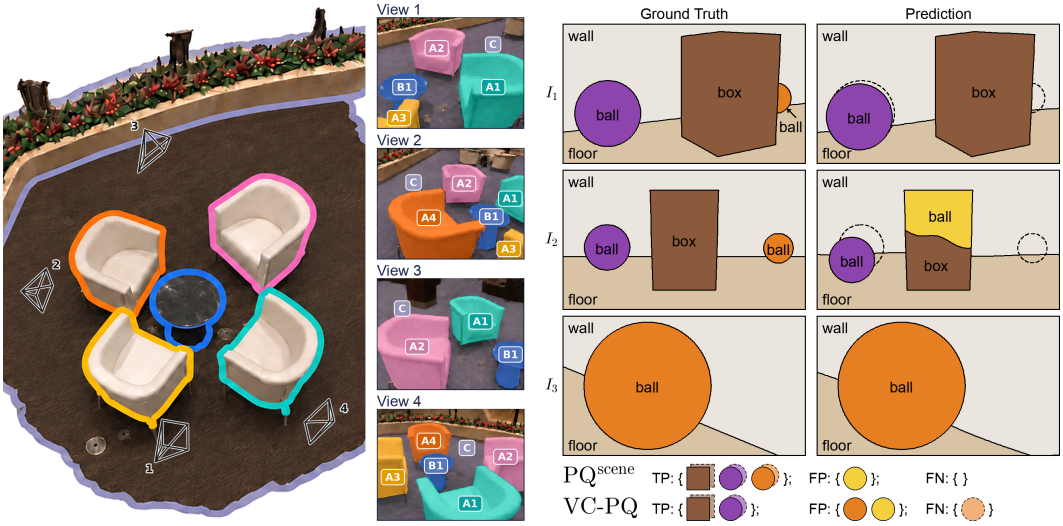
EVERY VIEW COUNTS: VIEW-CONSISTENT PANOPTIC QUALITY FOR MULTI-VIEW PANOPTIC SEGMENTATION

Anonymous authors
Paper under double-blind review



(a) (b)

Figure 1: (a) Multi-view panoptic segmentation: every view of a scene is labeled with a class and a scene-level instance ID. (b) A toy example of an input set of three views: the orange ball is predicted only in the view in which it appears large, so PQ^{scene} matches it by area whereas VC-PQ counts the two missed views and leaves it unmatched.

ABSTRACT

Multi-view panoptic segmentation assigns a semantic class and a scene-level instance ID to every pixel of an unordered set of images, and recent feed-forward 3D models predict these labels for the input views in a single forward pass. Their predictions, however, have been evaluated with the scene-level PQ (PQ^{scene}) borrowed from per-scene optimization methods, typically on rendered held-out views. PQ^{scene} tiles all views of a scene into a single image, so that a missed appearance or a change of ID lowers the score of the matched pair only in proportion to its area. We propose View-Consistent Panoptic Quality (VC-PQ), which extends PQ from a single image to a set of input views, counts equally every view in which an instance is visible, and penalizes a prediction that is not visible in the same views as its ground truth. A decomposition of VC-PQ attributes the score a method loses to mask accuracy, view consistency, and the matching threshold. A single additional parameter recovers the area weighting of tiling for comparison. Under a fixed evaluation protocol on ScanNet++ and ScanNetv2, recent feed-forward methods are evaluated with VC-PQ and PQ^{scene}, and the decomposition shows where each of them loses its score. Controlled perturbations of the ground truth show that VC-PQ responds to the number of views in which an instance is missed or changes ID, whereas PQ^{scene} responds to their area. The aim of this work is to make view consistency part of the evaluation of multi-view panoptic segmentation, with VC-PQ reported alongside PQ^{scene}. The evaluation code will be released.

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.

1 INTRODUCTION

Multi-view panoptic segmentation assigns a semantic class and a scene-level instance ID to every pixel of the images captured from several views of a scene, so that one object instance carries the same ID in every view in which it appears (Fig. 1a). Unlike in video segmentation and object tracking, the input views are captured in an arbitrary order, with no continuity from one view to the next, and therefore this consistency of identity across views is the demanding part of the task. The chairs in a lounge, for example, often look alike (Fig. 1a), so when the viewpoint changes, appearance alone may not tell whether the chairs in different views are the same instance.

Per-scene optimization methods lift the outputs of a 2D segmentation model into a 3D scene representation that is optimized for each scene given precomputed camera poses, and the shared representation enforces label consistency across views (Siddiqui et al., 2023; Bhalgat et al., 2023; Ye et al., 2024). Recent feed-forward 3D reconstruction models (Wang et al., 2024; 2025) predict the 3D geometry of a scene from the images alone, without precomputed camera poses. Building on their geometry-aware features, feed-forward methods for multi-view segmentation predict instance or panoptic labels with consistent IDs for the input views (Žust et al., 2025; Zhu et al., 2026b; Qu et al., 2026; Li et al., 2026b).

How these predictions are evaluated, however, has not followed the task. Predictions for this task have commonly been evaluated with the scene-level Panoptic Quality (PQ^{scene}) of Panoptic Lifting (Siddiqui et al., 2023), which applies PQ (Kirillov et al., 2019) to held-out views rendered from the per-scene representation, and feed-forward methods have adopted this protocol (Žust et al., 2025; Zhu et al., 2026b) although they predict on the input views. PQ^{scene} moreover tiles all views of a scene into a single image before applying PQ, so that a missed appearance, or a change of ID, in one view lowers the score of the matched pair only in proportion to the area of that appearance (Fig. 1b). The same instance appears at a smaller scale in a view captured farther from it and covers fewer pixels in a view where other objects occlude it, so the weight a view carries in its score depends largely on the camera pose within the scene.

We therefore propose View-Consistent Panoptic Quality (VC-PQ), which extends PQ from a single image to a set of input views and scores each instance jointly by the accuracy of its masks and by whether it is visible in the same views as its ground truth, with every such view weighted equally. Apart from a visibility threshold that disregards noisy appearances of only a few pixels, VC-PQ reduces to PQ for a single view. A decomposition of the score lost under VC-PQ attributes it to mask accuracy, view consistency, and the matching threshold, and a one-parameter family of pair scores isolates the area weighting of tiling.

We evaluate both metrics under a fixed protocol on ScanNet++ and ScanNetv2 that draws input sets of several sizes from every scene. Each method is evaluated with its native post-processing, and with an adapter where it does not output panoptic labels. Controlled perturbations of the ground truth show that VC-PQ responds to the number of views in which an instance is missed or changes ID, whereas PQ^{scene} responds to their area. As the input set grows, PQ^{scene} rises whereas VC-PQ falls on recent feed-forward methods, and the one-parameter family traces the fall to the equal weighting of views.

In summary, our contributions are as follows.

- A formulation of multi-view panoptic segmentation on the input views, with two requirements that a metric for this task should meet (Sec. 3).
- View-Consistent Panoptic Quality, which weights every view in which an instance is visible equally, penalizes a prediction that is not visible in the same views as its ground truth, and reduces to PQ for a single view, with a one-parameter family that isolates the area weighting of tiling (Sec. 4).
- An evaluation protocol on ScanNet++ and ScanNetv2, under which VC-PQ and PQ^{scene} are compared on recent feed-forward methods and their difference is attributed to each property of PQ^{scene} (Sec. 5.1–5.3).
- A decomposition of the lost score by mask accuracy, view consistency, and the matching threshold, which separates the errors within single images from those that arise only across views (Sec. 4.3, Sec. 5.4).

2 Related Work

2.1 Multi-view Panoptic Segmentation

Methods that assign semantic, instance, or panoptic labels to the views of a scene follow one of two approaches, per-scene optimization or feed-forward prediction. Per-scene methods lift the predictions of an external 2D segmenter into a 3D scene representation, a neural field or 3D Gaussians, that is optimized for each scene given precomputed camera poses, and evaluate the labels rendered from it (Siddiqui et al., 2023; Bhalgat et al., 2023; Zhu et al., 2024; Ye et al., 2024; Liu et al., 2025; Dhiman et al., 2026).

Recent feed-forward 3D reconstruction models have progressively removed both requirements. DUSt3R and its extensions (Wang et al., 2024; Leroy et al., 2024; Cabon et al., 2025) predict point maps without camera poses, and VGGT (Wang et al., 2025) predicts camera poses, depth maps, and point maps for a whole set of views in one forward pass. Feed-forward segmentation methods are built on these reconstruction models as a geometry backbone, predict the labels for the input views without a per-scene optimization, and differ in how an instance identity is derived from the geometry-aware features of the backbone. One group predicts a per-pixel or per-Gaussian instance embedding and clusters it into instances at inference, so that an ID is a cluster label and holds across views only as far as the clustering keeps an instance in one cluster (Li et al., 2026b; Zhu et al., 2026b; Koch et al., 2026; Jiang et al., 2026). The other shares a set of object queries across all views, so that an ID is a query index and is the same in every view by construction, and which query covers an instance in each view is left to a post-processing selection of masks or to training alone (Žust et al., 2025; Xu et al., 2025; Barberteguy et al., 2026; Qu et al., 2026; Li et al., 2026a). Among the feed-forward methods, PanSt3R, EPS3D, Pano3D, and SIU3R produce panoptic labels, and PanSt3R and EPS3D report panoptic quality with the scene-level protocol of per-scene methods, at novel views rendered from 3D Gaussians rather than on the input views (Žust et al., 2025; Zhu et al., 2026b).

2.2 Evaluation Metrics

Panoptic Quality (PQ) (Kirillov et al., 2019) scores a single image by matching predicted and ground-truth segments one-to-one and counting every segment equally regardless of its area. Panoptic Lifting (Siddiqui et al., 2023) extends it to a scene as PQ^{scene} by tiling the views of a scene into one image and reports it at held-out test views, and later per-scene and feed-forward methods adopt this metric (Bhalgat et al., 2023; Zhu et al., 2024; Liu et al., 2025; Dhiman et al., 2026; Žust et al., 2025). In video panoptic segmentation, PQ is extended by matching tubes over consecutive frames, as in VPQ (Kim et al., 2020; Qiao et al., 2021), or by combining an association term with a segmentation term without matching, as in STQ (Weber et al., 2021; Mei et al., 2022; Aygün et al., 2021); both pool the pixels of several frames, over a window for VPQ and over the whole sequence for STQ.

For the consistency of an instance across views, tracking evaluation offers a form that PQ does not have. It scores a given target by its overlap in each frame, averaged with equal weight over the frames, and by the fraction of frames in which it is tracked (Čehovin et al., 2016; Huang et al., 2021). Recent multi-view methods score a given instance in the same form, by its mean per-view IoU and by whether it is tracked in every view (Li et al., 2026b; Zhu et al., 2026a). Others match instances by IoU without classes and report a mean IoU and an F-score (Zhu et al., 2026b; 2025), and with the instance given or the class dropped, none of these scores panoptic labels with a class-wise matching and a count of unmatched instances. VC-PQ (Sec. 4) keeps the matching and unmatched terms of PQ and counts every view in which a matched pair is visible equally, in the form of these measures.

3 Problem Setting

3.1 Task formulation

Given N RGB images $I_1, \ldots, I_N$ of a static scene, captured from N views in an arbitrary order, the task is to predict for each view a panoptic label map Y_i.

