RaysUp: Ultra-light Universal Feature Upsampling via Geometry-Aware Ray Representation

Yuchuan Ding*, Linfei Li*, Lin Zhang†, and Ying Shen

School of Computer Science and Technology, Tongji University, Shanghai, China
{2534061,cslinfeili,cslinzhang,yingshen}@tongji.edu.cn

Abstract. Pre-trained Vision Foundation Models (VFMs) have become central to modern computer vision due to their powerful semantic representations and strong generalization ability. However, their patchified or pooled outputs are inherently low-resolution, limiting their effectiveness in tasks requiring fine-grained, pixel-level reasoning. Existing feature upsampling approaches either degrade semantic fidelity or rely on VFM-specific retraining and heavy architectures, hindering efficiency and scalability. To address these challenges, we propose RaysUp, an ultra-lightweight, task-agnostic, and VFM-agnostic feature upsampling framework that reconstructs high-resolution feature maps at arbitrary resolutions. Unlike conventional 2D interpolation or attention-based schemes, RaysUp lifts feature reconstruction into a geometry-aware ray domain. Specifically, we introduce a Spatially Decoupled Guidance Encoder for direction-aware guidance encoding, an Any-Resolution Cross-Attention mechanism for resolution-flexible reconstruction, and a novel Ray Positional Encoding (RayPE) that injects implicit 3D geometric priors via 6D Plücker ray coordinates. Finally, A Geometry-Aware Neighborhood Attention module further ensures content-adaptive bilateral aggregation while preserving geometric consistency. Extensive experiments across diverse dense prediction tasks demonstrate that RaysUp achieves state-of-the-art performance while using only 16% of the parameters of AnyUp and delivering approximately 7× faster inference. These results highlight a substantially improved accuracy–efficiency trade-off and establish RaysUp as a practical and scalable solution for universal feature upsampling. Code is available at https://github.com/MAP-RaysUp/RaysUp.

Keywords: Feature upsampling · Deep features · Ray representation

1 Introduction

Pre-trained Vision Foundation Models (VFMs), such as the DINO family [5, 34, 40], the CLIP series [7, 37, 50], the SigLIP series [44, 53], PE Spatial [3], and MAE [18], have emerged as central components in modern computer vision due

* Equal contributions (Co-first authors).
† Corresponding author.

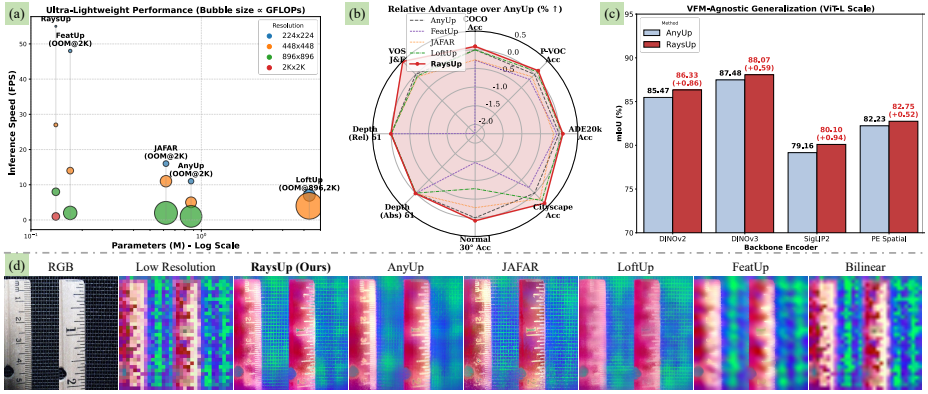

Fig. 1: We propose RaysUp, an (a) ultra-lightweight, (b) task-agnostic, and (c) VFM-agnostic upsampling framework, capable of upsampling backbone features to arbitrary resolutions while preserving (d) high semantic fidelity and geometric consistency.

to their large-scale pretraining on diverse datasets, which endows them with high-level semantic representations and strong generalization capability. These models are capable of extracting transferable and task-agnostic visual features, making them highly suitable as either frozen or fine-tuned backbone networks. Consequently, they have been extensively adopted across a wide range of downstream applications, including depth estimation [6, 28, 51, 52], open-vocabulary semantic segmentation [2, 20, 38, 48], 3D semantic field reconstruction [21, 22, 26, 36], and the evaluation of generative models [1, 49].

However, due to patchification or aggressive pooling operations in VFMs, their output features often have low spatial resolution, limiting their applicability in tasks that require fine-grained, pixel-level understanding. Traditional upsampling methods [12, 32], such as bilinear interpolation, are computationally efficient but often result in the loss of semantic information. More recently, learned upsampling methods, such as LiFT [43], FeatUp [16], LoftUp [19], and JAFAR [9], have demonstrated task-agnostic generalization. However, they require retraining for different VFMs or additional optimization at inference time, which restricts their broader applicability. A more recent approach, AnyUp [46], introduces an encoder-agnostic upsampling framework that generalizes across different VFMs, but its reliance on a heavy network architecture makes it difficult to balance generalization capability with inference efficiency.

To address the aforementioned challenges, we propose RaysUp, an ultra-lightweight, task-agnostic, and encoder-agnostic universal feature upsampling framework that enables feature maps extracted from arbitrary Vision Foundation Models to be reconstructed at any target resolution.

From a methodological perspective, RaysUp builds upon the classical Joint Bilateral Upsampling (JBU) [24] paradigm while introducing structural generalizations. Specifically, instead of directly employing the raw RGB image as guidance, RaysUp adopts a lightweight Spatially Decoupled Guidance Encoder

to generate guidance features. This encoder captures multi-scale spatial semantics and provides more informative structural priors than pixel-level color differences. Conditioned on these guidance representations, RaysUp formulates an Any-Resolution Cross-Attention mechanism, enabling adaptive reconstruction of backbone features at arbitrary output resolutions.

To overcome the limitations imposed by fixed 2D local neighborhoods in JBU and to enhance the spatial kernel's capacity to model underlying geometric structures, we further introduce Ray Positional Encoding (RayPE). Inspired by neural radiance field formulations [33], RayPE encodes guidance features using 6D Plücker ray coordinates, thereby implicitly injecting 3D geometric priors into the attention module. This mechanism generalizes conventional joint bilateral upsampling into a geometry-aware reconstruction process, effectively alleviating boundary blurring and structural drift.

Finally, RaysUp adopts Geometry-Aware Neighborhood Attention as a learnable bilateral aggregation operator, performing localized cross-attention between RayPE-encoded high-resolution guidance features and low-resolution input features. This design preserves the locality property of JBU while enabling content-adaptive learning of upsampling weights, achieving a favorable balance among geometric fidelity, semantic consistency, and computational efficiency.

As illustrated in Fig. 1, owing to our carefully designed architecture, RaysUp enables task-agnostic and VFM-agnostic feature upsampling with geometry-aware ray representations at arbitrary resolutions. Compared to the state-of-the-art method AnyUp, RaysUp achieves superior performance across multiple dense prediction tasks while using only approximately 16% of the parameters. Moreover, it attains an approximately 7× improvement in inference efficiency, substantially reducing both computational and memory overhead, thereby demonstrating an advantageous trade-off between accuracy and efficiency.

