ObliCity: A Benchmark and Baseline for Roof-to-Ground Projection Displacement Correction

Kai Li[a,b], Yupeng Deng[c,*], Ligao Deng[a], Zhihao Xi[c], Chenhao Wang[a], Jierui Zhang[d], Yingrui Ji[a], Yu Meng[c], Xiangyu Zhao[b]

[a]*School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, 1 East Yanqi Lake Road, Beijing, 100049, China,*
[b]*College of Computing, Department of Data Science, City University of Hong Kong, Kowloon Tong, Hong Kong, 999077, China,*
[c]*Aerospace Information Research Institute, Chinese Academy of Sciences, 9 Dengzhuang South Road, Beijing, 101408, China,*
[d]*Department of Electrical and Computer Engineering, The University of Hong Kong, Pokfulam, Hong Kong, 999077, China,*

Abstract

Oblique-view urban remote sensing imagery inevitably exhibits geometric projection displacements between building roofs and footprints, leading to significant distortions in spatial structure. Existing approaches either ignore these deformations or handle them implicitly within segmentation-based frameworks, where progress is dominated by general segmentation advances rather than improvements in geometric correction. In this work, we explicitly define roof-to-footprint offset vector (RFOV) extraction as an independent learning task that decouples geometric alignment from semantic segmentation. To support this task, we introduce the Oblique City dataset (ObliCity), the first large-scale benchmark that integrates high-resolution UAV imagery and globally distributed satellite data, covering diverse city morphologies and camera perspectives. Methodologically, we reformulate DragOSM into DragRoof, an ODE-based framework inspired by human annotation behavior. By simulating the continuous process of dragging roofs toward their footprints, DragRoof learns deterministic, geometry-consistent offset fields and adaptively determines convergence through an end token. Extensive experiments on ObliCity demonstrate that DragRoof achieves state-of-the-art RFOV extraction performance, requiring fewer inference steps while delivering superior directional and length accuracy. Our dataset and model establish a principled foundation for studying projection displacement correction in oblique remote sensing imagery. The source code and dataset will be avaliable at `https://github.com/likaiucas/DragRoof`.

Keywords: Offset learning, building extraction, off-nadir remote sensing image.

1. Introduction

Urban remote sensing images captured from oblique views inevitably suffer from geometric projection displacements between building roofs and their corresponding footprints (Li et al., 2024b). These roof-to-footprint displacements, caused by camera tilt and building height, distort the spatial structures of buildings and make it difficult to obtain accurate building geometry from monocular imagery. Currently, building detection and height estimation are two of the most prominent research directions. Existing mainstream building extraction studies, however, predominantly focus on roof segmentation (Xu et al., 2023; Wang et al., 2024, 2022; Yu et al., 2025) (Fig. 1(a)), while largely overlooking such projection-induced deformations. Many of these approaches treat the roof as a proxy (implicitly assuming roof–footprint perfectly overlapped, which no longer holds in modern ultra-high-resolution imagery) for the entire building or employ semantic segmentation to estimate per-pixel building height values (Cao and Huang, 2021; Chen et al., 2023; Sun et al., 2024; Gültekin et al., 2025) (Fig. 1(b)). Although effective for general building delineation, these rep-

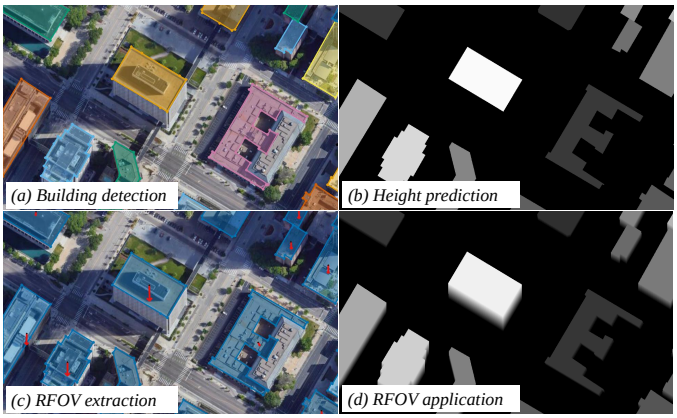


Figure 1: Current mainstream research on buildings mainly focuses on two directions: (a) Building detection, which emphasises semantic feature extraction, *e.g.*, roof instance segmentation and classification; and (b) Height prediction, which often relies heavily on deep learning models to directly regress physical height values (in meters) from local semantic cues. (c) We propose a new task that extracts the RFOV (red arrows) for each building in a given image. (d) Based on the RFOV features, we can more accurately describe building structures and positions; and, by leveraging the known pixel resolution of remote sensing imagery and camera geometry, the pixel-level height derived from RFOVs is more physically reliable than direct height regression in (b).

*Corresponding author
Email addresses: likai211@mails.ucas.ac.cn (Kai Li), dengyp@aircas.ac.cn (Yupeng Deng)

resentations (roof instances or semantic height maps) are insufficient to recover precise ground-level positions or the 3D structure of buildings.

Although RFOVs have been introduced in several off-nadir building footprint extraction tasks as auxiliary cues to assist roof segmentation (Wang et al., 2023; Pang et al., 2023; Li et al., 2021, 2024c,a, 2025a; Zhou et al., 2025), the innovations of these methods still primarily focus on advances in semantic segmentation. This is because the evaluation of footprint extraction relies on common mask-based metrics, *e.g.*, F1 score and *Intersection over Union*, which are largely insensitive to RFOV accuracy[1]. When the roof mask itself is not precisely delineated, even an improvement of 3–5 pixels in RFOV prediction yields negligible gains in mask metrics. This issue is particularly evident in cases where the roof mask is large but the RFOV magnitude is small. Consequently, progress in RFOV extraction has often been overshadowed by advances in roof segmentation performance.

This motivates us to decouple RFOV extraction from conventional segmentation frameworks, framing it as a standalone learning objective. *i.e.*, assuming that reliable building regions (roofs) are available as priors, we only estimate an RFOV for each building, as Fig. 1(c). Within these task settings, the future technical improvement on the RFOV extraction will benefit downstream applications (Fig. 1(d)).

However, existing datasets containing RFOV annotations are limited to BONAI (Wang et al., 2023) and OmniCity (Li et al., 2023). Both suffer from significant shortcomings: BONAI includes only satellite imagery from five major Chinese cities with 0.3–0.5 m resolution, while over 60% of the RFOVs in OmniCity are shorter than 10 pixels, making displacement cues weak. Moreover, ultra-long RFOV samples, which are commonly present in UAV imagery, remain scarce, and existing models still struggle to accurately predict their lengths. On the other hand, existing models capable of directly predicting RFOVs are mainly represented by OBM (Li et al., 2024a) and PolyFootNet (Li et al., 2025a). These methods were originally designed to facilitate the manual annotation of building footprints, and thus adopt simple yet spatially ambiguous bounding boxes as prompts for human–model interaction.

To address the aforementioned gaps, we meticulously annotated UAV imagery (0.1 m) from seven distinct regions across the Yangtze River Delta and re-annotated the urban subset of the open-source vector dataset IRSAMap (Meng et al., 2025). Together, these efforts constitute the ObliCity dataset, the first multi-scene, multi-platform dataset that integrates UAV and satellite imagery for RFOV extraction.

On the methodological side, inspired by the human annotation process in ObliCity, we model RFOV extraction as a continuous process that mimics how annotators drag the roof along the building facade boundaries toward its footprint until the projected edges align with visible image pixels. We formulate this trajectory as an ordinary ODE-based[2] inference process and instantiate it within the DragOSM framework; we name this model as DragRoof. Furthermore, to enable the model to intuitively determine whether the dragging process has reached the footprint position, we introduce an *end token* that serves as an indicator of convergence. DragRoof uses iterative inference to handle ultra-long RFOV cases.

Specifically, during training, DragRoof uniformly samples a time step $t \in [0, 1]$ to simulate the intermediate states of the roof being dragged along the facade, analogous to the human annotation process. The model is then trained to regress the displacement vector from the current sampled position to the footprint, while the end token decodes an *end flag* that is strongly correlated with the distance between the current mask and the footprint, allowing the model to sense convergence. During inference, DragRoof progressively refines the roof mask through a continuous correction process to obtain the final RFOV. DragRoof achieves state-of-the-art performance with only two-step inference.

