error as the origin point, relocated. Filling is therefore neither globally good nor bad; it is good where the hole is small enough that its neighbours are still the same surface. The measurements confirm the asymmetry: unconditional filling helps on OSD, whose dropouts are small and sit on the objects, and *loses to doing nothing* on OCID at high severity, where they are large and border-concentrated (Table 5).

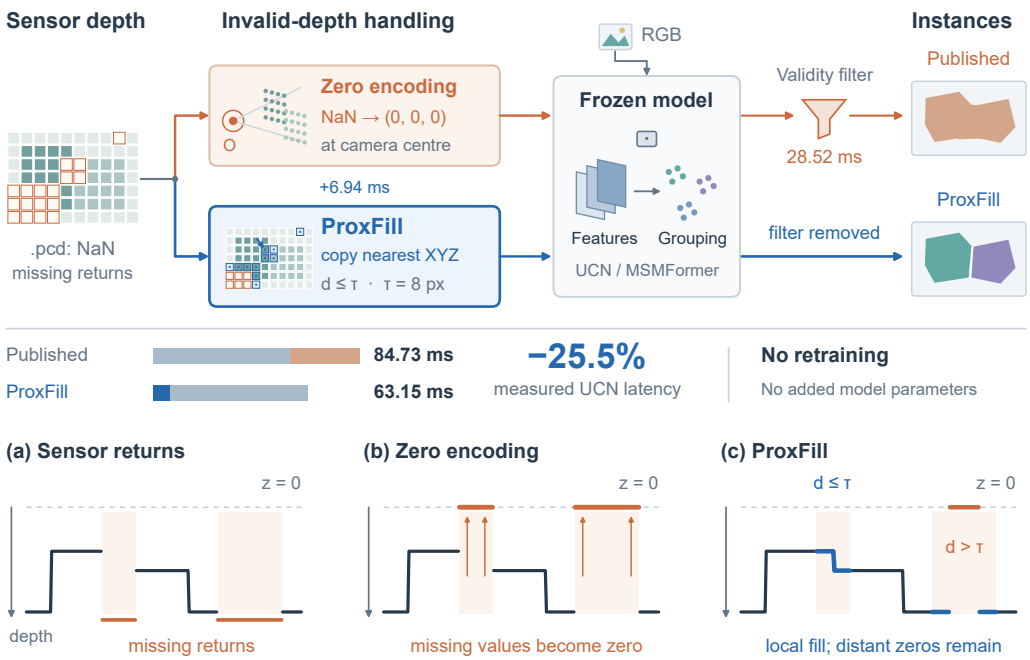


Figure 3: **Top: the published and corrected input pipelines.** The loader maps missing returns to the optical centre; PROXFILL instead copies nearby measured XYZ triplets. Both routes use the same released weights. Adding the fill and removing the depth-validity filter reduces the measured UCN stage sum from 84.73 to 63.15 ms (25.5%); file I/O and crop refinement are excluded (Table 13). The grids and instance masks are schematic. **Bottom: the gate in a depth cross-section. (a)** Missing returns leave gaps in the measured profile. **(b)** Zero encoding moves every missing return to $z = 0$. **(c)** Copying nearby returns preserves the local depth step; only the rims of a wide hole are filled, while its distant interior retains zeros. Diagram distances are not to scale.

Let $\mathcal{V} = \{(u, v) : p_z(u, v) > 0\}$ be the measured pixels. The nearest valid neighbour and its image-space distance define the gate:

$$\mathcal{N}(u, v) \in \arg\min_{\mathbf{q} \in \mathcal{V}} \|(u, v) - \mathbf{q}\|_2, \qquad d(u, v) = \|(u, v) - \mathcal{N}(u, v)\|_2. \tag{1}$$

$$\hat{\mathbf{p}}(u, v) = \begin{cases} \mathbf{p}(u, v) & \text{if } (u, v) \text{ has a return,} \\ \mathbf{p}\big(\mathcal{N}(u, v)\big) & \text{if } d(u, v) \le \tau, \\ \mathbf{0} & \text{otherwise,} \end{cases} \tag{2}$$

Measured returns and fully invalid inputs are unchanged. The implementation uses a labelled L2 approximation (Section C); $\tau = 0$ recovers the published behaviour and $\tau = \infty$ unconditional filling. We call this PROXFILL: proximity-gated filling of invalid depths.

Protocol for τ. We select $\tau = 8$ on 120 OCID ARID10 frames by maximising mean clean and $+20\%$ Object-F over six candidates (Eq. (3) and Table 15). ARID20, YCB10 and all of OSD are held out, making OSD a cross-dataset test. Nothing is trained and no parameters are added.

Is the scalar knife-edge? A method with one hyperparameter invites the question of whether its value was fitted to the test data, so we swept τ over fifteen values on both the selection split and a sensor never used for selection (Table 16). The optimum is a plateau, not a spike — on ARID10 every τ from 4 to 12 lands within 0.5 of the best — and the value chosen there, τ=8, is the outright argmax on OSD-111, which played no part in choosing it. The two datasets also disagree about the extremes in opposite directions: filling everything is catastrophic on ARID10 at $+20\%$ (76.54,

below the 83.94 of filling nothing), filling nothing is catastrophic on OSD (70.29 against 87.09). No fixed extreme serves both, which is the previous paragraph's asymmetry appearing as a curve.

6 EXPERIMENTS

Setup. We use the official public checkpoints of UCN (Xiang et al., 2021) and MSMFormer (Lu et al., 2024) without modification, and the standard UOIS evaluator (Overlap and Boundary P/R/F by Hungarian assignment, plus F@75, the fraction of ground-truth objects recovered at Overlap-F > 0.75). `UCN+` and `MSMFormer+` denote the published two-stage variants with crop refinement. Degradation uses the calibrated generator of Section 4.

A noise floor, measured rather than assumed. UCN samples mean-shift seeds from the torch RNG, so we reset that stream before every inference. This is necessary but not sufficient, because the clustering contains CUDA reductions that are not run-to-run reproducible: re-running the same comparison on the same frames with the same seed, per-frame Object-F still moves on a quarter to a half of frames, by up to 21 points on one frame. The *paired* quantity is stable. Over five repetitions the OSD-111 gain has standard deviation 0.17 clean (mean $+2.89$) and 0.64 at $+20\%$ (mean $+17.79$); over three on 400 held-out OCID frames, 0.035 clean (mean $+0.17$) and 0.094 at $+20\%$ (mean $+3.66$). Even the smallest effect we report, the OCID clean gain, is several times its own deviation and never changed sign. We quote this floor rather than assume a determinism we do not have.

Clean accuracy. Table 4 reports OSD-111 with every method run through one evaluator. The lower block is the claim: the same scalar, with τ fixed once on a disjoint split and never retuned, improves four published checkpoints — two UCN stages and two MSMFormer stages, two different architectures — on all five metrics, by $+2.51$ to $+6.04$ Object-F and $+6.93$ to $+15.11$ Boundary-F. Because nothing is trained and no parameter is added, each of those deltas isolates the input encoding. Against the unmodified reference points, the best corrected checkpoint takes the best recall and the best F@75 — the fraction of objects recovered at Overlap-F > 0.75, which counts whole objects rather than partial credit. One row beats it on Object-F: a DA-Fusion variant we trained ourselves, which reaches 90.26 against 90.05 using 462M parameters and 30,000 iterations of training, an order of magnitude more model for 0.21 of a point. We report it because it is the strongest thing we ran, and because it is in the table as the existence proof for Section 3 — an image encoding is immune to the flaw — rather than as a competitor at the same cost. Boundary-F is where it does not lead: methods with a refinement decoder or an explicit boundary refiner are up to 6.6 points better, because a distance threshold on filling does not sharpen a mask edge. Those cost a training run; this costs a distance transform.

Robustness. Table 5 sweeps both datasets under the calibrated protocol, each dataset from a single run so that its twelve cells are exactly paired. On OSD the published pipeline falls from 86.19 to 69.98 Object-F at $+20\%$ severity while PROXFILL goes from 89.17 to 88.85 — a macro gain of 10.6 Object-F and 17.7 Boundary-F. On OCID the published pipeline degrades more gently, from 89.32 to 86.49, and PROXFILL does not degrade at all: 89.64 clean, 89.66 at $+20\%$, which is above the published pipeline's *clean* accuracy. The unconditional-fill column is what makes the argument: on OSD it captures most but not all of the gain, and on OCID at $+20\%$ it drops to 85.32, *below* the 86.49 of leaving the holes alone. Filling is not what helps; filling only where a real measurement is close is.