In summary, our contributions are as follows:

- We propose RaysUp, an ultra-lightweight, task-agnostic, and VFM-agnostic framework for universal feature upsampling at arbitrary resolutions.
- A lightweight Spatially Decoupled Guidance Encoder is proposed to extract direction-aware spatial semantic guidance features. Building upon this, RaysUp employs an Any-Resolution Cross-Attention mechanism to enable adaptive reconstruction of backbone features at arbitrary resolutions.
- We propose Ray Positional Encoding (RayPE), which injects implicit 3D geometric priors via 6D Plücker ray coordinates, extending classical JBU into a geometry-aware reconstruction process that enhances boundary fidelity and mitigates structural drift.
- Extensive experiments demonstrate that, with only about 1/5 of the parameters of baseline methods, RaysUp achieves state-of-the-art performance across multiple dense prediction tasks and delivers approximately 7× faster inference, significantly reducing both computational and memory costs while balancing accuracy and practical efficiency.

2 Related Work

Feature upsampling aims to recover the spatial resolution of intermediate representations in deep neural networks and serves as a fundamental component in dense prediction tasks such as semantic segmentation [2, 20, 38, 48] and depth estimation [6,28,51,52]. Existing approaches can be broadly categorized into two paradigms: Discrete Reconstruction methods and Continuous Mapping methods.

Discrete Reconstruction. This category of methods primarily relies on interpolation, filtering, or convolutional decoding mechanisms to explicitly reconstruct high-resolution features from low-resolution inputs. Classical interpolation strategies [12,32], such as nearest-neighbor and bilinear, are computationally efficient and flexible across scales; however, they lack content adaptivity and often result in boundary blurring and loss of fine details. To enhance structural preservation, Joint Bilateral Upsampling (JBU) [24] introduces a high-resolution guidance image and performs structure-aware reconstruction through the joint weighting of spatial and range kernels. Subsequently, adaptive filtering principles were incorporated into neural network frameworks. For instance, CARAFE [45] dynamically predicts content-aware reassembly kernels for feature reconstruction; SAPA [31] improves spatial semantic consistency via similarity modeling; and FADE [30] enhances representational capacity by introducing a semi-shift operator into the decoding structure. In addition, IndexNet [29] and A2U [10] have demonstrated strong performance in task-specific scenarios such as image matting. Despite these advances, such approaches typically depend on encoder–decoder architectures and downstream supervision, and are often tailored to specific backbones or fixed upsampling ratios, thereby limiting their generalization capability. Moreover, the classical JBU formulation remains constrained by fixed 2D neighborhoods and RGB-difference-based range kernels, which are insufficient to model complex semantic relationships and underlying geometric structures.

In contrast, RaysUp revisits the classical JBU framework and systematically reformulates its spatial and range kernels to overcome the limitations of fixed 2D neighborhoods and RGB-based range modeling. By introducing a spatially decoupled guidance encoder and an attention-based cross-scale interaction, RaysUp unifies filter-style local aggregation with adaptive feature interaction. This design enables resolution-flexible feature upsampling while improving structural fidelity and semantic expressiveness.

Continuous Mapping. This line of research formulates feature upsampling as a resolution-agnostic continuous function approximation or cross-scale interaction problem, with an emphasis on enhancing representational capacity and scale flexibility. Inspired by implicit neural representations [27, 33], LiFT [43] models image reconstruction as a coordinate-conditioned continuous function, enabling arbitrary-resolution prediction. Building upon this paradigm, FeatUp [16] proposes a task-agnostic feature upsampling framework; however, its high-performance variant relies on per-image optimization, leading to complex training objectives and substantial deployment overhead. More recent approaches, such as JAFAR [9] and LoftUp [19], cast the upsampling process as cross-scale

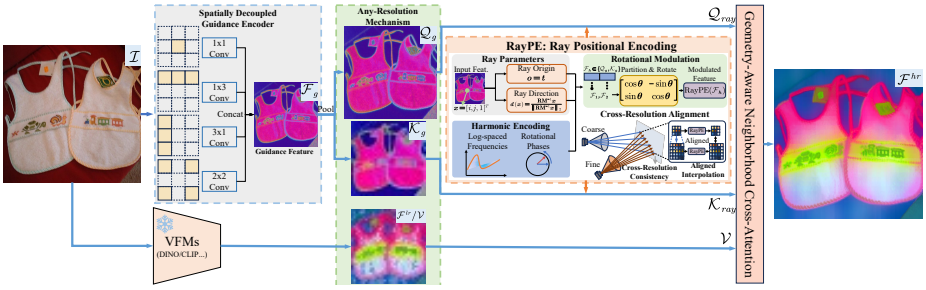

Fig. 2: Overview of RaysUp. Given an image $\mathcal{I}$ and a low-resolution VFM feature map $\mathcal{F}^{lr}$, RaysUp reconstructs a high-resolution feature map $\mathcal{F}^{hr}$ at an arbitrary target resolution. A lightweight Spatially Decoupled Guidance Encoder first extracts direction-aware guidance features $\mathcal{F}_g$, which are adaptively pooled to generate target-resolution queries $\mathcal{Q}_g$ and VFM-resolution keys $\mathcal{K}_g$. After RayPE, the resulting geometry-aware ray representations $\mathcal{Q}_{ray}$ and $\mathcal{K}_{ray}$ inject implicit 3D spatial priors into the attention computation. Finally, a geometry-aware Neighborhood Cross-Attention module aggregates low-resolution features $\mathcal{F}^{lr}$ as values to produce the reconstructed high-resolution features, ensuring cross-resolution spatial consistency while preserving underlying 3D geometric structure.

attention or coordinate-based mapping between high-resolution queries and low-resolution features, thereby achieving resolution-independent feature reconstruction. Although these methods improve expressiveness and scale generalization, they typically require retraining for specific visual encoders. Extending JAFAR, AnyUp [46] demonstrates that by constraining the receptive field of attention, a universal upsampling paradigm can be constructed, allowing a single trained upsampler to generalize across multiple vision encoders with different feature dimensions without retraining.

Following this attention-based trajectory, RaysUp introduces a lightweight architecture containing only 0.14M parameters, which can be seamlessly integrated with arbitrary vision encoders and supports universal feature upsampling at arbitrary resolutions. Notably, RaysUp achieves approximately 7× faster inference than AnyUp, while substantially reducing memory consumption.

3 RaysUp

3.1 Overview

As illustrated in Fig. 2, given an input RGB image $\mathcal{I} \in \mathbb{R}^{3\times H_{in}\times W_{in}}$ and a low-resolution feature map $\mathcal{F}^{lr} \in \mathbb{R}^{D_f\times H_{lr}\times W_{lr}}$ extracted from a Vision Foundation Model (VFM), where D_f denotes the feature dimension and (H_{lr}, W_{lr}) are typically small (e.g., 16×16 or 32×32), the goal of RaysUp is to reconstruct a high-resolution feature map $\mathcal{F}^{hr} \in \mathbb{R}^{D_f\times H_{any}\times W_{any}}$ at an arbitrary target resolution (H_{any}, W_{any}). To achieve this, RaysUp first employs a lightweight Spatially Decoupled Guidance Encoder to extract direction-aware guidance features from

the input image, producing $\mathcal{F}_g \in \mathbb{R}^{D_g \times H_{in} \times W_{in}}$, where D_g denotes the dimensionality of the guidance features. Then, an adaptive average pooling operation generates a target-resolution query $\mathcal{Q}_g \in \mathbb{R}^{D_g \times H_{any} \times W_{any}}$ and a VFM-resolution key $\mathcal{K}_g \in \mathbb{R}^{D_g \times H_{lr} \times W_{lr}}$. Both $\mathcal{Q}_g$ and $\mathcal{K}_g$ are subsequently encoded via Ray Positional Encoding (RayPE) to produce geometry-aware ray representations $\mathcal{Q}_{ray}$ and $\mathcal{K}_{ray}$, which embed implicit 3D spatial priors. Finally, a geometry-aware Neighborhood Cross-Attention mechanism aggregates the low-resolution features $\mathcal{F}^{lr}$ (acting as the value $\mathcal{V}$) into the high-resolution positions, yielding the reconstructed feature map $\mathcal{F}^{hr}$. This process ensures spatial consistency across resolutions while preserving the underlying 3D geometric structure.

