LIGHTWEIGHT 3D LATENT SEMANTICS FROM BAYESIAN INVERSION OF 2D FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present Casper3D, a lightweight probabilistic framework for converting noisy multi-view 2D foundation-model embeddings into a latent 3D semantic representation. We model view-level semantic features as noisy observations of an underlying 3D semantic state and infer this state with a set-based variational model that incorporates relative pose during multi-view reasoning. Casper3D is trained by predicting held-out semantic observations from novel viewpoints, while remaining aligned with visual and text semantic spaces for open-vocabulary 3D understanding. The framework is backbone-agnostic and applies to both language-aligned and self-supervised embeddings. We show that Casper3D produces more stable 3D semantics than multi-view pooling, especially in ambiguous and noisy settings.

1 INTRODUCTION

A central goal of perception is to recover stable properties of the physical world from partial and view-dependent measurements. Images are such measurements: they do not directly reveal the underlying three-dimensional scene, but rather record its projection under a particular camera pose, illumination, occlusion pattern, and surrounding context. Modern pretrained vision and vision-language models, such as DINO Caron et al. (2021); Oquab et al. (2023); Siméoni et al. (2025), CLIP Radford et al. (2021); Zhou et al. (2022); Lüddecke & Ecker (2022), and masked autoencoders He et al. (2022), transform these image measurements into powerful semantic feature spaces. These models have become widely used as backbones for 2D tasks including classification, segmentation, detection, and depth estimation Amir et al. (2021); Hamilton et al. (2022); Gu et al. (2021). Yet their representations remain fundamentally attached to image coordinates.

In contrast, the physical world is organized in three dimensions. A point on a table, a chair, or a wall exists independently of the camera from which it is observed. Its semantic identity is not created by a particular view, even though each view may reveal a partial and noisy signature of it. This distinction is crucial for 3D scene understanding: we would like to attach feature embeddings not to pixels, but to physical points $\{p_i \in \mathbb{R}^3\}_{i=1}^N$. For each point p_i, the desired embedding should represent an intrinsic property of that point in the scene: a view-invariant semantic state that persists across observations.

This motivates the following physical interpretation: given a 3D point p_i, there exists an unobserved latent vector $z_i \in \mathbb{R}^d$ that encodes the semantic content of the physical point. A 2D foundation model does not observe z_i directly. Instead, when p_i is seen from a camera view j, the model produces a feature observation $x_{ij} \in \mathbb{R}^C$ that depends both on the latent state z_i and on the viewing condition T_j. The image feature is therefore a measurement, not the object itself. It is affected by projection, scale, viewpoint, occlusion, local image context, and the inductive biases of the pretrained model.

Hence, lifting 2D features into 3D is not merely a geometric alignment problem. It is a Bayesian inverse problem: recover the hidden, view-independent semantic state of a physical point from multiple noisy, view-dependent measurements. Formally, given the observations $\mathcal{C}_i = \{(x_{ij}, T_j)\}_{j=1}^{N_i}$, we seek to infer the latent state that explains them. The desired 3D representation is not one of the observed 2D features, nor an arbitrary average of them, but the latent cause underlying all of them.

Most existing approaches adopt a lift-and-aggregate strategy Zhang et al. (2022b); Peng et al. (2023); Song et al. (2025). Using camera intrinsics, poses, and depth, each 3D point is projected into the images where it is visible. Features from a pretrained 2D model are sampled at the corresponding image locations and fused by average pooling, weighted averaging, voting, or similar deterministic

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.

rules: $e_i = \sum_{j=1}^{N_i} w_{ij} x_{ij}$. This approach is simple and often effective, but it implicitly assumes that the 3D feature can be estimated as a weighted mean of its 2D measurements.

Instead, we propose Casper3D (Canonical Semantic Probabilistic Embeddings for 3D), a probabilistic framework that formulates the lifting of pretrained 2D foundation-model features into 3D as a Bayesian inverse problem. For each 3D point p_i, we assume a latent semantic variable z_i, and model each 2D feature observation as a noisy observation generated by a view-dependent measurement process: $x_{ij} \sim p_\theta(x \mid z_i, T_j)$. The main objective is then to recover the posterior distribution $p(z_i \mid \mathcal{C}_i)$, which represents what can be inferred about the intrinsic 3D semantic state from all available views. This posterior formulation naturally distinguishes between the physical representation of the point and the observations induced by particular cameras. It also provides a principled way to represent uncertainty: points seen from many consistent views should yield concentrated posteriors, whereas points seen from few or conflicting views should remain uncertain.

Since the exact posterior is intractable, Casper3D introduces a lightweight view-conditioned variational model. The encoder receives an unordered set of 2D feature measurements and viewing conditions, $\mathcal{C}_i = \{(x_{ij}, T_j)\}_{j=1}^{N_i}$, and produces an amortized posterior approximation $q_\phi(z_i \mid \mathcal{C}_i)$. The decoder is conditioned on both the inferred latent state and a target viewing condition, and is trained to predict held-out 2D feature observations. In this way, the model is forced to separate the intrinsic component of the point representation, encoded in z_i, from the extrinsic component induced by viewpoint, encoded through T_j. We use held-out feature prediction as a self-supervised objective for learning a view-stable latent 3D semantic representation. Once inferred, the resulting 3D feature field can be used for multiple downstream tasks. We focus on open-vocabulary 3D semantic segmentation as a primary application. To summarize, our contributions are as follows: **(i)** We formulate 2D-to-3D foundation-model feature lifting as a Bayesian inverse problem, where each 3D point has an unobserved view-invariant semantic state, and each 2D feature is a noisy, view-dependent observation of that state. **(ii)** We introduce **Casper3D**, a probabilistic latent-variable framework for learning 3D feature fields from multi-view 2D pretrained features. It uses a lightweight view-conditioned variational model to infer posterior distributions over latent 3D point features from unordered, variable-size sets of 2D observations, and is trained through held-out view prediction to separate view-stable semantic content from view-dependent measurement effects. **(iii)** We demonstrate Casper3D on open-vocabulary 3D semantic segmentation, showing improved robustness and accuracy over aggregation-based feature lifting methods.

2 Related Work

Multi-view 2D feature lifting. A common approach to transfer 2D semantics to 3D is to associate image features with geometry and fuse them across views. SemanticFusion McCormac et al. (2017) uses SLAM Whelan et al. (2016), while recent open-vocabulary methods rely on pretrained 2D models. ConceptFusion Jatavallabhula et al. (2023) lifts SAM Kirillov et al. (2023) and global image features; FeatureRealisticFusion Mazur et al. (2023) uses an iMAP-like neural field Sucar et al. (2021); and OpenMask3D Takmaz et al. (2023), OpenIns3D Huang et al. (2024), and OV3D Zhou et al. (2025) lift SAM-based masks to 3D instances. PointCLIP Zhang et al. (2022b) and MV-CLIP Song et al. (2025) aggregate CLIP predictions across views. For point-level understanding, OpenScene Peng et al. (2023), CLIP-FO3D Zhang et al. (2023), Semantic Abstraction Ha & Song (2022), text-supervised methods Jiang et al. (2024); Zhang et al. (2025b), RegionPLC Yang et al. (2024), PLA/CLIP2Scene Ding et al. (2023); Chen et al. (2023), and DITR Knaebel et al. (2025) transfer 2D foundation-model knowledge to 3D. These methods typically *select, average, distill, or fuse* the observed 2D features. Casper3D takes a different view: the 2D features are noisy, pose-dependent measurements, and the 3D representation is an *unobserved latent semantic state* that must be inferred.

Bayesian inverse problems. Bayesian inverse problems recover hidden quantities from noisy observations rather than directly combining measurements. For example, dark-matter mapping infers an unobserved field from observational tracers Royo et al. (2026), while recent inverse methods use diffusion models Ho et al. (2020) and posterior samplers Zhang et al. (2025a); Wang & Bouman (2026). Casper3D brings this perspective to 2D-to-3D semantic lifting. Unlike classical inverse problems, our latent variable is an abstract foundation-model semantic state with no ground-truth

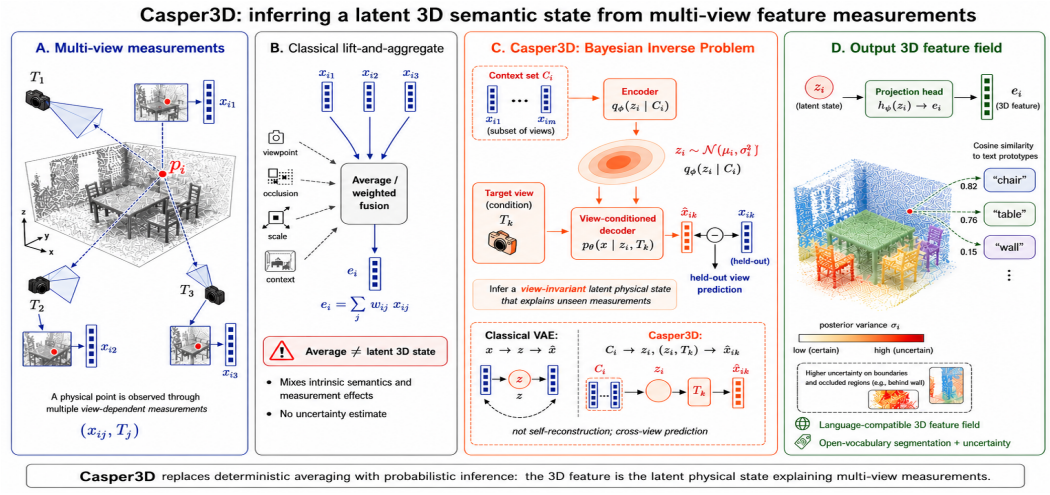


Figure 1: **Overview of Casper3D.**

3D target. We therefore learn a lightweight amortized posterior $q_\phi(z_i \mid \mathcal{C}_i)$ directly from multi-view observations.

