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

Supplementary Material

1. 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.

2. 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:

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

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:

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

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

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

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.

3. 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 Conv(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.

References