Le MuMo JEPA: Multi-Modal Self-Supervised Representation Learning with Learnable Fusion Tokens

Ciem Cornelissen Sam Leroux Pieter Simoens
IDLab, Department of Information Technology, Ghent University - imec
Belgium
{ciem.cornelissen, sam.leroux, pieter.simoens}@ugent.be

Abstract

Self-supervised learning has emerged as a powerful paradigm for learning visual representations without manual annotations, yet most methods still operate on a single modality and therefore miss the complementary structure available from heterogeneous sensors. We present Le MuMo JEPA, a self-supervised framework that learns unified representations from RGB images and aligned companion modalities. In our driving experiments, the second modality is camera-aligned LiDAR depth; we also evaluate RGB-thermal training and transfer on the Teledyne FLIR ADAS benchmark. Our approach extends LeJEPA to the multi-modal setting by learning fusion tokens *that act as a latent bottleneck between modality-specific patch stems inside a shared transformer. Our default model employs a pruned fusion strategy: after an initial cross-modal attention layer, modality-specific tokens are dropped, forcing cross-modal information into the shared fusion-token grid as an efficient latent bottleneck before Sketched Isotropic Gaussian Regularization (SIGReg) is applied to the joint multimodal CLS embedding. On Waymo, Le MuMo JEPA gives the strongest performance-efficiency trade-off on downstream patch probes among the from-scratch multi-modal baselines, improving CenterNet detection and dense depth while remaining competitive on segmentation. Under from-scratch training on nuScenes, Le MuMo JEPA remains the strongest model, and it also gives the best FLIR results, especially after Waymo-initialized fine-tuning. It also retains the best overall accuracy-efficiency balance in our study at substantially lower compute, memory, and estimated training time.*

1. Introduction

Many real-world perception systems rely on multiple sensors, with cameras providing dense texture and color information while complementary sensor modalities such as LiDAR depth or thermal infrared contribute geometry, temperature, or range cues. Learning representations that effectively combine these signals remains an open challenge, and most state-of-the-art multi-modal perception models [3, 20] are still trained in a fully supervised manner with costly large-scale annotations. Autonomous driving provides a natural testbed for this paradigm because modern perception stacks already rely on paired dense sensors and spatial correspondence across those streams matters directly for downstream reasoning.

Self-supervised learning (SSL) offers a compelling alternative by learning general-purpose representations from unlabeled data. Methods such as BYOL [14], DINO [10], MAE [16], and I-JEPA [1] have achieved strong image-understanding results. LeJEPA [4] further introduces *Sketched Isotropic Gaussian Regularization* (SIGReg), which constrains embeddings to follow an isotropic Gaussian distribution without relying on common SSL heuristics such as stop-gradients or teacher-student networks. For multi-modal learning, this is appealing because both modalities are pulled toward the same data-agnostic target geometry instead of being aligned only through pairwise contrastive matching, which typically depends more heavily on negative sampling and careful control of the modality gap.

However, existing JEPA-based methods operate exclusively on single-modality inputs. Extending self-supervised learning to RGB-plus-modality settings raises a more specific problem: the streams have very different structures; they must still be aligned well enough to support meaningful correspondences, and their interaction must be explicit and efficient, avoiding both the representational limits of weak late fusion and the quadratic computational cost of unrestricted all-to-all token mixing. We therefore frame both streams on a shared 2D spatial grid so that dense token-to-token alignment happens inside one transformer without the overhead of a separate sparse 3D backbone [20, 29]. Keeping both modalities in one token space also makes it easier to transfer from RGB-LiDAR training to RGB-thermal evaluation on FLIR [25].

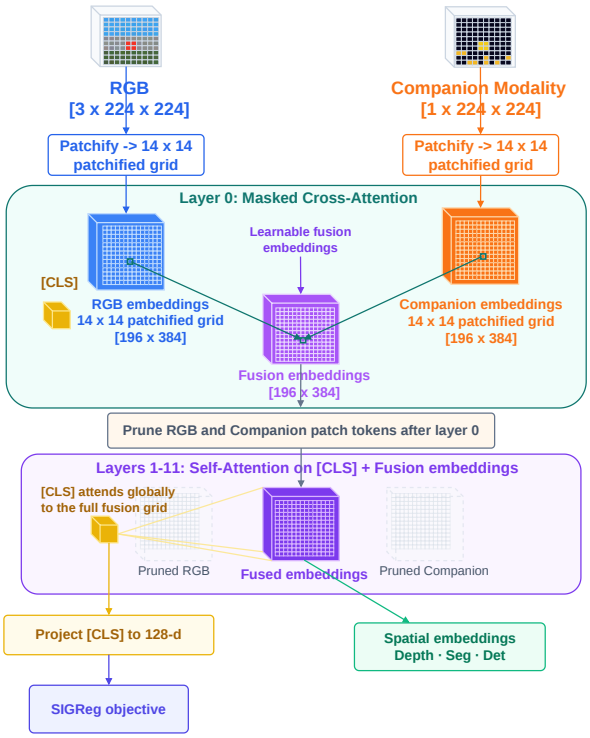

Figure 1. **Overview of Le MuMo JEPA.** The companion modality is represented as a spatially aligned signal and fused with RGB through learnable *fusion tokens*, which act as a latent bottleneck inside a shared transformer. The default training objective applies SIGReg to the joint multimodal CLS embedding.

In this work, we present Le MuMo JEPA, a self-supervised framework that learns unified RGB-plus-companion-modality representations. Our key insight is that SIGReg provides a natural *modality-agnostic* shared target: by encouraging both RGB and companion-modality representations to follow the same Gaussian reference distribution, the model learns a shared embedding geometry across modalities without imposing artificial token-wise pairing constraints.

Within a standard Vision Transformer [12], we introduce *learnable fusion tokens*, a set of tokens equal in number to the spatial patches that aggregate information from spatially corresponding RGB and companion-modality patches through attention. Conceptually, these tokens form a Perceiver-style latent bottleneck [17]: cross-modal communication is routed through a learned spatial memory buffer instead of allowing unrestricted all-to-all interaction between every RGB and companion-modality token.

We evaluate Le MuMo JEPA on Waymo [24], nuScenes [9], and RGB-thermal FLIR [25]. Waymo and nuScenes use matched from-scratch training within each dataset, while FLIR is reported both from scratch and under Waymo-initialized transfer and fine-tuning. Using frozen patch probes for CenterNet-style 3D object detection [30], depth estimation, and segmentation mIoU, we show that the default Le MuMo JEPA configuration consistently outperforms single-modality baselines and simpler early- or late-fusion alternatives. Waymo and nuScenes comparisons retrain each encoder from scratch within the target dataset rather than comparing against off-the-shelf checkpoints; FLIR additionally includes Waymo-to-FLIR transfer and fine-tuning. The largest gains appear in object-centric localization and depth, where explicit cross-modal fusion improves both geometry and downstream patch-probe performance over RGB-only JEPA [4], a DINOv3-style RGB baseline [23], MultiMAE [2], ImageBind [13], and simple fusion ablations.

Our contributions are:

- We extend LeJEPA to the multi-modal setting and show that SIGReg regularization supports joint RGB-plus-modality representation learning without auxiliary alignment labels.
- We introduce learnable fusion tokens with a pruned default design that acts as an efficient latent bottleneck, together with a persistent-routing ablation, and show that applying SIGReg to the joint multimodal CLS embedding yields the strongest default configuration in our study.
- We benchmark Le MuMo JEPA on Waymo, nuScenes, and FLIR against single-modality controls, early and late fusion ablations, MultiMAE, a DINOv3-style RGB baseline, and ImageBind under matched from-scratch protocols on Waymo and nuScenes, plus from-scratch and Waymo-initialized evaluation on FLIR, using the same frozen patch probes and compute profiling pipeline.