Different from OBM and PolyFootNet, DragRoof performs RFOV extraction using roof polygons or mask prompts as inputs, enabling better compatibility and integration with other mask–based methods, *e.g.*, HiSup (Xu et al., 2023) & SAM-PolyBuild (Wang et al., 2024).

In conclusion, the main contributions of this work are:

- We advocate for decoupling RFOV extraction from segmentation and formally define it as a standalone, critical computer vision task. We analyse how existing coupled approaches hinder progress and argue that this new formulation is essential for current studies.
- We constructed ObliCity, the first large-scale benchmark dataset, including high-resolution UAV and global satellite imagery, for the challenging RFOV extraction. We will continue to update ObliCity to grow in size and scope for evolving real-world conditions.
- Inspired by the human annotating process, we propose a strong baseline method, DragRoof, with ODE modelling, which demonstrates the feasibility of the RFOV extraction problem.

2. Related Work

In this section, we first review the evolution of building extraction methods from near-nadir to off-nadir imagery, and subsequently outline how our work bridges the existing gaps in this field.

Near-nadir Imagery. Research on building extraction initially focused heavily on near-nadir images, establishing a well-explored mainstream methodology. Prominent datasets, such as the WHU Building Dataset (Ji et al., 2018), SpaceNet (Van Etten et al., 2018), and WHU-Mix (Wei et al., 2023), have provided vast amounts of annotated building roofs. These datasets primarily consist of high-resolution remote sensing images captured from near-nadir perspectives, which successfully supported early-stage algorithm development (Xu et al., 2023; Wang et al., 2024, 2022; Yu et al., 2025). Consequently, the resulting methods are generally restricted to roof segmentation

[1] We mathematically probe this in Sec. 6.2.

[2] Ordinary Differential Equation (ODE)

under ideal near-nadir assumptions. While the geometric projection displacement inherent in off-nadir imagery was widely acknowledged during this stage, it remained largely unsolved due to its complexity.

Off-nadir Imagery. Driven by the continuous advancement of deep learning, researchers have increasingly turned their attention to building extraction in off-nadir settings (Wang et al., 2023; Pang et al., 2023; Li et al., 2021, 2024c,a, 2025a; Zhou et al., 2025). This shift is motivated by the practical advantages of off-nadir acquisition: relaxed satellite viewing angle constraints significantly improve Earth observation efficiency, and the oblique perspective captures rich facade information essential for reconstructing LoD-2 level building models (Gröger et al., 2012), including building heights. However, current methodologies still rely on conventional semantic segmentation metrics (*e.g.*, mask IoU) for evaluation, which fail to accurately reflect the actual quality of the extracted Roof-to-Footprint Offset Vectors (RFOVs) (detailed in Sec. 6.2). Furthermore, existing off-nadir datasets, such as BONAI (Wang et al., 2023) and OmniCity (Li et al., 2023), are constrained by the absence of 0.1m ultra-high-resolution UAV data, a lack of ultra-long RFOV samples, and relatively crude annotation boundaries.

To address these barriers, we formally decouple RFOV extraction into an independent computer vision task and introduce a comprehensive new dataset, ObliCity, to facilitate further research. Inspired by the natural human annotation process, we propose DragRoof as a robust baseline method tailored to effectively solve the RFOV extraction problem.

3. Dataset: ObliCity

In this section, we will first introduce why we newly annotated the ObliCity dataset

3.1. Collection & Annotation of ObliCity

Motivation. ObliCity is proposed to address the limitations of existing datasets for RFOV learning, which are mainly captured from a single urban environment and rely solely on satellite imagery. We newly annotated UAV images (0.1 m) to diversify the image resolution, and added RFOV annotations to the urban scenes of IRSAMap (Meng et al., 2025) to include worldwide samples.

Image collection. The images are collected from 3 parts: IRSAMap, BONAI (Wang et al., 2023) and UAV. Specifically, to further enrich scene diversity, we re-annotate the building regions in the IRSAMap (Meng et al., 2025) dataset with corresponding RFOVs. IRSAMap is a global satellite dataset covering 79 regions across various urban types, containing 5,869 finely annotated images of size 1024×1024. Among them, we re-labeled 666 off-nadir images with a spatial resolution of 0.5 m. The UAV dataset includes images collected from seven distinct urban areas across the Yangtze River Delta region. After standardised annotation and cropping, it consists of 1,008 images of size 1536×1536 with a spatial resolution of approximately 0.1 m. The BONAI dataset contains 3,300 satellite images of size 1024×1024 with a spatial resolution of 0.3–0.5 m,

Figure 2: Samples in ObliCity. The ObliCity dataset is designed for the task of extracting RFOVs (red arrows) for buildings with known roofs. Each sample is created by manually delineating the roof and recording the vector obtained from dragging it to its corresponding footprint position.

covering six major cities in China, *e.g.*, Chengdu, Xi'an, and Harbin.

Annotation. The annotation process for UAV imagery consists of three steps. For each unannotated image, (1) annotators manually delineate the roof of every building, (2) drag the roof polygon until its edge aligns with the visible portion of the corresponding footprint, and (3) compute the RFOV based on the displacement trajectory of this drag operation. For the IRSAMap dataset, we keep the roof annotations, and only steps (2) and (3) are performed. For BONAI, we keep it all. Inspired during this annotation process, we use an ODE-based modelling to design our DragRoof.

To ensure the quality of dataset, each image was independently checked by another annotator. Conflicts larger than 10 pixels were re-annotated.

3.2. Properties of ObliCity

The ObliCity dataset comprises diverse urban scenes and multiple spatial resolutions, consisting of three parts: BONAI, IRSAMap, and UAV. The BONAI subset is sampled from rapidly developing cities in China, while the IRSAMap subset provides global coverage, including cities across different continents. Influenced by diverse cultural and historical developments, its urban layouts are more complex and heterogeneous compared to those of BONAI.

In comparison, the UAV subset, captured at an ultra-high spatial resolution (0.1 m), contains fine-grained urban and architectural textures. Its buildings often exhibit significantly longer RFOVs than those in BONAI and IRSAMap. During tiling, a larger sliding window (1536×1536) is required to ensure complete building coverage. Besides, as UAV imagery is captured from lower altitudes with limited ground coverage per frame, frequent mosaicking is needed to obtain large-area views. Consequently, buildings within the same tile may have RFOVs pointing in different directions, a phenomenon rarely

Table 1: Statistics of the ObliCity dataset.

Data	Res. (m)	Size	Train		Test	
			# Img.	# Ins.	# Img.	# Ins.
BONAI	0.3–0.5	1024×1024	3,000	247,683	300	21,280
IRSAMap	0.5	1024×1024	600	52,244	66	3,893
UAV	0.1	1536×1536	827	16,599	181	4,936
Total			4,427	316,526	547	30,109

observed in BONAI or IRSAMap. This difference can be found in Fig. 2.

3.3. Dataset Splits

The ObliCity dataset is divided into training and testing subsets. For the BONAI part, we retain its original split. The IRSAMap subset is divided into 600 training images and 66 testing images. For the UAV subset, data are grouped by seven distinct acquisition regions, with five regions used for training and the remaining two for testing. Table 1 summarizes the detailed composition of the ObliCity dataset.

4. Method: from DragOSM to DragRoof

4.1. Problem Setup

Given an oblique remote sensing image I, the task of projection-difference extraction aims to derive the roof-to-footprint offset vector (RFOV) for each pre-located building instance. In other words, it serves as a post-detection task of building detection that provides a more precise description of structural characteristics, *e.g.*, geometric configuration and relative height.

