

GA-DPNet: A Geometry-Aware Dense Prediction Network for Monocular Height Estimation from Remote Sensing Images

Fan Yang¹, Yuecheng Yu^{1†}, Lizhou Huang^{2,3}, Yuanju Lu¹, Jiasong Shi^{2,3}, Jiemin Zhang^{2,3}

¹*School of Computer Science, Jiangsu University of Science and Technology, Zhenjiang, 212100, China*

²*State Key Laboratory of Chemistry for NBC Hazards Protection, Beijing 102205, China*

³*Chemical Defense Institute, Academy of Military Sciences, Beijing 102205, China*

Abstract

Monocular height estimation from remote sensing images plays a crucial role in urban planning, 3D reconstruction, and environmental monitoring. However, existing monocular height estimation networks primarily rely on implicit semantic features in remote sensing images, while neglecting the geometric structural information of ground objects. This limitation reduces the edge-preserving capability and physical plausibility of ground objects in complex scenes. To address this issue, we propose a Geometry-Aware Dense Prediction Network for Monocular Height Estimation from Remote Sensing Images (GA-DPNet). Firstly, building upon the introduction of DINOv2 as the encoder network, the Multi-Scale Geometric Alignment (MSGa) module is designed to reduce the inconsistency in global geometric space among features extracted at different levels by DINOv2. Secondly, the Geometry-Aware Feature Fusion Block (GAFFB) is designed, which includes Geometric Feature Extractor (GFE), Geometry-Aware Attention module (GAA), and Geometric Modulated Residual module (GMR). By extracting four geometric features including gradient, curvature, planarity, and edges to modulate attention weights, GAFFB improves the effectiveness of fusing multi-scale geometric information with semantic features during the decoding process. Finally, the Multi-component Geometric Constraint Loss (MGCL) function is designed, including geometric consistency loss and physical constraint loss, to enhance the geometric plausibility of the network's predictions. Experimental results on three public datasets, Vaihingen, Potsdam, and DFC2019, show that GA-DPNet achieves MAEs of 0.783, 1.022, and 0.835, respectively, demonstrating superior performance in terms of height estimation accuracy and edge preservation.

Keywords: height estimation; remote sensing images; dense prediction; geometry-aware; multi-scale fusion

1. INTRODUCTION

Height estimation from remote sensing images is one of the core tasks in computer vision and remote sensing. Height maps generated through height estimation are typically presented in the form of Digital Surface Models (DSM) or normalized Digital Surface Models (nDSM), which

[†]Corresponding author: Andrew Smith (Email: zhjyuyuecheng@163.com; ORCID: 0000-0002-7257-6880)

have significant application value in urban planning, disaster monitoring, 3D reconstruction, and other domains [1][2][3]. Traditional height map generation methods primarily consist of two major approaches, stereo image matching and Light Detection and Ranging (LiDAR) measurement. Stereo image matching methods achieve direct inference of depth information by computing the disparity between multi-view images. However, the acquisition of stereo images relies on precise camera calibration, and the image registration process requires images with good texture matching conditions [4]. As for LiDAR measurement methods, although LiDAR can provide high-precision 3D information, the complex data acquisition process requires expensive hardware equipment [5]. In contrast, monocular height estimation from remote sensing images can accomplish height inference using only ordinary optical images, offering significant advantages such as low data acquisition cost, high processing efficiency, and wide application scope [6].

Traditional monocular height estimation from remote sensing images mainly relies on hand-crafted features and rules such as shadow analysis and building geometry priors [7][8]. However, these methods have weak generalization ability in complex scenes due to the influence of lighting conditions and shadow visibility. With the rapid development of deep learning technology, particularly the continuous maturation of Convolutional Neural Network (CNN) and Vision Transformer (ViT) architectures [9], deep learning methods have become the mainstream approach for height estimation.

Early deep learning methods primarily adopted encoder-decoder architectures to leverage CNNs to obtain height information from monocular images in a regression manner. Mou et al. [10] designed a method for estimating target heights from monocular remote sensing images using an end-to-end fully residual convolutional-deconvolutional network. Similarly, Amirkolaei et al. [11] and Liu et al. [12] further improved height prediction accuracy by adopting residual learning and skip connection operations. To further preserve boundary details of objects, Xing et al. [13] introduced a gated feature aggregation module that estimates height information in a coarse-to-fine progressive manner by fusing multi-scale features. However, since the convolutional operations of CNNs can only capture feature information within local receptive fields, they struggle to effectively capture long-range spatial dependencies. Consequently, such methods find it difficult to effectively establish global spatial correlations when predicting heights of large-scale buildings and understanding the panoramic layout of buildings in complex scenes, which limits the global consistency of height prediction and scene understanding.