The panoptic label map Y_i assigns every pixel u of I_i a panoptic label $Y_i(u) = (c, z)$ (Kirillov et al., 2019), consisting of a semantic class c and an instance ID z. Instance IDs are defined at the scene level. That is, pixels sharing an ID, whether in the same or in different views, belong to one object instance and carry the same class. We refer to an instance by its ID and write $c(\cdot)$ for its class and $M_i(\cdot)$ for its mask in image I_i, so that $M_i(z) = \{\, u \mid Y_i(u) = (c(z), z) \,\}$, which is empty when the instance does not appear in I_i. Each stuff class is treated as a single instance.

Ground-truth instances are denoted by g and predicted instances by p, with their masks $M_i(g)$ and $M_i(p)$ defined from the respective label maps in the same way. An ideal prediction places predicted and ground-truth instances in one-to-one correspondence such that every corresponding pair (p, g) satisfies $c(p) = c(g)$ and $M_i(p) = M_i(g)$ for every image I_i.

3.2 Evaluation requirements and existing metrics

Evaluating multi-view panoptic segmentation means measuring how closely predicted instances approach the ideal correspondence of Sec. 3.1 across the input views. A metric that captures this ideal should therefore (i) reflect the mask accuracy of an instance in every view in which it appears, independent of its apparent size in that view, and (ii) penalize a predicted instance by how far the set of views in which it appears departs from that in which its ground-truth counterpart appears.

For a single image I_i, Panoptic Quality (PQ) (Kirillov et al., 2019) scores a predicted instance p against a ground-truth instance g of the same class by the intersection over union (IoU) of their masks, $\mathrm{IoU}_i(p, g) = |M_i(p) \cap M_i(g)|/|M_i(p) \cup M_i(g)|$, and matches predicted and ground-truth instances one-to-one among the pairs with $\mathrm{IoU}_i(p, g) > 0.5$, which makes the matching unique. Matched pairs, unmatched predicted instances, and unmatched ground-truth instances are counted over the dataset as true positives (TP), false positives (FP), and false negatives (FN). PQ then divides the summed IoU of the matched pairs, each taken in the image i that contains the pair, by the number of matches with a penalty of one half for every unmatched instance,

$$\mathrm{PQ} = \frac{\sum_{(p,g)\in \mathrm{TP}} \mathrm{IoU}_i(p, g)}{|\mathrm{TP}| + \frac{1}{2}|\mathrm{FP}| + \frac{1}{2}|\mathrm{FN}|}, \tag{1}$$

computed for each class independently and averaged over classes.

Panoptic Lifting (Siddiqui et al., 2023) introduced the scene-level PQ ($\mathrm{PQ}^{\mathrm{scene}}$), which recent feed-forward methods also report (Žust et al., 2025; Zhu et al., 2026b). It tiles the predictions and ground truth of all test views of a scene into a single image and applies standard PQ, so that a pair is scored by the IoU of the concatenated masks,

$$\mathrm{IoU}_{\mathrm{concat}}(p, g) = \frac{\sum_i |M_i(p) \cap M_i(g)|}{\sum_i |M_i(p) \cup M_i(g)|}, \tag{2}$$

where the sums run over the test views of the scene. The official implementation further discards, before matching, every class that covers less than 0.5% of the labeled pixels of the scene in either the prediction or the ground truth. The feed-forward methods that report $\mathrm{PQ}^{\mathrm{scene}}$ evaluate under this novel-view protocol, by feeding rendered held-out views to the network or by rendering their labels from a 3D field (Žust et al., 2025; Zhu et al., 2026b). We evaluate their predictions on the input views, as the task requires.

An input set, like a video clip, contains several views of one scene in which an instance keeps its ID from view to view, which suggests Video Panoptic Quality (VPQ) as a metric for this setting. VPQ (Kim et al., 2020; Qiao et al., 2021) matches predicted and ground-truth tubes within a window of k consecutive frames and averages the score over several window lengths. Within a window, a pair of tubes is scored by the same concatenated IoU of Eq. (2), with the sums running over the frames of the window.

In this setting, three properties of $\mathrm{PQ}^{\mathrm{scene}}$ no longer match the task (Fig. A2, Appendix D). First, Eq. (2) sums pixel areas over views, so that each view weighs in proportion to the area the pair covers in it. Accurate segmentation where an instance appears large then compensates for a complete miss where it appears small, as for the orange ball of Fig. 1b, and requirement (i) is not met. A view in which only one of the two appears has an empty intersection and adds its area to the denominator alone, so a missed or spurious appearance lowers Eq. (2) only in proportion to the area it covers

in that view, and requirement (ii) is not met either. Both requirements thus turn on the same area weighting, which PQ^{scene} and VPQ share.

Second, PQ^{scene} yields one value per scene, and results for a dataset are reported as the average of these values across scenes (Žust et al., 2025; Zhu et al., 2026b). The scene thus plays for PQ^{scene} the role that the image plays for PQ, yet PQ does not average over its unit, since the counts of TP, FP, and FN are pooled over all images before PQ is computed for each class (Kirillov et al., 2018). Averaging scene values instead gives every scene the same weight, so that a scene with only a few instances, all of them found, enters the average with a value close to the maximum. For this task, the labels of the N input views are predicted together, and the input set, rather than the scene, therefore takes the place of the single image of PQ.

Third, the 0.5% class filter of the official implementation excludes every class that falls below the threshold in the ground truth, where its instances are small or seldom visible, or in the prediction, so that the set of scored classes depends on the prediction under evaluation. Because the threshold is a share of all labeled pixels, a class that appears in only some of the views is more likely to fall below it in a scene with many views. Standard panoptic and instance segmentation (Kirillov et al., 2019; Lin et al., 2014) exclude no class on account of its size.

A metric for this task should therefore meet requirements (i) and (ii) and count over the dataset with the input set in place of the image.

4 View-Consistent Panoptic Quality

4.1 Definition

To this end, we propose View-Consistent Panoptic Quality (VC-PQ), which extends PQ from a single image to a set of views. VC-PQ takes the input set of N views as its unit of evaluation. Within each set, each pair of a predicted instance p and a ground-truth instance g of the same class receives a pair score $q(p, g) \in [0, 1]$.

An instance x, whether predicted or ground truth, is regarded as visible in view i when its mask $M_i(x)$ has at least η pixels, and its visibility support $V(x)$ is the set of these views, $V(x) = \{\, i \mid |M_i(x)| \geq \eta \,\}$, where η is a visibility threshold specified in pixels at the evaluation resolution and set to 100 following PanopticNeRF-360 (Fu et al., 2025).

For a predicted instance p and a ground-truth instance g, $\text{IoU}_i(p, g)$ is the per-view IoU of Sec. 3.2. Two quantities enter the pair score: the mean IoU over the views in which both are visible, each view counting equally, and a view-consistency factor given by the Jaccard index of their supports,

$$\overline{\text{IoU}}(p, g) = \frac{1}{|V(p) \cap V(g)|} \sum_{i \in V(p) \cap V(g)} \text{IoU}_i(p, g), \qquad \text{VC}(p, g) = \frac{|V(p) \cap V(g)|}{|V(p) \cup V(g)|}, \tag{3}$$

which address requirements (i) and (ii), respectively, with $\overline{\text{IoU}}(p, g)$ taken as zero when $V(p) \cap V(g)$ is empty. The pair score is the product of the two quantities,

$$q(p, g) = \overline{\text{IoU}}(p, g)\, \text{VC}(p, g) = \frac{1}{|V(p) \cup V(g)|} \sum_{i \in V(p) \cap V(g)} \text{IoU}_i(p, g), \tag{4}$$

that is, the IoU averaged over the union of the two supports, where the views outside their intersection count as zero.

Within each input set and class, predicted and ground-truth instances are matched one-to-one among the pairs with $q(p, g) > 0.5$, the same threshold as in PQ, which makes the matching unique since no instance can exceed this threshold with more than one counterpart (Appendix B). An identity split across views lowers the view-consistency factor of every fragment, and with it the pair score, which may fall to 0.5 or below so that the fragment is no longer matched as a TP. Matched pairs, unmatched predicted instances, and unmatched ground-truth instances are counted over the dataset as TP, FP, and FN, as in PQ. VC-PQ is then defined as PQ with the IoU of each matched pair replaced by its pair score,

$$\text{VC-PQ} = \frac{\sum_{(p,g) \in \text{TP}} q(p, g)}{|\text{TP}| + \frac{1}{2}|\text{FP}| + \frac{1}{2}|\text{FN}|}, \tag{5}$$

computed for each class independently and averaged over classes.

4.2 Relation to PQ$^{\text{scene}}$

VC-PQ addresses each of the three properties of Sec. 3.2 in a way that matches the task. In contrast to the area weighting of tiling, every view in the union of Eq. (4) contributes one term regardless of how large the instances appear in it, so that a missed or spurious appearance lowers the score by one term whatever its area. In place of per-scene averaging, matching within the input set and counting over the dataset keep the structure of PQ. The visibility threshold η, the only size-related criterion in VC-PQ, keeps tiny fragments of noisy annotation or prediction from enlarging the union in Eq. (4). It acts on individual appearances rather than on classes, so that an instance is excluded from evaluation only when its mask falls below η pixels in every view of the set. Apart from the visibility threshold, VC-PQ therefore differs from PQ only in how a pair is scored, and with a single view, where $q(p, g)$ equals $\mathrm{IoU}_i(p, g)$, it coincides with PQ.

The first of the three properties, the area weighting, can be varied on its own. Written per view, Eq. (2) is the average of the per-view IoU weighted by $a_i = |M_i(p) \cup M_i(g)|$, the union area of the pair in image I_i. Raising these weights to a power $\gamma \geq 0$ and restricting the sum to the support union of Eq. (4) gives a one-parameter family of pair scores,

$$q_\gamma(p, g) = \sum_{i \in V(p) \cup V(g)} w_i \, \mathrm{IoU}_i(p, g), \qquad w_i = \frac{a_i^\gamma}{\sum_{j \in V(p) \cup V(g)} a_j^\gamma}, \tag{6}$$

with $\mathrm{IoU}_i(p, g)$ taken as zero outside $V(p) \cap V(g)$. At $\gamma = 0$ every view weighs the same and q_γ coincides with q of Eq. (4), and at $\gamma = 1$ the weights are the areas a_i. We write VC-PQ$_\gamma$ for VC-PQ computed with q_γ in place of q, so that VC-PQ = VC-PQ$_0$ and VC-PQ$_1$ scores every pair by the area weighting of tiling. Between these ends the unit of evaluation, the counting over the dataset, and the absence of a class filter stay fixed. Sweeping γ therefore shows how much of a method's score rests on the views in which its instances appear large, which Sec. 5.3 measures.