2. Related Work

Self-Supervised Visual Representation Learning. Self-supervised visual learning spans contrastive methods such as SimCLR [11] and MoCo [15], non-contrastive methods such as BYOL [14] and VICReg [6], masked modeling methods such as MAE [16] and BEiT [5], and JEPA-style latent prediction [1, 7, 18]. LeJEPA [4] provides theoretical foundations for JEPA models through SIGReg, which matches the embedding distribution to $\mathcal{N}(0, \mathbf{I})$. Our work extends LeJEPA to multimodal RGB-plus-companion-modality inputs while retaining its regularization framework and JEPA-style predictive training.

Multi-Modal Sensor Fusion. Supervised camera-LiDAR fusion has been explored at various stages, from point-level sequential fusion [27] to late-stage bounding-box candidate fusion [21], alongside feature-space systems using transformers or bird's-eye view (BEV) grids such as

BEVFusion [19, 20] and TransFusion [3]. These methods rely on extensive 3D annotations. In contrast, we learn fused representations in a fully self-supervised manner, requiring no labeled data during pretraining.

Self-Supervised Multi-Modal Learning. Multi-modal self-supervised learning includes shared-embedding alignment methods such as ImageBind [13], masked reconstruction methods such as MultiMAE [2], and recent driving-oriented approaches such as ALSO, SLidR, UniPAD, and GeoMAE [8, 22, 26, 29]. Compared with reconstruction-based and pairwise contrastive objectives, Le MuMo JEPA uses SIGReg's isotropic Gaussian target as a shared regularizing anchor while learnable fusion tokens provide a structured mechanism for cross-modal interaction inside the transformer.

3. Method

We present Le MuMo JEPA, a multi-modal self-supervised framework that extends LeJEPA [4] to jointly learn from RGB images and aligned companion modalities. Figure 1 provides an overview of the architecture.

3.1. Preliminaries: LeJEPA and SIGReg

LeJEPA [4] is a Joint-Embedding Predictive Architecture that learns invariance across augmented views while preventing representation collapse through *Sketched Isotropic Gaussian Regularization* (SIGReg).

Let $\{\mathbf{x}_v^g\}_{v=1}^{V_g}$ denote the V_g global augmented image views of an input and $\{\mathbf{x}_u^\ell\}_{u=1}^{V_\ell}$ its V_ℓ local views. Each view is first processed by the shared encoder f_θ. The resulting representation is then mapped by the projector g_ϕ to latent embeddings $\mathbf{z}_v^g = g_\phi(f_\theta(\mathbf{x}_v^g))$ and $\mathbf{z}_u^\ell = g_\phi(f_\theta(\mathbf{x}_u^\ell))$ in $\mathbb{R}^d$. The training objective combines an invariance loss with SIGReg:

$$\mathcal{L}_{\text{LeJEPA}} = \lambda \cdot \mathcal{L}_{\text{SIGReg}}(\mathbf{Z}) + (1-\lambda) \cdot \mathcal{L}_{\text{inv}}, \qquad (1)$$

where λ is a single trade-off hyperparameter. In the multi-modal implementation used for Le MuMo JEPA, the target center is computed from the global views and the penalty is applied to all available views, so both global and local crops are pulled toward a shared center. SIGReg then prevents collapse by matching the empirical embedding distribution to $\mathcal{N}(0, \mathbf{I})$ through characteristic-function matching over random projections and fixed evaluation knots. This yields a regularizer with $\mathcal{O}(BK(T+d))$ complexity, where B is the batch size, K the number of random projection directions, T the number of evaluation knots, and d the embedding dimension, with no stop-gradients or teacher-student networks required; the detailed invariance and SIGReg expressions are given in the supplementary material.

3.2. Multi-Modal Inputs

To enable a unified transformer-based encoder for both modalities, we represent camera images and aligned companion signals in a common 2D format.

Camera Input. RGB images are processed using standard ViT patch embedding [12]. Each image is resized to 224×224 and divided into $N = 14 \times 14 = 196$ non-overlapping patches of size 16×16, which are linearly projected to the ViT embedding dimension. We apply the multi-crop augmentation strategy from LeJEPA [4]: V global crops (scale $[0.4, 1.0]$, size 224×224) and V_{local} local crops (scale $[0.05, 0.4]$, size 96×96), together with ColorJitter, RandomGrayscale, GaussianBlur, and RandomSolarize augmentations.

Companion-Modality Input. For RGB-depth experiments, we project the 3D LiDAR point cloud into the camera coordinate frame to obtain an *aligned depth map*. We render that depth map by writing points in depth-descending order so nearer points overwrite farther ones, normalized by a maximum range of $r_{\max} = 80\,\text{m}$; pixels with no LiDAR return remain zero. This discards some native 3D structure, but it gives strict pixel-level alignment with RGB, keeps both modalities inside a unified transformer tokenization scheme, and avoids a separate 3D backbone. For RGB-thermal experiments, the companion signal is the aligned thermal image itself, resized and patchified on the same spatial grid through its own modality-specific patch stem. Using a shared 2D tokenization lets the same dense ViT family cover both RGB-depth and RGB-thermal settings rather than mixing image-plane transformers with a separate sparse-3D architecture only for LiDAR.

Modality Embeddings. To distinguish between camera and companion-modality tokens in the shared transformer, we add learnable modality embeddings $\mathbf{e}_{\text{cam}}$ and $\mathbf{e}_{\text{mod}}$ to the respective patch embeddings before entering the transformer blocks. After tokenization, these become the camera token set $\mathbf{C}$ and companion-modality token set $\mathbf{M}$ used in Eq. (2).

3.3. Learnable Fusion Tokens

A central design choice in multi-modal transformers is how and where to fuse information across modalities. Rather than relying on full-token concatenation or late fusion, we follow the logic of a latent bottleneck [17]: a learned token set mediates cross-modal exchange, replacing expensive camera-to-LiDAR all-pairs interaction with routing through compact fusion tokens that act as a spatial memory buffer.

We introduce *learnable fusion tokens* that provide a structured mechanism for cross-modal interaction. Given

N spatial patch positions, we create N learnable fusion token embeddings $\{\mathbf{f}_i\}_{i=1}^{N}$, each associated with a spatial position and represented in the same embedding space $\mathbb{R}^D$ as the ViT tokens. We keep the number of fusion tokens equal to the patch count because the current design preserves one latent per spatial location: this keeps the cross-modal pairing explicit in the first fusion layer and retains the full $14 \times 14 = 196$ spatial grid that can be read out directly by the downstream patch probes. For the from-scratch experiments in this paper, these fusion tokens are initialized with a truncated normal distribution with standard deviation 0.02, matching the ViT token initialization used elsewhere in the encoder. Together with a shared CLS token and the N camera and N companion-modality patch tokens, the full token sequence entering the transformer is:

$$[\texttt{CLS}(1),\ \mathbf{F}(N),\ \mathbf{C}(N),\ \mathbf{M}(N)], \tag{2}$$

where $\mathbf{F}$, $\mathbf{C}$, $\mathbf{M}$ denote fusion, RGB, and companion-modality token sets respectively, for a total of $1+3N$ tokens (589 tokens for the 14×14 patch grid used here). We extend the positional embedding to cover all tokens.