Significance, on every held-out split separately. OCID bundles ARID10, ARID20 and YCB10, and τ was fitted on ARID10 alone, so splitting the held-out frames by sub-collection costs nothing and turns one held-out set into three — one of which, YCB10, carries a different object inventory. Table 6 crosses all three with all three metrics under a paired image-level bootstrap. Under degradation every interval on every split excludes zero. On clean input the OCID intervals straddle it, which is the mechanism asserting itself rather than a weakness: OCID objects carry essentially no invalid depth (Table 14), so a threshold worth three points on OSD should be worth nothing there. Across three independent corruption draws on OSD-111 the clean gain is $+2.93$ (range $+2.74$ to $+3.05$) and the $+20\%$ gain $+18.04$; every degraded entry in Table 6 exceeds the run-to-run floor above by more than an order of magnitude. ARID20 and YCB10 do share OCID's sensor with the selection

Table 4: OSD-111, clean input. Every row is our own run under one shared evaluator, so the columns are comparable to each other; they are not comparable to numbers reported in the original papers, which is the point of Section 4. *Cost* is what the row spends beyond a released checkpoint. The lower block is the result: one unchanged scalar improves four published checkpoints spanning two architectures, on every metric, while adding no parameters and training on nothing — so each Δ is attributable to the input encoding alone. The highest clean Object-F in the table belongs to a 462M-parameter Swin-L model we trained for 30,000 iterations, which it wins by 0.21.

Method	depth encoding	cost	Obj-P	Obj-R	Obj-F	Bnd-F	F@75
Reference points, reproduced but not modified here							
ZISVFM (Zhang et al., 2025)	RGB + VFM	zero-shot	78.22	74.55	75.07	51.18	67.21
UncOS (Fang et al., 2024)	RGB + VFM	zero-shot	87.61	85.85	86.48	81.01	80.16
UCN+ + BPR	point cloud	COCO refiner	87.40	88.28	87.80	78.84	80.25
DA-Fusion	depth image	462M, 10k it.	89.79	87.98	88.82	80.15	83.02
DA-Fusion	depth image	462M, 30k it.	**90.77**	89.87	**90.26**	**83.99**	86.28
Point-cloud pipelines: released checkpoint, then the same checkpoint corrected							
MSMFormer	point cloud	—	79.54	86.45	82.81	60.61	79.41
PROXFILL	point cloud	+0 params	87.96	89.83	88.85	75.72	84.31
Δ			+8.42	+3.38	+6.04	+15.11	+4.90
UCN	point cloud	—	84.27	88.42	86.26	67.38	79.78
PROXFILL	point cloud	+0 params	86.93	91.26	89.03	74.44	85.47
Δ			+2.66	+2.84	+2.77	+7.06	+5.69
MSMFormer+	point cloud	—	87.07	86.28	86.50	68.30	80.55
PROXFILL	point cloud	+0 params	88.85	89.25	89.01	76.21	86.20
Δ			+1.78	+2.97	+2.51	+7.91	+5.65
UCN+	point cloud	—	87.18	87.43	87.28	70.50	82.45
PROXFILL	point cloud	+0 params	89.39	**90.75**	90.05	77.43	**87.81**
Δ			+2.21	+3.32	+2.77	+6.93	+5.36

Table 5: Robustness under the calibrated protocol, invalid handling as the only variable. Object-F / Boundary-F; OSD-111 is UCN, OCID is the 1,570 held-out ARID20+YCB10 frames. Unconditional filling is the control that matters: it trails PROXFILL on OSD, whose dropouts are small and sit on the objects, and on OCID at +20% it falls *below doing nothing* (85.32 against 86.49), because nearest-valid fill drags a surface across OCID's large border dropouts. No fixed policy serves both datasets; the distance gate does.

	OSD-111			OCID held out		
Severity	published	uncond. fill	PROXFILL	published	uncond. fill	PROXFILL
clean	86.19 / 67.41	88.66 / 73.64	**89.17 / 74.62**	89.32 / 78.81	89.04 / 79.75	**89.64 / 79.64**
+5%	84.34 / 63.71	88.29 / 73.03	**89.19 / 74.58**	88.69 / 76.52	88.46 / 78.70	**89.63 / 79.55**
+10%	80.80 / 58.27	87.88 / 71.46	**88.95 / 74.05**	88.18 / 74.78	87.64 / 77.37	**89.62 / 79.39**
+20%	69.98 / 47.15	86.88 / 69.02	**88.85 / 73.68**	86.49 / 70.17	85.32 / 73.83	**89.66 / 79.04**
macro	78.37 / 56.38	87.68 / 71.17	**89.00 / 74.10**	87.79 / 73.82	87.14 / 76.63	**89.64 / 79.33**

split: this shows τ is not fitted to one scene collection, and OSD remains the only fully independent sensor.

Transfer. Table 10 applies the same τ to MSMFormer with nothing retuned. The published model loses 14.51 Object-F between clean and +20% severity and the published two-stage variant loses 15.05; with PROXFILL both are flat to within 0.8, and the PROXFILL two-stage model at +20% (89.18) is above the published one on undegraded input (86.50). PROXFILL also beats unconditional filling at every severity and on both stages, which is the pattern Table 5 showed on a different backbone and dataset.

Cost. PROXFILL is a distance transform and a gather: 6.94 ms on CPU against an 84.73 ms pipeline. It also retires the depth-validity post-filter, which alone costs 28.52 ms — dropping that filter changes Object-F by −0.15 and saves four times what PROXFILL spends, so the corrected

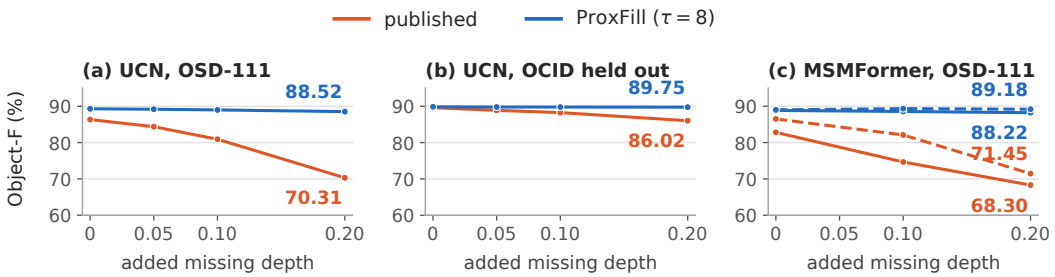


Figure 4: Robustness under the calibrated protocol. Every published pipeline loses accuracy monotonically as depth degrades; with PROXFILL the curves are flat — on OCID from 89.85 to 89.75 across the whole range. Dashed lines in **(c)** are the two-stage MSMFormer+ variant, whose PROXFILL form at 20% added missing depth (89.18) sits above the published one on undegraded input (86.50). Object-F; end values labelled.

pipeline is both more accurate and 25.5% faster with no added parameters. Table 13 breaks the per-frame budget down by stage.

When the flaw matters. The governing variable is not the frame's invalid fraction but the fraction *inside ground-truth objects*: 0.028 on OSD and 0.000 on OCID, which is why OSD gains three points on clean input and OCID gains a third of one. Table 14 stratifies 261 held-out frames by that quantity. The gain is significant exactly where objects carry missing depth and absent where they do not. The relation is a threshold rather than a trend (Spearman $+0.309$, Pearson $+0.048$), so it predicts *whether* the flaw matters on a new sensor without predicting a magnitude.

A learned decision does not do better. The obvious objection to a single threshold is that we did not try hard enough to learn one. We did, three times, each fitted on a domain disjoint from the split it was scored on, and all three lose: a per-instance gate over 21 descriptors (74.45 macro Object-F against 75.45 for doing nothing), a 108K-parameter depth-completion front-end, and a classifier trained directly on the measured change in Object-F when a hole's fill decision is flipped (0.730 under frame-grouped cross-validation against 0.766 for the scalar). Section K gives each in full. The depth-completion result is the informative one: it reaches 35% lower masked depth L_1 than nearest-valid fill and *still* loses downstream on all three splits, at every confidence threshold. Reconstructing the missing geometry more accurately does not segment better — which is the same conclusion the SHUFFLED control reached from the other direction. We do not claim τ is optimal: a greedy per-hole oracle reaches $+0.53$ to $+5.81$ over it, and none of the three learners got near that.