4.3 Diagnostic decomposition

The pair score of Eq. (4) is the product of two factors, each addressing one of requirements (i) and (ii), so that the score a prediction does not receive can be attributed to the requirement it fails to meet. Averaging $\mathrm{VC}(p, g)$ over the matched pairs would not show this, since the instances lost to view consistency are precisely those that fail to match. Instead, we attribute the whole of the score not received, $1 - $ VC-PQ, to the two requirements and to the matching threshold,

$$\begin{aligned} 1 - \text{VC-PQ} = {} & \underbrace{\frac{\sum_{\mathrm{TP}} \big(1 - \overline{\mathrm{IoU}}\big) + \frac{1}{2}|\mathrm{FN}_{\overline{\mathrm{IoU}}}| + \frac{1}{2}|\mathrm{FP}_{\overline{\mathrm{IoU}}}|}{D}}_{\varepsilon_{\overline{\mathrm{IoU}}}} \\ & + \underbrace{\frac{\sum_{\mathrm{TP}} \overline{\mathrm{IoU}} \, (1 - \mathrm{VC}) + \frac{1}{2}|\mathrm{FN}_{\mathrm{VC}}| + \frac{1}{2}|\mathrm{FP}_{\mathrm{VC}}|}{D}}_{\varepsilon_{\mathrm{VC}}} + \underbrace{\frac{\frac{1}{2}|\mathrm{FN}_q| + \frac{1}{2}|\mathrm{FP}_q|}{D}}_{\varepsilon_q}, \end{aligned} \tag{7}$$

where D denotes the denominator of Eq. (5), the sums run over the matched pairs, and $\mathrm{FN}_{\overline{\mathrm{IoU}}}$, $\mathrm{FN}_{\mathrm{VC}}$, and FN_q partition FN as follows, and likewise for FP with the roles of p and g exchanged.

An unmatched instance is classified by three tests applied in turn to its candidates, the predicted instances of its class in the input set: $\overline{\mathrm{IoU}}(p, g) > 0.5$, then $\mathrm{VC}(p, g) > 0.5$, then $q(p, g) > 0.5$. A ground-truth instance g belongs to $\mathrm{FN}_{\overline{\mathrm{IoU}}}$ if no candidate passes the first test, to $\mathrm{FN}_{\mathrm{VC}}$ if some does but none of those passes the second, and to FN_q if some passes both and fails only at the third. Since a pair with $q(p, g) > 0.5$ is always matched and both factors of Eq. (4) are at most one, every unmatched instance falls in exactly one of the three (Fig. A3, Appendix E).

For a matched pair, the cross term $(1 - \overline{\mathrm{IoU}})(1 - \mathrm{VC})$ of $1 - q$ is counted with the mean IoU in Eq. (7), so that $\varepsilon_{\mathrm{VC}}$ is a lower bound on the score lost to view consistency, and ε_q measures the cost of combining the two factors by a product. The three terms thus separate the errors that occur within a single image from those that arise only across views, since $\varepsilon_{\mathrm{VC}}$ and ε_q vanish when the set contains a single view and $1 - \mathrm{PQ} = \varepsilon_{\overline{\mathrm{IoU}}}$ up to the visibility threshold. Raising either factor would change the matching itself, so the terms attribute the lost score rather than predict a gain.

Table 1: PQ^{scene} and VC-PQ of the same predictions for input sets of N views, all classes; ScanNet++ pools the DSLR and iPhone runs. –: out of memory at $N \geq 20$.

		PQ^{scene}				VC-PQ			
Dataset	Method	N=5	N=10	N=20	N=40	N=5	N=10	N=20	N=40
ScanNet++	PanSt3R v1	43.7	45.7	48.9	50.4	22.0	19.2	17.9	17.4
	PanSt3R v2	46.8	51.3	54.7	57.7	24.5	22.2	20.7	20.0
	EPS3D	13.2	13.6	–	–	2.4	2.0	–	–
	IGGT	17.8	17.1	–	–	5.7	4.2	–	–
ScanNetv2	PanSt3R v2	50.1	52.5	55.2	57.4	43.7	41.5	40.6	40.6
	EPS3D	21.7	21.5	–	–	9.2	8.2	–	–

5 Experiments

5.1 Evaluation protocol

We evaluate on the validation splits of ScanNet++ (Yeshwanth et al., 2023) and ScanNetv2 (Dai et al., 2017). ScanNet++ provides 50 validation scenes, each captured twice, with a DSLR camera and with an iPhone,[1] and we use its panoptic benchmark of 100 classes, 84 things and 16 stuff. ScanNetv2 provides 312 validation scenes with 20 classes, 18 things and 2 stuff.

For each scene we sample 120 views with a fixed seed and split them at random into input sets of N views, so that the views of a set are spread over the scene with no continuity from one view to the next (Sec. 3.1). Every method receives the same input sets and predicts a panoptic label map for each view of a set. We take $N = 10$ as the default, following the multi-view pose evaluation of feed-forward reconstruction on 10 random frames per sequence (Wang et al., 2024; 2025), and vary $N \in \{5, 10, 20, 40\}$ below and above it in Tables 1 and 3.

Panoptic Lifting computes PQ^{scene} over the held-out views of each scene, one quarter of its frames, and averages it over the 26 scenes of its benchmark. To score the same input as VC-PQ, we treat each input set as the views of a scene, as feed-forward reconstruction does, tile it into a single image as in the official implementation (Siddiqui et al., 2023), and sum the counts of the sets of a scene into one value per scene before averaging over scenes. For both metrics, void pixels are handled as in PQ (Kirillov et al., 2019): predicted pixels on ground-truth void are removed from the prediction, and an unmatched predicted instance with more than half of its pixels on void is not counted as FP. Further details of the protocol are given in Appendix A.

5.2 Evaluating feed-forward methods

We evaluate four publicly available feed-forward methods: PanSt3R (Žust et al., 2025) in its two released versions (v1 and v2), EPS3D (Zhu et al., 2026b), IGGT (Li et al., 2026b), and SegVGGT (Qu et al., 2026). PanSt3R and EPS3D output panoptic labels and are run with their native post-processing. IGGT outputs class-agnostic instances, which we label with the open-vocabulary model LSeg (Li et al., 2022) as EPS3D does. SegVGGT outputs thing instances, which we turn into panoptic labels with the panoptic inference of Mask2Former (Cheng et al., 2022) and evaluate over its thing classes (Table 3 and Appendix C). The two metrics score the same predictions of each method, so the comparison between them does not depend on how the methods were trained.

Table 1 reports PQ^{scene} and VC-PQ of the same predictions at all input set sizes. EPS3D and IGGT score low under both metrics, which we attribute to how they form instances (Appendix C). PanSt3R v2 scores above v1 under both metrics, and for both versions the two metrics move in opposite directions as N grows. PQ^{scene} rises from 46.8 at $N = 5$ to 57.7 at $N = 40$ for v2 on ScanNet++ and from 50.1 to 57.4 on ScanNetv2, whereas VC-PQ falls for every method, from 24.5 to 20.0 and from 43.7 to 40.6 for v2. Overall, VC-PQ is lower than PQ^{scene} for every method and every N. Sec. 5.3 traces both differences to the three properties of PQ^{scene}.

[1] One iPhone capture is excluded because its ground truth is unusable, leaving 49 iPhone scenes.

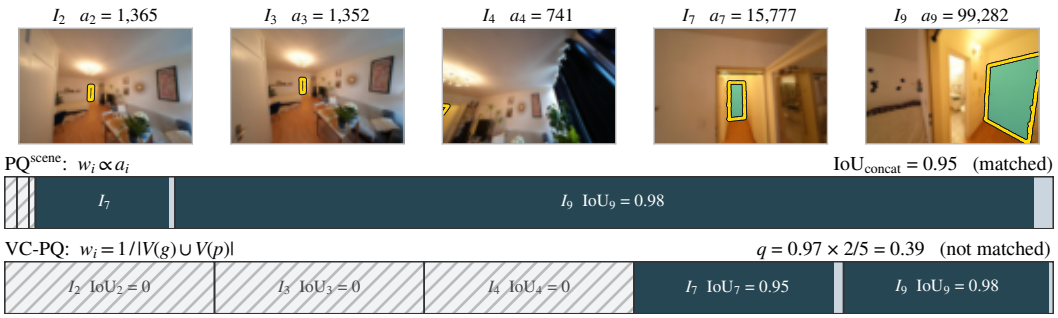


Figure 2: A door (PanSt3R v2, ScanNet++ DSLR, N=10) visible in five views and predicted in the two largest: PQ$^{\text{scene}}$ matches the pair at IoU$_{\text{concat}} = 0.95$, VC-PQ leaves it unmatched at $q = 0.39$. The union area a_9 in I_9 is 84% of the total, so this view alone contributes 0.82 to IoU$_{\text{concat}}$, whereas VC-PQ counts the door as missed in three of its five views, VC $= 2/5$.

Table 2: VC-PQ with the three properties of PQ$^{\text{scene}}$ switched on in every combination (ScanNet++, N=10, all classes). †Differs from PQ$^{\text{scene}}$ of Table 1 only by the visibility threshold. Right block: Shapley attribution of the difference between the two named columns.

	VC-PQ							PQ$^{\text{scene}\dagger}$	PQ$^{\text{scene}\dagger}$ − VC-PQ, Shapley			
area weighting	–	✓	–	–	✓	✓	–	✓		✓		
per-scene avg.	–	–	✓	–	✓	–	✓	✓			✓	
0.5% filter	–	–	–	✓	–	✓	✓	✓				✓
PanSt3R v1	19.2	25.0	24.2	26.0	29.7	35.6	37.2	45.8	+26.6	+7.3	+7.7	+11.5
PanSt3R v2	22.2	28.9	25.6	32.7	31.2	42.7	42.3	51.4	+29.2	+7.8	+6.0	+15.3
EPS3D	2.0	2.8	7.0	2.5	8.8	3.4	11.1	13.9	+11.9	+1.7	+7.6	+2.7
IGGT	4.2	6.2	7.0	6.0	9.5	8.7	12.6	17.1	+12.9	+3.0	+5.4	+4.5

5.3 Comparison with PQ$^{\text{SCENE}}$

Sec. 3.2 locates the failure of both requirements in the area weighting of tiling, and Fig. 2 shows one instance of the predictions of Table 1 on which the two metrics disagree for this reason. To isolate the effect of that weighting, we perturb the ground truth by removing an instance from, or changing its ID in, a fixed share of the views in which it is visible, chosen among the smallest, the largest, or random views (Table A11). VC-PQ is the same regardless of the mask area in the affected views, whereas PQ$^{\text{scene}}$ falls from 91.4 to 39.3 as those views change from the smallest to the largest.

To locate the difference between the two metrics more precisely, Table 2 switches the three properties of PQ$^{\text{scene}}$ on within VC-PQ in every combination over all classes. With all three on, VC-PQ reproduces PQ$^{\text{scene}}$ to within about 0.3 (†), so that the three properties account for the difference between the two metrics. Switching any one property on raises the score of every method, whichever of the other two properties are already on. The right block splits the difference between VC-PQ and the score with all three on over the three properties by Shapley attribution (Shapley, 1953), so that the three shares sum to the difference. For PanSt3R v2 the difference of 29.2 splits into 7.8 from the area weighting, 6.0 from per-scene averaging, and 15.3 from the filter. For EPS3D and IGGT the largest share comes instead from per-scene averaging, but over thing classes (Table A10) that share becomes the smallest for every method, so it comes from stuff classes. Otherwise the filter takes a large share of the difference for every method, the largest for PanSt3R.