To overcome the limitations of CNNs in global modeling capability, Vision Transformers have been introduced into height estimation tasks to capture long-range dependencies between patches through their self-attention mechanism. When constructing a Transformer-based height estimation network, Chen et al. [14] transformed height estimation into a classification-regression task by utilizing a head-tail cutting strategy and distribution-based constraints, reducing the adverse impact of long-tail distribution on height estimation. Similar to [14], Chen et al. [15] achieved dynamic interaction between pixel-level features and patch-level features by designing a Multilevel Interactive Backbone, better capturing long-range dependencies between patches. Furthermore, to mitigate the loss of local information during feature extraction, Wu et al. [16] employed a multi-scale vision encoder-decoder and a bilateral feature pyramid fusion strategy, enhancing the network's comprehensive capability to simultaneously capture global and local information. However, the introduction of Transformers generates a receptive field accumulation effect [17], which may lead to different parts of the same building being recognized as spatially separated independent units. The fundamental reason for this phenomenon is that deep-layer features tend to encode complete buildings as a whole, while shallow-layer features primarily respond to local texture and edge information. Consequently, during multi-scale fu-

sion, this inconsistency in spatial cognition leads to spatial misalignment and feature conflicts, which inevitably affects the localization accuracy of ground object edges and the overall geometric consistency of ground objects.

In fact, height variations within the same semantic objects are typically gradual and continuous, while height variations between different semantic objects are more drastic. Evidently, introducing semantic information as prior knowledge into height estimation networks can contribute to improving model performance. The multi-task learning framework proposed by Carvalho et al. [18], which simultaneously estimates object heights and object categories, demonstrated the promoting effect of semantic information on height estimation. To better learn the height differences of ground objects across different categories, Xing et al. [19] designed a feature separation-fusion module that utilizes attention mechanisms to enhance the representation capability of cross-task features, achieving effective separation and fusion of height features and semantic features. Similarly, Gao et al. [20] further introduced cross-task and cross-pixel contrastive loss to enhance feature discriminability by maximizing mutual information between different task features. Different from the previous two methods, Du et al. [21] directly incorporated land cover priors as additional input. Subsequently, they designed an attention mechanism based on land cover data within skip connections, enabling the model to consider category information of ground objects during elevation computation. Peng et al. [22] proposed a Cross-Task Mutual Enhancement Network (CTME-Net), which achieves bidirectional feature interaction and collaborative optimization between semantic segmentation and height estimation through a designed cross-task attention module, further exploiting the latent associations between the two tasks. However, these semantic prior-based methods primarily assist height prediction by identifying category information such as buildings, roads, and vegetation, while neglecting the intrinsic geometric structural information of ground objects. In remote sensing images, ground objects possess explicit geometric attributes, including roof planarity, edge height discontinuities, and curvature changes of adjacent surfaces. In multi-task learning frameworks, failing to consider the impact of these geometric features on height estimation leads to homogenization of pixel height estimation within the same semantic regions, ignoring the objectively existing height variations within the same category of ground objects. Therefore, considering the suppressive effect of semantic labels on detail variations within homogeneous regions, Lei et al. [23] designed a single-task framework that employs progressive relation mining and edge-preserving modules to capture subtle height variations within the same semantic regions.

Most of the aforementioned methods treat height estimation as a regression problem. Since most of these methods adopt MSE or smoothing L1 as the loss function, these methods can only focus on pixel-level numerical errors. This results in the inability of these methods to fully consider the physical constraints and geometric properties of ground objects in the real world, including the flatness of building roofs, the rationality of terrain slopes, and the high continuity of adjacent areas. The core idea of GA-DPNet is to integrate geometric information throughout the entire network architecture, enhancing the network's ability to perceive the geometric structure of ground objects in feature processing, multi-scale feature fusion, and loss function design. The main contributions of this paper are as follows:

- 1) A Multi-Scale Geometric Alignment (MSGa) module is proposed. It leverages the global perception capability of deep-layer features to generate unified spatial attention signals. This enables features at different levels to achieve consistency in global spatial cognition, effectively alleviating the spatial misalignment problem in multi-scale fusion.