7 CONCLUSION

The gap between a sensor reading and a sensor failing to read is one bit, and RGB-D UOIS pipelines throw it away at load time. Everything in this paper follows from putting it back. The tested robustness ordering is substantially affected by input encoding; independent pixel removal poorly matches the observed sensor holes; and local invalid-point handling recovers several points on two published checkpoints takes one threshold, no training, and less compute than the post-processing it replaces.

The wider lesson is about measurement rather than method: a benchmark number is a claim about a model only to the extent that the input pipeline is faithful, and here an unfaithful convention was shared widely enough to look like a property of the models being compared.

Where this applies. The result is bounded, and we state the bounds exactly. It applies to point-cloud encodings; a model encoding depth as an image is already insulated, and for it the reading is the diagnosis, not the repair. It helps in proportion to how much invalid depth falls *on the objects* — the quantity Table 14 separates by, and one any new sensor can be measured on in advance.

USE OF AI ASSISTANCE

Generative AI tools assisted with hypothesis refinement, experimental design, method and evaluation implementation, synthetic depth-corruption generation, data formatting, numerical analysis and interpretation, mathematical notation, literature retrieval, manuscript drafting, figure design and layout. Quantitative claims are based on archived experimental runs, not generated evidence. The artifact provides numerical checks against those records, and references were checked against publisher or original-paper records. The authors remain responsible for the final text, claims and artifacts, including all AI-assisted content. An initially proposed inpainting explanation was tested and revised in light of the ablation in Section 3.

REPRODUCIBILITY STATEMENT

The UCN, MSMFormer and UOIS transfer tests use unchanged public checkpoints. τ selection is confined to OCID ARID10 — ARID20, YCB10 and all of OSD are held out. The anonymised artifact accompanying this submission contains the method, the calibrated corruption generator and validator, subset Overlap-metric code, and archived records supporting the reported comparisons. The numerical audit checks 281 asserted values against these records; additional scripts regenerate the supplementary tables and verify the new transfer and locality experiments against their per-frame records. These checks require no GPU, dataset or checkpoint. Full inference reruns require the public datasets, released model code and checkpoints. The clean accuracy tables retain the depth-validity filter; the faster operating point removes it (Table 13). Inference is not bit-reproducible, so paired differences and the reported repetition variability should guide comparisons.

REFERENCES

Seunghyeok Back, Joosoon Lee, Taewon Kim, Sangjun Noh, Raeyoung Kang, Seongho Bak, and Kyoobin Lee. Unseen object amodal instance segmentation via hierarchical occlusion modeling. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

Rui Cao, Chuanxin Song, Biqi Yang, Jiangliu Wang, Pheng-Ann Heng, and Yun-Hui Liu. Adapting segment anything model for unseen object instance segmentation. *arXiv preprint arXiv:2409.15481*, 2024.

Hongjie Fang, Hao-Shu Fang, Sheng Xu, and Cewu Lu. TransCG: A large-scale real-world dataset for transparent object depth completion and a grasping baseline. *IEEE Robotics and Automation Letters*, 7(3):7383–7390, 2022.

Xiaolin Fang, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Embodied uncertainty-aware object segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024. arXiv:2408.04760.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019. arXiv:1903.12261.

HyunJun Jung, Shun-Cheng Wu, Patrick Ruhkamp, Guangyao Zhai, Hannah Schieber, Giulia Rizzoli, Pengyuan Wang, Hongcheng Zhao, Lorenzo Garattoni, Sven Meier, Daniel Roth, Nassir Navab, and Benjamin Busam. HouseCat6D: A large-scale multi-modal category level 6D object perception dataset with household objects in realistic scenarios. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. arXiv:2212.10428.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Yangxiao Lu, Yuqiao Chen, Nicholas Ruozzi, and Yu Xiang. Mean shift mask transformer for unseen object instance segmentation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2760–2766, 2024. doi: 10.1109/ICRA57147.2024.10610943.

Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

Chuong V. Nguyen, Shahram Izadi, and David Lovell. Modeling Kinect sensor noise for improved 3D reconstruction and tracking. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pp. 524–530, 2012.

Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *European Conference on Computer Vision (ECCV)*, 2020. arXiv:2007.10042.

Yesol Park, Hye-Jung Yoon, Juno Kim, and Byoung-Tak Zhang. DA-Fusion: Deformable attention-based RGB-D fusion transformer for unseen object instance segmentation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7490–7496, 2025. doi: 10.1109/ICRA55743.2025.11128151.

Andreas Richtsfeld, Thomas Mörwald, Johann Prankl, Michael Zillich, and Markus Vincze. Segmentation of unknown objects in indoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

Shreeyak Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. ClearGrasp: 3D shape estimation of transparent objects for manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. arXiv:1910.02550.

Markus Suchi, Timothy Patten, David Fischinger, and Markus Vincze. EasyLabel: A semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

Chufeng Tang, Hang Chen, Xiao Li, Jianmin Li, Zhaoxiang Zhang, and Xiaolin Hu. Look closer to segment better: Boundary patch refinement for instance segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9(1):23–34, 2004.

Pengyuan Wang, HyunJun Jung, Yitong Li, Siyuan Shen, Rahul Parthasarathy Srikanth, Lorenzo Garattoni, Sven Meier, Nassir Navab, and Benjamin Busam. PhoCaL: A multi-modal dataset for category-level object pose estimation with photometrically challenging objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. arXiv:2205.08811.

Yu Xiang, Christopher Xie, Arsalan Mousavian, and Dieter Fox. Learning RGB-D feature embeddings for unseen object instance segmentation. In *Conference on Robot Learning (CoRL)*, volume 155, pp. 461–470. PMLR, 2021. URL https://proceedings.mlr.press/v155/xiang21a.html.

Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. Unseen object instance segmentation for robotic environments. *IEEE Transactions on Robotics*, 2021. arXiv:2007.08073.

Lu Zhang, Siqi Zhang, Xu Yang, Hong Qiao, and Zhiyong Liu. Unseen object instance segmentation with fully test-time RGB-D embeddings adaptation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. arXiv:2204.09847.

Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single RGB-D image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. arXiv:1803.09326.

Ying Zhang, Maoliang Yin, Wenfu Bi, Haibao Yan, Shaohan Bian, Cui-Hua Zhang, and Changchun Hua. ZISVFM: Zero-shot object instance segmentation in indoor robotic environments with vision foundation models. *IEEE Transactions on Robotics*, 2025. arXiv:2502.03266.

APPENDIX

A MORE QUALITATIVE EXAMPLES

Every panel below is a real frame, rendered by the same script and selected the same way: the frames on which the published pipeline and PROXFILL disagree most, out of the first 111 (OSD) or 120 (OCID) of the split. Columns are RGB; the pixels where the sensor returned nothing; ground truth; the published pipeline; PROXFILL.

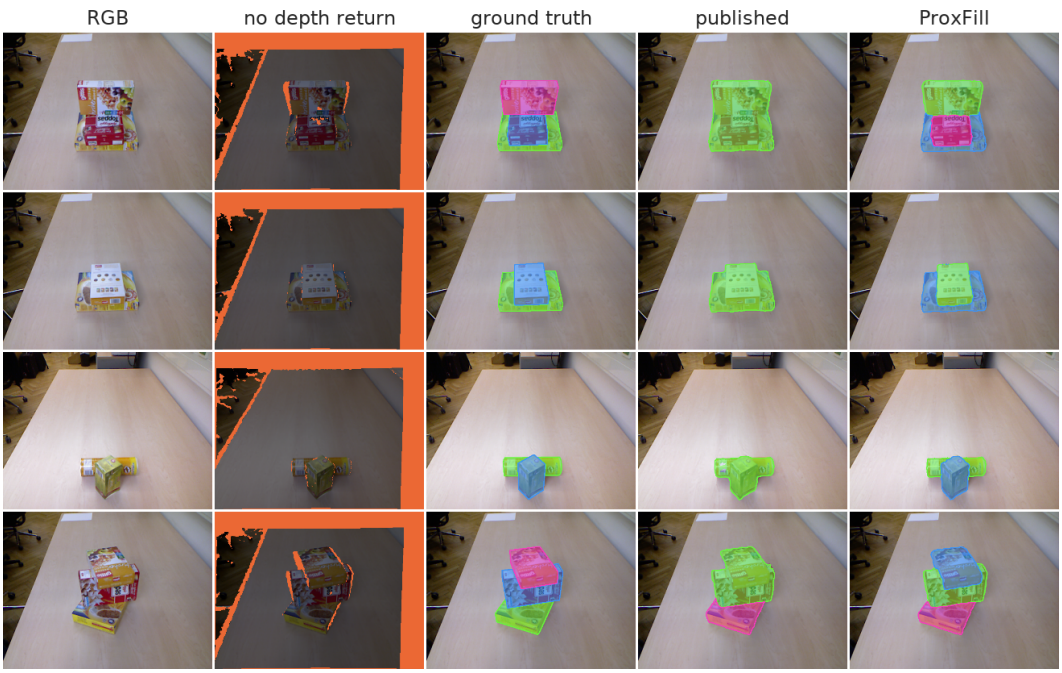


Figure 5: **Unmodified OSD, no added degradation.** This is the figure to read if the concern is that our corruption generator manufactures the effect. Nothing is corrupted here: the invalid pixels are the ones the sensor itself failed to return, and the published pipeline still merges touching objects across them. Object-F per row: $45.08 \rightarrow 95.45$, $51.05 \rightarrow 95.89$, $59.59 \rightarrow 95.52$, $65.14 \rightarrow 95.87$.