The opposite trends with N arise from two of these properties, the area weighting and per-scene averaging. Sweeping γ for PanSt3R v2 (Fig. A1), the score at $\gamma = 0$ falls from 24.5 at $N = 5$ to 20.0 at $N = 40$ on ScanNet++, whereas at $\gamma = 1$ it stays between 28.8 and 29.2, so the fall of VC-PQ with N comes with the equal weighting of views. Under per-scene averaging, in contrast, VC-PQ rises with N, from 25.3 to 28.2, as PQ$^{\text{scene}}$ does (Appendix D).

Table 3: Decomposition of $1 - \text{VC-PQ}$ by Eq. (7) for input sets of N views, thing classes; VC-PQ and the three terms sum to 100.

	N=5				N=10				N=20				N=40			
Method	VC-PQ	$\varepsilon_{\overline{\text{IoU}}}$	ε_{VC}	ε_q	VC-PQ	$\varepsilon_{\overline{\text{IoU}}}$	ε_{VC}	ε_q	VC-PQ	$\varepsilon_{\overline{\text{IoU}}}$	ε_{VC}	ε_q	VC-PQ	$\varepsilon_{\overline{\text{IoU}}}$	ε_{VC}	ε_q
ScanNet++																
PanSt3R v1	21.8	73.1	4.1	1.0	18.7	74.0	4.7	2.5	17.4	74.0	4.3	4.3	16.9	73.6	3.9	5.6
PanSt3R v2	24.7	68.7	5.2	1.4	22.3	67.5	7.2	3.0	20.6	66.7	7.5	5.2	19.9	66.3	7.0	6.8
EPS3D	0.6	98.8	0.5	0.1	0.3	99.1	0.4	0.2	–	–	–	–	–	–	–	–
IGGT	4.9	93.0	1.8	0.3	3.7	93.7	1.9	0.7	–	–	–	–	–	–	–	–
ScanNetv2																
PanSt3R v2	40.6	49.4	8.0	2.0	38.1	49.7	8.4	3.7	37.1	50.0	6.9	6.0	37.1	50.1	5.8	7.1
SegVGGT	48.7	44.0	5.8	1.5	47.0	44.2	5.9	2.9	45.8	44.1	5.4	4.7	44.9	44.5	5.0	5.6

5.4 Error decomposition

Table 3 applies the decomposition of Eq. (7) to each method over thing classes, for which an ID across views is defined. The first term collects the errors within single images and the other two those that arise only across views (Sec. 4.3), so that the table reads as two shares. For every method and every N the largest term is $\varepsilon_{\overline{\text{IoU}}}$, above 90 for EPS3D and IGGT and 44 to 74 for the others, and it changes little with N. Most of the score is thus lost within single images. The two terms across views are what VC-PQ is designed to penalize. ε_{VC} collects the instances whose views disagree severely, with the prediction and the ground truth sharing at most half of the views in which either appears. ε_q collects the instances whose views disagree and whose masks are inaccurate, neither enough to fail on its own, so that the product of the two falls below the matching threshold. Together, the two terms measure how much of the score a method loses to the multi-view setting.

As N grows, ε_{VC} changes little and in no consistent direction, so the severe disagreements across views remain a roughly constant share of the score. ε_q instead rises for every method and dataset, from 1.4 to 6.8 for PanSt3R v2 on ScanNet++ and from 2.0 to 7.1 on ScanNetv2, so more instances fall below the matching threshold on the combination of the two factors. Since $\varepsilon_{\overline{\text{IoU}}}$ also changes little, the fall of VC-PQ with N is accounted for by this term. Appendix E traces the rise to the predictions rather than to the metric, since an instance is matched with the same probability at every N given the fraction of the input set in which it is visible, and larger sets hold more instances that are visible in only a few views (Figs. A4 to A6).

The decomposition thus shows in which respect each method loses its score, and hence in which direction it should improve. PanSt3R v2 improves on v1 in $\varepsilon_{\overline{\text{IoU}}}$, whereas its ε_{VC} and ε_q are larger. On ScanNetv2 SegVGGT is ahead of PanSt3R v2 in all three terms, most in $\varepsilon_{\overline{\text{IoU}}}$. EPS3D and IGGT lose nearly all of their score before view consistency enters.

6 Conclusion

We have formulated multi-view panoptic segmentation on the input views and proposed VC-PQ, which counts every view in which an instance is visible equally and penalizes a prediction that is not visible in the same views as its ground truth. On recent feed-forward methods the two metrics move in opposite directions as the input set grows, which varying the area weighting through γ traces to the equal weighting of views. The decomposition separates the score lost within single images from the part that the multi-view setting costs. We hope that VC-PQ will be reported alongside PQ^{scene}, and that the decomposition serves to locate where a method loses its score and thus to guide its development in the right direction.

VC-PQ is defined for a static scene and a fixed set of input views, and the evaluation covers indoor scans of two datasets and four methods. In indoor scans the ground-truth masks are projected from 3D annotations and are therefore noisy, and VC-PQ, which counts every view equally, can be sensitive to such noise in the mask of an individual view (Table A7). The visibility threshold η and a slight area weighting through γ, which shifts weight toward the views in which an instance appears large, are the two means the metric provides against such noise.

AI USE STATEMENT

In this work, we used generative AI tools to assist in implementing the evaluation code, to search for related work, and to translate and edit the text. The other tasks with required disclosure were not used or are not applicable to this work. We have reviewed all AI-assisted work and take responsibility for the final content of this paper.

ETHICS STATEMENT

This work proposes an evaluation metric and uses the public datasets ScanNet++ (Yeshwanth et al., 2023) and ScanNetv2 (Dai et al., 2017) under their terms of use. No new data were collected.

REPRODUCIBILITY STATEMENT

Sec. 4 defines VC-PQ, Sec. 5.1 the evaluation protocol, and Appendix A the input sets, resolutions, the settings of every compared method, and the bootstrap. The evaluation code, including the lists of views that form every input set, will be released.

REFERENCES

Mehmet Aygün, Aljoša Ošep, Mark Weber, Maxim Maximov, Cyrill Stachniss, Jens Behley, and Laura Leal-Taixé. 4D panoptic LiDAR segmentation. In *CVPR*, 2021. arXiv:2102.12472.

Victor Barberteguy, Ahmet Iscen, Mathilde Caron, Alireza Fathi, Gül Varol, and Cordelia Schmid. Pano3D: Unified 3D reconstruction and panoptic segmentation. In *ECCV*, 2026. arXiv:2606.14307.

Yash Bhalgat, Iro Laina, João F. Henriques, Andrew Zisserman, and Andrea Vedaldi. Contrastive lift: 3D object instance segmentation by slow-fast contrastive fusion. In *NeurIPS*, 2023.

Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. MUSt3R: Multi-view network for stereo 3D reconstruction. In *CVPR*, 2025. arXiv:2503.01661.

Luka Čehovin, Aleš Leonardis, and Matej Kristan. Visual object tracking performance measures revisited. *IEEE TIP*, 25(3):1261–1274, 2016. arXiv:1502.05803.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017.

Ankit Dhiman, Srinath R, Jaswanth Reddy, Lokesh R. Boregowda, and Venkatesh Babu Radhakrishnan. UniC-Lift: Unified 3D instance segmentation via contrastive learning. In *AAAI*, 2026. arXiv:2512.24763.

Bradley Efron. Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82(397):171–185, 1987.

Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. PanopticNeRF-360: Panoramic 3D-to-2D label transfer in urban scenes. *IEEE TPAMI*, 47 (10):8804–8822, 2025. arXiv:2309.10815.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE TPAMI*, 43(5):1562–1577, 2021. arXiv:1810.11981.

Minchao Jiang, Xiaoxuan Ma, Shunyu Jia, Haoru Wang, Zhang Liang, and Wentao Zhu. InstanceSplat: Instance-aware feed-forward 3D Gaussian splatting for scene understanding, 2026. arXiv:2608.07144.

Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *CVPR*, 2020. arXiv:2006.11339.

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019.

Alexander Kirillov et al. COCO panoptic API (panopticapi). `https://github.com/cocodataset/panopticapi`, 2018. Reference implementation of Panoptic Quality.

Sebastian Koch, Johanna Wald, Hidenobu Matsuki, Pedro Hermosilla, Timo Ropinski, and Federico Tombari. Unified semantic transformer for 3D scene understanding. *Trans. Mach. Learn. Res.*, 2026. arXiv:2512.14364.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3D with MASt3R. In *ECCV*, 2024. arXiv:2406.09756.

Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. arXiv:2201.03546.

Changyang Li, Xueqing Huang, Shin-Fang Chng, Huangying Zhan, Qingan Yan, and Yi Xu. FAST3DIS: Feed-forward anchored scene transformer for 3D instance segmentation, 2026a. arXiv:2603.25993.

Hao Li, Zhengyu Zou, Fangfu Liu, Xuanyang Zhang, Fangzhou Hong, Yukang Cao, Yushi Lan, Manyuan Zhang, Gang Yu, Dingwen Zhang, and Ziwei Liu. IGGT: Instance-grounded geometry transformer for semantic 3D reconstruction. In *ICLR*, 2026b. arXiv:2510.22706.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

Xianzhu Liu, Xin Sun, Haozhe Xie, Zonglin Li, Ru Li, and Shengping Zhang. Multi-view consistent 3D panoptic scene understanding. In *AAAI*, volume 39, pp. 5613–5621, 2025. doi:10.1609/aaai.v39i6.32598.

Jieru Mei, Alex Zihao Zhu, Xinchen Yan, Hang Yan, Siyuan Qiao, Yukun Zhu, Liang-Chieh Chen, Henrik Kretzschmar, and Dragomir Anguelov. Waymo Open Dataset: Panoramic video panoptic segmentation. In *ECCV*, 2022. arXiv:2206.07704.

Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. ViP-DeepLab: Learning visual perception with depth-aware video panoptic segmentation. In *CVPR*, 2021. arXiv:2012.05258.

Jinyuan Qu, Hongyang Li, and Lei Zhang. SegVGGT: Joint 3D reconstruction and instance segmentation from multi-view images. In *ECCV*, 2026. arXiv:2603.19926.

David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3D semantic segmentation in the wild. In *ECCV*, 2022.

Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, 1953.

Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3D scene understanding with neural fields. In *CVPR*, 2023.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *CVPR*, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *CVPR*, 2024.

Mark Weber, Jun Xie, Maxwell Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, Aljoša Ošep, Laura Leal-Taixé, and Liang-Chieh Chen. STEP: Segmenting and tracking every pixel. In *NeurIPS*, 2021. arXiv:2102.11859; NeurIPS Datasets and Benchmarks Track.

Qi Xu, Dongxu Wei, Lingzhe Zhao, Wenpu Li, Zhangchi Huang, Shunping Ji, and Peidong Liu. SIU3R: Simultaneous scene understanding and 3D reconstruction beyond feature alignment. In *NeurIPS*, 2025. arXiv:2507.02705.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3D scenes. In *ECCV*, 2024.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *ICCV*, 2023.

Chenming Zhu, Peizhou Cao, Jingli Lin, Wenbo Hu, Yunlong Ran, Jiangmiao Pang, Tai Wang, and Xihui Liu. G^2TAM: Geometry grounded track anything model. In *ICML*, 2026a. arXiv:2607.03789.