Set-based variational models and held-out prediction. Our input is an unordered set with a variable number of views. Deep Sets Zaheer et al. (2017), Set Transformers Lee et al. (2019), and SetVAE Kim et al. (2021) provide permutation-invariant and variational tools, while ViTs typically rely on order-dependent positional encodings Dosovitskiy et al. (2020). In Casper3D, camera pose is not only positional information but part of the measurement process, motivating pose-aware set inference and view-conditioned decoding. Our held-out feature prediction is also related to novel-view synthesis Liu et al. (2023), but we predict foundation-model features rather than RGB or geometry, using unseen views to identify a stable latent 3D semantic state.

3D representation learning. Local Bai et al. (2020); Huang et al. (2021); Qin et al. (2023) and global Xie et al. (2020); Yu et al. (2022); Zhang et al. (2022a) point-cloud descriptors learn viewpoint-robust 3D representations (see Han et al. (2018)). Their goal, however, is different from ours. Casper3D does not learn an arbitrary 3D descriptor: it infers the view-independent counterpart of a *given* 2D foundation-model representation while preserving its semantic geometry. For example, CLIP-based Casper3D features remain compatible with CLIP's text space. Thus, instead of replacing the original representation with a new 3D embedding space, Casper3D estimates the latent 3D semantic state that best explains multiple observations within the original foundation-model space.

2D augmentation vs. 3D viewpoint invariance. Robustness to crops, resizing, or photometric augmentations does not imply invariance to changes in 3D viewpoint, which introduce occlusion, foreshortening, illumination changes, and different context. Prior studies El Banani et al. (2024); Ruan et al. (2024); Michalkiewicz et al. (2025) also report limited viewpoint robustness in 2D foundation models. Casper3D therefore explicitly models each feature as a pose-dependent observation and uses relative-pose reasoning and held-out-view prediction to recover the semantic information that remains stable across viewpoints.

3 METHOD

Casper3D (**Canonical Semantic Probabilistic Embeddings for 3D**) is a probabilistic method for lifting pretrained 2D foundation-model features into 3D. Its central idea is that a physical 3D point has an underlying semantic state that is never observed directly, but only through different view-dependent 2D measurements. This creates a fundamental difficulty: there is no ground-truth latent 3D feature to supervise. We do not know what the "correct" internal 3D representation of CLIP, DINO, or any other foundation model should be. Learning a 3D feature field therefore cannot be posed as standard supervised regression. Instead, we formulate it as a Bayesian inverse problem: recover a distribution over latent 3D semantic states that are compatible with the available multi-view measurements. This is the main departure from deterministic feature lifting. Existing approaches

often collapse the observed 2D features into a single embedding through averaging or weighted fusion. Casper3D instead treats these features as noisy, view-dependent measurements and asks: *which latent 3D semantic state could have generated them?*

3.1 A Latent Physical State, Not an Averaged Feature

As motivated above, Casper3D introduces a latent variable z_i for each 3D point $p_i \in \mathbb{R}^3$. z_i is not a supervised target, but an inferred physical semantic state: the hidden representation that explains how the point is observed across views. Since this state is unobserved and may be ambiguous, we infer a posterior distribution rather than a single deterministic embedding:

$$q_\phi(z_i \mid \mathcal{C}_i) = \mathcal{N}\left(z_i; \mu_i, \mathrm{diag}(\sigma_i^2)\right), \tag{1}$$

where ϕ denotes the complete set of learnable parameters used to infer the posterior from the context observations. We detail these components and their parameterization next.

The posterior mean $\mu_i \in \mathbb{R}^d$ represents the inferred latent state given the available observations, while $\sigma_i \in \mathbb{R}^d$ represents uncertainty in this estimate. When observations are sparse, noisy, or contradictory, the posterior may remain broader, reflecting ambiguity in the unobserved latent state.

3.2 Learning Without a Ground-Truth 3D Latent Space

A key challenge is that z_i cannot be supervised directly: there is no ground-truth view-independent CLIP-like or DINO-like representation associated with a physical 3D point. We therefore identify the latent state indirectly through cross-view prediction. During training, the observations associated with point i are divided into a context set $\mathcal{C}_i$ and a held-out set $\mathcal{H}_i$. The context observations are used to infer z_i, while the held-out observations provide a self-supervised signal for testing whether this state remains predictive from unseen viewpoints.

Standard Set Transformer Lee et al. (2019); Kim et al. (2021) components provide the permutation-aware machinery for unordered observations through the Multihead Attention Block (MAB; Appendix A, Figure 3a) and Pooling by Multihead Attention (PMA; Figure 3b). Our contribution is the probabilistic 3D inference architecture built around them: Casper3D introduces relative-pose representations of the measurement geometry, pose-aware reasoning across views, and held-out view-conditioned prediction to identify an unobserved 3D semantic state.

Relative-pose encoding. Because the observed feature of a physical point depends on the camera from which it is measured, camera geometry must be explicitly modeled during latent learning. We therefore represent each observation relative to the other context views.

Let $x_{ij} \in \mathbb{R}^C$ denote the feature of physical point i observed from view j, with camera pose $T_j \in SE(3)$. Rather than using absolute poses, which depend on an arbitrary scene coordinate frame, we use pairwise relative transforms $T_{k\to l} = T_l^{-1} T_k$. For N context observations, we construct

$$T_{\mathcal{C}\to\mathcal{C}} = \{T_{k\to l}\}_{k,l} \in SE(3)^{N\times N}. \tag{2}$$

Each transform is embedded by a shared pose encoder g_η, and the embeddings associated with observation k are aggregated using the standard PMA operator (Appendix A, Figure 3b):

$$p_k = \mathrm{PMA}\left(\{g_\eta(T_{k\to l})\}_{l=1}^N\right). \tag{3}$$

The resulting pose token p_k summarizes the position of view k relative to the other context views and is combined with its semantic feature: $h_k = \mathrm{MLP}\left([x_{ik}, p_k]\right)$.

Pose-aware attention. The relative-pose embeddings are then passed to our *Pose-Aware Attention Block* (PAB; Appendix A, Figure 3c). Learning a common latent state requires reasoning not only about semantic similarity, but also about the geometry relating different measurements. PAB uses the standard MAB (Figure 3a) as its content-attention backbone and incorporates relative camera geometry directly into the interaction between observations. For an attention head of dimension d_h, the attention logit between observations k and ℓ becomes

$$a_{kl} = \frac{Q(h_k)^\top K(h_l)}{\sqrt{d_h}} + b(g_\eta(T_{k\to l})). \tag{4}$$

Thus, relative geometry enters the network twice: first through the observation token h_k, and again through the pairwise attention bias. Unlike a standard MAB, PAB can therefore determine how observations interact using both semantic similarity and their relative viewing geometry.

Pose-aware variational encoder. The relative-pose representation and PABs together form the encoder used to estimate the posterior over the latent semantic state. We stack L PABs and then use the standard PMA operator to pool the resulting unordered set into a fixed-size representation, with one output token per learned PMA seed (Appendix A, Figure 3d):

$$H^{(\ell+1)} = \mathrm{PAB}^{(\ell)}\left(H^{(\ell)}, T_{\mathcal{C}\to\mathcal{C}}\right), \qquad s_i = \mathrm{PMA}\left(H^{(L)}\right). \tag{5}$$

Two encoder heads parameterize the posterior:

$$q_\phi(z_i \mid \mathcal{C}_i) = \mathcal{N}\left(z_i; \mu_\rho(s_i), \mathrm{diag}\left(\sigma_\rho^2(s_i)\right)\right). \tag{6}$$

The posterior therefore represents both the estimated view-independent semantic state and its uncertainty given potentially incomplete or inconsistent multi-view observations.

Pose-conditioned held-out prediction. The latent state itself is never observed, so the posterior cannot be trained against a ground-truth 3D foundation-model feature. We instead identify it through posterior prediction: the latent state inferred from the context views must explain measurements of the same physical point from held-out viewpoints that were not used to infer it.

For a target view $k \in \mathcal{H}_i$, its pose is encoded relative to the context views using the same relative-pose encoder: $r_k = \mathrm{PMA}\left(\{g_\eta(T_{k\to j})\}_{j\in\mathcal{C}_i}\right)$. Given $z_i \sim q_\phi(z_i \mid \mathcal{C}_i)$, the pose-conditioned decoder (Appendix A, Figure 3e) predicts the foundation-model feature expected from that unseen viewpoint: $\hat{x}_{ik} = D_\theta(z_i, r_k)$.