3.2 Lightweight Spatially Decoupled Guidance Encoder

Guidance encoders for dense feature extraction conventionally adopt standard 1×1 or 3×3 convolutional operators [9, 19, 46]. However, statistical analysis of the learned 3×3 convolutional kernels in JAFAR [9] reveals a markedly anisotropic spatial distribution (Appendix **??**). Specifically, the average kernel magnitude matrix exhibits consistently larger weights at the four corner positions, while the central cross region (i.e., horizontal and vertical axes) shows relatively smaller magnitudes. This observation indicates that local feature aggregation is not spatially isotropic.

Motivated by this empirical regularity, we design a lightweight spatially decoupled guidance encoder to enhance directional interpretability and structural disentanglement. Concretely, we decompose the implicit receptive field into multiple orthogonal directional components and model them in parallel. The total output channel dimension D_g is evenly divided across branches, such that each branch is responsible for learning only $D_g/4$ channels. Given an input image $\mathcal{I} \in \mathbb{R}^{3 \times H_{in} \times W_{in}}$, the branch-wise formulations are defined as:

$$\begin{aligned} \mathcal{F}_g^c &= \phi\left(\mathrm{GN}\left(\mathrm{Conv}_{1\times1}^{D_g/4}(\mathcal{I})\right)\right), \mathcal{F}_g^h = \phi\left(\mathrm{GN}\left(\mathrm{Conv}_{1\times3}^{D_g/4}(\mathcal{I})\right)\right), \\ \mathcal{F}_g^v &= \phi\left(\mathrm{GN}\left(\mathrm{Conv}_{3\times1}^{D_g/4}(\mathcal{I})\right)\right), \mathcal{F}_g^{diag} = \phi\left(\mathrm{GN}\left(\mathrm{Conv}_{2\times2,d=2}^{D_g/4}(\mathcal{I})\right)\right), \end{aligned} \tag{1}$$

where $\mathrm{Conv}_{k\times k}^{D_g/4}$ denotes a convolution with $D_g/4$ output channels, and $\mathrm{Conv}_{2\times2,d=2}$ represents a 2×2 dilated convolution with dilation rate 2. $\mathrm{GN}(\cdot)$ denotes Group Normalization [47], and $\phi(\cdot)$ is the SiLU activation function [14]. To further enhance representational capacity, each branch adopts a shallow residual structure:

$$\tilde{\mathcal{F}}_g^b = \mathcal{X}^b + \phi\left(\mathrm{GN}\left(\mathrm{Conv}_b(\mathcal{X}^b)\right)\right), \quad b \in \{c, h, v, diag\}, \tag{2}$$

where $\mathcal{X}^b$ denotes the output of the first convolution in branch b. Finally, the directional features are concatenated along the channel dimension to form the guidance feature representation:

$$\mathcal{F}_g(\mathcal{I}) = \mathrm{Concat}(\tilde{\mathcal{F}}_g^c, \tilde{\mathcal{F}}_g^h, \tilde{\mathcal{F}}_g^v, \tilde{\mathcal{F}}_g^{diag}) \in \mathbb{R}^{D_g \times H_{in} \times W_{in}}. \tag{3}$$

From a parameterization standpoint, a standard 3×3 convolution contains $27D_g$ parameters, whereas our proposed spatially decoupled encoder requires only $3 \cdot \frac{D_g}{4}(1+3+3+4) = 8.25D_g$. Accordingly, the spatially decoupled guidance encoder not only substantially enhances anisotropic modeling capability but also reduces parameter overhead by approximately 69.4%, demonstrating clear advantages in both parameter efficiency and computational cost.

3.3 Any-Resolution Cross-Attention Mechanism

To enable feature upsampling at arbitrary resolutions, we adopt a resolution-decoupled cross-attention mechanism [9]. Specifically, the guidance encoded features $\mathcal{F}_g(\mathcal{I})$ are used to construct multi-scale attention inputs. Queries $\mathcal{Q}_g \in \mathbb{R}^{D_g \times H_{any} \times W_{any}}$ are generated at the upsampled resolution:

$$\mathcal{Q}_g = \text{AdaptiveAvgPool}(\mathcal{F}_g(\mathcal{I}), H_{any}, W_{any}), \tag{4}$$

while keys $\mathcal{K}_g \in \mathbb{R}^{D_g \times H_{lr} \times W_{lr}}$ are produced at the backbone feature resolution:

$$\mathcal{K}_g = \text{AdaptiveAvgPool}(\mathcal{F}_g(\mathcal{I}), H_{lr}, W_{lr}), \tag{5}$$

and the value features are taken directly from the output of VFMs:

$$\mathcal{V} = \mathcal{F}^{lr} \in \mathbb{R}^{D_f \times H_{lr} \times W_{lr}}. \tag{6}$$

This mechanism enables high-resolution queries to aggregate information from semantically aligned low-resolution features without requiring explicit interpolation, thereby supporting flexible cross-resolution feature propagation.

3.4 RayPE: Ray Positional Encoding

Traditional feature upsampling methods [9, 16, 19, 43, 46] operate within the 2D image grid, implicitly assuming that Euclidean proximity reflects 3D geometric consistency. However, under perspective projection, adjacent pixels in the image may correspond to spatially distant 3D points (e.g., at depth discontinuities), whereas distant pixels may lie on the same physical surface. Therefore, spatial interpolation in image space cannot guarantee geometric consistency.

Inspired by Neural Radiance Fields (NeRF) [33], RaysUp lifts feature reconstruction into a projective ray domain by associating each pixel with its camera ray. Let each pixel correspond to a ray $\boldsymbol{r}(\boldsymbol{x}) = (\boldsymbol{o}, \boldsymbol{d}(\boldsymbol{x}))$, where $\boldsymbol{o}$ is the camera origin and $\boldsymbol{d}(\boldsymbol{x})$ is the normalized direction. In this formulation, the upsampling can then be interpreted as transporting features from a coarse sampling of rays (backbone resolution) to a denser sampling (target resolution).

However, directly adopting a NeRF-style rendering pipeline incurs substantial computational cost. To address this, RaysUp introduces Ray Positional Encoding (RayPE), which encodes guidance features by mapping camera geometry into a high-dimensional rotational space. This modulation aligns features along

the 3D ray manifold during upsampling. Unlike conventional 2D positional encodings [41, 42], RayPE explicitly binds feature aggregation to the underlying 3D ray geometry rather than relying on pixel-plane proximity.