Attention Masking Strategies. We consider the default pruned design together with a persistent-routing ablation that controls how fusion tokens interact with modality tokens across layers.

(a) Pruned Fusion. In the first transformer layer (layer 0), fusion token $\mathbf{f}_i$ is allowed to attend to its spatially corresponding camera patch $\mathbf{c}_i$ and companion-modality patch $\mathbf{m}_i$, as well as to itself and the CLS token. After layer 0, all $2N$ camera and companion-modality tokens are pruned from the sequence, leaving only $1 + N$ tokens (CLS + fusion). Because the fusion tokens attend to the modality-specific patches in layer 0, gradients still flow back through that cross-attention path to update both the RGB and companion-modality patch stems even though those tokens are pruned later. This creates an efficient latent bottleneck and reduces the computational cost of subsequent layers from $\mathcal{O}((1+3N)^2)$ to $\mathcal{O}((1+N)^2)$ per layer, a $\sim9\times$ reduction in attention cost, while forcing the model to compress the useful cross-modal evidence into the shared fusion-token grid early.

(b) Persistent Fusion (ablation). Fusion tokens maintain attention to their paired camera and companion-modality patches throughout *all* transformer layers. This preserves the full token set and allows deeper cross-modal reasoning at higher computational cost. Unless noted otherwise, Le MuMo JEPA refers to the default FT-Pruned SIGReg configuration, i.e., the pruned fusion-token encoder trained with the joint-CLS objective below.

3.4. Training Objective

The default Le MuMo JEPA training objective keeps the encoder and token routing identical to Sec. 3.3 but applies JEPA supervision only to the joint multimodal CLS embedding. For each paired input, the encoder processes the full multimodal token sequence once and produces a single fused CLS embedding that is passed through the projection head. The spatial fusion tokens are *not* projected individually. Instead, they are updated indirectly because the CLS token aggregates information from the fusion-token grid inside the same transformer, so the SIGReg and invariance gradients backpropagate through the CLS-to-fusion attention pathway. These fusion tokens are exactly the dense spatial features exposed to the downstream patch probes. Let $\mathbf{Z}^{(\text{joint})}$ denote the resulting set of projected views from the multimodal crops. The default loss is then

$$\mathcal{L}_{\text{MM}} = \lambda \cdot \mathcal{L}_{\text{SIGReg}}(\mathbf{Z}^{(\text{joint})}) + (1-\lambda) \cdot \mathcal{L}_{\text{inv}}^{(\text{joint})}. \tag{3}$$

Here, $\mathcal{L}_{\text{inv}}^{(\text{joint})}$ is the mean-squared invariance term that pulls the projected global and local fused-crop embeddings toward the shared global-view center. The computationally heavier FT-Pruned + SIGReg (3-pass) ablation keeps the same pruned encoder but averages SIGReg over three forwards: joint RGB+companion-modality, RGB-only with the companion modality zeroed out, and companion-modality-only with RGB zeroed out. Alternative supervision variants are evaluated in the ablation study in Tabs. 2 and 6, with the full 3-pass expression given in the supplementary material.

4. Experiments

4.1. Experimental Setup

Dataset. We use Waymo [24] as the main driving benchmark, nuScenes [9] as a second from-scratch driving benchmark, and the Teledyne FLIR ADAS dataset [25] for RGB-thermal experiments that include both from-scratch training and Waymo→FLIR transfer/fine-tuning. Following the data preparation used for the Waymo experiments, we keep the full segment set but subsample the synchronized stream to 2 Hz, with camera-view supervision attached wherever labels are available for the retained frames and probe caches. For nuScenes, the camera-view segmentation targets are generated by projecting the official lidarseg point labels into the image plane, so these masks are inherently sparser and noisier than the denser Waymo camera-view labels. For the patch-based 3D detection probes, Waymo annotations are remapped to the three foreground classes used by our probe export: car, pedestrian, and cyclist. Across datasets, the reported supervision covers 3D boxes, depth, and segmentation on Waymo; 3D boxes, depth, and projected segmentation on nuScenes; and 2D detection on FLIR, which does

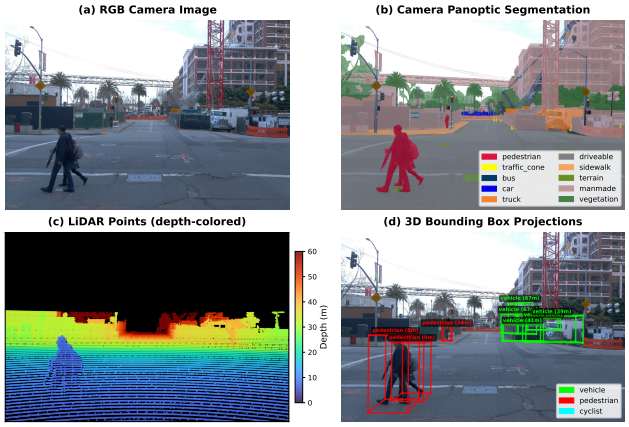

Figure 2. **Waymo dataset showcase.** Example synchronized supervision used in our experiments: (a) RGB image, (b) camera-view segmentation, (c) the aligned companion-modality signal shown in depth form for the driving setting, and (d) projected 3D bounding boxes.

not provide segmentation labels. Figure 2 shows the synchronized RGB, dense companion-modality, and projected-box views used throughout the Waymo experiments.

Implementation Details. Unless noted otherwise, all main experiments use a ViT-Small/16 backbone [12] implemented with `timm` [28] and batch size 64. Waymo and nuScenes scratch runs use 5 SSL epochs followed by 5 probe-training epochs. During self-supervised pretraining, the encoder sees 224×224 global crops and 96×96 local crops; frozen-probe evaluation instead uses deterministic clean probe views at 640×640. For from-scratch FLIR, we use a longer 20-epoch schedule; the Waymo→FLIR frozen-transfer block keeps the pretrained encoder fixed, and the end-to-end FLIR fine-tuning block uses the separate schedule summarized in the supplementary material. In practice, both the SSL loss and the frozen-probe validation curves largely stabilized within this budget, so we keep the shorter schedule rather than stretching all runs to much longer image-only SSL recipes. The default Le MuMo JEPA model is the pruned fusion-token encoder trained with SIGReg on the joint multimodal CLS embedding. Baseline-specific settings are deferred to the supplementary material.

Patch Probes. The main paper reports only patch-level probes. These probes are trained on frozen patch embeddings and include a CenterNet-style 3D detection head [30], dense depth prediction, semantic segmentation, and 2D detection heads. They should therefore be read as representation-quality tests of the learned spatial embedding geometry rather than as end-to-end finetuned perception-system numbers. In the frozen-probe benchmarks, every SSL encoder is evaluated with the same probe architecture and optimization protocol, which isolates encoder quality rather than detector-specific tuning; the FLIR fine-tuning rows later in this section provide a separate practical check that the same pretrained encoder also improves end-to-end downstream performance. For dual-stream multimodal baselines, the reported benchmarks use the camera-aligned RGB patch readout so that the downstream probe interface and dimensionality remain fixed across methods. In pilot ablations, simple post-hoc patch fusion strategies such as concatenation, averaging, and learned projection severely destabilized segmentation transfer, while CenterNet and high-resolution depth results changed only marginally. We therefore report CenterNet-style 3D detection together with Depth MAE and Seg. mIoU on Waymo, the same patch-level metrics on nuScenes, and 2D detection on FLIR. Detailed probe implementation notes are provided in the supplementary material.