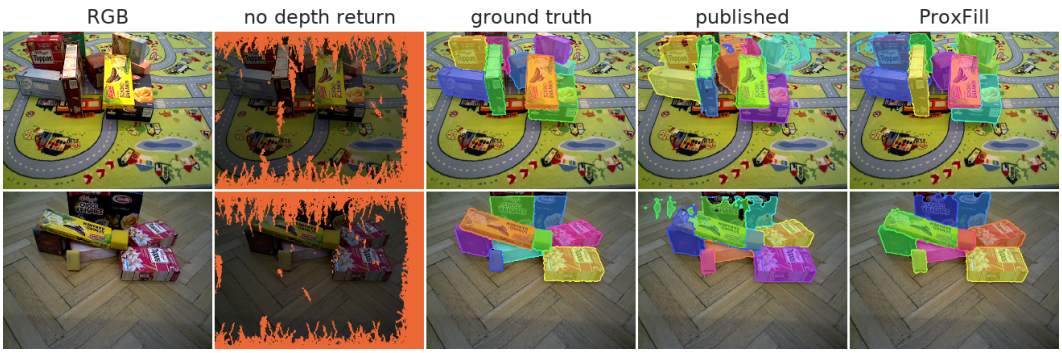


Figure 6: OCID, $+20\%$ added missing depth. OCID's dropouts concentrate at the frame border ($7.10\times$ border affinity), so the published pipeline survives clean input and loses objects only once degradation reaches the surfaces themselves. Object-F per row: $64.72 \rightarrow 91.56$, $78.92 \rightarrow 95.87$.

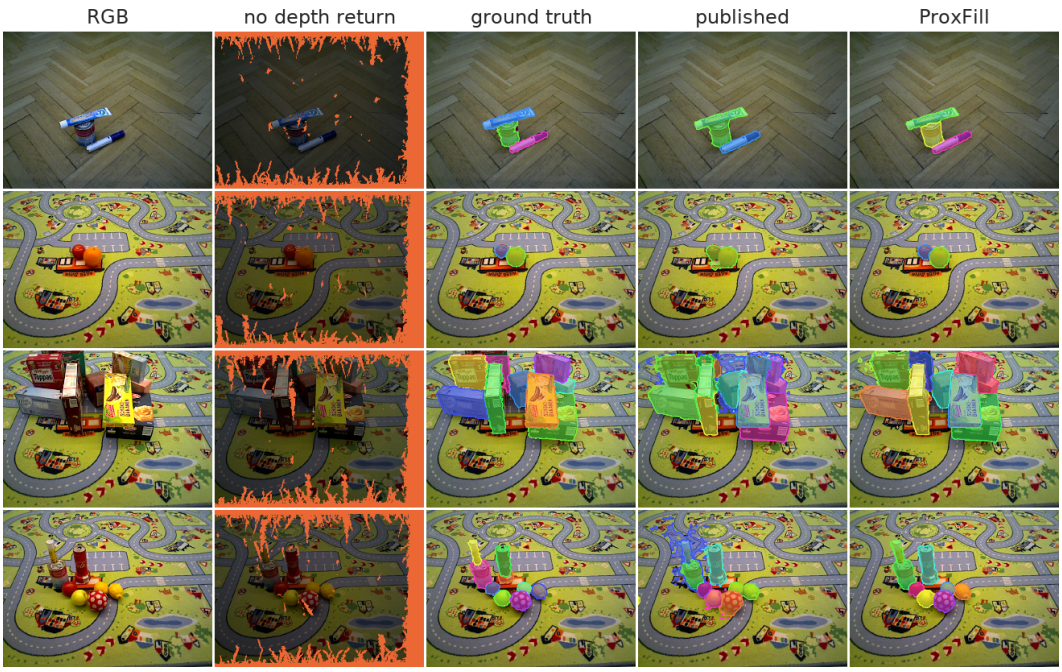


Figure 7: OCID, $+10\%$ added missing depth — half the severity of Fig. 6, and the failure is already present. Object-F per row: $61.02 \rightarrow 93.91$, $59.78 \rightarrow 92.22$, $62.94 \rightarrow 91.57$, $66.99 \rightarrow 87.72$.

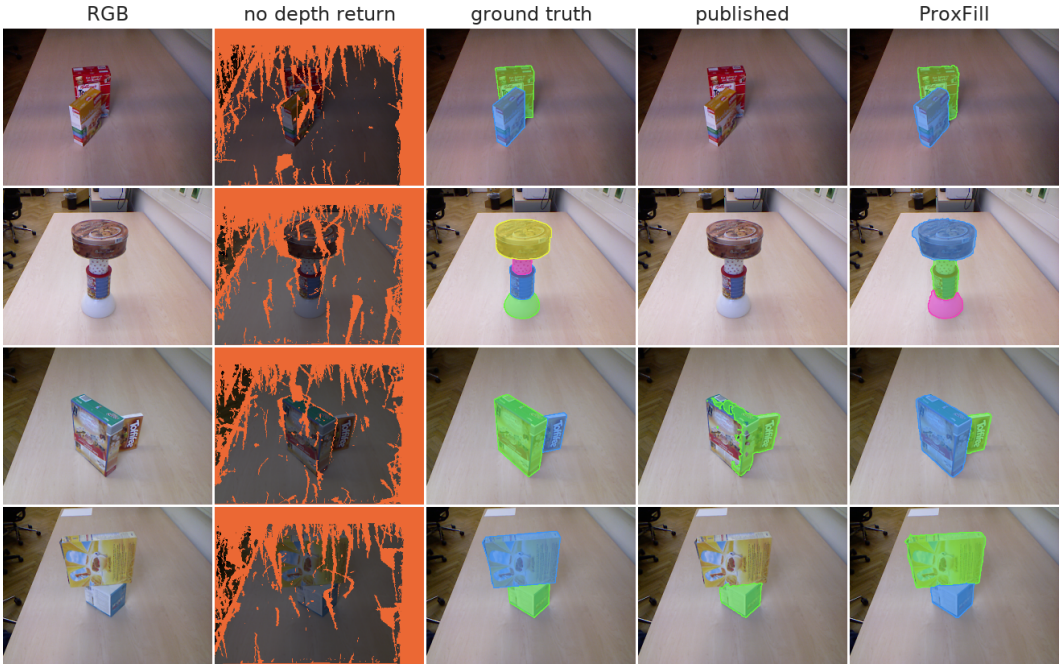


Figure 8: OSD, $+20\%$ added missing depth. The first two rows are frames where the published pipeline returns *no* instances at all and scores 0.00: the depth-validity post-filter deletes every mask, because every mask sits on invalid depth. Object-F per row: $0.00 \rightarrow 95.60$, $0.00 \rightarrow 88.50$, $24.34 \rightarrow 96.93$, $35.44 \rightarrow 97.59$.

B HELD-OUT SIGNIFICANCE, EVERY SPLIT AND METRIC

The threshold is fixed before any frame in this section is scored. OCID ARID10 is used for selection; ARID20, YCB10 and the full OSD-111 set are reserved for evaluation. The two OCID sub-collections share a sensor with the selection split, while OSD supplies the cross-dataset test. We keep the sub-collections separate so that pooling cannot hide which part of OCID carries a gain.

Table 6: Held-out significance. Δ over the published pipeline, paired image-level bootstrap, $10{,}000$ resamples. τ=8 was selected on OCID ARID10, which appears in no row here. OCID is shown both split by sub-collection and pooled, so the pooling can be seen not to be hiding a weak member; YCB10 carries a different object inventory from the ARID splits. On clean OCID, the separate Object-F and F@75 intervals include zero; Boundary-F and pooled Object-F remain positive. The intervals are pointwise and image-level.