Ray Geometry Parameterization. Given a pixel $\boldsymbol{x} = [i, j, 1]^T$ in guidance features $\mathcal{Q}_g \in \mathbb{R}^{D_g \times H_{any} \times W_{any}}$ or $\mathcal{K}_g \in \mathbb{R}^{D_g \times H_{lr} \times W_{lr}}$, the corresponding 3D ray is uniquely determined by the camera intrinsic matrix $\mathbf{M}$ and extrinsic parameters $\mathbf{T} = [\mathbf{R}|\boldsymbol{t}]$. The ray origin (camera center) and normalized direction in world coordinates are defined as:

$$\boldsymbol{o} = \boldsymbol{t}, \quad \boldsymbol{d}(\boldsymbol{x}) = \frac{\mathbf{R}\mathbf{M}^{-1}\boldsymbol{x}}{\|\mathbf{R}\mathbf{M}^{-1}\boldsymbol{x}\|_2}. \tag{7}$$

Then, the origin and direction are concatenated to form a unified ray descriptor:

$$\boldsymbol{r} = \mathrm{Concat}(\boldsymbol{o}, \boldsymbol{d}(\boldsymbol{x})) \in \mathbb{R}^6. \tag{8}$$

This parameterization provides a resolution-independent 3D geometric reference shared across feature maps.

Multi-band Harmonic Phase Encoding. To capture geometric variations at multiple spatial scales, we construct a log-spaced frequency set $\{\omega_f\}_{f=0}^{N-1}$:

$$\omega_f = \exp\left(\log\frac{2\pi}{\lambda_{\max}} + \frac{f}{N-1}\Big(\log\frac{2\pi}{\lambda_{\min}} - \log\frac{2\pi}{\lambda_{\max}}\Big)\right), \tag{9}$$

where N denotes the total number of frequency bands, index $f \in \{0, \ldots, N-1\}$ specifies the f-th frequency level, and $\lambda_{\max}$ and $\lambda_{\min}$ represent the maximum and minimum spatial wavelengths, which determine the lowest and highest angular frequencies in the encoding. For each spatial location, the rotational phase $\boldsymbol{\Theta} \in \mathbb{R}^{N\times 6}$ is computed by multiplying each frequency ω_f with the corresponding component r_d of ray descriptor $\boldsymbol{r}$:

$$\boldsymbol{\Theta}_{f,d} = \omega_f\, r_d, \quad f = 0, \ldots, N-1,\ d = 1, \ldots, 6. \tag{10}$$

Then, the frequency and coordinate axes are flattened into a phase vector:

$$\boldsymbol{\theta} = \mathrm{Flatten}(\boldsymbol{\Theta}) \in \mathbb{R}^{6N}, \tag{11}$$

which encodes multi-scale geometric information along the ray, providing a unified reference for aligning high- and low-resolution features in 3D space.

Ray-driven Rotational Modulation. For each attention-head guidance feature tensor $\mathcal{F}_h \in \{\mathcal{Q}_g, \mathcal{K}_g\}$ with dimensionality d_{head}, we first split the feature into two halves:

$$\mathcal{F}_h = [\mathcal{F}_1, \mathcal{F}_2], \quad \mathcal{F}_1, \mathcal{F}_2 \in \mathbb{R}^{\lfloor d_{\text{head}}/2 \rfloor}. \tag{12}$$

To accommodate the rotate-half structure of attention heads, we require $6N \leq \lfloor d_{\text{head}}/2 \rfloor$, where $6N$ corresponds to the number of flattened RayPE phases. If $6N < \lfloor d_{\text{head}}/2 \rfloor$, the remaining feature dimensions are padded with zero phase, equivalent to an identity rotation. Following the idea of RoPE [41], Ray-driven

rotational modulation is applied element-wise to each pair of feature dimensions:

$$\begin{bmatrix} \tilde{\mathcal{F}}_1 \\ \tilde{\mathcal{F}}_2 \end{bmatrix} = \begin{bmatrix} \cos\boldsymbol{\theta} & -\sin\boldsymbol{\theta} \\ \sin\boldsymbol{\theta} & \cos\boldsymbol{\theta} \end{bmatrix} \begin{bmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \end{bmatrix}. \tag{13}$$

Finally, the RayPE encoded feature is obtained by concatenation:

$$\tilde{\mathcal{F}}_h = \mathrm{Concat}(\tilde{\mathcal{F}}_1, \tilde{\mathcal{F}}_2). \tag{14}$$

As the rotation angles are computed from the 3D ray coordinates and multi-scale harmonic frequencies, RayPE extends RoPE [41] from 1D sequential positions to 3D geometric positions, ensuring alignment of cross-resolution features along the shared 3D ray manifold and enforcing geometric consistency during upsampling.

3.5 Geometry-Aware Neighborhood Cross-Attention

To reconstruct dense features while preserving geometric consistency, we employ geometry-aware Neighborhood Cross-Attention [17]. Instead of global attention, each high-resolution query aggregates information from a local neighborhood defined in the low-resolution feature space. Let $\mathcal{Q}_{ray}$ and $\mathcal{K}_{ray}$ denote RayPE-modulated queries and keys, respectively, and $\mathcal{V}$ the value features (VFM features). For a query position (i, j), feature reconstruction is formulated as:

$$\mathcal{F}^{hr}_{i,j} = \sum_{(u,v)\in\mathcal{N}_{i,j}} \mathrm{Softmax}\left(\frac{(\mathcal{Q}_{ray})^{\top}_{i,j}(\mathcal{K}_{ray})_{u,v}}{\sqrt{d}}\right)\mathcal{V}_{u,v}, \tag{15}$$

where $\mathcal{N}_{i,j}$ denotes a $k \times k$ local neighborhood centered at the corresponding key location. To bridge resolution discrepancies between high-resolution (H_{any}, W_{any}) queries and low-resolution (H_{lr}, W_{lr}) keys, dilation factors are defined as:

$$d_h = \max(1, \lfloor H_{any}/H_{lr} \rfloor), d_w = \max(1, \lfloor W_{any}/W_{lr} \rfloor). \tag{16}$$

The dilated neighborhood induces a locally consistent sampling on the underlying ray manifold, ensuring that attention is performed among rays with similar orientations. Consequently, feature aggregation respects angular consistency, preventing interactions between geometrically unrelated rays while maintaining cross-resolution correspondence.

Under a local planar approximation, this process can be interpreted as discrete feature propagation along smoothly varying ray directions, enabling geometry-consistent cross-resolution information transfer. Meanwhile, the computational complexity is reduced from global attention $\mathcal{O}(H_{any}W_{any}\cdot H_{lr}W_{lr})$ to $\mathcal{O}(H_{any}W_{any}\cdot k^2)$, thereby improving efficiency while preserving geometric consistency.