The decoder therefore acts as a learned measurement model: given the latent physical state and a target viewing condition, it predicts the semantic measurement expected from that viewpoint. Training follows directly from this held-out predictive formulation. Since the latent state is unobserved, we optimize the negative evidence lower bound (ELBO) Kingma & Welling (2013) over the held-out observations:

$$\mathcal{L}_{\mathrm{ELBO}}(i) = \mathcal{L}_{\mathrm{heldout}}(i) + \beta\,\mathrm{KL}\left[q_\phi(z_i \mid \mathcal{C}_i) \,\|\, p(z_i)\right], \tag{7}$$

where the first term encourages samples from the posterior inferred from the context views to explain held-out measurements. Here $p(z_i) = \mathcal{N}(0, I)$ is the standard Gaussian prior. In practice, we instantiate this predictive term with cosine distance in the foundation-model feature space:

$$\mathcal{L}_{\mathrm{heldout}}(i) = \frac{1}{|\mathcal{H}_i|}\sum_{k\in\mathcal{H}_i}\left[1 - \cos\left(\hat{x}_{ik}, x_{ik}\right)\right]. \tag{8}$$

Because x_{ik} is never provided to the encoder, the network cannot solve the task by copying an observed feature. It must instead infer a latent state that explains measurements across viewpoints. Together, relative-pose encoding, pose-aware attention, variational posterior estimation, and view-conditioned held-out prediction instantiate the Bayesian inverse formulation of Casper3D: recovering an unobserved 3D semantic state from multiple noisy, view-dependent measurements without requiring a ground-truth 3D feature.

Permutation structure. The context observations form an unordered set. Reordering the input views therefore permutes both the semantic observation tokens and the associated pairwise relative-pose tensor consistently, while the inferred physical state should remain unchanged. Let $\mathfrak{S}_n$ denote the symmetric group of all permutations of n elements, let $\mathcal{X}$ denote the semantic observation-token space, and let $P \in \mathbb{R}^{n\times n\times d_p}$ denote the pairwise relative-pose representation. For any $\pi \in \mathfrak{S}_n$, define

$$P' = \pi P \pi^\top,$$

where the operation denotes the corresponding permutation of the two view-index axes of P. We formalize the resulting permutation properties below; full proofs are provided in Appendix B.

Proposition 1 (Permutation Equivariance of the PAB Block)**.** Let $\mathrm{PAB}(X, P)$ denote the proposed Pose-Aware Attention Block. Then, for any $\pi \in \mathfrak{S}_n$ and any $X \in \mathcal{X}^n$,

$$\mathrm{PAB}(\pi \cdot X, P') = \pi \cdot \mathrm{PAB}(X, P).$$

Proposition 2 (Permutation Invariance of the Encoder)**.** Let $\mathrm{Enc}(X, P)$ denote the proposed pose-aware encoder. Then, for any $\pi \in \mathfrak{S}_n$ and any $X \in \mathcal{X}^n$,

$$\mathrm{Enc}(\pi \cdot X, P') = \mathrm{Enc}(X, P).$$

3.3 Preserving the Foundation-Model Semantic Space

The latent semantic state is defined to have the same dimensionality as the observed foundation-model features; that is, we set $d = C$, so that $z_i \in \mathbb{R}^C$. Its semantic structure is learned primarily through the variational held-out prediction objective, which encourages z_i to capture the view-independent information needed to explain unseen foundation-model observations. We additionally use lightweight auxiliary alignment objectives to preserve compatibility with the original foundation-model semantic space.

Language-Compatible 3D Features. For language-aligned foundation models such as CLIP, we additionally encourage the inferred 3D representation to remain compatible with the corresponding text embedding space. To do so, we construct a pseudo-label directly from the frozen 2D foundation-model observations, without using ground-truth semantic labels.

For each physical 3D location i, let $\{x_{ij}\}_{j=1}^{N_i}$ denote its features observed from N_i views, and let s_c be the normalized text embedding of class c, obtained from its class name using the LAION prompt templates. Let $\mathcal{Y}$ denote the set of semantic classes. Each view independently predicts its most compatible class: $\hat{c}_{ij} = \arg\max_{c \in \mathcal{Y}} \cos(x_{ij}, s_c)$.

The pseudo-label for the corresponding 3D location is then obtained by majority voting across its available views: $c_i = \arg\max_{c \in \mathcal{Y}} \sum_{j=1}^{N_i} \mathbf{1}[\hat{c}_{ij} = c]$.

Given this voted class-text target, we define the temperature-scaled probability

$$p_\psi(c_i \mid z_i) = \frac{\exp(\cos(z_i, s_{c_i})/\tau)}{\sum_{c' \in \mathcal{Y}} \exp(\cos(z_i, s_{c'})/\tau)}, \tag{9}$$

and optimize the auxiliary text-alignment loss

$$\mathcal{L}_{\mathrm{text}}(i) = -\log p_\psi(c_i \mid z_i). \tag{10}$$

This auxiliary objective preserves the language alignment of vision-language backbones, ensuring that the resulting Casper3D features remain directly queryable with text. It is not required by the core formulation: when the underlying 2D foundation model is not language-aligned, such as DINO, we omit this term by setting $\lambda_{\mathrm{text}} = 0$.

Weak feature-space anchoring. We also use the mean lifted feature as a weak anchor:

$$a_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{ij}. \tag{11}$$

The alignment loss is $\mathcal{L}_{\mathrm{avg}}(i) = 1 - \cos(z_i, a_i)$.

This term should not be interpreted as assuming that the average feature is the correct 3D representation. Rather, it keeps the inferred feature from drifting too far from the original pretrained feature space. The average feature serves only as a weak anchor, while the latent state is learned through posterior-predictive consistency.

3.4 Full Objective

The full training loss is

$$\mathcal{L}(i) = \mathcal{L}_{\mathrm{heldout}}(i) + \beta\, \mathrm{KL}\left[q_\phi(z_i \mid \mathcal{C}_i) \,\|\, p(z_i)\right] + \lambda_{\mathrm{text}}\, \mathcal{L}_{\mathrm{text}}(i) + \lambda_{\mathrm{avg}}\, \mathcal{L}_{\mathrm{avg}}(i). \tag{12}$$

The role of each term is distinct. The view-prediction term learns the latent state by asking it to explain unseen measurements. The KL term regularizes the unobserved latent space. The text term

preserves semantic compatibility with language, while the average-alignment term softly anchors the representation to the original feature space.

Together, these terms allow Casper3D to learn a 3D feature field without ever observing ground-truth 3D features.

3.5 Inference

At test time, the available observations selected for a physical 3D point are used as context; in the default evaluation setting, we use up to 20 context views per point. The encoder estimates the posterior parameters: $(\mu_i, \sigma_i) = \mathrm{Encoder}_\phi(\mathcal{C}_i)$. The posterior mean μ_i is used as the inferred view-independent 3D representation.

For open-vocabulary segmentation, each point is then assigned to the closest text prototype: $\hat{c}_i = \arg\max_c \cos(\mu_i, s_c)$. The posterior parameter σ_i provides an uncertainty estimate over the inferred physical state.

4 Experiments

We evaluate Casper3D on *annotation-free open-vocabulary 3D semantic segmentation*. Our goal is to test whether probabilistic multi-view inference produces better 3D semantic features than direct 2D-to-3D feature aggregation. For each 3D point, we collect its 2D features and relative camera poses from multiple views, infer a latent 3D embedding, and classify it by cosine similarity to text prototypes.

4.1 Experimental Setup

Datasets. We evaluate on three standard indoor benchmarks: **ScanNet** Dai et al. (2017), **ScanNet200**, and **Matterport3D** Chang et al. (2017). ScanNet contains 1 201 training and 312 validation RGB-D scenes with camera geometry for multi-view lifting; we map the ScanNet class *other furniture* to *other*. ScanNet200 extends ScanNet to 200 classes, while Matterport3D provides larger and more complex indoor scenes.

Task and metrics. For each 3D point, we infer a semantic embedding from all valid multi-view observations and classify it using text similarity. We report **mIoU** and **mAcc**. Unless stated otherwise, all methods use the same text prompts.

Implementation details. We keep only geometrically valid reprojections using $|\hat{d} - d(u,v)| < 0.05\, d(u,v)$. We train on one NVIDIA RTX A6000 with batch size 512, learning rate 5×10^{-5}, cosine decay, and 10 000 warm-up steps. We use $\beta = 10^{-4}$, $\lambda_{\text{text}} = 0.1$, $\lambda_{\text{avg}} = 0.05$, and KL annealing over 100 000 steps. Each point uses up to 20 context and 20 held-out views. The text temperature is 0.05, and TIPSv2 Cao et al. (2026) is our default 2D teacher. Further details are given in Appendix C.

4.2 Main Results: Annotation-Free Open-Vocabulary 3D Segmentation

Comparison with prior methods. Table 1 (top) compares Casper3D with zero-shot 2D fusion approaches. Casper3D consistently improves over prior annotation-free methods, showing that directly lifting or averaging strong 2D features is not enough. Instead, our results support inferring a latent 3D semantic state from noisy multi-view observations.

Optional 3D distillation. For image-free inference, we optionally distill the frozen Casper3D feature cloud into a MinkowskiNet18A Choy et al. (2019) without semantic labels. This *3D distillation* stage is separate from Casper3D and transfers its features to a point-cloud-only model; details are given in Appendix D. Table 1 (bottom) compares this approach with other methods using a 3D backbone.

Strong features without a 3D backbone. The results show that Casper3D already performs strongly without 3D distillation, showing that the main gain comes from probabilistic multi-view inference. Adding 3D distillation further improves performance on all datasets, showing that Casper3D is also complementary to 3D backbone training. Additional qualitative results are provided in Appendix E.2.