- 2) A Geometry-Aware Feature Fusion Block (GAFFB) is designed. It employs a Geometric Feature Extractor (GFE) to explicitly extract four types of geometric features: gradient, curva-

ture, planarity, and edges. These features are used to generate geometric weights. A Geometry-Aware Attention module (GAA) and a Geometric Modulated Residual module (GMR) are then designed to integrate these geometric weights into attention modulation and feature propagation processes, achieving deep fusion of geometric information with semantic features.

3) A Multi-component Geometric Constraint Loss (MGCL) function is constructed. It includes basic regression loss, geometric consistency loss, and physical constraint loss, comprehensively ensuring the rationality of prediction results from three aspects: pixel accuracy, geometric structure, and physical laws.

2. RELATED WORK

2.1. DINOv2

Compared with CNNs, ViT leverages its global modeling capability to capture spatial relationships at arbitrary distances [24]. Global modeling capability is crucial for understanding large-scale spatial structures in remote sensing images [25]. However, the training of ViT relies on a large amount of labeled data. To reduce the model's dependence on labeled data, DINO [26] employs self-distillation to train Vision Transformers. In DINO, the student network learns from the output of the teacher network, while the teacher model's parameters are updated through the Exponential Moving Average (EMA) method. To avoid the loss of feature discriminability caused by mode collapse, DINO adopts centering and sharpening operations. On one hand, this prevents excessive amplification of feature dimensions, and on the other hand, it enhances feature distinctiveness. The features learned by DINO can automatically focus on salient objects, and the attention maps can accurately identify foreground and background. Compared with supervised pre-trained models, DINO demonstrates better performance on downstream tasks.

As an upgraded version of DINO, DINOv2 [27] utilizes a larger-scale training dataset and exhibits better adaptability to objects at different scales. Therefore, DINOv2 can be directly applied to various downstream tasks without extensive fine-tuning. In recent years, DINOv2 has been widely used as a backbone network for semantic segmentation and scene classification of remote sensing images [28][29][30]. Consequently, we select DINOv2 as the feature extraction network to fully leverage its powerful representation capability and generalization performance for improving the accuracy of height estimation from remote sensing images.

2.2. Dense Prediction

Monocular height estimation from remote sensing images belongs to dense prediction tasks. Dense prediction methods adopting the U-Net [31] architecture achieve multi-scale feature fusion by directly transmitting shallow features from the encoder to corresponding layers in the decoder through skip connections. However, skip connections only employ simple feature concatenation, which may lead to feature conflicts caused by semantic inconsistency. To better fuse multi-scale features, progressive decoding strategies have been widely adopted. Feature Pyramid Network (FPN) [32] progressively transmits deep semantic features to shallow features through a top-down pathway, while fusing encoder features through lateral connections at corresponding feature layers. Path Aggregation Network (PANet) [33] adds a bottom-up pathway on the basis of FPN, achieving bidirectional information flow and further enhancing the effectiveness of feature fusion. Dense Prediction Transformer (DPT) [34] introduces Transformers while retaining CNNs, and applies the progressive decoding strategy to the Transformer architecture as well. The decoder of DPT adopts a bottom-up progressive fusion strategy, where each fusion

module includes a Residual Convolutional Unit (RCU) to fuse features through two convolutions and residual connections. This design can progressively refine features, recovering local details while maintaining global consistency. The success of DPT has inspired a series of subsequent works. Song et al. [35] combined knowledge distillation methods with DPT and designed a depth estimation model based on self-supervised learning. Gąsienica-Józkowy et al. [36] implemented a monocular distance and height estimation method for edge devices based on the DPT model.

In remote sensing image height estimation tasks, ground objects range from meter-scale buildings to kilometer-scale terrains. On one hand, since Transformers can effectively characterize long-range spatial dependencies, DPT can better capture the overall undulation trends of large-scale terrains. On the other hand, since the progressive decoding strategy of DPT can fuse multi-scale features in stages, height estimation using DPT can maintain the global consistency of ground objects while accurately capturing local detail information such as building edges and terrain discontinuities.