3.6 Training Objective

Following standard practices for feature upsampling [9, 46], RaysUp is trained with a reconstruction objective. Given a high-resolution image $\mathcal{I}_{hr}$, a low-resolution

counterpart $\mathcal{I}_{lr}$ is generated by downsampling $\mathcal{I}_{hr}$ with a random scale factor $s \in [2, 4]$. Both images are then fed into a frozen vision encoder to obtain the target high-resolution features $\mathcal{F}_{tgt} \in \mathbb{R}^{D_f \times H_{any} \times W_{any}}$ and the source low-resolution features $\mathcal{F}^{lr} \in \mathbb{R}^{D_f \times H_{lr} \times W_{lr}}$. Conditioned on $\mathcal{F}^{lr}$ and the high-resolution guidance image $\mathcal{I}_{hr}$, the RaysUp upsampler predicts the reconstructed high-resolution feature map $\hat{\mathcal{F}}^{hr} \in \mathbb{R}^{D_f \times H_{any} \times W_{any}}$.

To enforce feature consistency, the training loss $\mathcal{L}$ is defined as a combination of cosine similarity loss $\mathcal{L}_{cos}$ and L_2 distance loss $\mathcal{L}_{L2}$:

$$\mathcal{L} = \mathcal{L}_{cos}(\hat{\mathcal{F}}^{hr}, \mathcal{F}_{tgt}) + \mathcal{L}_{L2}(\hat{\mathcal{F}}^{hr}, \mathcal{F}_{tgt}). \tag{17}$$

4 Experiments

4.1 Experimental Setup

Implementation. RaysUp was trained on the standard ImageNet dataset [11]. The model was optimized using the AdamW optimizer for 100,000 iterations with a batch size of 4 and an initial learning rate of 2×10^{-4}. To construct training pairs, target high-resolution images were uniformly resized to a spatial resolution of 448×448. For improved training efficiency, the guidance input to the upsampler was downsampled to 224×224. Owing to the adopted training strategy and the lightweight architectural design, the entire training process completed in approximately one hour on a single NVIDIA A100 GPU. Additional implementation details were provided in the supplementary material.

Baselines. To comprehensively evaluate the performance and generalization capability of RaysUp, it was systematically compared against several representative methods, including: traditional interpolation (e.g., Bilinear), fixed-ratio task-agnostic methods requiring encoder-specific retraining (e.g., FeatUp [16], LoftUp [19]), arbitrary-resolution encoder-dependent methods (e.g., JAFAR [9]), and arbitrary-resolution encoder-agnostic methods (e.g., AnyUp [46]).

4.2 Task-agnostic Performance

Following the baseline protocol [9, 16, 19, 46], RaysUp employed DINOv2-S [34] as the default backbone for both training the upsampler and extracting features for downstream tasks.

Semantic Segmentation. Following the experimental protocol of JAFAR [9], RaysUp employed a 1×1 convolutional layer as a linear probe for semantic label prediction. Evaluation was conducted on COCO-Stuff [4], ADE20K [54], Pascal-VOC [15], and Cityscapes [8] datasets. Performance was reported in terms of mean Intersection over Union (mIoU) and pixel accuracy (Acc).

Depth and Surface Normal Estimation. Following Probe3D [13], RaysUp was evaluated on dense geometric prediction tasks using NYUv2 [39]. For both normal and depth estimation, RMSE served as the primary metric. For normals, we additionally reported accuracy under angular thresholds of 11.25°,

Table 1: Task-agnostic Performance.

Semantic Segmentation

Method	COCO-Stuff [4]		Pascal-VOC [15]		ADE20K [54]		Cityscape [8]	
	mIoU↑	Acc↑	mIoU↑	Acc↑	mIoU↑	Acc↑	mIoU↑	Acc↑
Bilinear	59.58	79.42	81.70	95.44	40.47	74.13	59.72	92.55
FeatUp [16]	61.89	81.10	83.37	96.01	<u>42.33</u>	75.65	60.18	93.05
LoftUp [19]	<u>62.23</u>	<u>81.38</u>	<u>84.50</u>	<u>96.30</u>	42.17	<u>75.79</u>	**62.09**	<u>93.54</u>
JAFAR [9]	61.79	81.11	83.89	96.11	42.16	75.56	61.40	93.47
AnyUp [46]	62.14	<u>81.38</u>	84.18	96.20	42.15	75.71	60.62	93.26
RaysUp	**62.32**	**81.47**	**84.64**	**96.34**	**42.34**	**75.81**	<u>61.88</u>	**93.64**

Surface Normal and Depth Estimation

Method	Surface Normal				Depth (Abs)		Depth (Rel)	
	RMSE↓	11.25° ↑	22.5° ↑	30° ↑	RMSE↓	δ_1 ↑	RMSE↓	δ_1 ↑
Bilinear	28.23	0.4933	0.6974	0.7718	0.4789	0.7987	0.3348	0.9243
FeatUp [16]	28.94	0.4846	0.6888	0.7619	0.4781	**0.8117**	0.3393	0.9205
LoftUp [19]	28.45	0.4861	0.6931	0.7689	0.4828	0.7958	0.3353	0.9221
JAFAR [9]	<u>27.80</u>	0.4948	0.6986	0.7740	<u>0.4693</u>	0.8071	0.3255	0.9301
AnyUp [46]	27.83	<u>0.4962</u>	<u>0.7014</u>	<u>0.7767</u>	0.4781	0.8019	<u>0.3244</u>	<u>0.9306</u>
RaysUp	**27.69**	**0.4986**	**0.7030**	**0.7775**	**0.4658**	<u>0.8103</u>	**0.3195**	**0.9309**

Video Object Segmentation and Open-Vocabulary Segmentation

Method	Video Obj. Seg.			Open-Voc Seg.							
	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	COCO-Stuff [4]		Pascal-VOC [15]		ADE20K [54]		Cityscape [8]	
				mIoU↑	Acc↑	mIoU↑	Acc↑	mIoU↑	Acc↑	mIoU↑	Acc↑
Bilinear	63.21	66.53	64.87	25.73	47.91	59.71	85.87	19.60	42.32	34.91	58.52
FeatUp [16]	65.45	72.03	68.74	26.67	48.13	54.96	84.02	19.78	<u>43.09</u>	35.23	57.44
LoftUp [19]	67.32	74.53	70.92	**27.54**	48.38	**65.21**	**87.89**	**21.10**	**43.28**	**38.25**	59.95
JAFAR [9]	67.18	<u>74.62</u>	70.90	27.13	48.47	63.52	87.43	<u>20.57</u>	42.76	<u>38.00</u>	**60.32**
AnyUp [46]	<u>67.55</u>	74.42	<u>70.98</u>	<u>27.30</u>	**48.62**	<u>63.67</u>	<u>87.50</u>	20.67	42.77	37.56	<u>60.03</u>
RaysUp	**68.14**	**74.81**	**71.47**	27.11	<u>48.57</u>	63.28	87.34	20.54	42.71	37.24	59.68

22.5°, and 30°. For depth, we reported the δ_1 metric, which measured the proportion of pixels with a prediction-to-ground-truth ratio below 1.25.

Video Object Segmentation and Open-Vocabulary Segmentation. RaysUp was evaluated on the DAVIS [35] dataset for video object segmentation. Specifically, the first-frame object mask was propagated temporally by computing dense cross-frame feature similarities. Performance was measured using $\mathcal{J}$ Mean (region similarity IoU), $\mathcal{F}$ Mean (boundary contour accuracy), and their combined metric, $\mathcal{J}\&\mathcal{F}$ Mean, providing a comprehensive assessment of regional consistency and boundary reconstruction quality. For zero-shot open-vocabulary segmentation, the upsampled features were integrated into the ProxyCLIP [25] framework. By leveraging spatial correspondences from pre-trained vision encoders to enhance CLIP [37] representations, we evaluated segmentation performance on COCO-Stuff [4], Cityscapes [8], ADE20K [54], and Pascal-VOC [15].