Held-out split (frames)	Metric	clean Δ (95% CI)	$+20\%$ Δ (95% CI)
OSD-111 (111)	Obj-F	$+2.86$ $[+1.52, +4.49]$	$+18.24$ $[+14.29, +22.53]$
	Bnd-F	$+7.19$ $[+4.70, +9.98]$	$+25.64$ $[+21.94, +29.56]$
	F@75	$+5.10$ $[+1.97, +8.92]$	$+30.56$ $[+23.90, +37.30]$
OCID ARID20 (1,066)	Obj-F	$+0.32$ $[-0.02, +0.67]$	$+2.77$ $[+2.22, +3.27]$
	Bnd-F	$+0.74$ $[+0.42, +1.06]$	$+8.47$ $[+7.83, +9.05]$
	F@75	$+0.25$ $[-0.20, +0.70]$	$+4.58$ $[+3.77, +5.36]$
OCID YCB10 (504)	Obj-F	$+0.11$ $[-0.11, +0.35]$	$+3.92$ $[+3.27, +4.61]$
	Bnd-F	$+0.69$ $[+0.30, +1.10]$	$+9.73$ $[+8.71, +10.83]$
	F@75	-0.41 $[-1.20, +0.35]$	$+5.65$ $[+4.06, +7.30]$
OCID pooled (1,570)	Obj-F	$+0.25$ $[+0.01, +0.51]$	$+3.14$ $[+2.72, +3.55]$
	Bnd-F	$+0.72$ $[+0.47, +0.98]$	$+8.87$ $[+8.33, +9.40]$
	F@75	$+0.04$ $[-0.35, +0.44]$	$+4.92$ $[+4.17, +5.67]$

What is resampled. For each image we subtract the published score from the score with $\tau = 8$. We resample those paired differences $10{,}000$ times with replacement, using NumPy seed 7, and report the 2.5th and 97.5th percentiles of the bootstrap means. Images, rather than instances, are the sampling unit. These are pointwise intervals without a multiple-comparison correction; they also do not model dependence between successive frames in the same scene. Thus the small clean OCID effect deserves more caution than the large degraded-depth gains.

A separate source of variation. Resampling images does not measure inference nondeterminism. We therefore also repeat inference on the same frames. The torch stream is reset before each UCN variant, but the CUDA clustering reductions can still change individual masks. Table 7 reports the paired quantity across runs. The OSD summary uses five archived repetitions; OCID uses three repetitions on a 400-frame subset of the held-out collection.

Table 7: Repeatability of the paired Object-F gain. OSD uses all 111 frames; OCID uses the same 400 held-out frames in each repetition. SD is across runs, not a confidence interval across images.

Split	Severity	Runs	Mean Δ	SD	Range
OSD-111	clean	5	+2.89	0.168	$[+2.63, +3.05]$
OSD-111	$+20\%$	5	+17.79	0.636	$[+17.06, +18.55]$
OCID held out	clean	3	+0.17	0.035	$[+0.14, +0.21]$
OCID held out	$+20\%$	3	+3.66	0.094	$[+3.55, +3.73]$

C WHERE THE INVALID POINTS ARE PUT

The placement experiment holds the checkpoint, RGB image and input invalid mask fixed. The depth-validity filter is disabled for every variant: its decision depends on $z \leq 0$, so retaining it would make a change of sentinel also a change of post-processing. This is why the absolute scores here differ from the filtered pipeline in the main table.

Table 8: Where the invalid points are put, UCN on OSD-111, post-filter disabled. Same invalid input pixels in every row. Median-plane and shuffled placements use the target camera ray, while ProxFill copies the neighbouring XYZ vector. SHUFFLED carries PROXFILL's own depth values permuted across pixels, so it preserves the depth multiset, including zeros, while changing spatial assignment and ray coordinates. Object-F / Boundary-F.

Placement of invalid points	what it is	clean	$+20\%$
$\mathrm{XYZ} = (0, 0, 0)$	published; a surface at the lens	81.65 / 57.80	70.19 / 45.33
plane at median depth	coherent surface, wrong place	85.61 / 67.32	62.54 / 39.25
SHUFFLED	right depths, wrong pixels	86.09 / 70.85	64.04 / 42.41
PROXFILL	locally correct depth	**89.10 / 74.34**	**88.65 / 72.94**

What each control changes. The median-plane control assigns the frame's median valid depth to every invalid pixel, then recovers XYZ on that pixel's camera ray. Intrinsics are fitted from the valid points of the same frame. The shuffled control first constructs ProxFill's depth values, including the zeros left in wide holes, permutes that multiset across the original invalid pixels, and backprojects the result. ProxFill itself copies the full XYZ vector of its selected neighbour. The shuffled comparison therefore changes both the locations of remaining zeros and the target-ray coordinates; it tests spatial assignment broadly, rather than isolating a single factor.

The implementation. The organized tensor has shape $3 \times H \times W$. After the benchmark loader replaces NaNs, a point is invalid when its z channel is nonpositive. OpenCV's labelled L2 distance transform uses a 5×5 chamfer mask; the resulting approximation to Euclidean pixel distance supplies both the threshold and the source labels. All three coordinates come from the same measured source point. Measured points pass through unchanged, and wide-hole interiors retain zeros.

```
invalid = xyz[2] <= 0
out = xyz.copy()
if invalid.any() and not invalid.all() and tau > 0:
    distance, labels = labelled_L2_transform(invalid)
    target = invalid & (distance <= tau)
    source = valid_pixel_lookup(labels[target])
    out[:, target] = xyz[:, source]
return out
```

The early return handles empty and fully invalid inputs without inventing a source measurement. Setting $\tau = 0$ disables filling; an effectively infinite threshold fills every invalid pixel whenever at least one valid measurement exists. The radius is in image pixels, so resolution is part of the protocol. No RGB, predicted mask, ground-truth label or trained parameter enters the fill decision.

D LOCALITY WITH AN IDENTICAL FILL MASK

The preceding shuffled-depth experiment moves the zeros that remain after filling. To isolate spatial assignment, we now freeze ProxFill's filled-pixel set $T = \{p : z_p \leq 0,\ d(p) \leq 8\}$ before constructing any control. Every filled variant changes exactly T: valid measurements and the unfilled hole interiors remain bitwise unchanged. The input RGB, checkpoint and corruption are shared.

Two matched contrasts. Shuffling the copied XYZ triplets only within T preserves the full multiset of three-dimensional points while breaking the association between a point and its neighbouring image region. A second pair backprojects either the nearest-neighbour depths or a permutation of those same positive depths onto the target camera rays. Within this pair, both the fill mask and depth multiset are identical, and only the depth-to-ray assignment changes. These interventions retain the original zeros outside T.

Table 9: Controls with identical fill masks. OSD cells are mean ± sample standard deviation across three inference runs on all 111 frames. OCID uses 200 uniformly spaced ARID20/YCB10 frames and one run. All scores are percentages; the depth-validity filter is off throughout.

	Clean			+20% missing		
Placement	Obj-F	Bnd-F	F@75	Obj-F	Bnd-F	F@75
OSD-111, three inference repetitions						
Unfilled zeros	81.82±0.04	57.75±0.17	80.01±0.10	70.38±0.23	45.29±0.11	58.53±0.49
PROXFILL: local XYZ	88.98±0.17	74.36±0.20	85.37±0.53	88.51±0.11	72.86±0.14	84.86±0.42
Nearest depth, target ray	89.15±0.01	74.51±0.02	85.95±0.04	88.71±0.10	73.15±0.04	85.49±0.31
Shuffled XYZ triplets	84.91±0.02	67.53±0.33	77.40±0.22	54.60±0.64	34.57±0.62	38.38±0.53
Shuffled depth, target ray	85.14±0.22	68.01±0.34	78.37±0.37	57.94±0.17	37.20±0.22	42.68±0.37
OCID held out, 200 frames						
Unfilled zeros	90.12	79.34	81.93	87.16	71.14	77.56
PROXFILL: local XYZ	90.34	79.96	82.79	89.71	78.39	82.05
Shuffled XYZ triplets	90.13	78.26	82.02	82.56	63.93	71.90

What the controls establish. On OSD, ray-projected nearest depth and copied local XYZ give similar accuracy, while both shuffled variants lose ground, especially with added missing depth. This supports the importance of local image–geometry correspondence; it does not show that copying the source ray is necessary. A global permutation is a deliberately strong nonlocal perturbation, so this comparison does not establish optimality over other local completion rules.