2.3. *Geometry-Aware*

Geometric features describe the spatial structure and shape information of images, and can explicitly characterize spatial attributes such as object positions, boundary shapes, and surface curvatures, providing important spatial prior knowledge for computer vision tasks. In the field of depth estimation, researchers have explored the utilization of geometric information from multiple perspectives. Godard et al. [37] introduced image gradient loss, effectively improving the sharpness of depth edges. Ramamonjisoa et al. [38] significantly improved reconstruction accuracy in boundary regions by jointly constraining normal vectors, depth, and occlusion contours. Patil et al. [39] introduced piecewise planarity priors and optimized depth estimation tasks in indoor scenes by leveraging planar coefficients of coplanar pixels based on coplanar pixel identification. Yang et al. [40] treated depth estimation and surface normal estimation as joint tasks, and utilized the Transformer architecture to enhance the algorithm's capability to capture global geometric structural information. It can be seen that the reasonable application of geometric constraints can effectively improve the accuracy of recovering 3D spatial information from 2D images.

Similar to depth estimation methods, geometric features also significantly promote the understanding of spatial structures of buildings. However, most existing monocular height estimation methods from remote sensing images still primarily rely on semantic features, neglecting the important role of geometric features in estimating heights at key regions such as building edges and roof contours. Although some methods have introduced gradient-based or edge-based constraint terms [23], they lack comprehensive modeling and adaptive fusion of multiple geometric features such as gradient, curvature, planarity, and edges, and cannot fully exploit the complementarity and synergy among different geometric features. Therefore, we explicitly extract and fuse multiple types of geometric features, integrating geometric feature weights into attention mechanisms and feature propagation processes to enhance the network's perception of ground object geometric structures.

3. METHODOLOGY

3.1. *Overall Framework*

The overall architecture of GA-DPNet is shown in Figure 1. GA-DPNet adopts an encoder-decoder architecture, which consists of three stages: feature extraction, feature processing, and

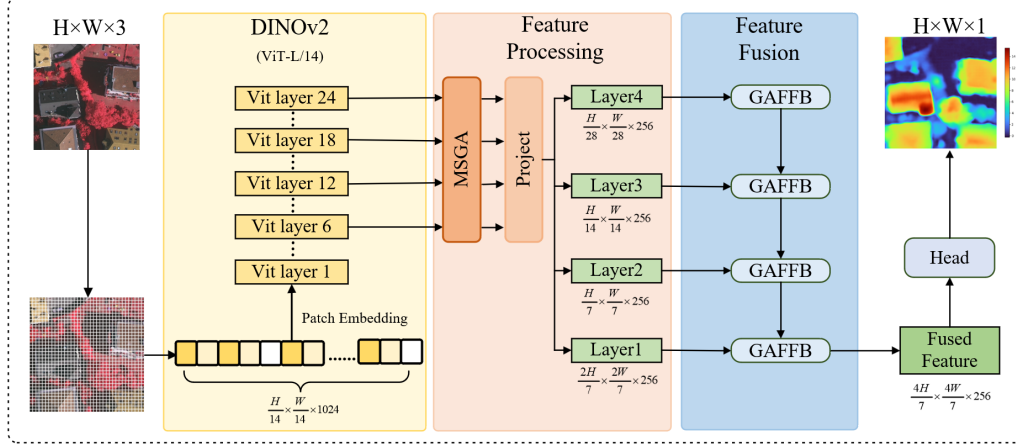


Figure 1. Overall architecture of GA-DPNet.

feature fusion. In GA-DPNet, geometric modeling mechanisms are explicitly introduced at multiple levels, enabling the network to explicitly model the geometric structural attributes of ground objects while focusing on semantic information. Thus, GA-DPNet can maintain the global consistency of height estimation results while improving the sharpness of ground object edges and local details.

For a given input RGB remote sensing image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, the goal of GA-DPNet is to predict the corresponding height map $\mathbf{H} \in \mathbb{R}^{H \times W \times 1}$. Specifically, in the feature extraction stage, the pre-trained DINOv2-Large is used as the backbone network to extract intermediate features at four scales, capturing features from fine-grained details to high-level semantics. In the feature processing stage, to address the inconsistency in spatial cognition among features at different levels, we design a Multi-Scale Geometric Alignment (MSGA) module. Firstly, leveraging the global perception capability of deep-layer features, MSGA generates a unified spatial attention prior and propagates it to all levels. Subsequently, through channel projection and size adjustment, MSGA continues to construct a spatially aligned and channel-unified feature pyramid. In the feature fusion stage, following a bottom-up progressive fusion strategy, we design a Geometry-Aware Feature Fusion Block (GAFFB). Through the synergy of the Geometric Feature Extractor (GFE), Geometry-Aware Attention module (GAA), and Geometric Modulated Residual module (GMR), GAFFB is utilized for deep coupling of geometric priors with semantic features. Finally, the height prediction head generates full-resolution height predictions after performing progressive upsampling and convolutional processing on the fused features.