Analysis. The quantitative results (Tab. 1) indicate that, owing to our Geometry-Aware Ray Representation, RaysUp achieves the best or second-best performance across most metrics, with particularly strong results in dense geometric prediction tasks such as surface normal and depth estimation. In com-

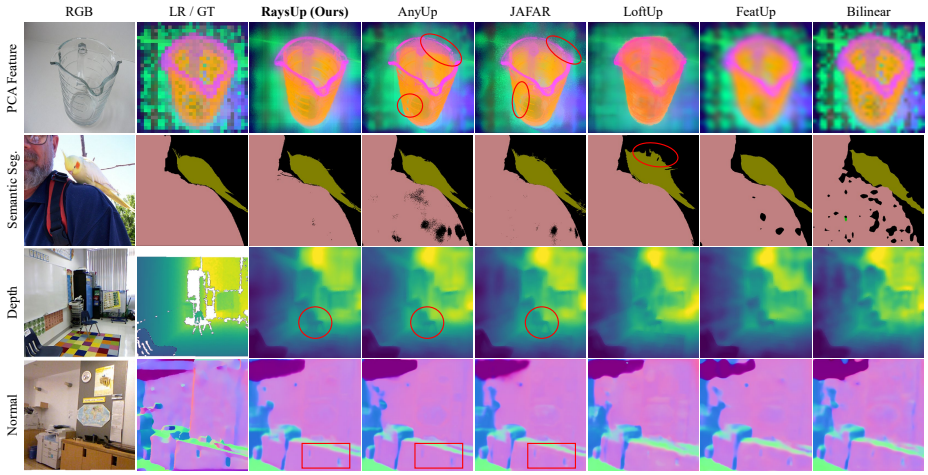

Fig. 3: Qualitative Results for Task-agnostic Performance. Leveraging the geometry-aware Ray representation, RaysUp demonstrates enhanced geometric consistency and achieves superior performance across downstream tasks.

parison, LoftUp [19] shows slightly superior performance in certain semantic segmentation and open-vocabulary tasks, primarily due to the incorporation of SAM [23] masks as auxiliary supervision during training. Qualitative analysis (Fig. 3) further demonstrates that, unlike AnyUp [46], JAFAR [9], and FeatUp [16] which exhibit holes, or LoftUp [19] which produces blurred boundaries, RaysUp better preserves geometric consistency. Overall, even without additional segmentation supervision, RaysUp maintains overall superior performance, with pronounced advantages in geometric consistency and video tasks, highlighting its stronger task-agnostic generalization capability.

4.3 VFM-agnostic Performance

Among the baseline methods, only AnyUp [46] possesses VFM-agnostic capability. Therefore, it was selected as the primary comparison. Consistent with AnyUp, RaysUp was trained on DINOv2-S and evaluated for its generalization across different VFMs (DINOv2 [34], DINOv3 [40], SigLIP2 [44], and PE Spatial [3]) and various model scales (ViT-S, ViT-M, and ViT-L). As shown in Tab. 2, RaysUp consistently outperforms AnyUp across all tested architectures and model sizes, demonstrating that the proposed ray-based upsampling approach effectively preserves high-level semantic information and fine-grained spatial structures without relying on the underlying backbone network.

4.4 Ultra-lightweight Performance

Tab. 3 presents a comprehensive evaluation of computational efficiency, memory footprint, and inference speed for RaysUp in comparison with existing up-

Table 2: VFM-agnostic Performance. Semantic segmentation results (mIoU/Acc) on Pascal-VOC [15] and depth estimation results (RMSE) on NYUv2 [39].

		DINOv2		DINOv3	
Size	Method	mIoU (Acc) ↑	RMSE (abs/rel) ↓	mIoU (Acc) ↑	RMSE (abs/rel) ↓
ViT-S	AnyUp [46]	84.18 (96.20)	0.478 (0.324)	81.97 (95.77)	0.506 (0.345)
	RaysUp	**84.64 (96.34)**	**0.465 (0.319)**	**82.60 (95.97)**	**0.502 (0.342)**
ViT-M	AnyUp [46]	85.44 (96.57)	0.415 (0.289)	86.51 (96.84)	0.455 (0.296)
	RaysUp	**86.03 (96.70)**	**0.400 (0.276)**	**87.20 (97.02)**	**0.452 (0.289)**
ViT-L	AnyUp [46]	85.47 (96.56)	0.393 (0.259)	87.48 (97.07)	0.402 (0.257)
	RaysUp	**86.33 (96.81)**	**0.376 (0.256)**	**88.07 (97.20)**	**0.398 (0.255)**
		SigLIP2†		PE Spatial	
Size	Method	mIoU (Acc) ↑	RMSE (abs/rel) ↓	mIoU (Acc) ↑	RMSE (abs/rel) ↓
ViT-B	AnyUp [46]	78.70 (94.60)	0.845 (0.522)	77.41 (94.63)	0.726 (0.461)
	RaysUp	**79.37 (94.84)**	**0.843 (0.519)**	**77.47 (94.69)**	**0.706 (0.448)**
ViT-L	AnyUp [46]	79.16 (94.73)	0.726 (0.474)	82.23 (95.87)	0.635 (0.404)
	RaysUp	**80.10 (94.97)**	**0.725 (0.472)**	**82.75 (95.98)**	**0.613 (0.388)**

† A ViT-S version of SigLIP2 is unavailable.

Table 3: Ultra-lightweight Performance. Comparison of Parameters (M), GFLOPs, GPU memory (GB), and inference FPS at various upsampling resolutions.

		224×224			448×448			896×896			$2K \times 2K$
Method	Params (M)	GFLOPs ↓	Mem ↓	FPS ↑	GFLOPs ↓	Mem ↓	FPS ↑	GFLOPs ↓	Mem ↓	FPS ↑	FPS ↑
FeatUp [16]	0.17	31.79	1.73	48	127.02	4.08	14	507.95	15.49	2	OOM
LoftUp [19]	4.30	397.32	2.88	7	1970.70	6.39	4	OOM	OOM	OOM	OOM
JAFAR [9]	0.62	91.88	2.01	16	366.52	5.98	11	1465.08	21.86	2	OOM
AnyUp [46]	0.87	84.21	2.19	11	328.55	6.73	5	1306.27	24.85	1	OOM
RaysUp	**0.14**	**10.17**	**1.26**	**55**	**40.67**	**2.69**	**27**	**162.68**	**8.47**	**8**	**1**

sampling methods across multiple resolutions. RaysUp demonstrates a clear advantage in all metrics. At a standard input resolution of 224×224, RaysUp requires only 0.14M parameters, consumes 10.17 GFLOPs, and uses 1.26GB of GPU memory, achieving 55 FPS, substantially outperforming alternatives such as FeatUp [16], LoftUp [19], JAFAR [9], and AnyUp [46]. As the target resolution scales to 448×448 and 896×896, RaysUp maintains a low computational burden while delivering high inference speeds (27 and 8 FPS, respectively), whereas other methods experience dramatic slowdowns or run out of memory (OOM) at higher resolutions. Even at extreme resolutions of $2K \times 2K$, RaysUp remains runnable (1 FPS), highlighting its scalability.