Runsong Zhu, Shi Qiu, Qianyi Wu, Ka-Hei Hui, Pheng-Ann Heng, and Chi-Wing Fu. PCF-Lift: Panoptic lifting by probabilistic contrastive fusion. In *ECCV*, 2024. arXiv:2410.10659.

Runsong Zhu, Shi Qiu, Zhengzhe Liu, Ka-Hei Hui, Qianyi Wu, Pheng-Ann Heng, and Chi-Wing Fu. Rethinking end-to-end 2D to 3D scene segmentation in Gaussian splatting. In *CVPR*, 2025. arXiv:2503.14029.

Runsong Zhu, Jiaxin Guo, Xiaoyang Guo, Zhengzhe Liu, Ka-Hei Hui, Wei Yin, Kai Chen, Wei Chen, Weiqiang Ren, Yunhui Liu, Pheng-Ann Heng, and Chi-Wing Fu. EPS3D: End-to-end feed-forward 3D panoptic segmentation. In *ICML*, 2026b. arXiv:2606.08980.

Lojze Žust, Yohann Cabon, Juliette Marrie, Leonid Antsfeld, Boris Chidlovskii, Jerome Revaud, and Gabriela Csurka. PanSt3R: Multi-view consistent panoptic segmentation. In *ICCV*, 2025.

A Evaluation protocol details

ScanNet++ has 50 validation scenes, each captured with a DSLR camera at 920×613 and with an iPhone at 960×720, and ScanNetv2 has 312 validation scenes at 1296×968. From every capture we sample 120 views with seed 42, shuffle them, and slice the sequence into input sets of N consecutive elements, which gives 24, 12, 6, and 3 sets per scene for $N = 5, 10, 20, 40$. Within a set the views are ordered by their capture index. At $N = 10$ a set holds on average 51.2 ground-truth instances above the visibility threshold on the DSLR captures, 44.0 on the iPhone captures, and 13.9 on ScanNetv2. For comparison, Panoptic Lifting reports PQ^{scene} on 26 scenes of HyperSim, ScanNet, and Replica (6, 12, and 8 scenes), with one quarter of the frames of each scene held out as test views (Siddiqui et al., 2023).

PanSt3R v1 assigns pixels to queries by simulated annealing, which is run once per input set without a fixed seed, so a second run of the same set can give a different label map. Between two such runs on the same input sets, PQ^{scene} differs by at most 1.0 and VC-PQ by at most 0.1 at $N = 10$ on either camera. PanSt3R v2 is run with its released two-pass assignment and class-score threshold of 0.1. EPS3D receives the full frames of a set and does not run at $N \geq 20$ on a 24 GB GPU, nor does the clustering of IGGT. PanSt3R and EPS3D are trained on ScanNet++, EPS3D also on ScanNet, and IGGT on ScanNet++ among other datasets. The training split of EPS3D on ScanNet++ is not specified, so an overlap with the validation scenes cannot be excluded, and the comparison across methods does not control for training data. IGGT outputs an instance embedding per pixel and clusters it over the views of a set. Each cluster is given the class whose text embedding is closest to the average of the LSeg (Li et al., 2022) pixel features over the cluster mask, with the same prompts as EPS3D. SegVGGT is run from its released checkpoint on ScanNetv2, and its instance masks and class scores are turned into a panoptic label map with the panoptic inference of Mask2Former (Cheng et al., 2022), and pixels that no mask claims are void. Its masks are binarized at 0.4, masks with fewer than 200 pixels are dropped, the 600 highest-scoring masks are kept, and no threshold is applied to the class score. Table A1 lists the 41 prompts that the open-vocabulary methods receive on ScanNetv2, namely the names of the other 19 classes and, in place of otherfurniture, the 22 ScanNet200 categories that the official label mapping assigns to it.

Since the DSLR and iPhone runs of ScanNet++ capture the same scenes, confidence intervals are obtained by a bootstrap over the 50 physical scenes that keeps both captures of a drawn scene

Table A1: The 41 prompts given to the open-vocabulary methods on ScanNetv2, in alphabetical order within each group (read down each column). The first group are the names of the other 19 classes, each scored as its own class. The second group are the 22 ScanNet200 categories (Rozenberszki et al., 2022) that the official label mapping assigns to otherfurniture, all scored as otherfurniture.

Scored as its own class (19)			
bathtub	counter	picture	table
bed	curtain	refrigerator	toilet
bookshelf	desk	shower curtain	wall
cabinet	door	sink	window
chair	floor	sofa	
Scored as otherfurniture (22)			
bench	keyboard piano	radiator	tv stand
closet	ladder	recycling bin	wardrobe
clothes dryer	music stand	seat	washing machine
crate	ottoman	stand	water cooler
furniture	piano	trash bin	
ironing board	rack	trash can	

together, and on ScanNetv2 it draws the 312 scenes. For each of $B = 10{,}000$ draws both metrics are recomputed from the per-set counts with the aggregation of Sec. 5.1, and the interval is the 95% BCa interval (Efron, 1987), whose acceleration constant is estimated by a leave-one-scene-out jackknife. The tables of the paper give point estimates, and Table A3 gives the intervals at $N = 10$.

B Additional definitions

PQ factors into a segmentation quality and a recognition quality, $\mathrm{PQ} = \mathrm{SQ} \times \mathrm{RQ}$, with SQ the mean IoU over the matched pairs and RQ the F1 score of the matching (Kirillov et al., 2019). VC-PQ factors in the same way with the pair score in place of the IoU,

$$\text{VC-SQ} = \frac{\sum_{(p,g)\in \mathrm{TP}} q(p,g)}{|\mathrm{TP}|}, \qquad \text{VC-RQ} = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + \frac{1}{2}|\mathrm{FP}| + \frac{1}{2}|\mathrm{FN}|},$$

so that VC-PQ = VC-SQ × VC-RQ for each class. As for PQ, the product holds per class and not for the class averages. Table A15 in Appendix F gives the two factors at every N.

The matching of Sec. 4 is unique because no instance can have a pair score above 0.5 with two counterparts. For a predicted instance p and two ground-truth instances g and g' of the same class, the masks $M_i(g)$ and $M_i(g')$ are disjoint in every view, so $\mathrm{IoU}_i(p,g) + \mathrm{IoU}_i(p,g') \le 1$ for every i. Summing over the views of $V(p)$ and dividing by $|V(p) \cup V(g)| \ge |V(p)|$ gives $q(p,g) + q(p,g') \le 1$, so at most one of the two exceeds 0.5. The same argument applies to two predicted instances and one ground-truth instance.

The 0.5% class filter of $\mathrm{PQ}^{\mathrm{scene}}$ (Sec. 3.2) is not described in the Panoptic Lifting paper. It is `get_non_robust_classes_for_image` in the official code, applied with a threshold of 0.005 after ground-truth void pixels are removed. The terms of Eq. (7) are computed for each class and averaged over classes as VC-PQ is, which preserves the identity. The identity requires the score of Eq. (4), the product of the two quantities of Eq. (3), and does not hold for q_γ of Eq. (6) with $\gamma > 0$.

C Evaluating feed-forward methods, additional results

Table A2 gives the scores of Table 1 at $N = 10$ for each camera of ScanNet++, and Table A3 the confidence intervals of the pooled values. The aggregation is that of Table 1 applied to the runs of one camera, and the iPhone values are over the 49 scenes with usable ground truth. Table A4 splits the scores at $N = 10$ into thing and stuff classes and adds SegVGGT on ScanNetv2. Thing and stuff values of $\mathrm{PQ}^{\mathrm{scene}}$ apply the 0.5% filter over all classes, with the labeled pixels of the set as denominator, and then average the classes of the subset. A scene run without any class of the

Table A2: PQ^{scene} and VC-PQ on ScanNet++ at N=10 for each camera, aggregated as in Table 1. The DSLR and iPhone runs capture the same 50 scenes, and the iPhone values exclude the scene without usable ground truth (Sec. 5.1).

	DSLR		iPhone	
Method	PQ^{scene}	VC-PQ	PQ^{scene}	VC-PQ
PanSt3R v1	48.8	20.7	42.5	17.5
PanSt3R v2	54.1	22.1	48.5	22.5
EPS3D	13.2	1.9	14.1	2.1
IGGT	16.6	3.8	17.7	4.7

Table A3: 95% BCa confidence intervals (Efron, 1987) for the scores of Table 1 at N=10. On ScanNet++ the bootstrap draws the 50 physical scenes with replacement and keeps the DSLR and iPhone runs of a drawn scene together; on ScanNetv2 it draws the 312 scenes. *The bound lies at the most extreme of the B=10,000 draws.

	ScanNet++		ScanNetv2	
Method	PQ^{scene}	VC-PQ	PQ^{scene}	VC-PQ
PanSt3R v1	45.7 [43.3, 48.1]	19.2 [17.2, 21.2]	–	–
PanSt3R v2	51.3 [48.9, 53.8]	22.2 [20.5, 24.1]	52.5 [51.0, 54.0]	41.5 [39.8, 43.3]
EPS3D	13.6 [12.3, 15.2]	2.0 [1.8, 2.1]*	21.5 [20.8, 22.2]	8.2 [8.0, 8.4]
IGGT	17.1 [15.1, 19.7]	4.2 [3.6, 4.8]	–	–

subset is left out of that column. Table A5 gives, for the thing classes of ScanNet++ at $N = 10$, the counts behind the low scores of EPS3D and IGGT in Sec. 5.2. EPS3D splits half of the ground-truth instances over two or more of its predictions, and only about a fifth of its predictions carry the class of the instance they overlap most. IGGT predicts fewer than half as many instances as the ground truth holds, and scoring with all classes merged raises its VC-PQ from 4.2 to 14.0. The first three columns are the matched and unmatched instances under VC-PQ, and the ratio of predicted to ground-truth instances is $(|\mathrm{TP}| + |\mathrm{FP}|)/(|\mathrm{TP}| + |\mathrm{FN}|)$. The class-agnostic VC-PQ merges all classes, things and stuff, into one before matching. For each predicted instance we find the ground-truth instance it overlaps most, with pixels summed over the views of the set and void excluded, and count whether the two classes agree. A prediction that overlaps only void counts as a disagreement. A ground-truth instance counts as split when two or more predicted instances each cover at least 10% of its pixels. The last two columns are counted on the label maps without the visibility threshold, and the two cameras are averaged with weights given by their numbers of input sets, 600 and 588.

D Comparison with PQ^{scene}, additional results

Table A6 counts, on the runs of Table A11, the appearances that the four methods miss and those they cover with an instance other than the one that covers the object in most views of the same input set, as a share of all appearances, the same denominator as the perturbation rates of Table A11. Table A7 gives the erosion perturbation of Sec. 5.3. Every thing mask is eroded by r pixels with a 3×3 structuring element, the eroded pixels become void, and the image border is not eroded. The last row is the share of appearances that fall below the visibility threshold after erosion and thus leave the support of the perturbed instance. Table A8 gives the two aggregations of Sec. 5.3 for PanSt3R v2 at every N. Per-scene averages the class-averaged value of each scene run, and pooled sums the counts of all sets before the class average. The first and fourth rows are the values of Table 1, the last row is PQ^{scene} without the 0.5% filter under pooled counts, and the per-scene rows average the 99 scene runs. Table A9 repeats VC-PQ at $N = 10$ for $\eta = 50$, 100, and 200 pixels, aggregated as in Table 1. Its last row is the share of ground-truth instances, over thing and stuff classes of both cameras, whose mask falls below η in every view of the set and which are therefore excluded from evaluation. At $\eta = 50$ no instance is excluded, because appearances below 50 pixels are already removed when the label maps are prepared. Fig. A1 sweeps γ of Eq. (6) for PanSt3R v2 with one curve per N on both datasets. Table A10 repeats Table 2 over thing classes, the classes over which Table 3 decomposes the score, so that the two tables can be read on the same classes. Over thing

Table A4: The scores of Table 1 at $N{=}10$ over all classes, thing classes, and stuff classes. The identity of an instance across views is defined only for things, since each stuff class forms a single instance, so that on stuff the view-consistency factor measures only the mismatch between the sets of views in which a class appears. SegVGGT predicts instances only and enters in the thing columns.