3.2. Encoding and Feature Processing

As a self-supervised learning model based on the Vision Transformer (ViT), DINOv2 adopts an improved self-distillation training strategy. Unlike traditional supervised learning methods, DINOv2 can acquire universal visual representation capabilities by learning the intrinsic structures and patterns of images. This enables DINOv2 to demonstrate excellent performance across various downstream tasks [41]. In GA-DPNet, we select DINOv2-Large as the backbone network of the encoder to accommodate training on a single GPU. DINOv2 contains 24 Transformer layers, each capable of representing 1024-dimensional features. For a given RGB remote

sensing image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, DINOv2 first divides the input image into non-overlapping patches with a size of 14×14 . Subsequently, each patch is mapped to a high-dimensional feature space through a linear projection layer. To obtain uniformly distributed and semantically hierarchical feature representations of the image, we adopt an interval sampling strategy. That is, intermediate features are extracted from different depth layers of DINOv2 to ensure that the intermediate features can better represent the semantic information in the image. Specifically, we select intermediate features extracted from the 6th, 12th, 18th, and 24th layers as multi-scale features $\{\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4\}$. Among them, the 6th layer represents shallow features, the 12th and 18th layers represent middle-layer features, and the 24th layer represents deep features. Shallow features retain more spatial detail information, which is beneficial for edge and texture recovery. Middle-layer features provide a good balance between semantic and geometric information. Deep features contain high-level semantic information, which helps understand the overall structure of the image.

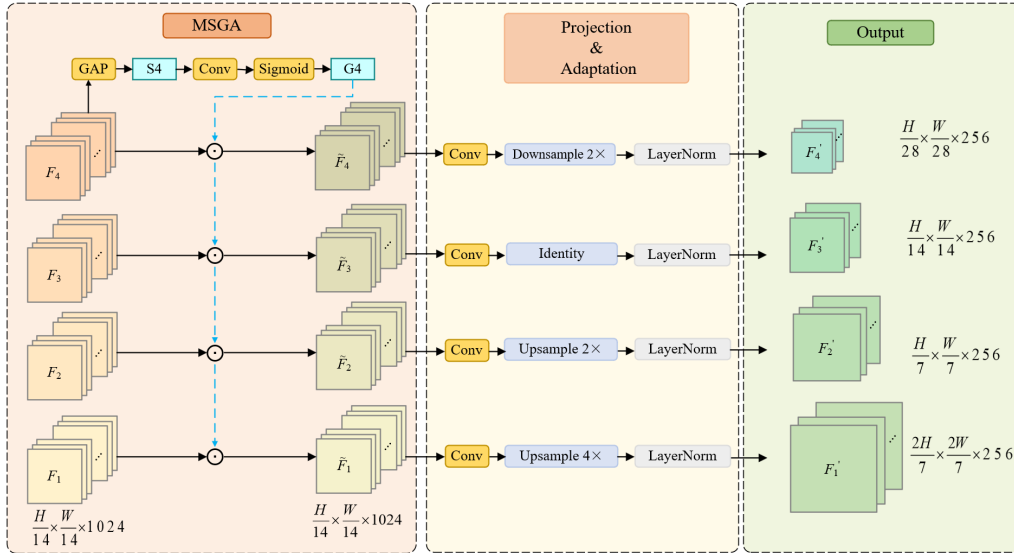


Figure 2. Feature processing network structure diagram.