Method	Test-Time Images	ScanNet		ScanNet200		Matterport3D	
		mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
2D fusion zero-shot							
MaskCLIP-3D Zhou et al. (2022)	✓	9.7	21.6	–	–	–	–
MSeg Voting Lambert et al. (2020)	✓	45.6	54.4	–	–	33.4	39.0
OpenScene-2D Peng et al. (2023)	✓	50.0	62.7	–	–	32.3	40.0
CLIP-FO3D (proj.) Zhang et al. (2023)	✓	27.6	47.7	–	–	–	–
2D fusion w/ training							
Ours	✓	**64.6**	**75.5**	**11.0**	**18.1**	**50.4**	**65.1**
3D point-cloud training							
OpenScene + 3D Distill. Peng et al. (2023)	✗	52.9	63.2	7.3	–	41.9	51.2
PLA Ding et al. (2023)	✗	–	–	1.8	–	–	–
RegionPLC Yang et al. (2024)	✗	–	–	6.5	–	–	–
OV3D Jiang et al. (2024)	✗	57.3	72.9	8.7	–	45.8	62.4
CLIP-FO3D Zhang et al. (2023)	✗	30.2	49.1	–	–	–	–
PGOV3D Zhang et al. (2025b)	✗	59.5	73.2	9.3	17.1	–	–
Ours + 3D Distill.	✗	**65.8**	**76.7**	**14.2**	**22.6**	**53.2**	**66.6**
Fully supervised upper bound							
Fully supervised	✗	72.0	80.7	23.9	32.9	55.7	67.4

Table 1: Annotation-free open-vocabulary 3D semantic segmentation on ScanNet, ScanNet200, and Matterport3D. Casper3D improves over prior zero-shot and trained 2D-fusion methods and remains competitive with methods that additionally optimize a 3D backbone.

Component	Variant	ScanNet mIoU ↑
Text alignment		
Without text alignment	No text loss	61.1
With text alignment	**Default**	**64.6**
Feature anchoring		
No feature anchor	None	59.9
Majority-vote anchor	Majority-view	63.2
Lowest-entropy anchor	Lowest-entropy view	58.5
All-view anchor	**Default**	**64.6**
Prediction objective		
Input reconstruction	Context reconstruction	53.8
Held-out prediction	**Default**	**64.6**
Held-out + reconstruction	Joint objective	63.5
Latent formulation		
Deterministic latent	Autoencoder	58.7
Variational without KL	No KL	Unstable
Variational + KL	**Default**	**64.6**
Pose conditioning		
No pose	No pose / no held-out	48.2
Basic pose encoding	Relative-pose tokens	57.0
Pose-aware attention	**Default**	**64.6**
Aggregation		
Feature averaging	Deterministic average	50.0
Casper3D	**Full model**	**64.6**

Table 2: **Component ablations on ScanNet.** Ablations examine the contribution of the main architectural and training choices under the same evaluation protocol.

4.3 Ablations and Analysis

Component ablations. Table 2 isolates the contribution of the main design choices in Casper3D. Held-out prediction substantially outperforms input reconstruction, the variational formulation improves over a deterministic latent, and pose-aware attention improves over basic pose encoding. The complete model also substantially outperforms direct feature averaging. The default configuration and hyperparameter sensitivity analysis are reported in Appendix C.1.

Effect of the 2D foundation model. Table 3 evaluates Casper3D with different 2D foundation-model features. Performance improves as the underlying 2D representation becomes stronger, from 60.8 mIoU with MaskCLIP to 64.6 with TIPSv2. This shows that Casper3D can benefit from improved 2D semantic features without changing the 3D inference architecture.

Latent structure versus deterministic averaging. In Appendix E.2, Figure 6 compares the semantic geometry induced by simple average aggregation and by the latent representation inferred by Casper3D. Across multiple representative scenes, latent embeddings exhibit tighter class-local neighborhoods and markedly reduced inter-class overlap, while average aggregation leads to fragmented clusters and stronger semantic mixing.

Robustness to sparse views. Figure 2 evaluates Casper3D as the number of available input views decreases. For this analysis, the averaging baseline corresponds to our re-implementation of OpenScene-

2D foundation model			Efficiency comparison		
2D Feature	**mIoU**	**Δ**	**Method**	**Training**	**Inference**
MaskCLIP Zhou et al. (2022)	60.8	–	OpenScene-3D Peng et al. (2023)	8×A100 - 24.7 h	RTX 2080Ti - 80 ms
CLIP-DINOiser Wysoczańska et al. (2024)	62.3	+1.5	PLA Ding et al. (2023)	8×A100 - 11.5 h	RTX 2080Ti - 80 ms
OpenSeg Ghiasi et al. (2022)	63.4	+2.6	RegionPLC Yang et al. (2024)	8×A100 - 12.5 h	RTX 2080Ti - 80 ms
TIPSv2 Cao et al. (2026)	**64.6**	+3.8	**Casper3D**	1×RTX A6000 - 41.6 h	RTX A6000 - **14.4 ms**

Table 3: **Effect of the 2D foundation model and computational efficiency.** Left: ScanNet mIoU with different 2D foundation-model features. Right: training and inference cost for representative open-vocabulary 3D methods.

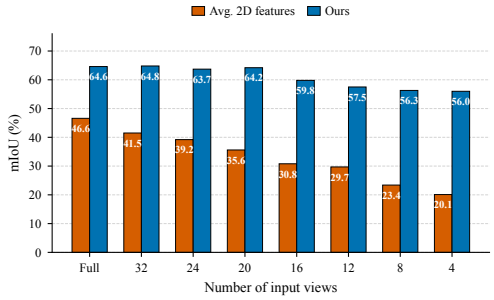


Figure 2: **Robustness to sparse views.** ScanNet mIoU of Casper3D and our re-implementation of OpenScene-style 2D feature averaging as the number of input views decreases.

style deterministic feature aggregation, which obtains 46.6 mIoU when using all available views. Casper3D remains substantially more stable as view coverage becomes sparse, retaining 56.0 mIoU with only four views. In contrast, directly averaging 2D features degrades rapidly as view coverage becomes sparse, highlighting the benefit of our inferred 3D feature field.

Efficiency. Table 3 compares the computational requirements of Casper3D with representative open-vocabulary 3D methods. Casper3D trains on a single RTX A6000 and achieves an inference latency of 14.4 ms, compared with the reported 80 ms of the considered baselines. A detailed computational profile of Casper3D, including FLOPs, parameter count, peak GPU memory, and the exact inference measurement protocol, is provided in Appendix C.2.

Outdoor transfer. Our main benchmarks are indoor RGB-D datasets; however, Casper3D does not assume an indoor environment. As a preliminary cross-domain study, we also pretrain Casper3D on outdoor KITTI Geiger et al. (2013) sequences and observe improved transfer to ScanNet200. Details are provided in Appendix E.1.

5 Conclusion

We introduced Casper3D, a probabilistic framework that formulates multi-view 2D-to-3D feature lifting as a Bayesian inverse problem. Rather than averaging observed 2D embeddings, Casper3D treats them as noisy, view-dependent measurements of an unobserved, view-independent semantic state. A pose-aware variational encoder infers this state, while a view-conditioned decoder learns the measurement process through held-out prediction, providing a principled alternative to deterministic feature fusion.

Across three open-vocabulary 3D segmentation benchmarks, Casper3D improves over deterministic feature aggregation and remains substantially more robust under sparse view coverage. The inferred representation remains compatible with the original foundation-model semantic space, preserving language compatibility when available and enabling optional label-free distillation into a 3D backbone. More broadly, Casper3D suggests viewing cross-view feature lifting as latent state inference rather than direct observation fusion. This perspective may extend to other inverse problems in which an unobserved state must be recovered from heterogeneous, view-dependent, partial, or noisy measurements.

AI USE STATEMENT

We used generative AI tools solely to assist with the preparation of this manuscript, including refining sentence-level phrasing and style, formatting tables, and checking notation and expository consistency. All research ideas, methodological choices, experimental design, implementation, results, and scientific interpretations were developed by the authors. AI-assisted suggestions were manually reviewed, and all technical statements, equations, and reported results were checked by the authors. We take full responsibility for the final content of this work, including any produced with AI assistance.

ETHICS STATEMENT

This work does not involve the collection of new human-subject data or personally identifiable information. Our experiments use established 3D benchmark datasets and pretrained foundation models according to their respective licenses and terms of use. As with other methods that rely on pretrained vision and vision-language models, Casper3D may inherit biases or failure modes present in the underlying 2D representations. The method is intended for research on 3D semantic representation learning, and we do not study safety-critical deployment settings in this work.

REPRODUCIBILITY STATEMENT

We provide the source code and configuration files used for the experiments in the supplementary material. The main paper describes the model, training objective, evaluation protocol, and datasets, while Appendix C provides the architectural details, optimization settings, preprocessing procedure, and hyperparameters used in the reported experiments. Controlled ablations and hyperparameter sensitivity are also reported in the appendix. Formal permutation properties of the set-based architecture are stated in the main paper and proved in Appendix B. Together with the released code, these details are intended to enable reproduction of the reported results.

REFERENCES

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.

Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6358–6366. IEEE, 2020.

Bingyi Cao, Koert Chen, Kevis-Kokitsi Maninis, Kaifeng Chen, Arjun Karpur, Ye Xia, Sahil Dua, Tanmaya Dabral, Guangxing Han, Bohyung Han, et al. Tipsv2: Advancing vision-language pretraining with enhanced patch-text alignment. *arXiv preprint arXiv:2604.12012*, 2026.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7020–7030, 2023.