Metric Definitions. All main detection numbers come from the CenterNet-style probe. We report XY-match mAP and XZ-match mAP on Waymo and nuScenes, corresponding to Bird's-Eye-View XY-plane and elevation XZ-plane matching respectively, together with Depth MAE and Seg. mIoU where available, and mAP50 and Car mAP50 on FLIR; the exact metric definitions are given in the supplementary material.

Baselines and Run Selection. For the Waymo and nuScenes patch-probe comparisons, all encoders in the main tables are trained from scratch within the target dataset and evaluated under the same frozen-probe protocol. We compare against LeJEPA and LiDAR-only JEPA controls, a DINOv3-style RGB baseline [23], MultiMAE and ImageBind multimodal baselines [2, 13], and simple early- and late-fusion ablations implemented in the same training pipeline. We exclude methods that rely on native 3D sparse-convolution or point-cloud backbones [8, 22, 26, 29] so the comparison remains apples-to-apples among unified 2D transformer encoders. For dual-stream multimodal baselines, this protocol keeps the downstream probe architecture fixed by using a camera-aligned patch readout rather than introducing a method-specific post-hoc fusion block only at evaluation time. Foundation-model baselines such as ImageBind and MultiMAE are retrained strictly from scratch for the corresponding benchmark rather than evaluated as off-the-shelf pretrained models. FLIR is the only section that additionally reports Waymo-initialized transfer and fine-tuning. ImageBind serves as the RGB-depth dual-encoder contrastive baseline, while MultiMAE appears as self-supervised and multitask reconstruction variants. This keeps the comparison tied to objective families under identical data and compute constraints.

Table 1. **Self-supervised single-modality encoders vs. encoder-frozen controls on Waymo.** All rows train the probes. Rows labeled "encoder frozen" keep a randomly initialized encoder fixed, whereas the other rows train the encoder self-supervised before frozen-probe evaluation. We report XY mAP, Depth MAE, and Seg. mIoU.

Method	mAP XY ↑	Depth MAE ↓	Seg. mIoU ↑
LeJEPA [4]	**19.3**	4.704	**0.261**
LeJEPA encoder frozen [4]	13.8	6.734	0.126
LiDAR-only	15.4	**2.982**	0.151
LiDAR-only encoder frozen	8.0	5.397	0.089

Table 2. **Focused fusion comparison on Waymo for the fusion variants.** We compare the default Le MuMo JEPA setting against early and late fusion, persistent routing, VICReg, and the 3-pass SIGReg variant.

Method	mAP XY ↑	Depth MAE ↓	Seg. mIoU ↑
Early Fusion RGBD	18.1	4.767	0.248
Late Fusion	18.7	4.802	0.251
FT-Pruned + VICReg [6]	22.8	2.911	0.248
FT-Persistent + SIGReg	23.1	2.846	0.271
FT-Pruned + SIGReg (3-pass)	23.2	**2.777**	0.274
Le MuMo JEPA	**23.6**	2.860	**0.275**

4.2. Patch-Level Results

Table 1 compares self-supervised single-modality encoders against their randomly initialized, encoder-frozen counterparts. Allowing the encoder to train under the LeJEPA objective substantially improves both modalities over keeping the encoder frozen at random initialization, which shows the value of learning the representation rather than relying only on the downstream probe. The difference between RGB and LiDAR is also expected: RGB is stronger on localization and segmentation cues, whereas LiDAR is naturally stronger on depth reconstruction.

Table 2 then isolates the architectural comparison. All three token-based fusion variants are much better than early or late fusion on depth. The key takeaway from this table is that the default Le MuMo JEPA configuration is the strongest overall row: it gives the best XY mAP and Seg. mIoU while staying close to the best depth row and ahead of the persistent variant on the overall accuracy-efficiency balance. We attribute this to an explicit information bottleneck: because the modality-specific tokens disappear after the first layer, the model is forced to compress the useful cross-modal evidence into the shared fusion-token grid early, whereas persistent routing can spend capacity on repeatedly revisiting redundant paired tokens. The smaller but consistent gap between FT-Pruned + VICReg and the default SIGReg row suggests that the isotropic Gaussian target is also a better regularizing anchor for the joint multimodal CLS embedding than variance-and-covariance matching alone, likely because it more directly discourages modality-specific anisotropy in the shared latent space.

Table 3 broadens the comparison to the full baseline set. This is the paper's primary patch-probe benchmark. Because Table 3 compares encoders that are all trained from scratch on the same Waymo setup, the ranking is easier to interpret: Le MuMo JEPA achieves the best performance across all four spatial metrics in this comparison. These absolute detection values are intentionally conservative because they come from frozen patch probes on top of fixed self-supervised features rather than from end-to-end detector fine-tuning. The takeaway from this table is that the multimodal JEPA objective yields the strongest overall object-centric representation in the comparison.

MultiMAE-MT remains the strongest reconstruction-style reference on Waymo, which is consistent with the extra segmentation supervision available in that pretraining setup, while ImageBind remains a useful contrastive multimodal control but does not lead any of the main Waymo columns. That the multimodal foundation baselines do not clearly surpass the RGB-only LeJEPA control likely reflects the higher data demands of contrastive and masked-reconstruction objectives under strictly from-scratch training on our Waymo subset. LiDAR-only and the DINOv3-style RGB baseline both remain clearly below Le MuMo JEPA, indicating that neither single-modality geometry nor stronger RGB-only pretraining is sufficient to match joint representation learning.

4.3. nuScenes Results

Table 4 shows that the default model remains the strongest row when trained from scratch on nuScenes. The same fusion-token design that wins on Waymo also gives the best XY-match mAP, the best XZ-match mAP, the strongest segmentation score, and by far the best depth error on nuScenes. The extra segmentation supervision used by MultiMAE-MT is less explicit here than on Waymo, which is unsurprising given that the nuScenes camera-view segmentation targets are projected from lidarseg labels and therefore noisier than the denser Waymo masks.

4.4. FLIR Results

Table 5 exposes two useful patterns. First, the Waymo-pretrained Le MuMo JEPA encoder transfers more effectively to FLIR than the other multimodal baselines without any FLIR fine-tuning. Second, the gap widens further after fine-tuning: Le MuMo JEPA reaches the best 2D detection transfer by a clear margin, while training from scratch on FLIR remains weak and noisy for all methods, likely because FLIR is much smaller than Waymo.

Table 3. **Patch-probe comparison on Waymo with all encoders trained from scratch.** CenterNet and dense prediction heads are trained on frozen patch embeddings taken from models that were all self-supervised from scratch on Waymo. "Training Data" denotes the modalities seen during SSL pretraining; for dual-stream baselines, the frozen patch probes use the camera-aligned RGB patch readout described in Section 4.1. Best results are in **bold**; second best are underlined. ↑ means higher is better and ↓ means lower is better.

Method	Training Data	mAP XY ↑	Depth MAE ↓	mAP XZ ↑	Seg. mIoU ↑
LeJEPA [4]	RGB	19.3	4.704	4.9	0.261
DINOv3 [23]	RGB	15.2	5.314	3.5	0.239
LiDAR-only	Depth	15.4	2.982	4.0	0.151
MultiMAE-SS [2]	RGB+Depth	13.5	4.441	2.7	0.221
MultiMAE-MT [2]	RGB+Depth	13.7	3.583	2.9	0.262
ImageBind [13]	RGB+Depth	13.4	4.309	3.0	0.243
Le MuMo JEPA	RGB+Depth	**23.6**	**2.860**	**7.2**	**0.275**

Table 4. **Results on the nuScenes dataset.** We report CenterNet XY-match mAP, XZ-match mAP, semantic segmentation, and Depth MAE.