These results confirm that RaysUp achieves an ultra-lightweight design that balances parameter efficiency, computational cost, and high-speed inference, making it suitable for dense prediction tasks across a wide range of resolutions while maintaining practical usability on modern GPUs.

Table 4: Ablation Study. Default settings of RaysUp are marked with an asterisk(*).

	DINOv2	SigLIP2	PE Spatial	DINOv3	Avg.	Params (M)
Guidance Encoder						
Single-Branch	84.37	79.17	77.34	86.63	81.87	0.266
Dual-Branch	84.49	79.25	77.39	87.02	82.03	0.66
Multi-Branch	84.52	79.33	77.42	87.11	82.09	0.268
Decoupled-Branch *	**84.64**	**79.37**	**77.47**	**87.20**	**82.17**	**0.14**
Guidance Feature Dim						
$D_g = 128$	84.26	**79.72**	77.35	86.78	82.02	**0.06**
$D_g = 256$ *	**84.64**	79.37	77.47	87.20	82.17	0.14
$D_g = 512$	84.61	79.46	77.49	87.24	**82.20**	0.40
$D_g = 768$	84.59	79.41	**77.50**	**87.26**	82.19	0.89
Guidance Conv Blocks						
$L = 1$ *	84.64	79.37	77.47	87.20	82.17	**0.14**
$L = 2$	**84.66**	79.59	77.53	87.42	82.30	0.19
$L = 3$	84.64	**79.68**	77.95	87.51	**82.44**	0.27
$L = 4$	84.40	79.60	**78.08**	**87.66**	82.43	0.36
Positional Encoding						
None	83.51	77.70	76.47	86.54	81.05	0.14
RoPE [41]	84.43	78.89	77.24	87.15	81.92	0.14
SinRays [42]	84.12	78.45	76.81	86.98	81.59	0.47
RayPE *	**84.64**	**79.37**	**77.47**	**87.20**	**82.17**	**0.14**
Image Pose (T)						
Identity *	84.64	79.37	77.47	87.20	82.17	**0.14**
DA3-Small [28]	84.69	**79.46**	77.85	87.65	82.41	0.14
DA3-Base [28]	**84.87**	79.21	77.90	87.78	**82.44**	0.14
DA3-Large [28]	84.76	79.06	**78.03**	**87.84**	82.42	0.14

4.5 Ablation Study

We conducted a series of ablation experiments to evaluate the effectiveness of the core components within the RaysUp framework. All experiments were systematically performed across four representative VFMs, including DINOv2 [34], SigLIP2 [44], PE Spatial [3], and DINOv3 [40].

Effectiveness of Spatially Decoupled Guidance Encoder. As shown in Tab. 4, compared to Single-Branch, Dual-Branch, and Multi-Branch designs, the proposed Decoupled-Branch achieves the highest performance across all four VFMs, with an average of 82.17%, while drastically reducing the parameter count to only 0.14M. This indicates that decoupled modeling of spatial guidance features not only more effectively captures multi-scale spatial semantic information but also enables an ultra-lightweight design, substantially improving efficiency for downstream tasks. Regarding the guidance feature dimension and the number of convolutional blocks, performance varies across different VFMs; considering the trade-off between accuracy and efficiency, RaysUp adopts a guidance feature dimension of $D_g = 256$ with a single convolutional block.

Effectiveness of RayPE. As shown in Tab. 4, omitting positional encoding leads to a significant drop in performance (average 81.05%). While existing methods such as RoPE [41] and SinRays [42] provide moderate improvements, RayPE achieves the highest performance across all VFMs without introducing

additional parameters. This indicates that RayPE effectively injects implicit 3D geometric priors, enhancing geometric consistency, boundary fidelity, and structural reconstruction. Moreover, incorporating DA3 [28] pose information can further improve performance in certain scenarios; however, considering computational efficiency, RaysUp adopts an Identity pose configuration.

5 Conclusion

We introduced RaysUp, an ultra-lightweight, task-agnostic, and encoder-agnostic framework for universal feature upsampling, capable of reconstructing backbone features at arbitrary resolutions. By integrating a Spatially Decoupled Guidance Encoder, an Any-Resolution Cross-Attention mechanism, and Ray Positional Encoding (RayPE), RaysUp injects implicit 3D geometric priors to achieve geometry-aware high-resolution feature reconstruction. Extensive experiments demonstrate that RaysUp delivers superior performance across multiple dense prediction tasks, using only about 1/5 of the parameters of baseline methods and achieving approximately 7× faster inference, while effectively balancing accuracy and computational efficiency.

Acknowledgements

This work was supported in part by the New Generation Artificial Intelligence-National Science and Technology Major Project under Grant 2025ZD0123701, in part by the National Natural Science Foundation of China under Grant 62476202 and 62272343, and in part by the Fundamental Research Funds for the Central Universities.

References

1. Asim, M., Wewer, C., Wimmer, T., Schiele, B., Lenssen, J.E.: MEt3R: Measuring Multi-View Consistency in Generated Images. In: CVPR. pp. 6034–6044 (2025)
2. Barsellotti, L., Bianchi, L., Messina, N., Carrara, F., Cornia, M., Baraldi, L., Falchi, F., Cucchiara, R.: Talking to DINO: Bridging Self-Supervised Vision Backbones with Language for Open-Vocabulary Segmentation. In: ICCV. pp. 22025–22035 (2025)
3. Bolya, D., Huang, P.Y., Sun, P., Cho, J.H., Madotto, A., Wei, C., Ma, T., Zhi, J., Rajasegaran, J., Rasheed, H., et al.: Perception Encoder: The Best Visual Embeddings Are Not at the Output of the Network. arXiv preprint arXiv:2504.13181 (2025)
4. Caesar, H., Uijlings, J., Ferrari, V.: COCO-Stuff: Thing and Stuff Classes in Context. In: CVPR. pp. 1209–1218 (2018)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging Properties in Self-Supervised Vision Transformers. In: ICCV. pp. 9650–9660 (2021)