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 3070–3079. IEEE, 2019.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Pla: Language-driven open-vocabulary 3d scene understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7010–7019, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21795–21806. IEEE, 2024.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European conference on computer vision*, pp. 540–557. Springer, 2022.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

Huy Ha and Shuran Song. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. *arXiv preprint arXiv:2207.11514*, 2022.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv preprint arXiv:2203.08414*, 2022.

Xian-Feng Han, Shi-Jie Sun, Xiang-Yu Song, and Guo-Qiang Xiao. 3d point cloud descriptors in hand-crafted and deep learning age: State-of-the-art. *arXiv preprint arXiv:1802.02297*, 2018.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 4265–4274. IEEE, 2021.

Zhening Huang, Xiaoyang Wu, Xi Chen, Hengshuang Zhao, Lei Zhu, and Joan Lasenby. Openins3d: Snap and lookup for 3d open-vocabulary instance segmentation. In *European Conference on Computer Vision*, pp. 169–185. Springer, 2024.

Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.

Li Jiang, Shaoshuai Shi, and Bernt Schiele. Open-vocabulary 3d semantic segmentation with foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21284–21294, 2024.

Jinwoo Kim, Jaehoon Yoo, Juho Lee, and Seunghoon Hong. Setvae: Learning hierarchical composition for generative modeling of set-structured data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15059–15068, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Karim Knaebel, Kadir Yilmaz, Daan de Geus, Alexander Hermans, David Adrian, Timm Linder, and Bastian Leibe. Dino in the room: Leveraging 2d foundation models for 3d segmentation. *arXiv preprint arXiv:2503.18944*, 2025.

John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2879–2888, 2020.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9298–9309, 2023.

Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7086–7096, 2022.

Kirill Mazur, Edgar Sucar, and Andrew J Davison. Feature-realistic neural fusion for real-time, open set scene understanding. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8201–8207. IEEE, 2023.

John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pp. 4628–4635. IEEE, 2017.

Mateusz Michalkiewicz, Sheena Bai, Mahsa Baktashmotlagh, Varun Jampani, and Guha Balakrishnan. Not all views are created equal: Analyzing viewpoint instabilities in vision foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9113–9123, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 815–824, 2023.

Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, Slobodan Ilic, Dewen Hu, and Kai Xu. Geotransformer: Fast and robust point cloud registration with geometric transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9806–9821, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Diego Royo, Brandon Zhao, Adolfo Muñoz, Diego Gutierrez, and Katherine L Bouman. Mapping dark-matter clusters via physics-guided diffusion models. *arXiv preprint arXiv:2603.14503*, 2026.

Shouwei Ruan, Yinpeng Dong, Hanqing Liu, Yao Huang, Hang Su, and Xingxing Wei. Omniview-tuning: Boosting viewpoint invariance of vision-language pre-training models. In *European Conference on Computer Vision*, pp. 309–327. Springer, 2024.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Dan Song, Xinwei Fu, Ning Liu, Wei-Zhi Nie, Wen-Hui Li, Lan-Jun Wang, You Yang, and An-An Liu. Mv-clip: Multi-view clip for zero-shot 3d shape recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(9):8767–8779, 2025.

Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6229–6238, 2021.

Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2306.13631*, 2023.

Frederic Wang and Katherine L Bouman. Sample-efficient evidence estimation of score based priors for model selection. *arXiv preprint arXiv:2602.20549*, 2026.

Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.

Monika Wysoczańska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzciński, and Patrick Pérez. Clip-dinoiser: Teaching clip a few dino tricks for open-vocabulary semantic segmentation. In *European Conference on Computer Vision*, pp. 320–337. Springer, 2024.

Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European conference on computer vision*, pp. 574–591. Springer, 2020.

Jihan Yang, Runyu Ding, Weipeng Deng, Zhe Wang, and Xiaojuan Qi. Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19823–19832, 2024.

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19291–19300. IEEE, 2022.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20895–20905, 2025a.

Junbo Zhang, Runpei Dong, and Kaisheng Ma. Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2048–2059, 2023.

Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems*, 35:27061–27074, 2022a.

Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8552–8562, 2022b.

Shiqi Zhang, Sha Zhang, Jiajun Deng, Yedong Shen, Mingxiao Ma, and Yanyong Zhang. Pgov3d: Open-vocabulary 3d semantic segmentation with partial-to-global curriculum. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 5070–5079, 2025b.

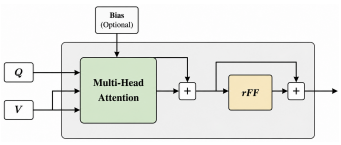
Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European conference on computer vision*, pp. 696–712. Springer, 2022.

Mingquan Zhou, Chen He, Ruiping Wang, and Xilin Chen. Ov3d-cg: Open-vocabulary 3d instance segmentation with contextual guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5305–5314, 2025.

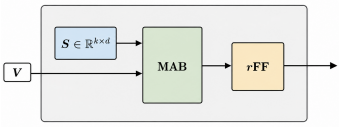
Supplementary Material

A Detailed Architecture

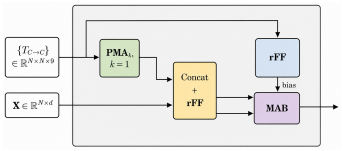
Figure 3 provides a block-level view of the architecture introduced in Sec. 3.2. The model follows four stages: relative-pose encoding, pose-aware attention, variational inference, and pose-conditioned held-out prediction.



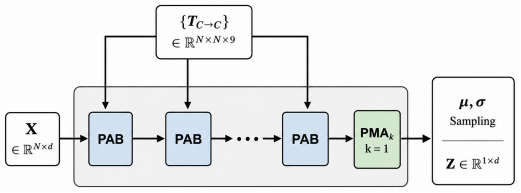
(a) Multihead Attention Block (MAB).



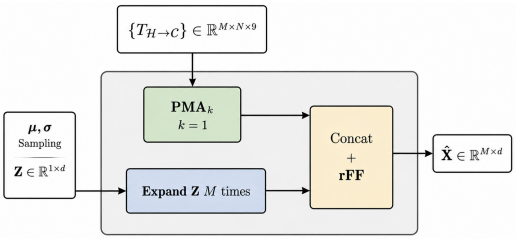
(b) Pooling by Multihead Attention (PMA). S: learned seed vectors.



(c) Pose-Aware Attention Block (PAB).



(d) Pose-aware variational encoder.



(e) Pose-conditioned held-out decoder.

Figure 3: **Detailed architecture of Casper3D.** The encoder combines semantic observations with relative camera geometry and processes them using Pose-Aware Attention Blocks (PABs), followed by permutation-invariant attention pooling to parameterize the latent posterior. The decoder combines a sampled latent state with the relative pose of a held-out target view to predict the corresponding foundation-model feature.

Relative-pose encoding. For each context observation, the camera pose is expressed relative to the other context views rather than in an absolute scene coordinate frame. Each pairwise transform is embedded by a shared pose encoder, and the resulting embeddings are aggregated with PMA to produce a pose token. This token is combined with the corresponding foundation-model feature before entering the encoder. Relative geometry is therefore associated explicitly with the semantic observation from which it was obtained.

Pose-aware attention. The same relative geometry is also injected directly into the attention operation through the PAB shown in Figure 3c. In addition to the standard content-based query–key interaction, each pair of observations receives a learned bias derived from their relative camera transform. The architecture therefore uses geometry in two complementary ways: inside the observation representation and directly in the cross-view attention weights.

Pose-aware variational encoder. The encoder consists of a stack of PABs followed by attention pooling, as illustrated in Figure 3d. The pooled representation is mapped to the posterior mean μ_i and uncertainty parameter σ_i of the posterior $q_\phi(z_i \mid \mathcal{C}_i)$. The latent variable represents the semantic information shared across the available views, while its posterior uncertainty reflects ambiguity or inconsistency in the observed context.

Pose-conditioned decoder. For a held-out target observation, its camera pose is encoded relative to the context views and aggregated into a target-view representation. The decoder in Figure 3e combines this representation with the sampled latent state to predict the foundation-model feature expected from that viewpoint. Importantly, the target feature itself is never provided to the encoder.

Consequently, successful prediction requires the latent state to capture information that transfers across viewpoints rather than simply reconstructing an observed feature.

Permutation structure. Reordering the context observations permutes the semantic tokens and the pairwise relative-pose tensor consistently. Each PAB is therefore permutation equivariant, while the subsequent PMA operation produces a permutation-invariant representation.

B PROOFS

Lemma 1 (MAB Equivariance in Queries and Invariance to Key–Value Ordering)**.** Let $\mathrm{MAB}(Q, V)$ denote the Multihead Attention Block of the Set Transformer, where $Q \in \mathbb{R}^{n\times d}$ is the query input and $V \in \mathbb{R}^{m\times d}$ is the key–value input. Let $\pi \in \mathfrak{S}_n$ be a permutation of the query indices and $\rho \in \mathfrak{S}_m$ a permutation of the key–value indices. Then

$$\mathrm{MAB}(\pi \cdot Q, V) = \pi \cdot \mathrm{MAB}(Q, V),$$

and

$$\mathrm{MAB}(Q, \rho \cdot V) = \mathrm{MAB}(Q, V).$$

These properties follow directly from the multi-head attention equations and are established for Set Transformer in Lee et al. (2019).