	PQ^{scene}			VC-PQ		
Method	all	things	stuff	all	things	stuff
ScanNet++						
PanSt3R v1	45.7	42.1	55.6	19.2	18.7	21.7
PanSt3R v2	51.3	48.4	59.4	22.2	22.3	21.8
EPS3D	13.6	1.3	45.5	2.0	0.3	10.7
IGGT	17.1	11.8	32.5	4.2	3.7	6.7
ScanNetv2						
PanSt3R v2	52.5	44.1	80.6	41.5	38.1	71.5
EPS3D	21.5	2.8	80.9	8.2	1.4	69.3
SegVGGT	–	46.3	–	–	47.0	–

Table A5: Predicted instances of thing classes on ScanNet++ ($N{=}10$, DSLR and iPhone). Counts of matched and unmatched instances under VC-PQ, the number of predicted instances per ground-truth instance (both above the visibility threshold), VC-PQ with all classes, things and stuff, merged, the share of predicted instances whose class agrees with the ground-truth instance they overlap most, and the share of ground-truth instances split over two or more predicted instances.

Method	$\lvert TP\rvert$	$\lvert FP\rvert$	$\lvert FN\rvert$	pred. / GT	class-agnostic VC-PQ	class agrees (%)	GT split (%)
PanSt3R v1	11,044	11,905	39,334	0.46	29.1	56	21
PanSt3R v2	15,351	21,625	35,027	0.73	31.9	52	23
EPS3D	346	38,581	50,032	0.77	4.5	22	49
IGGT	2,756	18,758	47,622	0.43	14.0	31	17

classes the share of per-scene averaging is the smallest of the three for every method, 20, 12, 24, and 22% of the difference for PanSt3R v1, PanSt3R v2, EPS3D, and IGGT, whereas over all classes (Table 2) it is the largest for EPS3D and IGGT. The eight combinations and the Shapley attribution are defined as in Table 2, and the row with all three properties switched on differs from PQ^{scene} over thing classes (Table A4) only by the visibility threshold. Fig. A2 shows the three properties of PQ^{scene} of Sec. 3.2 on real data. In (a), PQ^{scene} at the operating point of Table A11 is 91.4 when the smallest appearances are removed and 39.3 when the largest are, whereas VC-PQ is 80.9 in both cases. In (b), the same counts give 42.7 when pooled over all input sets and 51.3 when averaged per scene run, and floor, wall, and ceiling account for 7% of the first score and 29% of the second. A per-scene average counts a class once in every scene run in which it is scored, so raising the PQ of floor by 10 points raises the score by 0.62 and that of socket by 0.02, whereas under pooling any class raises it by 0.11. In (c), 0.5% of the labelled pixels of an input set of 40 views is about 88,000 pixels on the DSLR captures, 16% of one image, and 58 of the 98 classes are removed in more than half of the input sets that contain them. Table A11 gives the perturbation of Sec. 5.3 at its operating point, with the three variants of each perturbation and the four scores. We take the ground-truth label maps of the ScanNet++ DSLR runs at $N = 10$, restricted to thing classes, apply a perturbation, and score the result against the original, so that the unperturbed maps score 100 under every metric. An appearance is a pair of an instance and a view in which it is visible, and the two perturbations act on appearances. A missed appearance removes the instance from the view, and an identity split gives its pixels in that view a new ID of the same class. Both act on the same number of appearances per instance, about 30% of the views in which it is visible, which amounts to 28% of all appearances at the operating point. They differ only in which views are chosen, namely the views in which the instance appears smallest, those in which it appears largest, or views drawn at random. The operating point lies below the share of appearances that PanSt3R v2 misses on the same runs (55%) and far above the share of appearances it covers with an ID other than the one it gives the instance in most views of the same input set (1.1%). In addition to the two metrics, the table reports $VC\text{-}PQ_1$, which applies the area weighting alone, and per-image PQ, which scores each view independently and is

Table A6: Errors of the predictions on the runs of Table A11 (ScanNet++ DSLR, N=10, thing classes), counted over the same appearances. An appearance is covered when a predicted instance overlaps it with an IoU above 0.5 in that view. The first column is the share of appearances that are not covered, and the second the share of appearances covered by a predicted instance other than the one that covers the instance in the largest number of views of the same input set.

Method	not covered (%)	covered by another ID (%)
PanSt3R v1	63.9	0.4
PanSt3R v2	54.8	1.1
EPS3D	92.0	0.6
IGGT	80.7	1.2

Table A7: Scores of the ground truth of Table A11 with every thing mask eroded by r pixels, scored against the unperturbed ground truth. Eroded pixels become void, and the image border is not eroded. The last row is the share of appearances that fall below the visibility threshold after erosion.

erosion radius r (px)	1	2	4	8
$\mathrm{PQ}^{\mathrm{scene}}$	85.9	77.0	63.3	45.2
VC-PQ	61.3	40.2	20.3	8.7
$\mathrm{VC\text{-}PQ}_1$ (γ=1)	70.3	51.3	30.0	15.2
per-image PQ	61.7	41.9	24.3	12.3
appearances below η (%)	13.2	23.4	38.1	55.0

therefore unaffected by a change of ID across views. The VC-PQ column is identical across the three variants, as predicted before the run, and $\mathrm{VC\text{-}PQ}_1$ follows $\mathrm{PQ}^{\mathrm{scene}}$, which places the difference between the two metrics on the area weighting rather than on the counting or the visibility threshold.

The instance of Fig. 2 is taken from the ScanNet++ DSLR runs of PanSt3R v2 at $N = 10$, scored by both metrics without the 0.5% filter ($\eta = 100$, $\gamma = 0$). It was chosen by hand among the 971 thing instances of this cell that $\mathrm{PQ}^{\mathrm{scene}}$ matches and VC-PQ does not, as a legible example of the area difference, and it is not the highest-ranked case under either of the two orderings we tried.

E Error decomposition, additional results

The dependence of ε_q on N in Table 3 lies in the predictions rather than in the metric. Whether a candidate is matched depends on the fraction of the input set in which the instance is visible and not on N itself, and as N grows more instances are visible in a small fraction of the set (Figs. A4 to A6).

Fig. A3 shows the three tests as regions of the plane of $\overline{\mathrm{IoU}}(p, g)$ and $\mathrm{VC}(p, g)$ and places the unmatched instances of PanSt3R v2 behind Table 3 at $N = 10$ in them. Since $q(p, g)$ is the product of the two factors, the pairs with $q(p, g) = 0.5$ lie on the curve $\mathrm{VC}(p, g) = 0.5/\overline{\mathrm{IoU}}(p, g)$, and the region between this curve and the two lines at 0.5 holds the pairs that pass both tests but not their product. In the classification of unmatched instances (Sec. 4.3), the test on $\overline{\mathrm{IoU}}(p, g)$ precedes the test on $\mathrm{VC}(p, g)$ because $\mathrm{VC}(p, g)$ compares supports alone and equals one for any two instances that are visible in the same views, so that a candidate whose mask does not overlap with g carries no evidence about g, however consistent its support. For instance, an instance with one candidate at $\overline{\mathrm{IoU}} = 0.2$, $\mathrm{VC} = 1$ and another at $\overline{\mathrm{IoU}} = 1$, $\mathrm{VC} = 0.5$ falls in $\mathrm{FN}_{\mathrm{VC}}$, whereas testing view consistency first would place it in $\mathrm{FN}_{\overline{\mathrm{IoU}}}$. The candidates of an unmatched ground-truth instance are all predicted instances of the same class in the input set, matched or not, and likewise for an unmatched predicted instance with the roles of p and g exchanged.

Table A12 gives the decomposition of Table 3 over all classes, and its VC-PQ column is that of Table 1. SegVGGT predicts thing classes only and is given over thing classes alone, in Table 3. Table A13 gives the counts behind Table 3, namely the ground-truth and predicted instances and the unmatched ones grouped by the first test that fails. Table A14 counts the predicted instances of PanSt3R v2 that fail the view-consistency test by the number of views in which they are visible, where $|V(p)|$ counts the views in which the predicted mask, without its pixels on ground-truth void,

Table A8: PQ^{scene} and VC-PQ of PanSt3R v2 on ScanNet++ for input sets of N views under two aggregations. Per-scene averages the class-averaged value of each scene run, and pooled sums the counts of all input sets before the class average. The first and fourth rows are the values of Table 1.

Score	0.5% filter	aggregation	N=5	N=10	N=20	N=40
PQ^{scene}	on	per-scene	46.8	51.3	54.7	57.7
PQ^{scene}	on	pooled	41.3	42.7	40.9	44.2
VC-PQ	off	per-scene	25.3	25.6	26.9	28.2
VC-PQ	off	pooled	24.5	22.2	20.7	20.0
PQ^{scene}	off	pooled	28.6	28.6	28.6	29.1

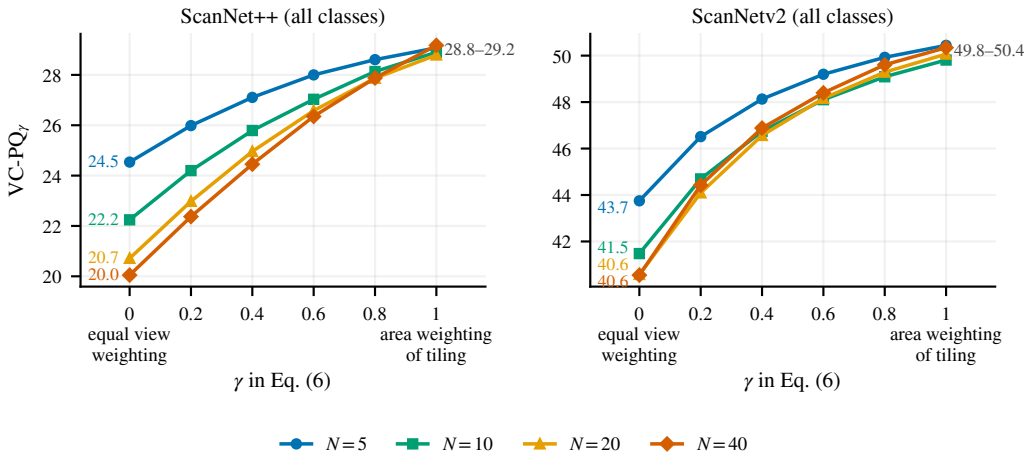


Figure A1: $VC\text{-}PQ_\gamma$ of PanSt3R v2 against γ in Eq. (6), one curve per input set size N, over all classes. The predictions are identical along each curve; only the weighting of views changes. $\gamma = 0$ is VC-PQ and $\gamma = 1$ applies the area weighting of tiling.

has at least η pixels. Fig. A4 and Fig. A5 give, for the instances that are visible in at least two views and have a same-class counterpart with $\overline{\text{IoU}}(p, g) > 0.5$, the share that is matched as a function of the fraction $|V|/N$ of the input set in which the instance is visible, for $\gamma = 0$ and $\gamma = 1$. Fig. A6 gives the distribution of that fraction over the ground-truth instances at each N.