In this work, we propose to solve this problem by supervised learning of an RFOV extraction model with dataset $\mathcal{D}$, *i.e.*, $\mathcal{D} = \{(I_i, \mathbf{R}_i, \vec{\mathbf{O}}_i); i = 1, ..., N\}$, where $\mathbf{R}_i$ is pre-extracted building roofs in I_i, and $\vec{\mathbf{O}}_i$ is the RFOVs. We use $\{\mathbf{r}_j, \vec{o}_j; j = 1, ..., M\}$ to index each instance of $\mathbf{R}_i, \vec{\mathbf{O}}_i$ in I_i, and the target function of model is,

$$\vec{o}_j = \text{MODEL}(I_i, \mathbf{r}_j). \tag{1}$$

An example of this relationship is demonstrated in Fig. 1(c).

4.2. Preliminary: DragOSM

DragOSM was originally proposed to address the misalignment between historical building labels in maps and newly acquired imagery. Because these labels exhibit random positional discrepancies relative to the true building locations, DragOSM introduces 2D Gaussian noise to the ground-truth annotations during training to simulate such randomness, while jointly learning to correct it. During inference, the correction process is performed in two stages: (1) a continuous denoising procedure first eliminates the positional error between the historical label and the building footprint; and (2) the denoised footprint is then used to generate a roof polygon consistent with the image semantics. Therefore, applying DragOSM to RFOV extraction requires interpreting the initial roof input as the "historical label" in DragOSM's formulation. This process can be mathematically described as an SDE[3],

$$d\mathbf{X}_t = \mathbf{f}(\mathbf{X}_t, t)\, dt + \sigma\, d\mathbf{W}_t, \tag{2}$$

where $\mathbf{X}_t \in \mathbb{R}^{2l}$ represents the concatenated coordinates of l polygon vertices at time t, $\mathbf{W}_t$ is a Wiener process, and σ controls the intensity of the positional noise. The term $\mathbf{f}(\mathbf{X}_t, t)$ denotes the *alignment offset* generated by DragOSM. By discretizing Eq. 2 with step size Δt, we obtain the following iterative formulation:

$$\mathbf{X}_{t+\Delta t} = \mathbf{X}_t + \mathbf{f}(\mathbf{X}_t, t)\, \Delta t + \sigma \sqrt{\Delta t}\, \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3}$$

Let a_t absorb Δt and other scaling coefficients, and denote $\mathbf{q}(\mathbf{X}_t; \theta, I) = \mathbf{f}(\mathbf{X}_t, t)\, \Delta t$. Then Eq. (10) simplifies to the discrete update rule:

$$\mathbf{X}_{t+1} = \mathbf{X}_t + a_t\, \mathbf{q}(\mathbf{X}_t; \theta, I), \tag{4}$$

which mirrors the multi-step denoising process in DragOSM (Li et al., 2025b) with model weight θ, where $a_t = \delta^{t-1}$ is a geometrically decaying step size, and δ is a constant.

During both training and inference, the relationship between the model-level *alignment token* and the physical-level *alignment offset vector* is formulated as:

$$\vec{v}_e = \frac{\vec{v} - \vec{\bar{v}}}{\kappa}, \tag{5}$$

where κ is a scaling constant that normalises the physical vectors to a trainable range; $\vec{v}_e$ and $\vec{v}$ denote the encoded and decoded vectors, respectively; and $\vec{\bar{v}}$ represents the mean alignment centre in the latent space.

4.3. DragRoof

Motivation. Extracting RFOVs fundamentally differs from updating historical labels on new imagery, which requires an SDE model. Once the image and the corresponding building are fixed, the direction and magnitude of the RFOV become deterministic, directly governed by clear semantic and geometric cues in the image. This deterministic nature naturally aligns with an ODE-based formulation, *i.e.*, sampling the trajectory between roofs and footprints that simulates the human annotation procedure, and trains the model to predict the rest of the trajectory.

Model structure. As shown in Fig. 3, DragRoof adopts an similar structure as DragOSM (Li et al., 2025b). A Vision Transformer (ViT) (Dosovitskiy et al., 2020) is used as the visual encoder, while a multi-layer convolutional network encodes the input polygons $\mathbf{X}_t$. The encoded polygon features are concatenated with the two *alignment tokens* from the original DragOSM and fed into a two-way Transformer decoder to regress both the RFOV and the offset vector $\vec{v}_t$ from the current polygon $\mathbf{X}_t$ to the footprint. To allow the model to adaptively determine whether the current polygon has reached the

[3] Stochastic Differential Equation (SDE)

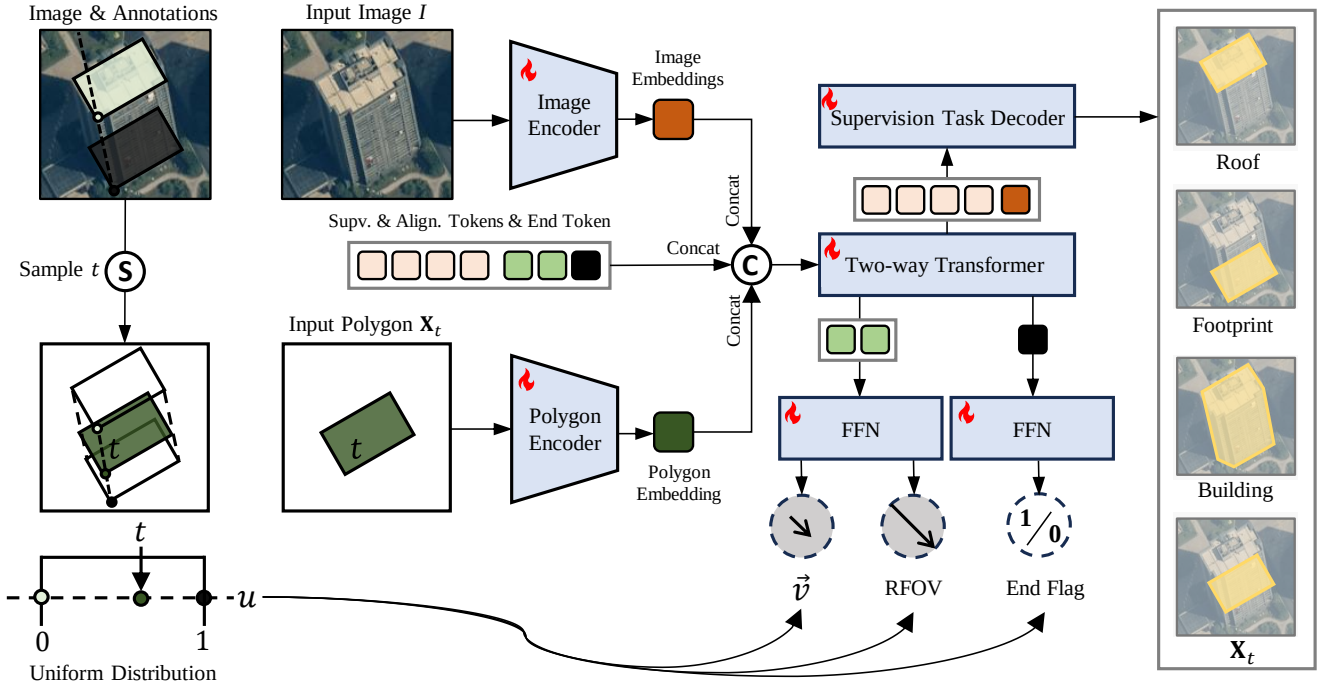


Figure 3: DragRoof in training. The system first samples a t to get the input polygon; then this polygon and image will be encoded as embeddings. These embeddings, together with the tokens, will transfer their features through a Two-way Transformer, and finally, they will be decoded into masks, offset vectors, and an end flag, whose ground truth values are generated during the sampling process. DragRoof simulates the human annotation process by iteratively adjusting roof positions toward footprints with facade constraints through an ODE-based sampling scheme.

footprint, we introduce an additional *end token*. The activation of this token is closely coupled with the magnitude of $\vec{v}_t$: it is decoded as 1 when $\|\vec{v}_t\|$ becomes sufficiently small, indicating convergence, and 0 otherwise. Finally, the alignment tokens and end token will be decoded separately with different Feed-Forward Networks (FFNs). Furthermore, the mask prediction branch of SAM is retained as an auxiliary task to segment the input polygon position, building roof, footprint, and overall building region, enabling DragRoof to better capture structural cues.