Lemma 2 (Joint Permutation Equivariance of MAB with Pairwise Bias)**.** Let $\mathrm{MAB}_B(Q, V)$ denote a Multihead Attention Block with additive pairwise attention bias $B \in \mathbb{R}^{N\times N}$, where $Q, V \in \mathbb{R}^{N\times d}$. Let $\pi \in \mathfrak{S}_N$ be a permutation and, by slight abuse of notation, let π also denote its corresponding permutation matrix. If the query and key–value inputs are permuted consistently, together with both axes of the pairwise bias, then

$$\mathrm{MAB}_{\pi B\pi^\top}(\pi Q, \pi V) = \pi\, \mathrm{MAB}_B(Q, V). \tag{13}$$

Proof. It suffices to prove the result for a single attention head, since multi-head attention concatenates several heads and applies shared linear maps, which preserve permutation equivariance. For notational simplicity, B denotes the bias associated with one attention head; the same argument applies independently to every head.

For one attention head with additive bias B, define

$$\mathrm{Att}_B(Q, V) = \mathrm{softmax}\left(\frac{QW_Q(VW_K)^\top}{\sqrt{d_h}} + B\right) VW_V, \tag{14}$$

where d_h denotes the dimensionality of one attention head. Let

$$A := \frac{QW_Q(VW_K)^\top}{\sqrt{d_h}} + B. \tag{15}$$

After jointly permuting the query and key–value inputs and both axes of the pairwise bias, the attention logits become

$$A' = \frac{(\pi Q)W_Q\big((\pi V)W_K\big)^\top}{\sqrt{d_h}} + \pi B\pi^\top \tag{16}$$

$$= \frac{\pi QW_Q(VW_K)^\top\pi^\top}{\sqrt{d_h}} + \pi B\pi^\top \tag{17}$$

$$= \pi\left(\frac{QW_Q(VW_K)^\top}{\sqrt{d_h}} + B\right)\pi^\top \tag{18}$$

$$= \pi A\pi^\top. \tag{19}$$

Since softmax is applied row-wise, simultaneous row and column permutations satisfy

$$\mathrm{softmax}\left(\pi A\pi^\top\right) = \pi\ \mathrm{softmax}(A)\pi^\top. \tag{20}$$

Therefore,

$$\mathrm{Att}_{\pi B\pi^\top}(\pi Q, \pi V) = \mathrm{softmax}\left(\pi A \pi^\top\right)(\pi V) W_V \tag{21}$$

$$= \pi\ \mathrm{softmax}(A)\pi^\top \pi V W_V \tag{22}$$

$$= \pi\ \mathrm{softmax}(A) V W_V \tag{23}$$

$$= \pi\ \mathrm{Att}_B(Q, V). \tag{24}$$

Thus, a single biased attention head is permutation equivariant when the query and key–value inputs are permuted consistently and the two corresponding axes of the pairwise bias are permuted in the same way.

The multi-head attention layer preserves this property because each head satisfies the same equivariance relation and the output projection is applied row-wise with shared parameters. The remaining operations in the MAB—residual connections, layer normalization, and row-wise feed-forward networks—are also applied identically to each row and therefore commute with the action of π. Hence,

$$\mathrm{MAB}_{\pi B\pi^\top}(\pi Q, \pi V) = \pi\, \mathrm{MAB}_B(Q, V). \tag{25}$$

□

Lemma 3 (Permutation Invariance of PMA)**.** Let PMA denote Pooling by Multihead Attention with a learned seed matrix S. Then, for any permutation $\pi \in \mathfrak{S}_n$ and any $X \in \mathbb{R}^{n\times d}$,

$$\mathrm{PMA}(\pi \cdot X) = \mathrm{PMA}(X).$$

Proof. Since rFF is applied row-wise with shared parameters, it is permutation equivariant:

$$\mathrm{rFF}(\pi \cdot X) = \pi \cdot \mathrm{rFF}(X).$$

Therefore,

$$\begin{aligned}\mathrm{PMA}(\pi \cdot X) &= \mathrm{MAB}\big(S, \mathrm{rFF}(\pi \cdot X)\big)\\ &= \mathrm{MAB}\big(S, \pi \cdot \mathrm{rFF}(X)\big)\\ &= \mathrm{MAB}\big(S, \mathrm{rFF}(X)\big)\\ &= \mathrm{PMA}(X).\end{aligned}$$

The third equality follows from Lemma 1. □

Proposition 1 (Permutation Equivariance of the PAB Block)**.** Let $X \in \mathbb{R}^{N\times d}$ denote an unordered set of multi-view semantic features, and let

$$P \in \mathbb{R}^{N\times N\times d_p}$$

denote the corresponding pairwise relative-pose representation, with

$$P_{k\ell} = g_\eta(T_{k\to\ell}).$$

For any permutation $\pi \in \mathfrak{S}_N$, let

$$P' = \pi P \pi^\top,$$

where the multiplication denotes the corresponding permutation of the first two, view-index, axes of P. Then

$$\mathrm{PAB}(\pi X, P') = \pi\, \mathrm{PAB}(X, P). \tag{26}$$

Proof. For each observation k, define its aggregated pose representation as

$$p_k = \mathrm{PMA}\left(\{P_{k\ell}\}_{\ell=1}^N\right). \tag{27}$$

When the observations are reordered by π, both view indices of the pairwise pose tensor are permuted. Consequently, the pose entries associated with each permuted observation are the same entries as before, but reordered along the second view index. Since PMA is permutation invariant by Lemma 3, the resulting collection of pose summaries is permuted only along its observation index:

$$p' = \pi p. \tag{28}$$

Define the intermediate pose-aware semantic representation

$$\Phi(X, P) := \mathrm{rFF}\left(\mathrm{Concat}(X, p)\right). \tag{29}$$

Because concatenation is performed observation-wise and rFF is applied row-wise with shared parameters,

$$\Phi(\pi X, \pi P \pi^\top) = \pi\, \Phi(X, P). \tag{30}$$

The pairwise pose representations are additionally mapped to additive attention biases. Let

$$B(P)_{k\ell} = b(P_{k\ell}), \tag{31}$$

where the same learned function b is applied independently to each ordered pair of observations. It follows immediately that

$$B(\pi P \pi^\top) = \pi B(P) \pi^\top. \tag{32}$$

By definition of the PAB block,

$$\mathrm{PAB}(X, P) = \mathrm{MAB}_{B(P)}\left(\Phi(X, P), \Phi(X, P)\right). \tag{33}$$

Therefore,

$$\begin{aligned}
\mathrm{PAB}(\pi X, \pi P \pi^\top) &= \mathrm{MAB}_{B(\pi P \pi^\top)}\left(\Phi(\pi X, \pi P \pi^\top), \Phi(\pi X, \pi P \pi^\top)\right) && (34)\\
&= \mathrm{MAB}_{\pi B(P) \pi^\top}\left(\pi\Phi(X, P), \pi\Phi(X, P)\right) && (35)\\
&= \pi\, \mathrm{MAB}_{B(P)}\left(\Phi(X, P), \Phi(X, P)\right) \qquad \text{by Lemma 2} && (36)\\
&= \pi\, \mathrm{PAB}(X, P). && (37)
\end{aligned}$$

Hence, the PAB block is permutation equivariant. □

Proposition 2 (Permutation Invariance of the Encoder)**.** Let Enc denote the pose-aware encoder obtained by composing permutation-equivariant PAB blocks followed by a permutation-invariant PMA layer, as illustrated in Figure 3d. Then, for any permutation $\pi \in \mathfrak{S}_N$,

$$\mathrm{Enc}(\pi X, \pi P \pi^\top) = \mathrm{Enc}(X, P). \tag{38}$$

Proof. Let

$$H_0 := X,$$

and define the successive PAB representations by

$$H_\ell := \mathrm{PAB}_\ell(H_{\ell-1}, P), \qquad \ell = 1, \ldots, L. \tag{39}$$

Under a permutation of the observations, define

$$H'_0 := \pi X, \qquad P' := \pi P \pi^\top.$$

Since each PAB_ℓ is permutation equivariant by Proposition 1, we have

$$H'_\ell = \pi H_\ell, \qquad \ell = 1, \ldots, L, \tag{40}$$

by induction. In particular,

$$H'_L = \pi H_L. \tag{41}$$

The encoder output is obtained by applying PMA to the final set representation:

$$\mathrm{Enc}(X, P) = \mathrm{PMA}(H_L). \tag{42}$$

Using Lemma 3,

$$\begin{aligned}
\mathrm{Enc}(\pi X, \pi P \pi^\top) &= \mathrm{PMA}(H'_L) && (43)\\
&= \mathrm{PMA}(\pi H_L) && (44)\\
&= \mathrm{PMA}(H_L) && (45)\\
&= \mathrm{Enc}(X, P). && (46)
\end{aligned}$$

Therefore, the pose-aware encoder is permutation invariant with respect to the ordering of the context observations. □

C Implementation Details

2D semantic features. Our default 2D semantic teacher is TIPSv2. For each valid 2D reprojection of a 3D point, we extract a 1024-dimensional semantic feature and store it together with view identity and geometric metadata. In all main experiments reported in the paper, Casper3D operates on these TIPSv2 features, although the architecture itself is backbone-agnostic.

Geometric preprocessing. We preprocess each scene by reprojecting 3D points into all available RGB-D views using camera intrinsics and extrinsics. A reprojected observation is considered valid only if it satisfies geometric and visibility constraints. We use a relative depth consistency rule between the reprojected depth and the observed depth map, together with a border exclusion heuristic to remove unstable image-edge projections. The resulting valid observations are grouped by 3D point. For each 3D point, we store: its 3D coordinates, the list of valid view IDs, the corresponding 2D semantic features, and the associated camera extrinsics. We additionally precompute the average feature across valid observations for each point, which is used as a baseline and as an auxiliary alignment target during training.