6. Chen, S., Guo, H., Zhu, S., Zhang, F., Huang, Z., Feng, J., Kang, B.: Video Depth Anything: Consistent Depth Estimation for Super-Long Videos. In: CVPR. pp. 22831–22840 (2025)
7. Chuang, Y.S., Li, Y., Wang, D., Yeh, C.F., Lyu, K., Raghavendra, R., Glass, J.R., HUANG, L., Weston, J.E., Zettlemoyer, L., Chen, X., Liu, Z., Xie, S., tau Yih, W., Li, S.W., Xu, H.: Meta CLIP 2: A Worldwide Scaling Recipe. In: NeurIPS (2025)
8. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: CVPR. pp. 3213–3223 (2016)
9. Couairon, P., Chambon, L., Serrano, L., HAUGEARD, J.E., Cord, M., THOME, N.: JAFAR: Jack up Any Feature at Any Resolution. In: NeurIPS (2025)
10. Dai, Y., Lu, H., Shen, C.: Learning Affinity-Aware Upsampling for Deep Image Matting. In: CVPR. pp. 6841–6850 (2021)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR. pp. 248–255 (2009)
12. Duchon, C.E.: Lanczos Filtering in One and Two Dimensions. Journal of Applied Meteorology **18**(8), 1016–1022 (1979)
13. El Banani, M., Raj, A., Maninis, K.K., Kar, A., Li, Y., Rubinstein, M., Sun, D., Guibas, L., Johnson, J., Jampani, V.: Probing the 3D Awareness of Visual Foundation Models. In: CVPR. pp. 21795–21806 (2024)
14. Elfwing, S., Uchibe, E., Doya, K.: Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. Neural Networks **107**, 3–11 (2018)
15. Everingham, M., Eslami, S.M., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal Visual Object Classes Challenge: A Retrospective. IJCV **111**(1), 98–136 (2015)
16. Fu, S., Hamilton, M., Brandt, L.E., Feldmann, A., Zhang, Z., Freeman, W.T.: FeatUp: A Model-Agnostic Framework for Features at Any Resolution. In: ICLR (2024)
17. Hassani, A., Walton, S., Li, J., Li, S., Shi, H.: Neighborhood Attention Transformer. In: CVPR. pp. 6185–6194 (2023)
18. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked Autoencoders Are Scalable Vision Learners. In: CVPR. pp. 16000–16009 (2022)
19. Huang, H., Chen, A., Havrylov, V., Geiger, A., Zhang, D.: LoftUp: Learning a Coordinate-Based Feature Upsampler for Vision Foundation Models. In: ICCV (2025)
20. Jose, C., Moutakanni, T., Kang, D., Baldassarre, F., Darcet, T., Xu, H., Li, D., Szafraniec, M., Ramamonjisoa, M., Oquab, M., et al.: DINOv2 Meets Text: A Unified Framework for Image- and Pixel-Level Vision-Language Alignment. In: CVPR. pp. 24905–24916 (2025)
21. Jun-Seong, K., Kim, G., Yu-Ji, K., Wang, Y.C.F., Choe, J., Oh, T.H.: Dr. Splat: Directly Referring 3D Gaussian Splatting via Direct Language Embedding Registration. In: CVPR. pp. 14137–14146 (2025)
22. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: LERF: Language Embedded Radiance Fields. In: ICCV. pp. 19729–19739 (2023)
23. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment Anything. In: ICCV. pp. 4015–4026 (2023)
24. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint Bilateral Upsampling. ACM Trans. Graph. **26**(3), 96 (2007)

25. Lan, M., Chen, C., Ke, Y., Wang, X., Feng, L., Zhang, W.: ProxyCLIP: Proxy Attention Improves CLIP for Open-Vocabulary Segmentation. In: ECCV. pp. 70–88 (2024)
26. Li, L., Zhang, L., Wang, Z., Shen, Y.: GS3LAM: Gaussian Semantic Splatting SLAM. In: ACM MM. p. 3019–3027 (2024)
27. Li, L., Zhang, L., Wang, Z., Zhang, F., Li, Z., Shen, Y.: Representing Sounds as Neural Amplitude Fields: A Benchmark of Coordinate-MLPs and a Fourier Kolmogorov-Arnold Framework. AAAI **39**(23), 24458–24466 (2025)
28. Lin, H., Chen, S., Liew, J.H., Chen, D.Y., Li, Z., Zhao, Y., Peng, S., Guo, H., Zhou, X., Shi, G., Feng, J., Kang, B.: Depth Anything 3: Recovering the Visual Space from Any Views. In: ICLR (2026)
29. Lu, H., Dai, Y., Shen, C., Xu, S.: Index Networks. IEEE TPAMI **44**(1), 242–255 (2022)
30. Lu, H., Liu, W., Fu, H., Cao, Z.: FADE: Fusing the Assets of Decoder and Encoder for Task-Agnostic Upsampling. In: ECCV. pp. 231–247 (2022)
31. Lu, H., Liu, W., Ye, Z., Fu, H., Liu, Y., Cao, Z.: SAPA: Similarity-Aware Point Affiliation for Feature Upsampling. In: NeurIPS (2022)
32. McKinley, S., Levine, M.: Cubic Spline Interpolation. College of the Redwoods **45**(1), 1049–1060 (1998)
33. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: ECCV. pp. 405–421 (2020)
34. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P., Li, S., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning Robust Visual Features without Supervision. TMLR (2024)
35. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 DAVIS Challenge on Video Object Segmentation. arXiv preprint arXiv:1704.00675 (2017)
36. Qin, M., Li, W., Zhou, J., Wang, H., Pfister, H.: LangSplat: 3D Language Gaussian Splatting. In: CVPR. pp. 20051–20060 (2024)
37. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning Transferable Visual Models from Natural Language Supervision. In: ICML. pp. 8748–8763 (2021)
38. Shin, H., Kim, C., Hong, S., Cho, S., Arnab, A., Seo, P.H., Kim, S.: Towards Open-Vocabulary Semantic Segmentation Without Semantic Labels. NeurIPS **37**, 9153–9177 (2024)
39. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor Segmentation and Support Inference from RGBD Images. In: ECCV. pp. 746–760 (2012)
40. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: DINOv3. arXiv preprint arXiv:2508.10104 (2025)
41. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: RoFormer: Enhanced Transformer with Rotary Position Embedding. Neurocomputing **568**, 127063 (2024)
42. Sun, C., Yuan, Z., Xu, K., Mai, L., N, S., Chen, S., Marina, M.K.: Learning High-Frequency Functions Made Easy with Sinusoidal Positional Encoding. In: ICML (2024)

43. Suri, S., Walmer, M., Gupta, K., Shrivastava, A.: LiFT: A Surprisingly Simple Lightweight Feature Transform for Dense ViT Descriptors. In: ECCV. pp. 110–128 (2024)
44. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., et al.: SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. arXiv preprint arXiv:2502.14786 (2025)
45. Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C.C., Lin, D.: CARAFE: Content-Aware ReAssembly of FEatures. In: ICCV. pp. 3007–3016 (2019)
46. Wimmer, T., Truong, P., Rakotosaona, M.J., Oechsle, M., Tombari, F., Schiele, B., Lenssen, J.E.: AnyUp: Universal Feature Upsampling. In: ICLR (2026)
47. Wu, Y., He, K.: Group Normalization. In: ECCV. pp. 3–19 (2018)
48. Wysoczańska, M., Siméoni, O., Ramamonjisoa, M., Bursuc, A., Trzciński, T., Pérez, P.: CLIP-DINOiser: Teaching CLIP a few DINO Tricks for Open-Vocabulary Semantic Segmentation. In: ECCV. pp. 320–337 (2024)
49. Xie, X., Lessen, J.E., Pons-Moll, G.: MVGBench: A Comprehensive Benchmark for Multi-view Generation Models. In: ICCV. pp. 8207–8218 (2025)
50. Xu, H., Xie, S., Tan, X.E., Huang, P.Y., Howes, R., Sharma, V., Li, S.W., Ghosh, G., Zettlemoyer, L., Feichtenhofer, C.: Demystifying CLIP Data. arXiv preprint arXiv:2309.16671 (2023)
51. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. In: CVPR. pp. 10371–10381 (2024)
52. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth Anything V2. NeurIPS **37**, 21875–21911 (2024)
53. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid Loss for Language Image Pre-Training. In: ICCV. pp. 11941–11952 (2023)
54. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic Understanding of Scenes through the ADE20K Dataset. IJCV **127**(3), 302–321 (2019)