F Factorization of VC-PQ

Table A15 gives VC-PQ and its two factors of Appendix B for every input set size, aggregated as in Table 1.

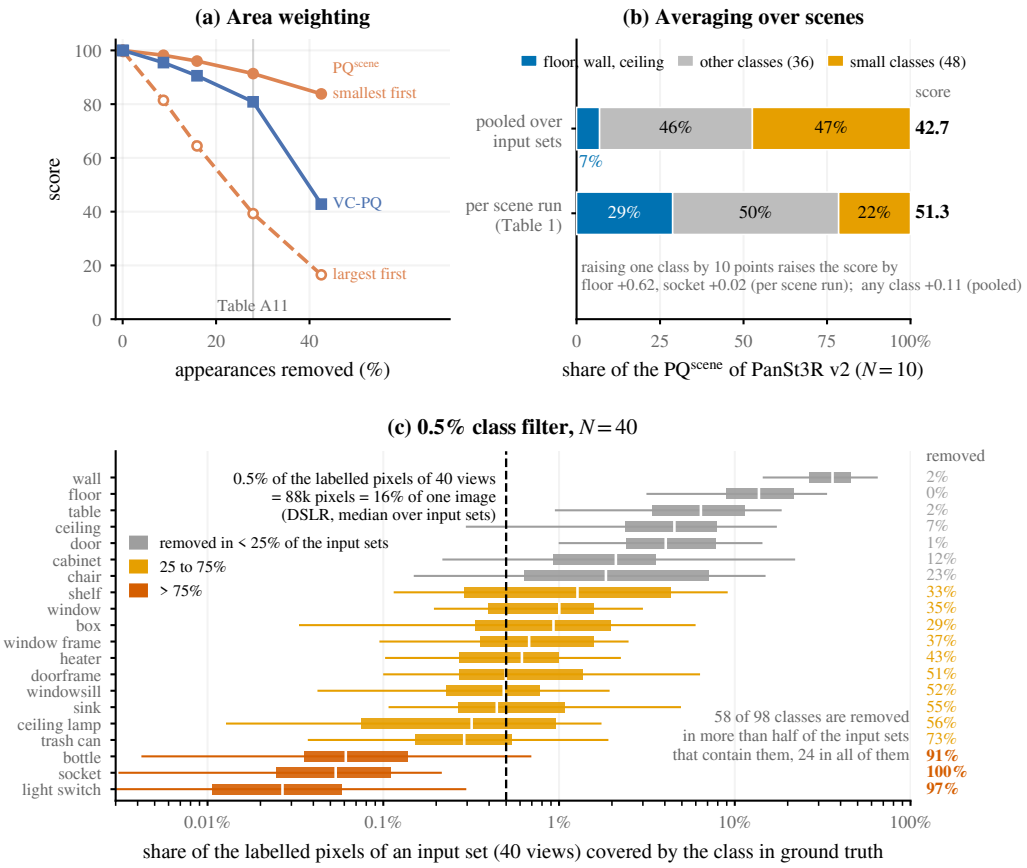


Figure A2: The three properties of PQ^{scene} of Sec. 3.2 on real data. (a) Area weighting: ground truth of the ScanNet++ DSLR runs (N=10, thing classes) with the same number of appearances removed, in the views in which an instance appears smallest or in those in which it appears largest, scored against the unperturbed ground truth; the vertical line marks the operating point of Table A11. (b) Averaging over scenes: the counts of PanSt3R v2 on ScanNet++ (N=10, all classes, 0.5% filter on), averaged over classes after pooling all input sets and after averaging per scene run (Table 1), split into the shares of floor, wall, and ceiling, of the small classes (median share of the labelled pixels of an input set below 0.5% at N=10), and of the other classes. (c) The 0.5% class filter at N=40: for the 20 classes present in the most input sets, the share of the labelled pixels of an input set that the class covers in the ground truth (box: quartiles, line: 5th to 95th percentile), and on the right the share of the input sets containing the class in which the filter removes it.

Table A9: VC-PQ on ScanNet++ at N=10 for three values of the visibility threshold η, aggregated as in Table 1. The last row is the share of ground-truth instances whose mask has fewer than η pixels in every view of the set and that are therefore excluded from evaluation.

	η=50	η=100	η=200
PanSt3R v1	18.5	19.2	20.2
PanSt3R v2	21.5	22.2	23.4
EPS3D	2.0	2.0	2.1
IGGT	4.0	4.2	4.5
ground-truth instances excluded (%)	0.0	2.5	6.9

Table A10: Table 2 restricted to thing classes. VC-PQ with the three properties of PQ^{scene} switched on in every combination, on ScanNet++ at $N=10$. †With all three switched on, the score differs by about 0.4 from PQ^{scene} over thing classes of Table A4; unlike PQ^{scene}, it applies the visibility threshold. The right block splits the difference between the two named columns over the three properties by Shapley attribution.

	VC-PQ							$PQ^{scene\dagger}$	$PQ^{scene\dagger}$ − VC-PQ, Shapley			
area weighting	–	✓	–	–	✓	✓	–	✓		✓		
per-scene avg.	–	–	✓	–	✓	–	✓	✓			✓	
0.5% filter	–	–	–	✓	–	✓	✓	✓				✓
PanSt3R v1	18.7	24.5	20.8	25.2	26.1	35.0	33.5	42.2	+23.5	+7.3	+4.8	+11.4
PanSt3R v2	22.3	28.9	23.1	33.1	28.3	42.6	39.8	48.5	+26.1	+7.5	+3.2	+15.4
EPS3D	0.3	0.7	0.4	0.5	0.8	1.1	0.8	1.7	+1.4	+0.6	+0.3	+0.5
IGGT	3.7	5.7	4.0	5.3	6.1	8.0	7.5	11.8	+8.1	+2.9	+1.8	+3.4

Table A11: Ground truth of ScanNet++ DSLR ($N=10$, thing classes) perturbed at one operating point and scored against the original, so that the unperturbed label maps score 100. In each group the same number of appearances, 28% of all, is removed (missed appearance) or given a new ID (identity split), and the three variants differ only in which views are chosen: the appearances of smallest area first, of largest area first, or at random (mean of three seeds). Per-image PQ scores each view on its own.

Perturbation	Variant	PQ^{scene}	VC-PQ	VC-PQ_1 ($\gamma=1$)	per-image PQ
none	–	100.0	100.0	100.0	100.0
missed appearance	smallest area first	91.4	80.9	92.9	86.7
	largest area first	39.3	80.9	50.7	86.7
	random	72.9	80.9	77.2	86.7
identity split	smallest area first	65.2	63.2	72.3	100.0
	largest area first	51.6	63.2	61.9	100.0
	random	55.8	63.2	65.9	100.0

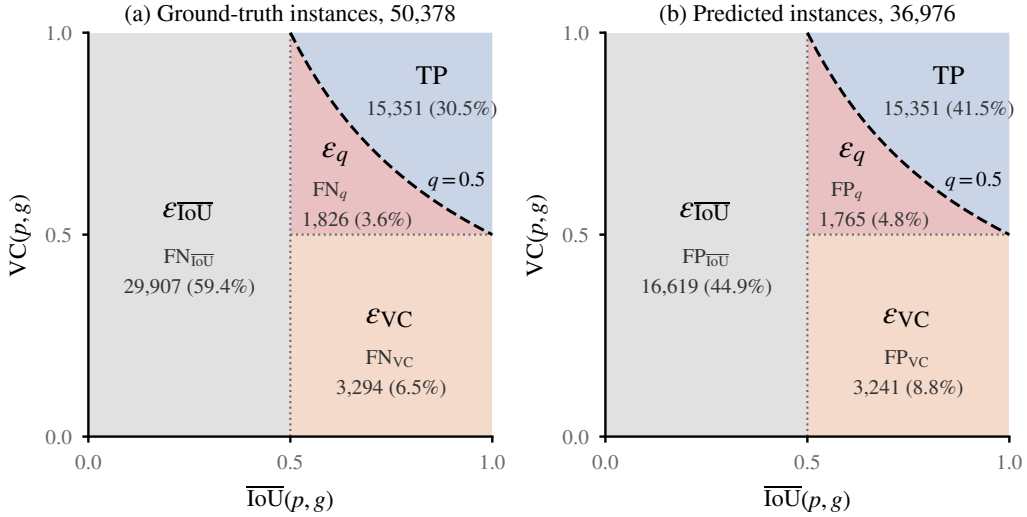


Figure A3: The plane of the two factors of Eq. (4), $\overline{\mathrm{IoU}}(p,g)$ and $\mathrm{VC}(p,g)$, divided by the three tests of Sec. 4.3, for PanSt3R v2 on ScanNet++ at $N=10$ (thing classes): (a) ground-truth instances, (b) predicted instances. Each region is labelled with the term of Eq. (7) that its unmatched instances enter, with their number and share of all instances; the dashed curve is $q(p,g)=0.5$. Matched pairs also enter $\varepsilon_{\overline{\mathrm{IoU}}}$ with $1-\overline{\mathrm{IoU}}(p,g)$ and $\varepsilon_{\mathrm{VC}}$ with $\overline{\mathrm{IoU}}(p,g)\,(1-\mathrm{VC}(p,g))$, which the regions do not show.

Table A12: The decomposition of Table 3 over all classes, for input sets of N views (ScanNet++ pools the DSLR and iPhone runs). VC-PQ and the three terms sum to 100 in each block, and VC-PQ equals Table 1. SegVGGT predicts thing classes only and is reported over thing classes alone (Table 3). –: as in Table 1.