4.4. ODE Modelling

Motivation. As introduced in Sec. 4.3, the core design of DragRoof is inspired by the human annotation process. In this section, we integrate this concept with the mathematical formulation of an ODE to model the training process.

Continuous formulation & objective. Let $\mathbf{R} \in \mathbb{R}^{2l}$ denote the roof polygon (with l vertices) and $\mathbf{F} \in \mathbb{R}^{2l}$ the corresponding footprint. We define a trajectory $\mathbf{X}_t$ that continuously moves the roof toward the footprint:

$$\frac{d\mathbf{X}_t}{dt} = \mathbf{v}_\theta(\mathbf{X}_t, t; I, \mathbf{R}), \quad \mathbf{X}_0 = \mathbf{R},\ \mathbf{X}_1 = \mathbf{F}, \tag{6}$$

where $\mathbf{v}_\theta$ is a learnable velocity field predicted by the Transformer decoder of DragRoof, conditioned on both the image and the current polygon state. For the footprint location has a clear semantic feature among the given image, to accelerate the inference, This field encodes the instantaneous motion that progressively aligns the roof with its ground footprint, *i.e.*, $\mathbf{v}_\theta(\mathbf{X}_t, t; I, \mathbf{R}) = \vec{v}_t$. Because the displacement between roof and footprint is deterministic, we can sample the $\mathbf{X}_t$ by,

$$\mathbf{X}_t = \mathbf{R} + \phi(t)\,(\mathbf{F} - \mathbf{R}), \quad \phi(0) = 0,\ \phi(1) = 1. \tag{7}$$

In practice, $\phi(t)$ can be a linear schedule ($\phi(t) = t$), and t is sampled from a unified distribution $\mathcal{U}(0, 1)$ to simulate the insufficient prediction of model, *i.e.*, when $t = 0$, it represents the initial state, footprint label on roof, and when $t = 1$, it represents the roof label accurately placed on the footprint location; then, naturally the $\vec{v}_t = \mathbf{F} - \mathbf{R} - \phi(t)\,(\mathbf{F} - \mathbf{R})$. The network is trained to match this target velocity via *conditional flow matching*:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t\sim\mathcal{U}(0,1)} \left\| \mathbf{v}_\theta(\mathbf{X}_t, t; I, \mathbf{R}) - \vec{v}_t \right\|_1. \tag{8}$$

From Eq. 7, each $\mathbf{X}_t$ will be semantically attached to the building selected in the given image I. Thus, a global supervision is added to regress the RFOV. As a result, the loss function is,

$$\mathcal{L} = \alpha\mathcal{L}_{\text{CFM}} + \beta\mathcal{L}_{\text{RFOV}} + \gamma\mathcal{L}_{\text{END}} + \sum_{n=1}^{4} \lambda_n \mathcal{L}_n, \tag{9}$$

where $\mathcal{L}_{\text{RFOV}}$ denotes the smooth L1 Loss (Girshick, 2015) for the global RFOV supervision, $\mathcal{L}_{\text{END}}$ denotes the Binary CrossEntropy Loss for end token flag, and $\sum_{n=1}^{4} \mathcal{L}_n$ is the CrossEntropy Loss (Shannon, 1948) for the auxiliary mask tasks mentioned in section 4.3. In training, we use constants $\alpha, \beta, \gamma, \lambda$ to balance the losses.

Training. As a result, the training pipeline is as Fig. 3: given an image and a building roof annotation, the model will first

sample a t to get an interval polygon. From here, the ground truth of $\vec{v}$, RFOV and End Flag, used for supervised learning.

Then, the image and polygon will be encoded as embeddings. They will consequently be concatenated with supervise mask tokens, align tokens and end tokens. The following Two-way Transformer will weave their features. Finally, these tokens will be decoded as masks, $\vec{v}$, RFOV and end flag. The $\vec{v}$ and RFOV are encoded in the same way as Eq. 5 in DragOSM.

Inference. At inference, the model integrates Eq. (6) numerically using a K-step iterative system to get the predictions $\{\hat{\text{RFOV}}_k\}_{k=1}^K$, end flag $\{\hat{e}_k\}_{k=1}^K$, and the predicted $\{\hat{\vec{v}}_k\}_{k=1}^K$:

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \mathbf{v}_\theta(\mathbf{X}_k, t_k; I, \mathbf{R}). \tag{10}$$

Starting from $\mathbf{X}_0 = \mathbf{R}$, integration proceeds until K steps. Finally, the global prediction $\hat{\text{RFOV}}$ is defined as:

$$\hat{\text{RFOV}} = \sum_{k=1}^{K} \frac{g_1(\hat{e}_k)\hat{\text{RFOV}}_k + g_2(\hat{e}_k)\hat{\vec{v}}_k}{\sum_{k=1}^{K} g_1(\hat{e}_k) + 1}, \tag{11}$$

where $g_1(\cdot)$ and $g_2(\cdot)$ are selection strategies based on the value of $\hat{e}_k$, and their outputs are 0 or 1. In practice, we simply set both $g_1(\hat{e}_k)$ and $g_2(\hat{e}_k)$ after $\hat{e}_k > \eta$ as 0, otherwise 1, which means the end of the prediction. Specifically, if the model converges within a single inference step, RFOV_1 is taken as the final output.

5. Experiment

5.1. Implementation Detail

We use ViT-Base (Dosovitskiy et al., 2020) initialized from DragOSM (Li et al., 2025b) as the backbone. Both DragOSM and DragRoof are fully fine-tuned on our new dataset. All experiments are conducted on a single server equipped with seven NVIDIA RTX GPUs (24 GB each). We employ SGD with a weight decay of 0.0001 and a momentum of 0.9. The training runs for 48 epochs, with the learning rate warming up from 0 to 0.0025 over the first 500 iterations and decaying by a factor of 0.1 at the 32nd and 44th epochs. The η for $g_1(\cdot)$ and $g_2(\cdot)$ is set to 0.5. Our model contains 90.1M trainable parameters. During inference, the model achieves 0.85 FPS on the full test set ($\approx$ 1.17 seconds per image) using a single NVIDIA RTX 3090 GPU.

5.2. Evaluation Protocol

The evaluation of the RFOV extraction focuses on the quality of the predicted offset vectors. We assess performance from three complementary aspects between the predicted and ground-truth vectors: vector Angular Error (AE), vector Length Error (LE), and End-Point Error (EPE). For each prediction vector $\vec{o}_p$ and the ground-truth vector $\vec{o}_g$, the AE (in radian) is defined as the smaller angle between them, *i.e.*, the one less than 180°, while AE and LE (in pixels) are measured as:

$$\text{LE} = \left| \|\vec{o_g}\|_2 - \|\vec{o_p}\|_2 \right|_2, \tag{12}$$

$$\text{EPE} = \left\| \vec{o_g} - \vec{o_p} \right\|_2, \tag{13}$$

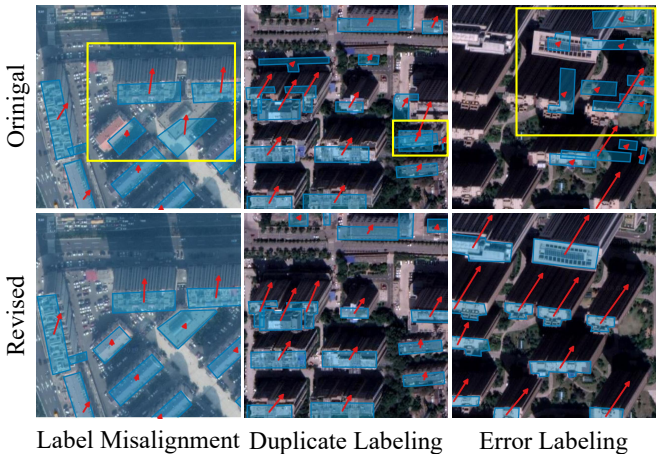


Figure 4: Examples of label revisions on the BONAI dataset. The selected cases show three common correction types: label-image misalignment, duplicate labeling, and error annotations. These revisions improve the consistency among roof masks, footprint masks, RFOVs, and the corresponding image content.

where $\|\cdot\|_2$ represents the 2nd Norm. Based on these three metrics, for a test set, we group the vectors based on the length of the ground truth. EPE as an example, we define EPE_i where $i = 10, 20, ..., 100$ to represent the average EPE of predicted vectors whose length of ground truth vector is between $[i-10, i)$; differently, EPE_∞ is used for the length interval $[100, \infty)$. Then, mEPE is defined as the mean of all EPE_i, and we use aEPE to denote the average EPE of all predictions in the test set. Finally, LE_i, AE_i, mLE, mAE, aLE, and aAE are defined in the same way.