Method	mAP XY ↑	mAP XZ ↑	Seg. mIoU ↑	Depth MAE ↓
MultiMAE-SS [2]	6.95	1.66	0.195	5.736
MultiMAE-MT [2]	6.67	1.62	0.192	5.624
ImageBind [13]	6.86	1.57	0.198	5.912
Le MuMo JEPA	**9.52**	**2.53**	**0.228**	**2.031**

Table 5. **FLIR results across training regimes.** We report 2D CenterNet mAP50 and Car mAP50. Rows are ordered from training from scratch on FLIR, to Waymo→FLIR transfer, to Waymo-pretrained fine-tuning on FLIR. Because FLIR does not provide segmentation labels, the scratch block excludes MultiMAE-MT.

Setting	Method	mAP50 ↑	Car mAP50 ↑
Scratch	MultiMAE-SS [2]	**0.51**	**3.86**
Scratch	ImageBind [13]	0.24	1.90
Scratch	Le MuMo JEPA	0.41	3.00
Waymo→FLIR	MultiMAE-SS [2]	0.54	4.21
Waymo→FLIR	MultiMAE-MT [2]	0.69	5.26
Waymo→FLIR	ImageBind [13]	0.72	5.49
Waymo→FLIR	Le MuMo JEPA	**1.56**	**10.22**
Fine-tune	MultiMAE-SS [2]	0.63	4.81
Fine-tune	MultiMAE-MT [2]	0.75	5.83
Fine-tune	ImageBind [13]	0.71	5.54
Fine-tune	Le MuMo JEPA	**2.39**	**12.88**

4.5. Fusion and Loss Ablations Through Compute Profiling

Table 6 makes the practical trade-off explicit. Relative to the 3-pass row, the default model cuts the estimated training time by almost 3× while also reducing peak VRAM and total FLOPs. Persistent paired fusion remains the expensive edge of the family even under the same joint-CLS loss because it retains paired cross-modal token interactions throughout all transformer layers instead of pruning the modality-specific tokens after the first layer; that deeper routing increases both activations and attention-state memory, and its modest depth gain does not offset the extra cost.

Table 6. **Encoder-side compute profile.** Rows are sorted by estimated total encoder training time. We report time in minutes, total encoder SSL FLOPs in PFLOPs, and peak reserved VRAM in GB. Time is estimated for the encoder-side workload on an H200 system with an AMD EPYC 9275F 24-Core Processor. The fusion-token rows use the default single-pass joint-CLS objective, with the 3-pass SIGReg variant shown separately as an ablation.

Method	Time (min) ↓	SSL FLOPs (PFLOPs) ↓	VRAM (GB) ↓
MultiMAE-SS [2]	**14.07**	**21.4**	**2.94**
Early Fusion RGBD	18.70	72.9	6.15
LeJEPA [4]	18.73	72.7	6.01
ImageBind [13]	23.65	80.4	9.49
Late Fusion	34.72	144.9	11.41
DINOv3 [23]	37.72	108.8	8.70
Le MuMo JEPA	55.09	86.0	12.11
FT-Pruned + VICReg	59.41	84.8	12.03
MultiMAE-MT [2]	63.49	56.9	7.45
FT-Pruned + SIGReg (3-pass)	165.98	258.0	33.98
FT-Persistent + SIGReg	170.63	218.0	36.08

The broader compute table also helps interpret the baselines. Early and late fusion are much cheaper than token-based multimodal routing, but Table 2 shows that the cheaper designs leave clear detection and depth performance on the table. The default Le MuMo JEPA row remains the best-performing point inside the stronger token-fusion family before the much more expensive 3-pass and persistent variants.

4.6. Qualitative Results

Figure 3 provides an embedding-space view that is consistent with the patch-probe tables: the fusion-token representation is more organized than the single-modality baseline when semantic grouping and geometry must be resolved jointly. In the same 2D t-SNE space, a 10-NN class-purity score also improves from 0.453 for the RGB-only LeJEPA baseline to 0.514 for Le MuMo JEPA, against random same-class baselines of 0.215 in both cases. That depth-oriented structure is not only visual: fitting a simple planar trend in the 2D t-SNE space of the plotted depth gradient gives an R^2 score of 0.463 for Le MuMo JEPA versus

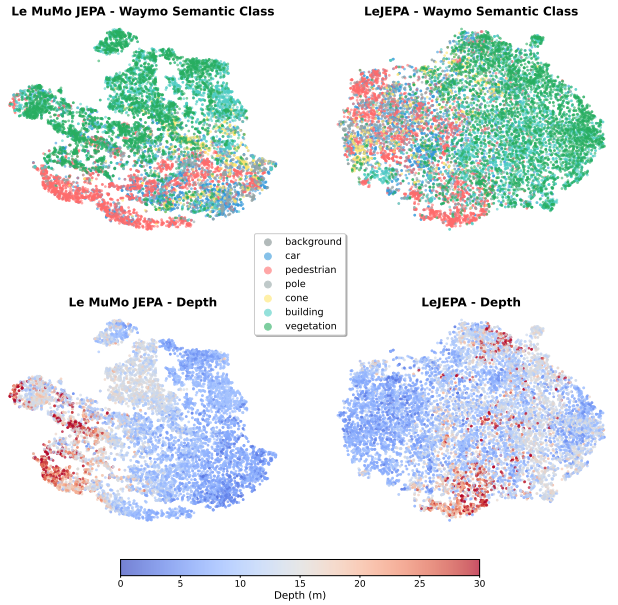

Figure 3. **Waymo patch-embedding visualization.** t-SNE projections of final-layer patch embeddings are shown with class-oriented structure and depth-oriented structure, illustrating how the learned patch space organizes both semantic grouping and geometric variation across methods. The left column uses Le MuMo JEPA fusion-token embeddings, and the right column uses LeJEPA patch embeddings. A planar fit to the plotted depth gradient gives an R^2 score of 0.463 for Le MuMo JEPA versus 0.086 for LeJEPA.

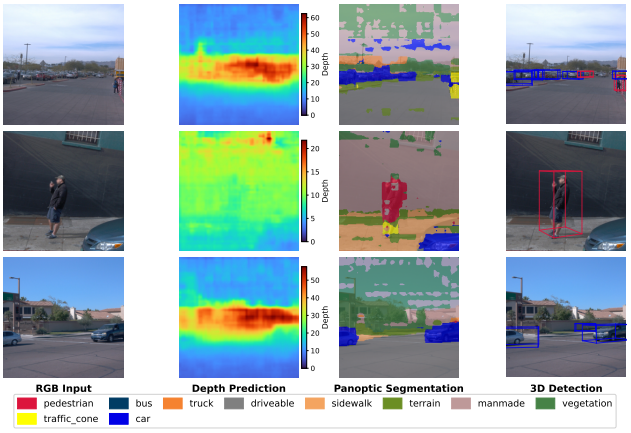

Figure 4. **Waymo qualitative probe output.** The figure shows the RGB input together with predictions from the three probe families used in the paper: dense depth estimation, segmentation, and 3D detection boxes. It provides a direct qualitative view of the same patch-level capabilities summarized by the quantitative probe tables.