Calibration and repetition. The OSD camera-ray fit is checked against valid input points on every frame (maximum residual below 0.1 pixel). The OCID organized clouds do not satisfy this pinhole fit to that tolerance, so we report only the exact XYZ-copy and triplet-shuffle contrast there. No approximate ray control is treated as exact. The three OSD runs keep the corruption, permutation and clustering seeds fixed; their standard deviations measure inference variation, not uncertainty over possible shuffles or scene sampling. No parameter is retuned.

E CROSS-BACKBONE TRANSFER UNDER DEGRADATION

The transfer test uses the same $\tau = 8$ on both official MSMFormer checkpoints, with the original predictor settings. No threshold is selected on OSD. We report all three metrics below so that stability in Object-F can be checked against boundaries and the fraction of objects recovered above the 0.75 overlap threshold.

Table 10: MSMFormer on OSD-111 under the calibrated protocol. Official checkpoints, the same $au = 8$, and no retuning. Both stages and all recorded severities are shown; bold marks the ProxFill rows.

Severity	Invalid handling	Object-F	Boundary-F	F@75
MSMFormer				
clean	published (zeros)	82.81	60.61	79.41
clean	unconditional fill	87.90	74.27	84.76
clean	PROXFILL	**88.85**	**75.72**	**84.31**
$+10\%$	published (zeros)	74.66	49.74	69.55
$+10\%$	unconditional fill	87.32	72.40	85.30
$+10\%$	PROXFILL	**88.56**	**75.59**	**83.58**
$+20\%$	published (zeros)	68.30	42.34	57.24
$+20\%$	unconditional fill	85.43	68.56	85.18
$+20\%$	PROXFILL	**88.22**	**74.01**	**84.37**
MSMFormer+				
clean	published (zeros)	86.50	68.30	80.55
clean	unconditional fill	88.82	75.96	86.23
clean	PROXFILL	**89.01**	**76.21**	**86.20**
$+10\%$	published (zeros)	82.10	63.06	72.95
$+10\%$	unconditional fill	88.78	75.64	86.31
$+10\%$	PROXFILL	**89.34**	**76.64**	**86.47**
$+20\%$	published (zeros)	71.45	50.81	55.27
$+20\%$	unconditional fill	88.27	75.18	87.27
$+20\%$	PROXFILL	**89.18**	**76.17**	**86.69**

Between clean input and $+20\%$, the published base and crop models lose 14.51 and 15.05 Object-F, respectively. Their ProxFill counterparts change by -0.63 and $+0.17$. Unconditional filling improves both pipelines, but the distance gate retains an advantage at each recorded severity. All rows within a stage use the same weights; the base and crop stages remain distinct inference configurations.

F TRANSFER TO AN INDEPENDENT DEPTH-SEEDING PIPELINE

UOIS-Net-3D (Xie et al., 2021) provides a different pipeline from UCN and MSMFormer: a depth-seeding network predicts foreground and centre offsets in 3D, initial mask processing cleans the clusters, and a separate RGB network refines the regions. We evaluate both the processed depth masks and the final RGB-refined output, with the same depth intervention applied before the depth-seeding network.

Table 11: Transfer to the official UOIS-Net-3D depth-seeding pipeline. Each block uses the same fixed weights and paired inputs; $\tau = 8$ is inherited from OCID ARID10. OSD uses all 111 frames; OCID uses 200 evenly spaced held-out ARID20/YCB10 frames. These are single-run percentages.

	Clean			+20% missing		
Depth handling	Obj-F	Bnd-F	F@75	Obj-F	Bnd-F	F@75
OSD-111: Depth seeding + initial mask processing						
Unfilled zeros	83.32	70.53	74.80	65.35	36.96	46.11
ProxFill, $\tau = 8$	82.67	65.74	76.02	82.16	62.22	75.20
Unconditional fill	71.24	57.12	74.63	68.86	53.62	75.18
OSD-111: Full pipeline with RGB refinement						
Unfilled zeros	84.01	75.64	73.79	70.00	48.48	56.01
ProxFill, $\tau = 8$	83.16	74.02	75.51	82.97	71.24	75.13
Unconditional fill	70.68	62.32	73.84	68.31	58.84	74.70
OCID held out (200): Depth seeding + initial mask processing						
Unfilled zeros	91.35	85.53	81.17	72.47	58.23	55.32
ProxFill, $\tau = 8$	89.79	82.43	78.79	87.64	79.55	78.00
Unconditional fill	87.88	80.46	76.16	85.69	77.38	74.89
OCID held out (200): Full pipeline with RGB refinement						
Unfilled zeros	89.62	82.89	80.53	75.68	63.45	60.24
ProxFill, $\tau = 8$	88.69	81.32	78.27	86.39	77.96	77.44
Unconditional fill	87.05	79.77	76.33	84.97	76.89	74.76

Weights and settings. We use the authors' released TOD-trained depth seeding checkpoint and OID-trained RGB refinement checkpoint. The settings follow the official example notebook: feature dimension 64, 200 mean-shift seeds, 10 iterations, bandwidth 0.02, connected-component tolerance 0.05, subsampling factor 5, and minimum cluster size 500 pixels. Initial masks use the published 9×9 opening/closing and largest-component processing; refinement uses 224×224 crops with padding fraction 0.25. Final closing is off for the OID checkpoint. These predictor settings are held fixed for every depth policy.

Input and comparison boundaries. RGB uses the official ImageNet normalization. The organized clouds are in metres; their Y coordinate is reversed to match the depth-seeding model's documented X-right, Y-up, Z-forward convention. This conversion is shared by all variants. No additional UCN depth-validity filter is applied. Ground-truth annotations enter only the common segmentation evaluator. The corruption, frame selection and $\tau = 8$ rule follow the preceding held-out protocol; nothing is fitted to these frames. Absolute scores should be compared within each block, since the pipelines use different training and post-processing.

A conditional gain. In the full OSD pipeline, clean Object-F changes from 84.01 to 83.16, whereas at +20% missing depth it changes from 70.00 to 82.97. Thus the strong recovery under degraded depth does not imply a gain in every clean-input metric. The single runs do not quantify inference variability.

G AN EXTERNAL DATASET WITH NON-UNIFORM TRANSFER

HouseCat6D (Jung et al., 2024) supplies different objects and capture scenes. We use the released instance masks to score full-frame segmentation of the annotated objects. Before inference, we select 20 evenly spaced frame indices from each of the five official test scenes. This 100-frame diagnostic is a subset of the 2,929 test frames, and is not reported as the full test split.

Table 12: HouseCat6D: a fixed 100-frame diagnostic subset (20 frames per test scene). VGA cells show mean ± sample standard deviation over three runs; native-resolution cells are one run. No setting is selected on these results. The filter threshold is inherited from OSD; "off" uses the same predictions before filtering. Scores are percentages.

	640 × 480, three runs			1096 × 852, one run		
Policy / filter	Obj-F	Bnd-F	F@75	Obj-F	Bnd-F	F@75
Zeros / off	52.60±0.37	24.54±0.16	40.53±0.31	42.63	21.42	26.30
Zeros / on	50.80±0.28	24.17±0.20	37.43±0.40	42.03	21.16	25.90
PROXFILL / off	51.04±0.16	18.45±0.14	45.90±0.26	54.93	21.73	38.20
PROXFILL / on	51.63±0.12	18.92±0.11	45.90±0.26	55.00	21.72	38.20
Full fill / off	48.37±0.36	17.51±0.15	40.93±0.76	48.07	16.28	35.00
Full fill / on	48.37±0.36	17.51±0.15	40.93±0.76	48.07	16.28	35.00
Per-scene change: ProxFill minus zeros, filter on						
Scene 1	+4.11	−3.94	+17.00	+17.55	+0.26	+17.00
Scene 2	−7.75	−9.74	+0.83	+9.57	+1.29	+11.00
Scene 3	+12.12	−1.06	+14.33	+21.26	+1.65	+14.00
Scene 4	+2.01	−3.81	+9.50	+13.57	+2.23	+13.00
Scene 5	−6.36	−7.68	+0.67	+2.89	−2.61	+6.50

Inputs and fixed choices. The predictor receives the complete RGB image and measured depth; rendered ground-truth depth, pose labels and instance crops are not used as inputs. Depth is converted from millimetres to metres and backprojected with the supplied intrinsics. In the instance map, background value 255 is mapped to zero and object IDs are checked against scene metadata. The VGA condition uses area resizing for RGB, nearest-neighbour resizing for depth and labels, and scaled pixel-centre intrinsics. The native condition keeps the released resolution. Both use the same UCN checkpoint and $\tau = 8$ pixels; the filter threshold 0.8 is carried over from OSD and is also ablated without another model call.