5.3. Revisions on BONAI

Although the BONAI dataset includes roof masks, footprint masks, and RFOVs for most of the buildings, the labels suffer from problems, *e.g.*, missing, mistaken labeling, and label-image misalignments. We use the tool of the Sec. 3 to revise the labels of BONAI to improve the quality of the training set. Fig. 4 illustrates representative revision examples, including newly added, deleted, and modified labels. Based on statistics, 634 labels are newly added, 1,320 labels are deleted, and 11,912 labels are modified.

5.4. Comparisons

DragRoof is tailored for RFOV extraction. The function of the model needs to follow the Eq. 1. In this paper, we pick methods with a similar function: (1) **LOFT (Wang et al., 2023)** is a two-stage model based on Mask RCNN (He et al., 2017). It supports manually providing a roof region of interest (roof bounding box), and uses its feature-level offset augmentation module to extract the related RFOV. (2) **PolyFootNet (Li et al., 2025a)** is built upon OBM (Li et al., 2024a). It uses a mix-of-expert (MoE) system, named reference offset augmentation module, to predict RFOV in one step with given roof bounding boxes, and applies self-offset attention to improve the angle of RFOV. (3) **DragOSM (Li et al., 2025b)** is trained for map update under SDE assumption. It receives historical labels as

Table 2: Main results on ObliCity test set and its test sub-sets.

Model	Type	E_i↓											mE↓	aE↓			
		10	20	30	40	50	60	70	80	90	100	∞		ObliCity	BONAI	UAV	IRSAMap
LOFT (Wang et al., 2023)	EPE	6.91	7.30	10.01	15.86	19.23	24.14	**28.34**	**38.24**	34.83	39.24	142.77	33.35	10.61	8.21	26.03	4.56
	LE	4.81	4.51	**6.60**	10.59	13.49	**18.23**	**21.40**	**30.90**	28.22	26.85	128.63	26.75	7.89	5.91	20.03	**3.48**
	AE	0.80	0.38	0.32	0.38	0.38	0.39	0.36	0.44	0.33	0.41	0.70	0.45	0.49	0.44	0.68	0.50
PolyFootNet (Li et al., 2025a)	EPE	6.44	7.19	9.53	13.89	21.82	28.21	32.31	42.57	**32.37**	**36.80**	142.28	33.95	10.60	8.13	26.39	**4.27**
	LE	5.30	5.46	7.02	10.84	16.91	23.36	27.56	36.93	**27.27**	29.97	132.64	29.39	8.86	6.78	22.15	3.57
	AE	0.56	0.30	0.29	0.31	0.39	0.40	0.35	0.43	0.33	0.33	0.65	0.39	0.37	0.30	0.61	0.40
DragOSM (Li et al., 2025b)	EPE	5.92	8.86	13.33	20.24	28.23	39.36	49.91	61.55	67.88	75.00	118.42	44.43	11.00	10.96	17.61	5.97
	LE	**3.63**	6.45	10.74	16.70	24.03	35.91	46.74	57.24	61.39	69.95	111.03	40.35	8.72	8.67	14.41	4.43
	AE	0.88	0.48	0.38	0.43	0.42	0.43	0.45	0.56	0.54	0.52	0.52	0.51	0.57	0.56	0.55	0.62
Ours	EPE	**5.01**	**5.95**	**9.14**	**11.96**	**15.71**	**22.22**	30.53	44.11	38.98	44.17	**89.01**	**28.80**	**7.99**	**7.67**	**14.08**	4.59
	LE	3.76	**4.47**	7.31	**9.86**	**12.87**	19.94	28.59	40.42	36.20	40.19	**83.61**	**26.11**	**6.61**	**6.36**	**11.60**	3.73
	AE	**0.53**	**0.25**	**0.21**	**0.19**	**0.17**	**0.16**	**0.21**	**0.29**	**0.18**	**0.17**	**0.21**	**0.24**	**0.32**	**0.30**	**0.38**	**0.37**

Bold highlights the best; ↓: lower is better; E_i, mE, and aE represent the range, mean, and average metrics of EPE, LE, and AE, respectively. For DragOSM, we evaluate all denoising configurations and report the best results under the optimal hyper-parameter setting (1-step denoising).

prompts to predict two offsets to get roofs and footprints on updated images.

LOFT and PolyFootNet were originally designed to assist the manual annotation of building footprint masks, and therefore use boxes as inputs to simplify human interactions. In contrast, DragOSM and DragRoof are mask-based label correction models that take roof masks as input.

5.5. Results & Analyses

In Table 2, we evaluate the RFOV extraction performance of our proposed method on ObliCity. The results are reported separately: overall results and sub-set results.

Overall results. LOFT (Wang et al., 2023) predicts RFOVs by decoding the roof features cropped from the image feature map. However, these features are spatially limited and fail to cover the entire building region. As a result, although LOFT achieves slight advantages on a few specific metrics, its overall performance on the dataset remains the lowest. PolyFootNet (Li et al., 2025a) adopts a SAM-based architecture and refines the process of RFOV extraction. It employs a ROAM decoder, similar to a mixture-of-experts design, to handle RFOVs of varying lengths, and incorporates a SOFA module for global offset optimization. Consequently, PolyFootNet consistently ranks second across most evaluation metrics. DragOSM (Li et al., 2025b) takes roof masks as input, providing a more explicit spatial prior than the roof boxes used by other methods. However, since DragOSM is trained under an SDE-based 2D Gaussian assumption, the model may associate a given roof with an incorrect building footprint. As a result, only the global offset estimated during the first denoising step can effectively represent the RFOV, and the model reaches its upper-bound performance at this stage (refer to Sec. 6.1 and supplemental materials Sec. 6.3). Nevertheless, compared with LOFT and PolyFootNet, which rely on fixed roof boxes, DragOSM uses randomly distributed polygon inputs, leading to better performance in predicting ultra-long RFOVs (the E_∞ and the aE on UAV subset), although only its first step estimation is valuable.

In comparison, DragRoof leverages the fact that, given an image and a building prompt, the directional prior of the building offset is already implicitly encoded within the image. Accordingly, it adopts an ODE-based formulation for denoising, which enables the highest accuracy of predicting RFOV directions (better at all AEs). Moreover, the iterative inference ability allows DragRoof to achieve superior performance across almost all RFOV lengths. *e.g.*, its average EPE on the entire ObliCity dataset is 2.62 pixels lower than that of the second-best method, PolyFootNet.

Subset results. Across datasets, the UAV subset exhibits the highest aE, while IRSAMap shows the lowest, *e.g.*, aEPE of UAV predicted by DragRoof nearly doubled that on BONAI. This observation aligns with the characteristics of each dataset: the UAV subset is captured from aerial platforms, *e.g.*, drones, offering ultra-high spatial resolution. Consequently, the RFOVs in UAV imagery are much longer, *i.e.*, exceeding 100 pixels and sometimes approaching 1,000 pixels. In contrast, BONAI and IRSAMap mainly differ in sampling regions. BONAI focuses on densely populated metropolitan areas in China, whereas IRSAMap covers a diverse range of global cities. Overall, the difficulty of RFOV extraction appears to correlate more strongly with the length of the RFOV than with the architectural style: longer RFOVs are inherently more challenging to predict, while shorter ones are relatively easier.