0.086 for the RGB-only LeJEPA baseline.

Figure 4 complements the tables with direct Waymo prediction-level examples: the 3D boxes stay centered on vehicles, the depth prediction preserves the large layout changes between road and nearby objects, and the segmentation output maintains coherent road and sidewalk regions.

5. Conclusion

We presented Le MuMo JEPA, a multi-modal self-supervised framework that extends LeJEPA to jointly learn from RGB and aligned companion modalities. By combining camera-aligned inputs, learnable fusion tokens, and SIGReg, Le MuMo JEPA enables structured multimodal representation learning in a shared token space without labeled pretraining data.

Our experiments show that the default Le MuMo JEPA configuration is now the strongest overall configuration in the paper. On Waymo it leads the patch-probe benchmark, on nuScenes it also gives the best from-scratch row, and on FLIR it performs best under Waymo-initialized transfer and fine-tuning. Among the fusion-token variants, the pruned default model also offers the best accuracy-efficiency balance while remaining far more practical than persistent routing in FLOPs, VRAM, and estimated training time.

More broadly, these results suggest that efficient shared-token multimodal pretraining can become a practical foundation for future autonomous driving systems that must integrate heterogeneous sensors without relying on expensive label-heavy supervision at every stage.

5.1. Limitations and Broader Impacts

Our evaluation is limited to frozen probes, a small dataset suite, and ViT-Small/16-scale backbones, and the current encoder relies on camera-aligned depth maps that simplify fusion but discard part of the native 3D structure. The present formulation also benefits from reasonably aligned cross-modal observations, so robustness to calibration error, missing overlap, or weaker image-plane correspondence is not yet established. Future work should pair the fusion-token design with sparse-voxel or point-cloud backbones, expand the nuScenes evaluation beyond detection, and test larger modality families beyond RGB-depth and RGB-thermal. It should also verify how the learned fusion tokens scale to larger backbones. Stronger multimodal pretraining could reduce annotation cost and improve robustness across sensing conditions, but deployment-facing claims would still require much broader safety validation and failure analysis than this probe-based study provides.

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15619–15629, 2023. 1, 2

[2] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal multi-task masked autoencoders. In *Eur. Conf. Comput. Vis.*, pages 348–367, 2022. 2, 3, 5, 7

[3] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. TransFusion: Robust LiDAR-camera fusion for 3D object detection with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1090–1099, 2022. 1, 3

[4] Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. 1, 2, 3, 6, 7

[5] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *Int. Conf. Learn. Represent.*, 2022. 2

[6] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *Int. Conf. Learn. Represent.*, 2022. 2, 6

[7] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024. 2

[8] Alexandre Boulch, Corentin Sautier, Björn Michele, Gilles Puy, and Renaud Marlet. ALSO: Automotive LiDAR self-supervision by occupancy estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13455–13465, 2023. 3, 5

[9] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11621–11631, 2020. 2, 4

[10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Int. Conf. Comput. Vis.*, pages 9650–9660, 2021. 1

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Int. Conf. Mach. Learn.*, pages 1597–1607, 2020. 2

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021. 2, 3, 5

[13] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One embedding space to bind them all. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15180–15190, 2023. 2, 3, 5, 7

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent – a new approach to self-supervised learning. In *Adv. Neural Inform. Process. Syst.*, pages 21271–21284, 2020. 1, 2

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9729–9738, 2020. 2

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16000–16009, 2022. 1, 2

[17] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver IO: A general architecture for structured inputs & outputs. In *Int. Conf. Learn. Represent.*, 2022. 2, 3

[18] Yann LeCun. A path towards autonomous machine intelligence. *openreview.net*, 2022. 2

[19] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. BEVFusion: A simple and robust LiDAR-camera fusion framework. *Adv. Neural Inform. Process. Syst.*, 35: 10421–10434, 2022. 3

[20] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *Int. Conf. Robot. Autom.*, pages 2774–2781, 2023. 1, 3

[21] Su Pang, Daniel Morris, and Hayder Radha. CLOCs: Camera-LiDAR object candidates fusion for 3D object detection. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 10386–10393, 2020. 2

[22] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-LiDAR self-supervised distillation for autonomous driving data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9891–9901, 2022. 3, 5

[23] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. 2, 5, 7

[24] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo Open Dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2446–2454, 2020. 2, 4

[25] Teledyne FLIR LLC. Teledyne flir free adas thermal dataset v2. https://adas-dataset-v2.flirconservator.com/, 2022. Accessed: 2026-03-19. 1, 2, 4

[26] Xiaoyu Tian, Haoxi Ran, Yue Wang, and Hang Zhao. GeoMAE: Masked geometric target prediction for self-supervised point cloud pre-training. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13570–13579, 2023. 3, 5

[27] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. PointPainting: Sequential fusion for 3D object de-

tection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4604–4612, 2020. 2

[28] Ross Wightman. PyTorch image models. *GitHub repository*, 2019. https://github.com/rwightman/pytorch-image-models. 5

[29] Honghui Yang, Sha Zhang, Di Huang, Xiaoyang Wu, Haoyi Zhu, Tong He, Shixiang Tang, Hengshuang Zhao, Qibo Qiu, Binbin Lin, Xiaofei He, and Wanli Ouyang. UniPAD: A universal pre-training paradigm for autonomous driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14746–14757, 2024. 1, 3, 5

[30] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2, 5

Le MuMo JEPA: Multi-Modal Self-Supervised Representation Learning with Learnable Fusion Tokens

Supplementary Material

S1. Baseline Implementation Details

This supplement records the implementation choices behind the main baselines used in Section 4. The goal is not to reproduce every training flag inline in the main paper, but to make clear what each comparison represents and which prior work it is adapting.

S2. Detailed SIGReg Formulation

In the multimodal implementation used for Le MuMo JEPA, the target center is computed from the global views and the penalty is applied to all available views:

$$\bar{\mathbf{z}} = \frac{1}{V_g} \sum_{v=1}^{V_g} \mathbf{z}_v^g,$$
$$\mathcal{L}_{\text{inv}} = \frac{1}{V_g + V_\ell} \left(\sum_{v=1}^{V_g} \|\mathbf{z}_v^g - \bar{\mathbf{z}}\|^2 + \sum_{u=1}^{V_\ell} \|\mathbf{z}_u^\ell - \bar{\mathbf{z}}\|^2 \right). \tag{S1}$$

SIGReg then matches the empirical embedding distribution to $\mathcal{N}(0, \mathbf{I})$ by projecting embeddings onto K random directions and comparing the empirical characteristic function of those projections against the Gaussian target at T evaluation knots:

$$\hat{c}_{k,j} = \frac{1}{B} \sum_{n=1}^{B} \cos\left(t_j \mathbf{w}_k^\top \mathbf{z}_n\right),$$
$$\hat{s}_{k,j} = \frac{1}{B} \sum_{n=1}^{B} \sin\left(t_j \mathbf{w}_k^\top \mathbf{z}_n\right),$$
$$\mathcal{L}_{\text{SIGReg}}(\mathbf{Z}) = \frac{1}{K} \sum_{k=1}^{K} \sum_{j=1}^{T} \omega_j \left[\left(\hat{c}_{k,j} - e^{-t_j^2/2} \right)^2 + \hat{s}_{k,j}^2 \right]. \tag{S2}$$

These are the full expressions summarized more briefly in the main paper.