A limit to uniform transfer. The VGA results improve the fraction of detected objects but reduce Boundary-F; Object-F changes also differ across scenes and filter settings. At native resolution, Object-F and detection improve in all five scenes, while Boundary-F still decreases in Scene 5. The larger native-resolution gain illustrates that resolution is part of the method's operating conditions: keeping $\tau = 8$ after resizing changes the radius relative to the original image. This comparison does not isolate that radius change from changes in the predictor's image features. No synthetic corruption is added in either condition.

Uncertainty and task scope. The three VGA runs measure inference variation at fixed frame selection and seeds. Nearby frames within a scene are dependent, and five scenes provide limited independent coverage; these standard deviations are not scene-level confidence intervals. The masks originate from a pose dataset, so these results specifically measure the released annotated objects in full images. No threshold or checkpoint is selected using this diagnostic.

H PER-FRAME COST

The timing experiment measures one stage at a time. Each stage receives three warm-up calls and ten timed repeats on each of twenty OSD frames. CUDA is synchronized before and after timed calls; the fill itself runs on CPU. We first average repeats within a frame, then compute the mean and standard deviation across frames.

Table 13: Per-frame stage timings over 20 OSD frames, with 10 timed repeats and 3 warm-up calls per stage. SD is across the 20 per-frame mean times. The UCN backbone has 42,635,008 parameters; ProxFill adds none.

Stage	Mean (ms)	SD (ms)
PROXFILL (CPU)	6.94	0.81
backbone forward	8.94	0.06
mean-shift clustering	47.27	0.51
depth-validity filter	28.52	3.42

The totals quoted in the paper are sums of these measured stage means:

$$T_{\text{published}} = 8.94 + 47.27 + 28.52 = 84.73\,\text{ms},$$
$$T_{\text{ProxFill}} = 6.94 + 8.94 + 47.27 = 63.15\,\text{ms}.$$

They exclude dataset loading, file decoding and a separately measured crop-refinement pass. The reduction is 25.5% of the original stage sum. Keeping the filter is a separate operating point: the clean accuracy tables retain it, while the faster pipeline removes it. The recorded UCN+ ablation changes Object-F by -0.15 when it is removed.

I GAIN STRATIFIED BY INVALID DEPTH ON OBJECTS

The stratification asks where the gain occurs, using the fraction of ground-truth object pixels that have invalid depth. Ground truth is used only for this analysis. The pooled 261 frames comprise 111 OSD frames and 150 held-out OCID frames; the bin boundaries below are printed to three decimals so that the 0.005 cut is explicit.

Table 14: Gain by invalid fraction inside ground-truth objects, 261 held-out frames, clean input. Paired bootstrap, 10,000 resamples.

Object-region invalid	frames	published	PROXFILL	Δ (95% CI)
$[0.000, 0.005)$	154	86.69	87.34	$+0.65$ $[-1.42, +2.95]$
$[0.005, 0.020)$	38	87.74	91.59	$+3.85$ $[+0.69, +7.77]$
$[0.020, 0.040)$	50	85.12	88.30	$+3.17$ $[+1.42, +5.41]$
$[0.040, 1.000]$	19	84.73	86.23	$+1.50$ $[-0.62, +3.65]$

The two middle bins have positive intervals. The lowest bin and the highest bin include zero, and the highest bin contains only 19 frames. This supports an association between object-region missingness and benefit in the observed sample, without establishing a monotonic gain with missing fraction. The Pearson correlation in this diagnostic is $+0.048$; the reported rank statistic is $+0.309$.

J τ SENSITIVITY IN FULL

Selection and diagnosis are separate experiments. The original search uses 120 ARID10 frames and six candidates; it maximizes the mean of clean and $+20\%$ Object-F. Let $\bar{F}_{\mathcal{D}}(t, s)$ denote mean Object-F on these development frames at radius t and severity s. For $\mathcal{T} = \{0, 2, 4, 8, 16, \infty\}$, we select

$$\tau^\star = \arg\max_{t \in \mathcal{T}} \frac{\bar{F}_{\mathcal{D}}(t, 0) + \bar{F}_{\mathcal{D}}(t, 0.2)}{2} = 8. \tag{3}$$

Both conditions choose 8 in this run; no held-out score enters the objective.

Table 15: The original 120-frame ARID10 selection run. The criterion is the mean of clean and $+20\%$ Object-F over the six candidates.

τ	clean	$+20\%$	mean
0	87.05	84.89	85.97
2	86.78	86.07	86.43
4	87.09	87.05	87.07
8	**87.24**	**87.17**	**87.21**
16	87.00	86.03	86.52
∞	86.81	75.55	81.18

The later sweep adds intermediate and larger radii and uses 200 ARID10 frames. It is a sensitivity check, not a second selection step. The value 8 is retained even though 10 has the best mean in this larger ARID10 run. The held-out OSD sweep is reported in the same table for diagnosis only.

Table 16: All fifteen values from the diagnostic sweep. Object-F; mean is the average of clean and $+20\%$. ARID10 uses 200 frames and OSD uses all 111. Bold marks the best value in each score column. The selected threshold remains 8.

	OCID ARID10			OSD-111		
τ	clean	$+20\%$	mean	clean	$+20\%$	mean
0	86.75	83.94	85.34	86.35	70.29	78.32
1	86.87	84.95	85.91	87.80	78.98	83.39
2	86.79	85.44	86.11	87.30	84.23	85.76
3	**86.96**	85.41	86.19	89.06	87.73	88.40
4	86.76	85.95	86.36	89.02	88.08	88.55
6	86.69	86.23	86.46	88.87	**88.91**	88.89
8 (fixed)	86.84	86.64	86.74	88.98	88.84	**88.91**
10	86.95	**86.72**	**86.84**	88.95	88.81	88.88
12	86.86	86.64	86.75	88.96	88.45	88.71
16	86.68	85.70	86.19	**89.09**	87.95	88.52
20	86.72	84.54	85.63	88.45	87.76	88.11
24	86.81	84.42	85.62	88.13	86.81	87.47
32	86.47	83.36	84.92	87.72	85.66	86.69
48	86.35	78.13	82.24	87.15	85.29	86.22
∞	86.25	76.54	81.40	88.55	87.09	87.82

On the mean criterion, 8 is within 0.10 points of the best ARID10 value and has the best OSD value. Individual-condition maxima differ: OSD clean peaks at 16, whereas OSD at $+20\%$ peaks at 6. The conclusion is a useful range around the selected radius, rather than a unique optimum for every condition.

K LEARNED ALTERNATIVES, IN FULL

The three learned alternatives test different decisions: which predicted instances to keep, what depth to reconstruct, and which invalid components to fill. Their fitting domains and evaluation units differ, so the comparisons below state each protocol explicitly. In particular, ARID10 is the fitting domain for depth completion and hole classification; its scores are not held-out results for those learners.

K.1 A PER-INSTANCE GATE

The gate replaces a hand-tuned keep/drop decision with logistic regression over 21 descriptors: size and objectness, boundary agreement with depth and RGB, planar-fit residual, depth range, and the valid-pixel ratio used by the original filter. It is fitted on UOAIS-Sim predictions: 400 frames under five fitting conditions produce $2{,}000$ image-condition records and $28{,}268$ instances. The decision threshold is 0.05, selected on the fitting source by clean and macro degraded Object-F.

Table 17: The per-instance gate on its original archived prediction sets. Object-F; macro averages the seven non-clean diagnostic conditions. These are post-hoc tests on DA-Fusion predictions, separate from the calibrated UCN comparison.

Dataset	Decision	clean	macro
OSD	keep all instances	75.72	75.45
OSD	learned gate	74.96	74.45
OCID	keep all instances	84.65	84.28
OCID	learned gate	84.02	83.73

These archived tests use DA-Fusion predictions under the earlier diagnostic conditions: independent removal at 10% and 25%, depth noise, two image shifts, a central hole, and depth dropout. They are retained as a negative result for post-hoc instance selection, not substituted for the calibrated robustness protocol. The fitting source labels 94.9% of instances worth keeping; that decision is much less frequent on OSD. The gate loses to keeping all instances on both real collections, even though its fitting performance is high.

K.2 A LEARNED DEPTH-COMPLETION FRONT-END

The completion model has $108{,}196$ parameters. Its input is nearest-filled XYZ, validity, log distance to the nearest measurement, and RGB. Six 3×3 convolution blocks use width 48, GroupNorm and GELU, with dilations $(1, 2, 4, 8, 16, 1)$. Separate heads predict an XYZ residual and a confidence. The residual head starts at zero, and measured pixels are copied through unchanged by construction.