Visualization & case study[4]**.** To enhance the understanding of the mentioned results, we visualised them in Fig. 5. Overall, the main challenge in RFOV extraction lies in predicting ultra-long offsets. As shown in Fig. 5, our model produces RFOVs that are more consistent with the ground truth in both direction and length, with many predictions nearly overlapping the annotations. The mask prompt offers a more explicit and spatially precise guidance than the box prompt, enabling more accurate feature localization. In previous works, RFOV prediction was often treated as an auxiliary task coupled with instance segmentation, which limited its independent optimization. *e.g.*, in the

[4]More visualizations & generalization test in supplemental materials

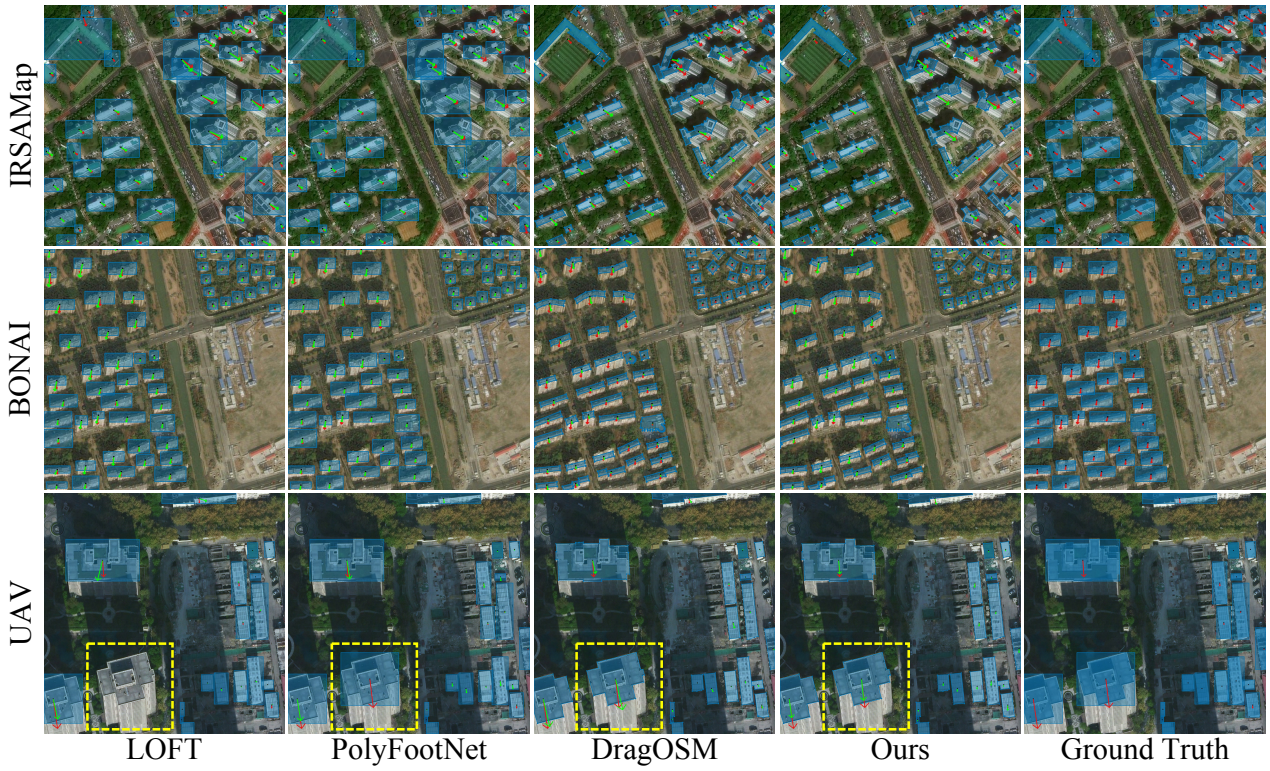


Figure 5: Visualized results on three parts of the ObliCity test set. Each example shows the input prompts (blue bounding boxes or polygons) and the model predictions (green arrows) overlaid on the ground-truth RFOVs (red arrows) for easier comparison across methods.

Table 3: Performance under varying flag thresholds on ObliCity.

Flag Threshold	0.5	0.6	0.7	0.8	0.9	0.95
aEPE	7.99	7.97	8.00	8.03	8.10	8.18
aLE	6.59	6.59	6.62	6.66	6.73	6.83
aAE	0.323	0.322	0.320	0.321	0.322	0.322

Table 4: The stop rate (%) in different flag thresholds over steps.

Step	Flag Threshold					
	0.5	0.6	0.7	0.8	0.9	0.95
1	96.80	94.72	91.30	85.28	70.39	48.95
2	98.82	97.79	95.94	92.57	82.46	62.71
3	99.42	98.72	97.58	95.40	88.32	72.52
4	99.70	99.20	98.33	96.69	91.02	77.29

failure case highlighted by the yellow box: LOFT, despite being provided with the ground-truth roof box, it misclassified the building as a negative sample, resulting in missing predictions. PolyFootNet, on the other hand, produced an incorrect RFOV, while DragOSM exhibited large directional deviations. In contrast, our model yields an almost perfectly aligned direction; however, the manually defined end flag threshold occasionally terminates inference too early, leading to slightly shorter RFOV lengths.

Ablations. Unlike the SDE assumption in DragOSM, where the footprint position is modelled as a Gaussian distribution centred at zero mean, the iterative correction process exhibits inherent stability. Specifically, as the number of iterations increases, the predicted displacement $\mathbf{X}_t$ tends to move toward the distribution centre. Even when $\mathbf{X}_t$ is initially assigned to an incorrect label position, the model can still converge toward the Gaussian mean under the 2D Gaussian assumption, enabling stable denoising for OSM-based inputs. In contrast, DragRoof adopts an ODE formulation with uniformly sampled priors, where $\mathbf{X}_t$ is distributed within the pixel range corresponding to the building area in the image. This means that if $\mathbf{X}_t$ falls outside the valid building region, the model, having never learned such cases, may produce unpredictable outputs that corrupt the ongoing inference process. To address this, we introduce an *End Flag* that enables the model to autonomously determine when to stop making predictions.

To verify our observation, we set different end-flag thresholds to control when DragRoof stops inference, *i.e.*, higher thresholds lead to later stopping steps. As shown in Table 3, a slight increase in the threshold initially improves performance; however, further increases gradually degrade it. Table 3 also reports the average number of steps before the model stops predicting under different thresholds, showing that DragRoof typically converges within two steps. In contrast, DragOSM (Li et al., 2025b) requires about five denoising steps to align the historical map with the updated image. As a result, combining the results in Table 2, where DragRoof outperforms DragOSM, this finding indicates that DragRoof achieves superior performance with fewer inference steps. This behaviour aligns well

Table 5: DragRoof with noised roof inputs.

IoU	mEPE	aEPE			
		Obli.	BON.	UAV	IRS.
1.00	28.80	7.99	7.67	14.08	4.59
0.88	29.26	8.09	7.82	14.02	4.57
0.65	29.36	8.16	7.89	14.02	4.71
0.32	30.80	8.53	8.20	14.23	5.45
0.18	33.18	9.04	8.71	14.81	5.92

Table 6: Decoupled vs. non-decoupled extraction.

Model	Roof			Footprint		
	Prec.	Rec.	F1	Prec.	Rec.	F1
LOFT	46.5	91.5	59.1	42.8	90.8	55.7
MLS.	56.5	86.1	66.2	53.9	84.6	63.8
Ours	69.4	76.8	71.9	64.6	73.4	67.7

with observations in image generation tasks, where ODE-based flow matching (Lipman et al., 2022) often attains comparable or better results than SDE-based DDPMs (Ho et al., 2020) using significantly fewer iterations.

On the other hand, both PolyFootNet and DragRoof are implemented upon SAM (Kirillov et al., 2023), but they differ in how RFOVs are extracted. PolyFootNet adopts a 4-head FFN decoder, resembling a MoE design. It first performs a forward inference to generate a coarse output and then refines the RFOV length through expert-specific predictions. In contrast, DragRoof requires only a single FFN and achieves more accurate RFOV estimation through two consecutive denoising steps, *i.e*., using only 1/4 parameters and the same number of inference steps, DragRoof attains a lower mean EPE (↓ 5.15 pixels) than PolyFootNet (Table 2).