Shared training defaults. For the Waymo experiments used in the paper, the shared defaults are a ViT-Small backbone, batch size 64, 5 self-supervised training epochs, $V = 2$ global crops, 8 local crops, projection dimension 16, learning rate 10^{-4}, and $\lambda = 0.1$ for SIGReg. The self-supervised encoder is trained with 224×224 global crops and 96×96 local crops, whereas frozen-probe evaluation uses deterministic clean probe views at 640×640. After encoder pretraining, the encoder is frozen and the probes are trained for 5 additional epochs, with validation every 100 steps on the full validation split. Patch probes remain enabled during evaluation, and the occupancy IoU uses a neutral empty-union policy. Unless noted otherwise, all baselines reuse the same data filtering, camera-view supervision, probe heads, validation cadence, and run-selection logic as the main method so that the comparison changes the representation learner rather than the downstream evaluation stack.

Modality-specific augmentations. The RGB stream receives the appearance-level augmentations listed in the main paper, namely ColorJitter, RandomGrayscale, GaussianBlur, and, in the official DINO-style branch, RandomSolarize. The companion modality does *not* receive those photometric perturbations. For aligned RGB-depth training, the crop rectangle is sampled once and applied to both RGB and depth, and a single horizontal-flip decision is shared between the two streams; after that synchronized spatial step, RGB receives the photometric augmentation stack while the depth map only undergoes resizing/pooling, dtype conversion, and the shared flip. For FLIR, the RGB and thermal crops are likewise sampled in a synchronized way and optionally flipped together, but the thermal branch uses only image conversion, float casting, and 1-channel normalization, while the RGB branch receives the JEPA/DINO-style photometric transforms. This preserves pixel alignment across modalities without applying color-style perturbations to depth or thermal inputs, where they would not have a physical interpretation.

Waymo subset construction. The Waymo setup used throughout the paper is defined by the shared data-preparation pipeline rather than by manual scene curation. For the reported runs, we keep the available segments for the selected split and subsample the synchronized stream from the native 10 Hz capture rate to 2 Hz, i.e., every fifth frame, exactly as described in the main paper. The concrete export used by a run records the resulting segment and frame counts in the generated metadata, so the subset definition is deterministic even though those counts are not repeated inline in every table.

Tuning policy. We do not run a separate downstream hyperparameter search for each baseline. The frozen-probe stage, deterministic probe views, validation cadence, run-selection rule, and probe heads are shared across methods,

while baseline-specific changes are limited to objective-intrinsic settings such as DINO temperatures, InfoNCE temperature, or MultiMAE mask ratio and decoder size. The goal is therefore to compare representation learners under the same short from-scratch budget rather than to maximize each baseline with method-specific probe engineering.

Single-modality JEPA baselines. RGB-only and **LiDAR-only** share the same basic encoder design, with the only architectural change being the input channel count: RGB uses a 3-channel input, whereas LiDAR depth uses a 1-channel input. The paired **RGB-only frozen** and **LiDAR-only frozen** settings keep the encoder fixed and train only the probe heads. These baselines therefore isolate cross-modal fusion from two different confounds at once: modality choice and encoder adaptation. In particular, the trainable single-modality rows test whether the gains in the main paper come merely from stronger encoder optimization, whereas the frozen rows test how much downstream performance is available without any encoder-side adaptation at all.

Early and late fusion. Early Fusion RGBD uses a single encoder over stacked RGB and aligned depth channels. **Late Fusion** uses separate modality encoders whose features are concatenated before probing. These two settings are simple in-house architectural ablations within our training pipeline rather than direct reimplementations of specific prior supervised fusion systems. Both baselines share the same Waymo data pipeline, probe heads, and compute profiling code as Le MuMo JEPA. They are included to separate the benefit of multimodal data itself from the benefit of the learnable fusion-token bottleneck: early fusion tests whether naive channel stacking is sufficient, and late fusion tests whether keeping the modalities separate until readout is already enough.

DINOv3-style baseline. The DINO baseline used in the main table is a scratch-trained RGB model with a DINOv3-style training objective rather than a frozen pretrained encoder. Its main hyperparameters are a DINO learning rate of 5×10^{-4}, prototype dimension 1024, iBOT output dimension 1024, teacher temperature 0.04, teacher warmup start 0.04, zero DINO warmup epochs, and zero frozen-last-layer epochs. The training setup additionally uses student temperature 0.1, teacher momentum 0.996, center momentum 0.9, and AdamW betas $(0.9, 0.95)$. These are not intended to reproduce the official DINOv3 recipe exactly. Instead, they are a tuned in-project configuration chosen to learn faster and remain competitive under the shorter from-scratch budget used throughout this paper. This makes it a stronger RGB-only architectural baseline than plain JEPA, without introducing LiDAR or explicit multi-modal fusion. Under the short from-scratch budget used in this paper, however, it still underperforms the simpler RGB-only LeJEPA control in the main table. Its role in the paper is to test whether a stronger modern RGB-only SSL objective can close the gap to multimodal learning when both are trained under the same from-scratch protocol.

ImageBind-style baseline. The **ImageBind** baseline uses paired RGB-depth encoders trained with a symmetric InfoNCE objective at temperature 0.07. This configuration disables local crops, uses clean probe views, runs at batch size 64 in the current rerun, and evaluates probes at high image resolution. For the comparison reported in the main table, this baseline is trained in the same project pipeline as the other methods rather than being treated as a frozen pretrained encoder. It therefore appears in both the main accuracy table and the compute table as a trainable multimodal baseline. Conceptually, this row is the contrastive multimodal reference in the paper: two modality-specific encoders are aligned through paired-view InfoNCE rather than through predictive fusion tokens and JEPA-style prediction. This is important for interpretation because it keeps the multimodal setting but changes the learning principle from predictive regularized representation learning to pairwise contrastive alignment.

MultiMAE baselines. MultiMAE-SS and **MultiMAE-MT** use a mask ratio of 0.75, decoder depth 2, and decoder width 256. These variants disable local crops because the decoder expects a fixed global patch grid. **MultiMAE-SS** is the self-supervised variant: it reconstructs multimodal RGB-depth content without using segmentation labels during representation learning. **MultiMAE-MT** is the multitask variant: it keeps the same reconstruction backbone but additionally enables semantic supervision through the auxiliary labels that the Waymo pipeline already prepares. These baselines are therefore intentionally stronger dense-prediction references than the simpler early- and late-fusion designs. They serve as reconstruction-style multimodal baselines whose inductive bias is closer to masked modeling than to either contrastive alignment or JEPA-style latent prediction.

Fusion-token ablations. The compute table in the main paper includes **FT-Pruned**, **FT-Pruned + VICReg**, **FT-Persistent**, and **FT-Pruned + SIGReg (3-pass)** in addition to the default **Le MuMo JEPA** model, which corresponds to the FT-Pruned SIGReg setting used throughout the main comparison tables. These scenarios all share the same fusion-token encoder family and differ mainly in their routing strategy and learning objective. In the paper, the synchronized accuracy comparison focuses on the main pruned model, the VICReg variant, the persistent-routing

variant, and the 3-pass objective, while the broader compute table shows the cost of different token-routing choices. The ablations are intended to answer two separate questions: whether explicit token routing matters relative to simpler fusion, and whether the gains are specific to SIGReg on the joint embedding rather than to the encoder family alone.