The fitting source is 400 ARID10 frames. The calibrated generator removes valid measurements, and supervision is restricted to those removed returns; native holes have no target. The training objective combines scene-normalized masked L_1 with 0.1 times a confidence binary cross-entropy. The confidence target decreases with the model's detached reconstruction error. The training script uses AdamW, a cosine schedule and gradient clipping; its defaults are batch size 2, learning rate 2×10^{-4}, weight decay 10^{-4} and $4{,}000$ iterations.

Table 18: Depth-completion confidence sweep, Object-F. The completion model is fitted on ARID10; the other two evaluation sets are held out. Confidence c controls which predicted points are accepted. Each row is a paired comparison.

Split (frames)	Severity	$\tau = 8$	$c = 0$	0.2	0.4	0.6	0.8
ARID10 (100)	clean	87.90	86.08	85.89	85.80	86.47	87.75
ARID10 (100)	$+20\%$	86.93	80.65	80.66	81.54	85.14	87.06
OCID held out (150)	clean	87.94	87.08	87.23	87.15	87.29	86.98
OCID held out (150)	$+20\%$	88.51	86.05	86.02	85.18	85.81	85.61
OSD (111)	clean	89.13	87.81	86.67	87.08	88.02	86.97
OSD (111)	$+20\%$	88.59	86.72	85.72	86.34	87.02	85.16

The full confidence sweep matters. At $c = 0.8$, completion slightly exceeds the scalar on degraded ARID10 (87.06 versus 86.93), which is its fitting domain. Every tested threshold remains below ProxFill on both held-out sets, under both conditions. Thus the held-out conclusion does not depend on choosing an inconvenient confidence threshold. The recorded lower masked depth error also measures reconstruction, not instance segmentation; the two objectives need not choose the same repair.

K.3 A UTILITY-SUPERVISED HOLE CLASSIFIER

This learner supervises the fill decision directly. It uses $4{,}072$ connected invalid components from 120 ARID10 frames at clean, $+10\%$ and $+20\%$ severity. A label is the sign of the Object-F change obtained by filling a component while retaining the reference decisions for all others. Each example is weighted by the absolute change, floored at 10^{-4}, so nearly neutral components do not dominate the objective.

The twelve descriptors cover area, distance to valid pixels, aspect ratio, solidity, border contact, rim depth statistics, rim validity, RGB variation and image position. The gradient-boosted model uses 200 depth-3 trees and learning rate 0.05. It reaches 0.957 weighted accuracy in sample, against 0.766 for the component threshold. With all components and severities from the same frame assigned to the same cross-validation group, the recorded five-fold accuracy is 0.730.

This is a decision-accuracy diagnostic, not an Object-F score of a deployed segmentation pipeline. It shows that fitting the available component descriptors is easy, while transferring that decision to unseen frames is harder. The depth discontinuity across the rim is informative, but the tested learner does not turn that feature into an improvement over the scalar under the grouped evaluation.

L THE GAIN IS NOT POST-HOC MASK EDITING

A reader may reasonably suspect that a few points of segmentation gain come from cleaning up an existing instance set. We therefore evaluate simple edits with ground truth available to the decision maker. On UCN's archived OSD-111 predictions, greedy instance dropping gains 0.83 Object-F. A drop/merge search on a separate 40-frame OCID subset gains 0.10. These are different evaluation sets, so the two numbers are not nested bounds on one experiment.

The metric for a chosen subset is exact: it follows from the pixel contingency table and reproduces the reference evaluator on the archived outputs. The search over subsets is greedy, however, so exact scoring does not make it an exhaustive oracle. The small gains characterize the editing procedures tested. They do not exclude stronger boundary refiners, which can be complementary to changing the input.

Room in the fill decision. A separate experiment searches decisions over connected invalid components. It starts by filling a whole component when its maximum distance to a valid pixel is at most 8. This component-level starting policy differs from the pixelwise ProxFill rule at the rims of large holes. We retain that distinction when reporting the recorded search gains.

Table 19: Ground-truth-guided component search on 60 frames per split. Cells report Object-F / Boundary-F; the final column is the paired Object-F gain. The starting policy accepts whole components with maximum distance at most 8, as specified in the text.

Split	Severity	component threshold	greedy search	Δ
OCID held out	clean	90.25 / 80.91	90.78 / 81.83	$+0.53$
OCID held out	$+20\%$	90.16 / 77.50	91.90 / 81.32	$+1.74$
OSD	clean	88.07 / 73.53	89.75 / 78.13	$+1.68$
OSD	$+20\%$	84.04 / 69.04	89.85 / 76.00	$+5.81$

The search considers up to the 20 largest components of at least 16 pixels, makes up to two passes, and accepts a flip only when measured Object-F improves. Each split uses 60 evenly spaced frames, chosen independently of the larger benchmark runs. These scores are therefore paired within the row and should not be subtracted from absolute scores in other tables. The search also uses ground truth and repeated model calls, so it is a diagnostic rather than a deployable method. Its 0.53–5.81 point gains establish attainable improvement for the tested component policy; a greedy search is not a proof of the global optimum.

M CORRUPTION GENERATOR AND EVALUATION DETAILS

What enters the shape bank. We use 8-connected components of the native invalid mask. Components smaller than 4 pixels are left native but are not resampled. A component whose bounding rectangle covers more than a quarter of the frame is also excluded. The retained binary patches carry their source-frame index. A 30×40 grid accumulates the invalid-pixel frequency and supplies a normalized placement prior. Neither operation uses object labels.

How a corrupted frame is formed. A sampled patch is mirrored horizontally with probability 0.5, then placed at a location drawn from the coarse prior. Placement is clamped so that the patch stays within the frame. Only newly covered valid pixels count toward the severity budget; existing invalid pixels remain invalid. We stop when the budget is reached or the placement-attempt limit is hit. Whole patches are pasted, so the achieved fraction can exceed the requested fraction slightly. Let $\mathcal{V}_0$ be the originally valid pixel set, $\mathcal{M}_{\text{add}}$ the union of pasted masks, and ρ_0 the native invalid fraction. The achieved severity and final invalid fraction are

$$\hat{s} = \frac{|\mathcal{M}_{\text{add}} \cap \mathcal{V}_0|}{|\mathcal{V}_0|}, \qquad \rho_{\text{final}} = \rho_0 + (1 - \rho_0)\hat{s}. \tag{4}$$

Thus the target s refers to additional loss among valid pixels, not total missingness. For a fully invalid frame we set $\hat{s} = 0$ and leave the input unchanged.

Table 20: Generator validation on 60 frames per dataset at $+10\%$ target severity. Statistics are averaged over frames. Mean area is the mean of each frame's component-area mean, not total invalid pixels divided by the mean number of components.

Dataset	Mask	missing	holes/frame	mean area (px)	border affinity
OSD	native	24.27%	48.9	2016.9	2.82×
OSD	calibrated	31.98%	89.7	1175.6	2.28×
OSD	i.i.d.	31.85%	14581.6	6.7	2.25×
OCID	native	11.45%	8.9	11865.6	7.11×
OCID	calibrated	20.34%	32.0	2132.0	4.34×
OCID	i.i.d.	20.32%	17271.7	3.6	4.08×

The validator uses 60 frames per dataset, with 1,419 retained OSD shapes and 244 OCID shapes. Border affinity divides invalid frequency in the outer 12-pixel band by invalid frequency over the frame. The paired calibrated and i.i.d. masks use the same target added fraction. Native masks have a smaller total missing fraction because no additional pixels have been removed.

Where calibration information comes from. The evaluated frame's own patches are excluded when sampling shapes. The location prior is still estimated from the dataset's unlabelled depth, including that frame. This is a sensor-calibrated benchmark protocol; it is not a corruption model learned entirely from a separate sensor or a physical rendering of specularity. Training the completion front-end uses an ARID10-only bank, keeping its fitting corruption separate from held-out scene statistics.

Pairing and record boundaries. The UCN corruption seed is 20260831 plus the dataset frame index, and its inference seed is reset to 20260917 before each variant. Variants of a frame receive the same corrupted input. The main robustness grid uses one run per dataset, whereas the significance, threshold-sweep and repeatability experiments are separate recorded runs. CUDA nondeterminism and different frame subsets make their absolute scores slightly different; comparisons should use the paired rows within each experiment. Added tables in this appendix are generated directly from those archived records, with source hashes retained alongside the LaTeX table files.