Impact of roof robustness. Our experiments are conducted under the setting that reasonably accurate roof extraction is available. However, this assumption may not always hold in practical scenarios, since roof masks are typically obtained by external algorithms. Therefore, we add random noise to the ground-truth annotations of the ObliCity dataset to simulate varying roof-extraction quality (IoU) and analyze its impact on RFOV estimation.

As Tab. 5, as the edge noise of the input roof masks increases, the IoU between the input roofs and the ground truth decreases, and the RFOV extraction error increases accordingly. However, the EPE degradation is relatively mild compared to the significant drop in roof quality. *e.g*., when the IoU drops to 0.65, the aEPE on ObliCity increases by only 0.17 (↑ 2.1%). This suggests that, under the decoupled setting, although random noise is not explicitly injected into roof masks during training, the learned model exhibits strong robustness to inherent noise in the input roof masks.

Down-stream validation. We conduct quantitative downstream evaluations on the ObliCity by comparing roof and footprint extraction across models with only image inputs. Specifically, we use an HTC (≈ 156M) trained only on the IRSAMap roof dataset to perform roof extraction, and then apply DragRoof to estimate RFOV. In contrast, non-decoupled baselines, *i.e*., LOFT [TPAMI 2023] and MLS-BRN [CVPR2024] predict roofs and RFOV, and rely on soft-NMS to retain multiple detection candidates. Although these methods preserve a large number of predictions (high Recall), their overall precision is relatively low. Our decoupled formulation separates roof extraction and RFOV estimation into two independent stages, enabling more focused, task-specific training. This design likely contributes to the higher Precision and F1-score achieved by our method in the downstream roof and footprint extraction tasks (Tab. 6).

6. Discussion

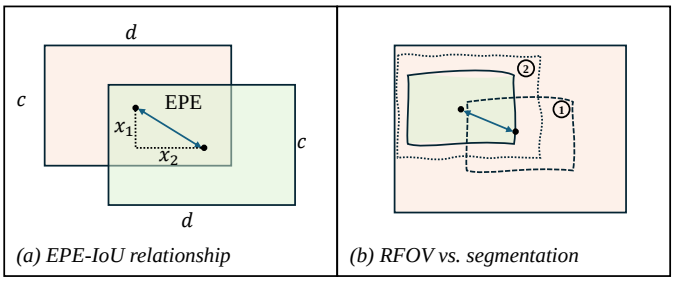
6.1. ODE Modelling for RFOV extraction

By simulating the human annotation process through ODE modeling, DragRoof achieves superior performance in both inference efficiency and RFOV accuracy. This effectiveness likely stems from the geometric path implicitly provided by the image itself: similar to how human annotators drag the roof mask along the building facade, DragRoof samples positions along this trajectory to mimic intermediate dragging states. The given image and facade structures thus provide explicit geometric guidance, which may serve as the key to its strong directional awareness. In contrast, DragOSM is trained under an SDE formulation that assumes a 2D Gaussian noise distribution over positional space, without establishing a one-to-one correspondence between the input roof and its footprint. Consequently, when surrounded by multiple adjacent buildings, the model often fails to identify the correct roof–footprint pairing, making only the first denoising step effective for valid RFOV prediction. We visualized this in Sec. 6.3.

6.2. RFOV Extraction vs. *Footprint Mask Metrics*

In previous studies, RFOV extraction was often treated as an auxiliary task for building footprint extraction in off-nadir remote sensing images, primarily aimed at improving footprint accuracy. Consequently, these works mainly evaluated performance using footprint mask metrics. The footprint mask itself, however, is obtained through a two-stage process: first, roof masks are produced via semantic segmentation; then, RFOVs are predicted to shift the roof masks, generating the final footprint masks. While it may appear that the quality of the footprint mask is directly correlated with RFOV accuracy, this is not necessarily the case.

As Fig. 6, there exist cases where improvements in RFOV prediction have little to no impact on footprint mask metrics. For case Fig. 6(a), we take footprint mask IoU as an example: assuming that the predicted and ground-truth footprint masks share identical rectangular outlines with side lengths c and d, the current EPE can be expressed as EPE $= \sqrt{x_1^2 + x_2^2}$, where (x_1, x_2) denotes the displacement between the predicted



(a) EPE-IoU relationship (b) RFOV vs. segmentation

Figure 6: Two typical cases of footprint mask metrics fail to reflect RFOV improvement.

and ground-truth footprint centers. Consequently, the footprint mask IoU is expressed as:

$$\text{IoU} = \frac{(c - x_1)(d - x_2)}{(c + x_1)(d + x_2) - 2x_1x_2}. \tag{14}$$

We fix the c, d, x_2 as constant in Eq. 14 to study the relationship between IoU and EPE with x_1, then:

$$\text{IoU} = \frac{c(d - x_2) - (d - x_2)x_1}{c(d + x_2) + (d - x_2)x_1}. \tag{15}$$

In this equation, $c > x_1$ and $d > x_2$; obviously, the EPE will drop with the decrease of x_1, and IoU will increase together, but the decrease of EPE and the increase of IoU were not linearly bound: when the area of the footprint mask is large enough (c, d is large), the gain from the decreased EPE will be very slight. That is one of the reasons why we append ultra-long RFOV annotations with UAV images.

Case Fig. 6(b) is another practical problem when using footprint metrics to measure off-nadir algorithms. One of the common discoveries in predicting roof masks is that the roof prediction is always smaller than the ground-truth roof within its boundary. This made even a fully correct RFOV is provided as ①, the footprint mask metrics will have no change, but if we use a more advanced roof segmentation method under the same RFOV as ②, the footprint mask IoU will be tremendously improved. However, the importance of accurate RFOV extraction goes far beyond achieving precise footprint localisation, as it provides richer geometric cues for understanding building structure, height, *etc*.. Moreover, in other non-ideal and unstable roof prediction cases, the impact of advances in RFOV extraction on footprint metrics becomes more complex and less predictable.

On the other hand, RFOV-bound mask metrics are highly sensitive to post-processing steps commonly used in instance segmentation, such as Non-Maximum Suppression. When a strict Non-Maximum Suppression strategy is applied to segmentation results, incorrectly removed masks will be directly reflected in the mask evaluation scores. However, this is not the case for RFOV. If predictions with longer offsets are removed, the evaluation will naturally shift toward shorter offsets, which inherently exhibit smaller errors. *e.g*., when the ground truth offset is 2 pixels, the error of a shorter offset prediction may fall within 1 pixel. Such coupling fails to provide a faithful and robust assessment of RFOV extraction performance.

That is why we said that RFOV extraction has often been overshadowed by advances in roof segmentation performance, and it is time to consider it independently.

6.3. *ODE* vs. *SDE*

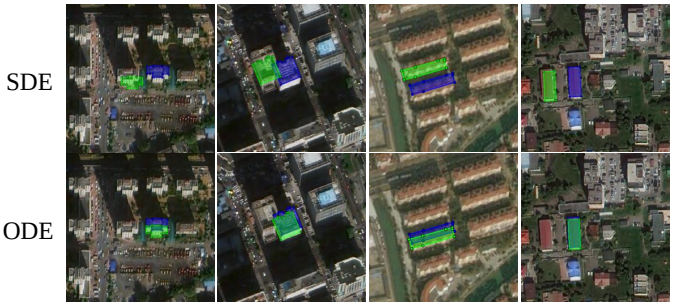


Figure 7: Comparison between DragOSM and DragRoof. The blue polygon is the location of input polygons, and the green polygons are the position of the final outputs under the predicted offset vectors.

DragOSM was originally proposed to address the misalignment between vector maps and newly captured imagery. Since the relative positions of building labels in historical maps are often uncertain, DragOSM models the spatial relationship between image buildings and historical annotations using a 2D Gaussian assumption during training. In contrast, DragRoof is specifically designed to handle projection differences in off-nadir imagery. It learns to correct such displacements by imitating how human annotators drag roofs along building facades toward their footprints. From a mathematical perspective, DragOSM and DragRoof can be respectively categorized as SDE-based and ODE-based formulations.