Training samples. Training is performed on scene chunks. Each training sample is constructed by first selecting a scene, then sampling a chunk of 2048 3D points from that scene. For each point, we sample up to 20 context views and up to 20 held-out target views. The fraction of observations allocated to context is capped by `max_context_frac`=0.6. When fewer observations are available, we use padding and binary masks. At training time, the model receives the context observations and is trained to predict the semantic embeddings of held-out observations.

Observation representation. Each context observation is represented by a 1024-dimensional semantic feature. Pairwise relative pose features are computed between context views and are used as pairwise attention biases in the encoder. Held-out target views are represented by their relative pose with respect to the context views and are used by the decoder for conditional prediction.

Network Architecture. Our architecture takes as input a set of 1024-dimensional semantic tokens. An input linear layer first maps these features to an encoder dimension of 1024. The encoder consists of 4 stacked PABs with 8 attention heads, LayerNorm, and dropout probability 0.05. Relative pose information is injected through a dedicated pose encoder that takes 12-dimensional pose descriptors as input and maps them to attention-space biases using a hidden size of 64 and output size 128, with dropout 0.05.

After encoder processing, we reduce the set representation using a learned seed-based reducer with 4 seeds. The reduced representation is then processed by two latent reduction layers of dimension 512. Two separate output heads map this representation to the posterior mean and uncertainty parameters, each in $\mathbb{R}^{1024}$, matching the dimensionality of the original foundation-model features. The decoder uses 4 attention heads, a hidden dimension of 512, 2 hidden layers, and dropout 0.05.

Latent objective. We train the variational model with a held-out semantic prediction objective and a KL regularizer. The held-out prediction loss is computed in feature space using cosine similarity. The KL term is weighted by 10^{-4} and linearly annealed over the first 100 000 optimization steps. We do not use free bits in the reported experiments.

Semantic alignment losses. In addition to the held-out semantic prediction objective, we use two auxiliary semantic alignment terms. First, we align the inferred 3D semantic representation with the mean aggregated 2D semantic embedding of the same 3D point, using weight $\lambda_{\text{avg}} = 0.05$. Second, we align the inferred representation with text space using text prototypes, with weight $\lambda_{\text{text}} = 0.1$. Text similarity uses temperature 0.05. Unless otherwise noted, text prototypes are built with the LAION-style prompt templates.

Optimization. We train with Adam using learning rate 5×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 10^{-4}. We use gradient clipping with maximum norm 1.0. The learning rate follows a warm-up-cosine schedule with 10 000 warm-up steps and minimum learning-rate ratio 0.01. Training runs for 50 epochs, with 10 000 optimization steps per epoch.

Batching and data loading. We use a batch size of 512 for training. Data loading uses 16 workers and prefetch factor 4. All experiments are run on an NVIDIA RTX A6000.

Voxelization and evaluation protocol. For 3D evaluation, we voxelize the point cloud using voxel size 0.02. This is used only for evaluation and comparison on standard 3D segmentation metrics. In the default evaluation setting, we use 20 context views per point.

Reproducibility. All architectural and optimization hyperparameters used in the main experiments are fixed in the public configuration file. In particular, the encoder depth, number of heads, latent reduction dimension, number of context and target views, optimizer settings, KL annealing schedule, and alignment-loss weights are kept constant across the main ScanNet experiments.

C.1 Default Configuration and Hyperparameter Sensitivity

Hyperparameter	Default	Hyperparameter	Default
Text-loss weight λ_{text}	0.1	Average-anchor weight λ_{avg}	0.05
Text temperature τ	0.05	KL weight β	10^{-4}
KL warm-up	100k	Learning rate	5×10^{-5}
LR warm-up	10k	Encoder depth	4
Attention heads	8	Latent reduction dimension	512
Maximum context views	20	Maximum target views	20
Maximum context fraction	0.6	Batch size	512

Table 4: Default Casper3D configuration used throughout the main experiments. Unless otherwise specified, these settings are kept fixed in all controlled ablations.

Hyperparameter sensitivity. Table 5 shows that Casper3D is generally robust around the selected configuration. Moderate text-alignment and feature-anchor weights perform similarly, while the KL weight and learning rate are comparatively more sensitive. Warm-up durations have limited impact, and encoder depths of four and six layers perform similarly, whereas a two-layer encoder is weaker.

Quantity	Tested values	ScanNet mIoU ↑
Text weight λ_{text}	0, .01, .05, **.1**, .5, 1	61.1, 62.2, 63.5, **64.6**, 63.4, 63.8
Average weight λ_{avg}	0, .01, **.05**, .1	59.9, 64.0, **64.6**, 63.3
KL weight β	10^{-5}, 5×10^{-5}, $\mathbf{10^{-4}}$, 5×10^{-4}, 10^{-3}	60.0, 64.0, **64.6**, 59.9, 40.4
KL warm-up	25k, 50k, **100k**, 200k	63.2, 64.3, **64.6**, 64.5
Learning rate	$\mathbf{5\times10^{-5}}$, 10^{-4}, 5×10^{-4}, 10^{-3}	**64.6**, 62.2, 54.4, 43.8
LR warm-up	5k, **10k**	64.1, **64.6**
Encoder depth	2, **4**, 6	60.8, **64.6**, 64.4

Table 5: Hyperparameter sensitivity on ScanNet. Bold values indicate the default configuration. Performance is stable around the selected settings, while the KL weight and learning rate exhibit greater sensitivity. Warm-up durations are comparatively robust, and four and six encoder layers perform similarly.

C.2 Detailed Efficiency Profile

Table 6 reports model size and runtime statistics. FLOPs, inference latency, and peak GPU memory are measured for a single forward pass with batch size 1, 20 context observations, and 20 held-out observations; inference latency is averaged over 100 runs after 10 warm-up iterations on a single GPU. Casper3D remains lightweight, requiring only 0.64G FLOPs, 14.4 ms inference time, and 150 MB peak GPU memory. This makes it practical as a semantic inference module and complementary to heavier 3D distillation pipelines.

Metric	Casper3D
FLOPs ↓	0.64G
Parameters ↓	35.6M
Peak GPU memory ↓	150 MB
Training time ↓	41.6 h
Inference time ↓	14.4 ms
Training / inference GPU	RTX A6000

Table 6: Detailed computational profile of Casper3D. Inference statistics are measured with batch size 1 using 20 context and 20 target observations on a single RTX A6000. Inference latency is averaged over 100 runs after 10 warm-up iterations.

D Optional 3D Distillation

The 3D-distillation stage is optional and is not part of the Casper3D objective. Casper3D first produces a semantic feature cloud

$$\mathbf{F}_{2\mathrm{D}} \in \mathbb{R}^{M \times C}, \tag{47}$$

with one multi-view-consistent embedding for each of the M 3D points. Here, C denotes the dimensionality of the original foundation-model embedding space. These features can be used directly when the corresponding images are available.

For image-free inference, we freeze this Casper3D feature cloud and distill it into a MinkowskiNet18A point-cloud encoder $E_{3\mathrm{D}}$. Given point coordinates

$$\mathbf{P} \in \mathbb{R}^{M \times 3}, \tag{48}$$

the point-cloud encoder produces

$$\mathbf{F}_{3\mathrm{D}} = E_{3\mathrm{D}}(\mathbf{P}) \in \mathbb{R}^{M \times C}. \tag{49}$$

The encoder is trained to reproduce the frozen Casper3D features using a cosine distillation objective,

$$\mathcal{L}_{\mathrm{distill}} = \frac{1}{M} \sum_{i=1}^{M} \left[1 - \cos\left(F_{2D}^{(i)}, F_{3D}^{(i)} \right) \right]. \tag{50}$$

No ground-truth semantic labels are used during this stage. Because the Casper3D features remain aligned with the original CLIP embedding space, the distilled point-cloud features retain language compatibility and can therefore be queried with arbitrary text prompts without requiring images at inference.

Thus, *Ours* uses the Casper3D semantic feature cloud directly, whereas *Ours + 3D Distillation* transfers these features into a point-cloud-only model.

E More results

E.1 Outdoor Pretraining and Cross-Domain Transfer

To examine whether Casper3D can learn transferable multi-view inference from outdoor observations, we perform a preliminary cross-domain experiment using the outdoor KITTI sequences recorded on `2011_09_26`. KITTI LiDAR measurements define the underlying 3D locations and are used to associate image features corresponding to the same physical points. We pretrain Casper3D on these outdoor observations and subsequently fine-tune it on ScanNet200.

Table 7 reports the resulting cross-domain transfer performance.

As shown in Table 7, outdoor KITTI pretraining improves ScanNet200 mIoU from 11.0 to 11.5 and mAcc from 18.1 to 18.7. Although this is not a direct evaluation on an outdoor semantic-segmentation benchmark, it provides preliminary evidence that Casper3D can learn transferable multi-view inference from outdoor observations.

Initialization	ScanNet200 mIoU ↑	ScanNet200 mAcc ↑
Base Casper3D	11.0	18.1
KITTI-pretrained Casper3D	**11.5**	**18.7**

Table 7: **Cross-domain transfer from outdoor KITTI pretraining.** Pretraining Casper3D on outdoor KITTI observations before fine-tuning on ScanNet200 improves both mIoU and mAcc.