	N=5				N=10				N=20				N=40			
Method	VC-PQ	$\varepsilon_{\overline{\mathrm{IoU}}}$	$\varepsilon_{\mathrm{VC}}$	ε_q	VC-PQ	$\varepsilon_{\overline{\mathrm{IoU}}}$	$\varepsilon_{\mathrm{VC}}$	ε_q	VC-PQ	$\varepsilon_{\overline{\mathrm{IoU}}}$	$\varepsilon_{\mathrm{VC}}$	ε_q	VC-PQ	$\varepsilon_{\overline{\mathrm{IoU}}}$	$\varepsilon_{\mathrm{VC}}$	ε_q
ScanNet++																
PanSt3R v1	22.0	72.8	4.0	1.1	19.2	73.6	4.6	2.6	17.9	73.7	4.1	4.2	17.4	73.6	3.7	5.3
PanSt3R v2	24.5	68.8	5.1	1.5	22.2	67.6	7.0	3.1	20.7	67.0	7.1	5.2	20.0	66.6	6.6	6.8
EPS3D	2.4	96.8	0.6	0.2	2.0	97.1	0.5	0.3	–	–	–	–	–	–	–	–
IGGT	5.7	92.2	1.8	0.4	4.2	93.2	1.8	0.7	–	–	–	–	–	–	–	–
ScanNetv2																
PanSt3R v2	43.7	46.7	7.5	2.0	41.5	46.9	7.9	3.7	40.6	47.2	6.5	5.7	40.6	47.2	5.5	6.7
SegVGGT	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Table A13: Counts behind Table 3 (thing classes): ground-truth and predicted instances, and the unmatched ones grouped by the first of the three tests, $\overline{\mathrm{IoU}}(p,g) > 0.5$, $\mathrm{VC}(p,g) > 0.5$, and $q(p,g) > 0.5$, that fails (Sec. 4.3).

		instances		unmatched ground truth			unmatched predictions		
Method	N	GT	pred.	$\lvert\mathrm{FN}_{\overline{\mathrm{IoU}}}\rvert$	$\lvert\mathrm{FN}_{\mathrm{VC}}\rvert$	$\lvert\mathrm{FN}_q\rvert$	$\lvert\mathrm{FP}_{\overline{\mathrm{IoU}}}\rvert$	$\lvert\mathrm{FP}_{\mathrm{VC}}\rvert$	$\lvert\mathrm{FP}_q\rvert$
ScanNet++									
PanSt3R v1	5	80,635	38,468	56,078	2,970	915	14,310	2,632	854
	10	50,378	22,949	36,256	1,828	1,250	9,114	1,619	1,172
	20	28,608	12,633	21,033	781	1,017	5,218	690	948
	40	15,058	6,493	11,235	312	659	2,772	268	601
PanSt3R v2	5	80,635	58,103	47,842	4,563	1,331	25,503	4,406	1,295
	10	50,378	36,976	29,907	3,294	1,826	16,619	3,241	1,765
	20	28,608	21,531	16,973	1,787	1,609	9,979	1,763	1,550
	40	15,058	11,587	8,972	820	1,048	5,549	811	1,009
EPS3D	5	80,635	65,001	78,475	992	200	62,989	859	185
	10	50,378	38,927	49,481	406	145	38,089	354	138
IGGT	5	80,635	40,029	71,204	2,315	649	30,764	2,199	599
	10	50,378	21,514	45,937	1,113	572	17,158	1,063	537
ScanNetv2									
PanSt3R v2	5	73,066	72,143	28,767	6,477	1,669	29,117	5,294	1,579
	10	44,726	46,158	17,365	3,798	2,065	19,481	3,250	1,929
	20	24,831	26,717	9,508	1,470	1,810	11,618	1,341	1,715
	40	12,721	13,915	4,867	473	1,111	6,140	439	1,066
SegVGGT	5	73,066	91,518	22,011	4,326	1,131	40,363	4,467	1,090
	10	44,726	58,006	13,097	2,710	1,572	26,322	2,864	1,473
	20	24,831	33,774	6,972	1,177	1,440	15,892	1,276	1,364
	40	12,721	18,205	3,451	463	940	8,952	521	865

Table A14: Predicted instances that fail the view-consistency test ($\mathrm{FP}_{\mathrm{VC}}$) for PanSt3R v2 on ScanNet++ (thing classes), by the number of views $|V(p)|$ in which they are visible.

		share by $\lvert V(p)\rvert$ (%)		
N	$\lvert\mathrm{FP}_{\mathrm{VC}}\rvert$	1	2	≥ 3
5	4,406	77.4	19.2	3.4
10	3,241	65.2	24.0	10.8
20	1,763	45.1	28.1	26.8
40	811	26.6	19.5	53.9

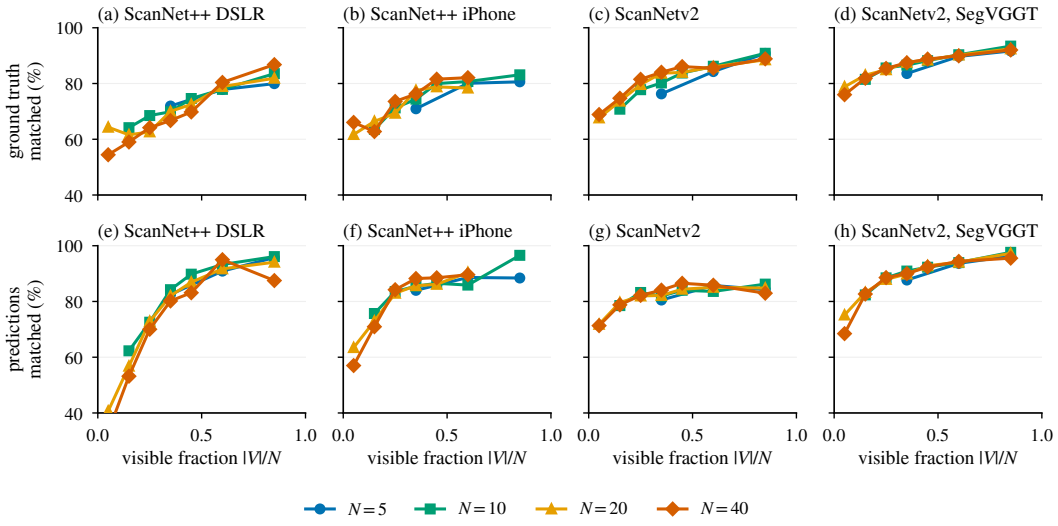


Figure A4: Share of matched candidates as a function of the fraction $|V|/N$ of the input set in which the instance is visible, for $\gamma = 0$ (VC-PQ), one curve per N, over thing classes visible in at least two views. Top: ground-truth instances, bottom: predicted instances. (a)–(c) and (e)–(g) are PanSt3R v2 on the three captures, (d) and (h) SegVGGT on ScanNetv2. A candidate is an instance with a same-class counterpart at $\overline{\mathrm{IoU}}(p, g) > 0.5$, and bins with fewer than 30 instances are omitted. The fraction is binned, with bin edges at 0.1, 0.2, 0.3, 0.4, 0.5, and 0.7, and each marker sits at the centre of its bin.

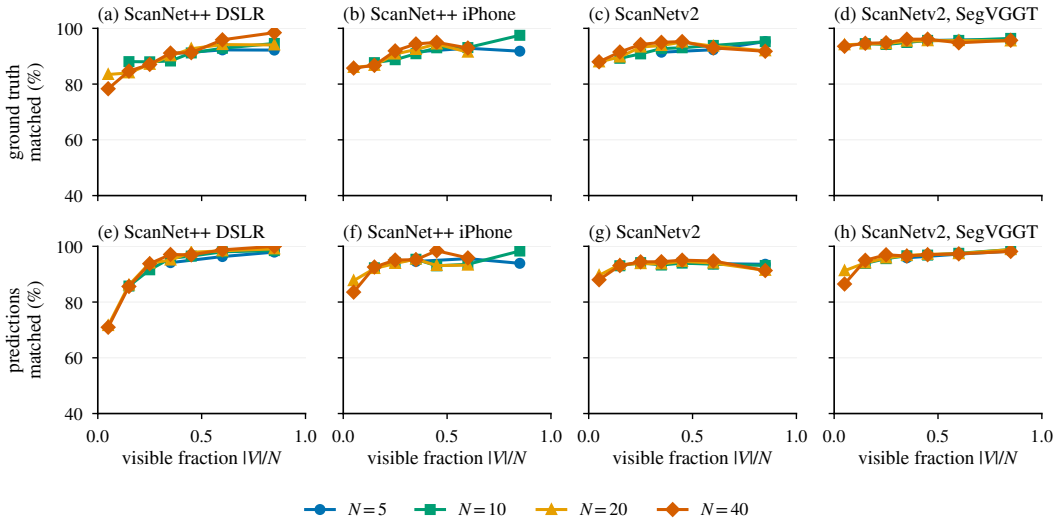


Figure A5: As Fig. A4, for $\gamma = 1$.

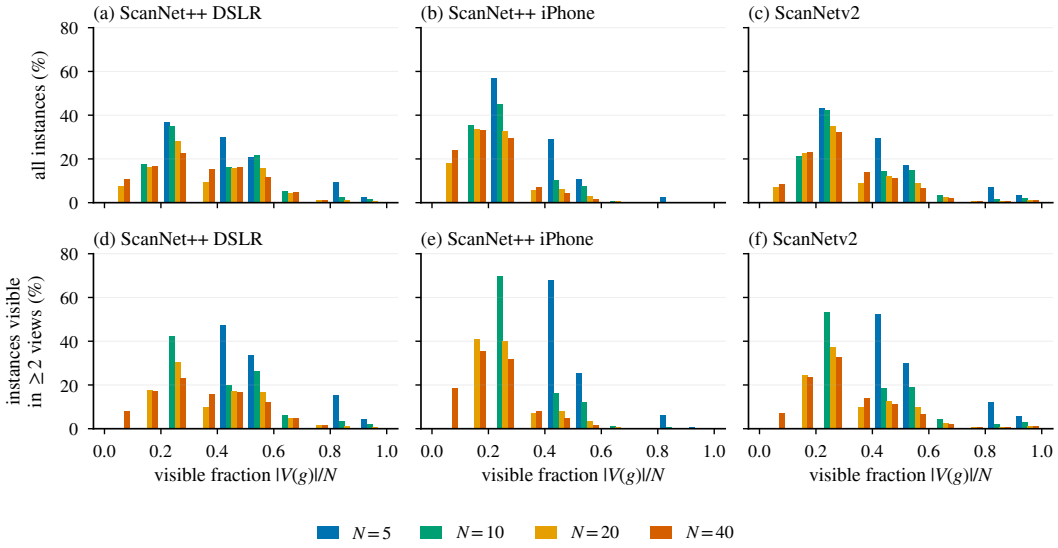


Figure A6: Distribution of the fraction $|V(g)|/N$ of the input set in which a ground-truth instance is visible, at each N, over thing classes of the three captures. Top: all scored instances, bottom: instances visible in at least two views.

Table A15: VC-PQ and its two factors VC-SQ and VC-RQ for input sets of N views, all classes, aggregated as in Table 1. The product VC-PQ = VC-SQ × VC-RQ holds for each class and not for the class averages shown here, and a class without a matched pair enters the average of VC-SQ with zero, as in PQ. –: as in Table 1.

		VC-PQ				VC-SQ				VC-RQ			
Dataset	Method	N=5	10	20	40	N=5	10	20	40	N=5	10	20	40
ScanNet++	PanSt3R v1	22.0	19.2	17.9	17.4	65.8	62.9	56.6	53.3	28.7	25.7	24.5	24.2
	PanSt3R v2	24.5	22.2	20.7	20.0	64.8	61.8	57.1	52.9	32.0	29.6	28.3	27.9
	EPS3D	2.4	2.0	–	–	15.2	14.4	–	–	3.3	2.9	–	–
	IGGT	5.7	4.2	–	–	36.4	32.4	–	–	7.4	5.7	–	–
ScanNetv2	PanSt3R v2	43.7	41.5	40.6	40.6	78.9	76.6	74.7	73.7	54.9	53.5	53.6	54.2
	EPS3D	9.2	8.2	–	–	47.2	45.8	–	–	12.2	11.2	–	–