However, learning under the SDE formulation corresponds to a Wiener process, whereas the ODE formulation follows a deterministic trajectory. As illustrated in Fig. 7, we visualize the starting and ending points of the dragging process to highlight this distinction. Under the SDE assumption, the absence of an explicit dragging path causes the input roof to be influenced by surrounding buildings, which may lead the model to misassociate the roof with incorrect footprint labels and drift toward neighboring structures. That is the reason why only the first step inference of DragOSM is reliable for extracting the RFOV.

In contrast, the ODE formulation leverages the well-defined geometric contours along building facades to provide a clear path condition during inference, effectively eliminating such mismatched correspondences.

6.4. *Value & Future*

Projection displacement in off-nadir building imagery is a common phenomenon in high-resolution remote sensing, and it becomes increasingly significant as image resolution improves. The diversity of camera viewing angles has rendered many traditional building extraction approaches less effective, yet most existing studies still overlook this displacement.

We argue that decoupling RFOV extraction as an independent task provides a more principled and practical framework

for modelling such geometric effects, making off-nadir building analysis more feasible and physically consistent. Freed from the constraints of semantic segmentation, RFOV can evolve into a standalone research direction that substantially advances urban remote sensing.

7. Conclusion

In this work, we formalised RFOV extraction as an independent task for correcting projection displacements in oblique remote sensing images. We introduced the ObliCity, the first large-scale benchmark that combines UAV and global satellite imagery with diverse resolutions and viewing angles. By incorporating imitation learning to emulate human annotation behaviors, we propose DragRoof, an ODE-based reformulation of DragOSM that learns deterministic, geometry-consistent offset fields, achieving state-of-the-art performance with fewer inference steps. We hope ObliCity and DragRoof will establish a strong foundation for future research on projection correction in monocular remote sensing imagery.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of generative AI use

During the preparation of this manuscript, the authors used generative AI tools (*e.g.*, ChatGPT) solely for language polishing and grammar improvement. The authors reviewed and edited the output and take full responsibility for the content of the manuscript.

References

Cao, Y., Huang, X., 2021. A deep learning method for building height estimation using high-resolution multi-view imagery over urban areas: A case study of 42 chinese cities. Remote Sens. Environ. 264, 112590.

Chen, S., Shi, Y., Xiong, Z., Zhu, X.X., 2023. Htc-dc net: Monocular height estimation from single remote sensing images. IEEE Trans. Geosci. Remote Sens. 61, 1–18.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale, in: ICLR.

Girshick, R., 2015. Fast r-cnn, in: ICCV, pp. 1440–1448.

Gröger, G., Kolbe, T.H., Nagel, C., Häfele, K.H., 2012. Ogc city geography markup language (citygml) encoding standard .

Gültekin, F., Koz, A., Bahmanyar, R., Azimi, S.M., Süzen, M.L., 2025. Fusing convolution and vision transformer encoders for object height estimation from monocular satellite and aerial images, in: Int. Conf. Comput. Vis. Worksh., pp. 3709–3718.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN, in: ICCV, pp. 2980–2988.

Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (Eds.), NeurIPS, Curran Associates, Inc.. pp. 6840–6851.

Ji, S., Wei, S., Lu, M., 2018. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. IEEE Transactions on geoscience and remote sensing 57, 574–586.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al., 2023. Segment anything, in: ICCV, pp. 4015–4026.

Li, K., Deng, Y., Chen, J., Meng, Y., Xi, Z., Ma, J., Wang, C., Wang, M., Zhao, X., 2025a. Polyfootnet: Extracting polygonal building footprints in off-nadir remote sensing images. IEEE Trans. Geosci. Remote Sens. 63, 1–16. doi:10.1109/TGRS.2025.3590054.

Li, K., Deng, Y., Kong, Y., Liu, D., Chen, J., Meng, Y., Ma, J., Wang, C., 2024a. Prompt-driven building footprint extraction in aerial images with offset-building model. IEEE Trans. Geosci. Remote Sens. , 1–16doi:10.1109/TGRS.2024.3487652.

Li, K., Weng, X., Deng, Y., Meng, Y., Pang, C., Xia, G.S., Zhao, X., 2025b. Dragosm: Extract building roofs and footprints from aerial images by aligning historical labels. arXiv preprint arXiv:2509.17951 .

Li, Q., Mou, L., Sun, Y., Hua, Y., Shi, Y., Zhu, X.X., 2024b. A review of building extraction from remote sensing imagery: Geometrical structures and semantic attributes. IEEE Trans. Geosci. Remote Sens. 62, 1–15. doi:10.1109/TGRS.2024.3369723.

Li, W., Lai, Y., Xu, L., Xiangli, Y., Yu, J., He, C., Xia, G.S., Lin, D., 2023. Omnicity: Omnipotent city understanding with multi-level and multi-view images, in: CVPR, pp. 17397–17407.

Li, W., Meng, L., Wang, J., He, C., Xia, G.S., Lin, D., 2021. 3d building reconstruction from monocular remote sensing images, in: ICCV, pp. 12548–12557.

Li, W., Yang, H., Hu, Z., Zheng, J., Xia, G.S., He, C., 2024c. 3d building reconstruction from monocular remote sensing images with multi-level supervisions, in: CVPR, pp. 27728–27737.

Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M., 2022. Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 .

Meng, Y., Deng, L., Xi, Z., Chen, J., Chen, J., Yue, A., Liu, D., Li, K., Wang, C., Li, K., et al., 2025. Irsamap: Towards large-scale, high-resolution land cover map vectorization. IEEE Trans. Geosci. Remote Sens. .

Pang, C., Wu, J., Ding, J., Song, C., Xia, G.S., 2023. Detecting building changes with off-nadir aerial images. Sci. China Inf. Sci. 66, 140306.

Shannon, C.E., 1948. A mathematical theory of communication. Bell Syst. Tech. J. 27, 379–423.

Sun, X., Huang, X., Mao, Y., Sheng, T., Li, J., Wang, Z., Lu, X., Ma, X., Tang, D., Chen, K., 2024. Gable: A first fine-grained 3d building model of china on a national scale from very high resolution satellite imagery. Remote Sens. Environ. 305, 114057.

Van Etten, A., Lindenbaum, D., Bacastow, T.M., 2018. Spacenet: A remote sensing dataset and challenge series. arXiv preprint arXiv:1807.01232 .

Wang, C., Chen, J., Meng, Y., Deng, Y., Li, K., Kong, Y., 2024. Sampolybuild: Adapting the segment anything model for polygonal building extraction. ISPRS J. Photogramm. Remote Sens. 218, 707–720. doi:https://doi.org/10.1016/j.isprsjprs.2024.09.018.

Wang, J., Meng, L., Li, W., Yang, W., Yu, L., Xia, G.S., 2023. Learning to Extract Building Footprints From Off-Nadir Aerial Images. IEEE TPAMI 45, 1294–1301.

Wang, L., Fang, S., Meng, X., Li, R., 2022. Building extraction with vision transformer. IEEE Trans. Geosci. Remote Sens. 60, 1–11. doi:10.1109/TGRS.2022.3186634.

Wei, S., Zhang, T., Ji, S., Luo, M., Gong, J., 2023. Buildmapper: A fully learnable framework for vectorized building contour extraction. ISPRS journal of photogrammetry and remote sensing 197, 87–104.

Xu, B., Xu, J., Xue, N., Xia, G.S., 2023. Hisup: Accurate polygonal mapping of buildings in satellite imagery with hierarchical supervision. ISPRS J. Photogramm. Remote Sens. 198, 284–296.

Yu, W., Zhang, T., Ji, S., Zhang, K., Liu, B., Liu, H., Gong, J., 2025. P2pformerv2: Improving primitive-based regular building contour extraction methods via contour feature enhancement. IEEE Trans. Geosci. Remote Sens. , 1–1doi:10.1109/TGRS.2025.3620903.

Zhou, Y., Jiang, W., Wang, B., 2025. Nesf-net: Building roof and facade segmentation based on neighborhood relationship awareness and scale-frequency modulation network for high-resolution remote sensing images. ISPRS J. Photogramm. Remote Sens. 226, 247–266.