E.2 Qualitative results

Semantic predictions. Figures 4 and 5 compare Casper3D predictions with the ground-truth semantic maps on representative ScanNet scenes. Overall, the qualitative results show that the inferred 3D feature field produces spatially coherent semantic predictions.

Latent semantic structure. Figure 6 further compares the geometry of the inferred latent representation with deterministic multi-view feature averaging. Across the representative scenes, averaged features exhibit substantial class overlap, whereas the latent embeddings form more compact class-consistent regions with clearer separation. This provides qualitative evidence that Casper3D reorganizes noisy multi-view observations into a more structured 3D semantic representation rather than merely reproducing their average.

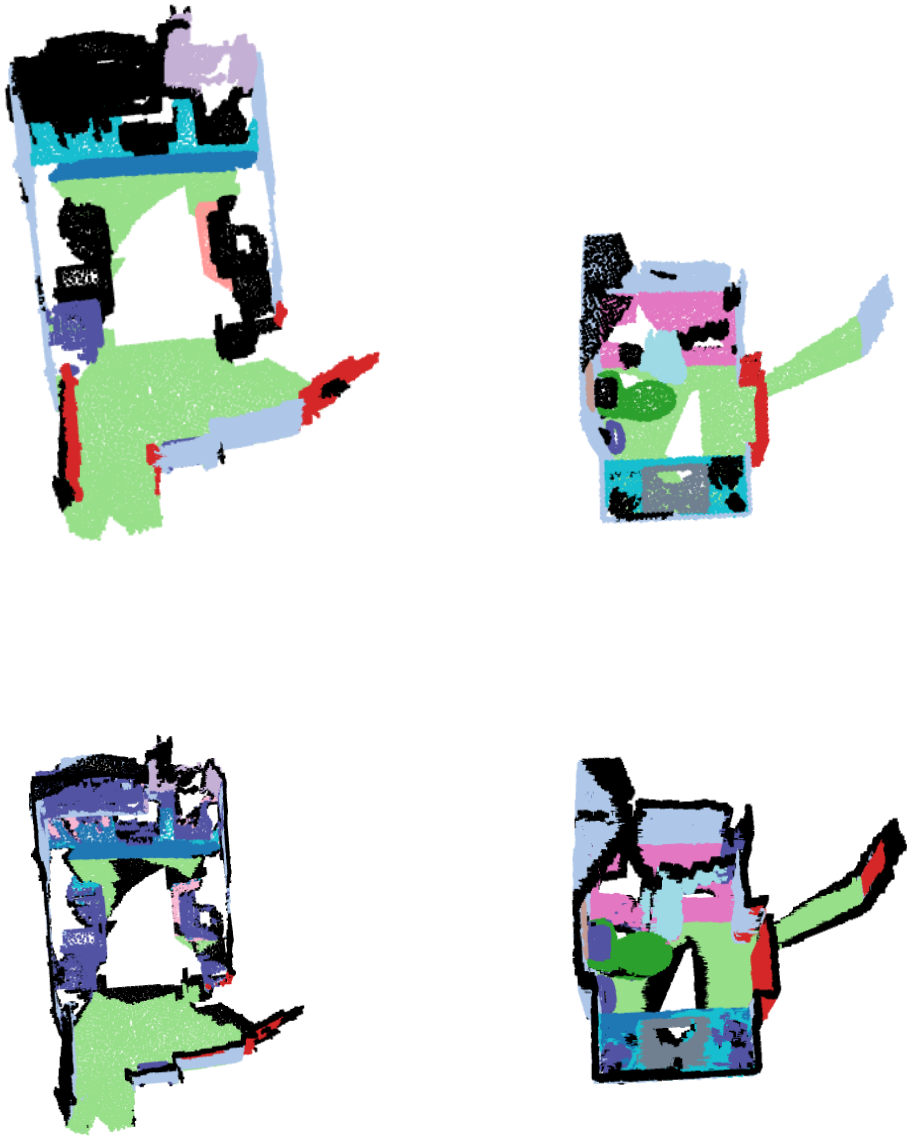


Figure 4: Qualitative comparison between ground-truth semantic maps and Casper3D predictions. Top: ground truth. Bottom: predicted outputs.

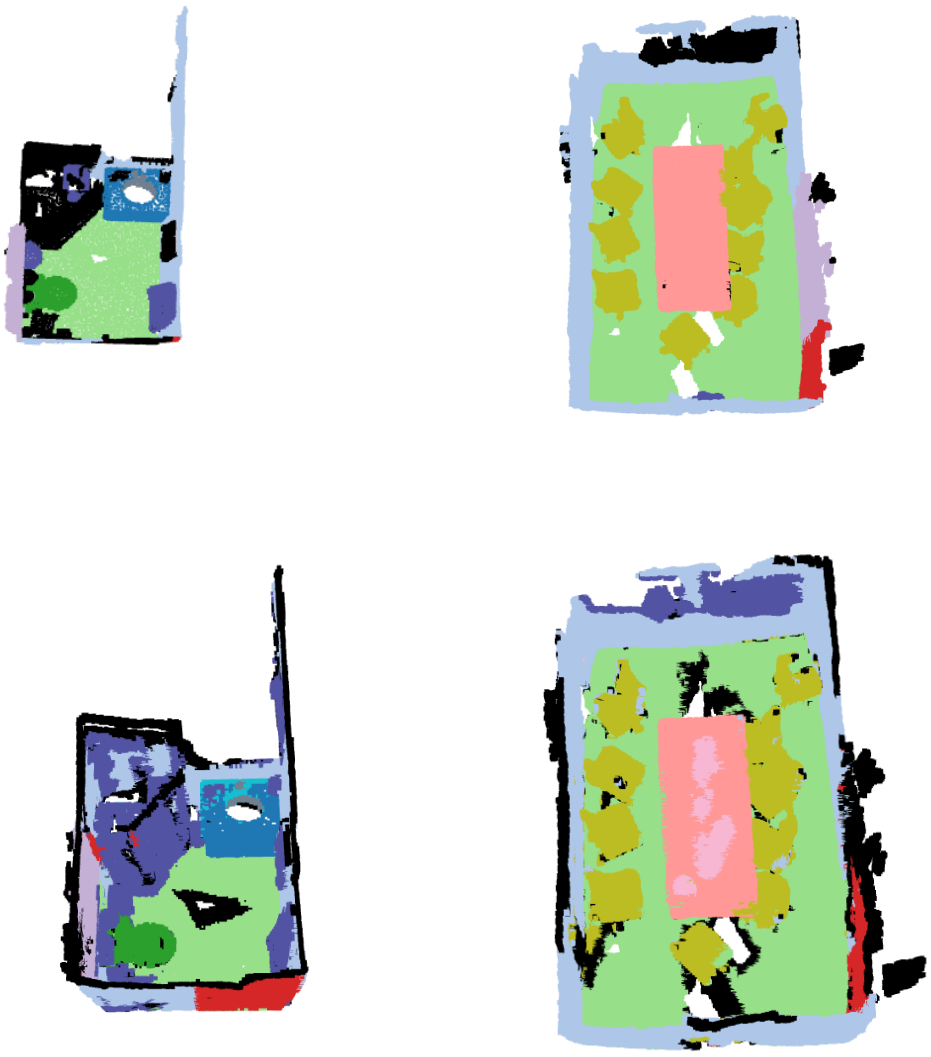
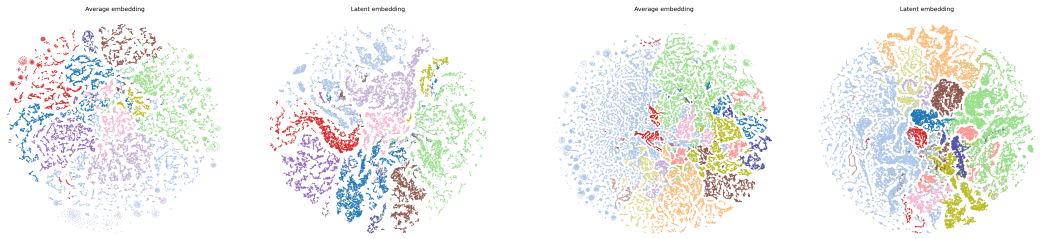
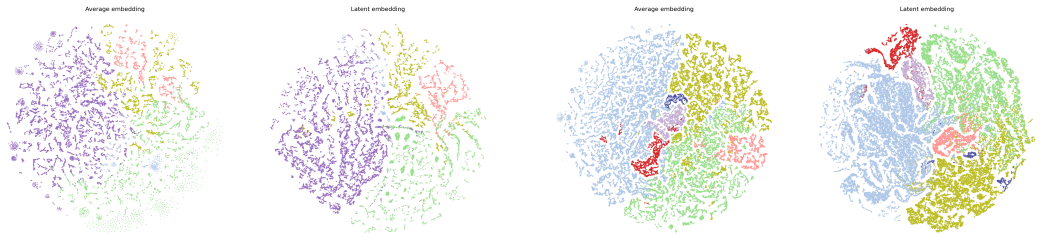


Figure 5: Additional qualitative comparison between ground-truth semantic maps and Casper3D predictions. Top: ground truth. Bottom: predicted outputs.



Average Latent Average Latent

(a) Scene 1 (b) Scene 2



Average Latent Average Latent

(c) Scene 3 (d) Scene 4

Figure 6: t-SNE visualizations of point embeddings on representative ScanNet scenes. Compared with deterministic average aggregation, the latent features inferred by Casper3D form tighter same-class clusters and exhibit less inter-class mixing, indicating a more structured semantic field.