Three-pass SIGReg ablation. The **FT-Pruned + SIGReg (3-pass)** row uses the same pruned fusion-token encoder as the default model, but it evaluates SIGReg on three forward passes instead of only the joint fused pass. For each paired sample, it computes: (i) a joint RGB+companion-modality pass, (ii) an RGB-only pass with the companion modality zeroed out, and (iii) a companion-modality-only pass with RGB zeroed out. If $\mathbf{Z}^{(\text{joint})}$, $\mathbf{Z}^{(\text{rgb})}$, and $\mathbf{Z}^{(\text{mod})}$ denote the projected CLS embeddings from those three passes, then the added three-pass regularizer is

$$\mathcal{L}_{\text{SIGReg}}^{(\text{3-pass})} = \frac{1}{3}\Big[\mathcal{L}_{\text{SIGReg}}(\mathbf{Z}^{(\text{joint})}) + \mathcal{L}_{\text{SIGReg}}(\mathbf{Z}^{(\text{rgb})}) + \mathcal{L}_{\text{SIGReg}}(\mathbf{Z}^{(\text{mod})})\Big]. \tag{S3}$$

The joint pass is reused from the main objective, so the extra compute comes primarily from the two masked single-modality forwards. This is the ablation referred to as "3-pass" in the Waymo and compute tables.

Compute-profile reporting. The compute table in the main paper is intentionally encoder-side only. Its time column is the estimated encoder-training runtime from the shared logging pipeline on an H200 system with an AMD EPYC 9275F 24-Core Processor, and the FLOP and VRAM columns come from the same training logs via the profiled encoder SSL FLOPs and peak reserved GPU memory fields. These numbers are meant to compare representation-learning cost under a common training stack; they do not include offline dataset preparation or the separate frozen-probe training stage.

Dataset-specific training details. nuScenes from scratch. These runs retrain the encoder directly on nuScenes for 5 SSL epochs and then train the frozen probes for 5 epochs with batch size 64, mirroring the short Waymo schedule under the same clean-view probe setup.

Waymo→FLIR frozen transfer. These runs use the dedicated probe-evaluation configuration rather than the SSL pretraining loop: the pretrained Waymo encoder is frozen, probe-only training is enabled, $V = 1$ and local crops are disabled, and only the downstream heads are optimized for 5 epochs with batch size 64. The frozen-transfer block uses learning rate 10^{-4}, probe learning rate 10^{-3}, patch-probe learning rate 10^{-3}, probe resolution 640×640, validation every 100 steps, and no LiDAR or copy-paste augmentation.

FLIR from scratch. The scratch FLIR rows use the longer 20-epoch FLIR schedule before the downstream evaluation stage, reflecting the smaller size of the RGB-thermal dataset.

End-to-end FLIR fine-tuning. The FLIR fine-tuning rows use the separate detection fine-tuning configuration with batch size 64, 30 epochs, encoder learning rate 2×10^{-5}, decoder learning rate 2×10^{-4}, AdamW with weight decay 0.05, validation every 100 steps, early-stopping patience 8, and random-resized-crop scale $[0.8, 1.0]$ in train mode. The optimizer uses a 5% linear warmup starting at 1% of the target learning rate, followed by cosine decay to 10^{-7}. For FLIR, this configuration uses the 2D CenterNet-style detection path, a 3-layer decoder with hidden width 512, an auxiliary global-view size of 224, and a clean probe/evaluation view of 640×640.

S3. Probe Implementation Details

This section focuses on the patch probes reported in the main paper. All probes are trained after self-supervised pretraining with the encoder frozen, so the tables should be read as representation-quality measurements rather than as full end-to-end finetuning results. The probe stage always uses the same deterministic 640×640 camera-view inputs so that differences in downstream numbers reflect the learned representation and not stochastic crop variation at evaluation time.

Patch probes. The patch-probe family contains a CenterNet-style 3D detection head with $2\times$ upsampling, segmentation heads, a dense depth-map probe with $4\times$ upsampling, and an occupancy-map probe. All of these heads are shallow readouts rather than standalone perception backbones. The stronger CenterNet-style 3D head keeps the spatial token grid, applies a 3×3 $\text{Conv}(\text{vit\_dim}, 256)$ adapter with batch normalization and ReLU, upsamples the 14×14 grid to 28×28 with a transposed convolution, and then uses separate 1×1 heads for heatmap, offset, size, depth, and yaw. The segmentation probe is strictly linear: a single 1×1 convolution projects to $C\,r^2$ channels, PixelShuffle upsamples by $r = 4$, and bilinear interpolation resizes the output to 224×224. The depth-map probe is still lightweight but not purely linear: it uses a 1×1 projection to $16\,r^2$ channels, PixelShuffle with $r = 4$, then a 3×3 refinement convolution and a final 1×1 depth head. The occupancy-map probe uses a 1×1 projection branch, a 1×1 skip branch, PixelShuffle with $r = 2$, two small 3×3 GroupNorm+GELU refinement blocks, and a final 1×1 prediction head. Its loss is BCE-with-logits with focal reweighting, plus a Dice term and a small consistency term for multi-channel outputs. For FLIR, the box-segmentation

head reuses the same linear SemanticSegProbe template at occupancy-grid resolution, and the 2D detection head is the 2D analogue of the same CenterNet-style spatial readout. For Waymo, the main paper reports the stronger CenterNet-style detector together with Depth MAE and Seg. mIoU. The detector probes operate on frozen spatial features rather than on the global CLS token, which is why they are a better test of whether the learned representation preserves object layout, localization cues, and cross-modal geometry.

Patch-readout protocol for dual-stream baselines. For dual-stream multimodal baselines such as ImageBind and MultiMAE, the reported patch-probe benchmarks use a fixed camera-aligned patch readout rather than an additional post-hoc probe-time fusion module. The reason is empirical rather than purely cosmetic: in pilot ablations, simple post-hoc multimodal patch fusion choices such as concatenation, averaging, and learned projection severely destabilized segmentation transfer, even when the higher-resolution depth probe and the CenterNet probe were less affected. We therefore treat those probe-time fusion variants as unstable evaluation choices in the current frozen-feature setting and keep the main comparison focused on a matched probe interface rather than on method-specific readout engineering.

Metric definitions and correspondence. The main detection numbers come from the CenterNet-style spatial box probe. For Waymo and nuScenes, the detection table columns are read from the CenterNet export as XY-match mAP and XY-match ADE, with XZ-match mAP additionally shown where that aggregate is available in the checked export. These 3D detection mAP values are computed with the shared center-distance AP evaluation used throughout the codebase rather than with an official leaderboard submission script: for each evaluated class, AP is computed under center-distance matching at 0.5, 1, 2, and 4 meters, averaged over those four thresholds, and then averaged over classes. In the Waymo patch-probe setting, the class average is over car, pedestrian, and cyclist; in the nuScenes from-scratch setting, it is over all evaluated classes in the export used for the table. The main depth number is the higher-resolution dense depth-map MAE from the dedicated depth-map probe, reported as Depth MAE in the paper tables. The main segmentation number is reported in the paper as Seg. mIoU. For Waymo, this value comes from the semantic-segmentation probe metric logged in the selected runs, while the nuScenes table uses the corresponding segmentation field from the export used for that table. For FLIR, the main table switches to the available 2D detection outputs, namely CenterNet mAP50 and Car mAP50, because those are the directly comparable detection metrics logged for that dataset; in that setting, mAP50 is the mean AP at IoU 0.5 over all FLIR detection classes and Car mAP50 is the car-specific AP50.