It is worth noting that although multi-scale features exhibit progressive relationships at the semantic level, the Transformer architecture exhibits a receptive field accumulation effect. This leads to significant differences in the response patterns of different layers to the same spatial object [42]. Specifically, after multiple self-attention computations, the effective receptive field of deep-layer features can cover the entire image region. Thus, deep-layer features tend to encode the entire target as a semantic unit holistically. In contrast, the limited receptive field of shallow features primarily responds to low-level visual patterns such as local textures and edges. Consequently, shallow features may recognize different parts of the same target as spatially separated independent units. This spatial cognitive inconsistency is particularly prominent in remote sensing scenarios. This is because ground objects in remote sensing images often span a large spatial range, and the difference between deep and shallow features perceived by the model will be significantly amplified. However, existing feature pyramid-based multi-scale alignment methods do

not consider the above differences. Obviously, due to the spatial localization discrepancies of ground object boundaries across different layers, the geometric features extracted at each level and their corresponding spatial relationships inevitably exhibit deviations. This leads to feature conflicts during multi-scale fusion, thereby affecting the accuracy of edge localization and geometric consistency at the global level. Therefore, using the unified spatial attention extracted from deep-layer features as a prior, MSGA propagates the spatial guidance map to all levels to ensure that the salient features identified at each layer remain spatially consistent. The processing flow of MSGA is shown in Figure 2, which includes three stages: spatial saliency extraction, learnable guidance map generation, and feature modulation.

Due to the integration of global context information, the deep feature $\mathbf{F}_4$ will not be easily misled by local noise and can determine the location of the building more reliably. As shown in formula (1), in the spatial saliency extraction stage, MSGA obtains the spatial saliency distribution after aggregating cross-channel information using global channel average pooling for $\mathbf{F}_4$:

$$\mathbf{S}_4(i, j) = \frac{1}{1024} \sum_{c=1}^{1024} \mathbf{F}_4(c, i, j) \quad (1)$$

where (i, j) is the spatial position, c is the channel index, and $\mathbf{S}_4$ reflects the activation intensity of the (i, j) position. In fact, target areas such as buildings usually have higher activation values in multiple channels, while background areas generally have weaker activation values. It can be seen that after performing channel averaging operations on this information, the stability of spatial saliency estimation can be improved.

In the generation stage of learnable guidance map, the saliency distribution will be transformed into a guidance signal that can be used for feature modulation. As shown in formula (2), we generate a normalized spatial guidance map through a learnable nonlinear transformation:

$$\mathbf{G}_4 = \sigma(\mathbf{W}_s * \mathbf{S}_4 + \mathbf{b}_s) \quad (2)$$

where $\mathbf{G}_4$ is the spatial guidance map representing spatial importance weights, $*$ denotes 1×1 convolution, $\mathbf{W}_s$ and $\mathbf{b}_s$ are learnable parameters, and σ is the Sigmoid activation function. When we introduce learnable parameters in formula (2) instead of directly using $\mathbf{S}_4$, the model can adaptively adjust the discrimination criteria for spatial saliency during training. When the model is used for building height estimation, it will also assign higher weights to ground object regions. Thus, if the value of $\mathbf{G}_4$ is close to 1, it indicates the presence of key ground object information such as buildings at that location. Conversely, if the value of $\mathbf{G}_4$ is close to 0, it indicates that the location is a non-key ground object region such as background. Since features at different levels of DINOv2 have the same spatial resolution, $\mathbf{G}_4$ can directly modulate features at all levels through a broadcasting mechanism to achieve spatial alignment of features at different levels. As shown in formula (3), feature modulation is achieved through element-wise weighting:

$$\tilde{\mathbf{F}}_i = \mathbf{F}_i \odot (1 + \alpha \cdot \mathbf{G}_4), \quad i \in \{1, 2, 3, 4\} \quad (3)$$

Where $\odot$ represents element-wise multiplication, α is the hyperparameter of the modulation intensity, and $\tilde{\mathbf{F}}_i$ is the modulated feature at each level. Experiments show that if the α value is too large, it will destroy the original representation of the pre-trained features, while if the α value is too small, it cannot provide sufficient spatial guidance. Thus we adopt grid search to select the optimal value within the range 0.1, 0.2, 0.3, 0.4, 0.5. Experimental results show that when the

value of α is 0.3, the feature response of important areas such as buildings can be increased by up to 30%. At this time, feature modulation can effectively guide shallow attention to focus on the target area without excessively changing the distribution of original features.

After MSGA processing, the features of each layer, namely $\{\tilde{\mathbf{F}}_1, \tilde{\mathbf{F}}_2, \tilde{\mathbf{F}}_3, \tilde{\mathbf{F}}_4\}$, reach agreement on spatial attention. This means that features at different levels maintain the same spatial position response to the area of interest of the same ground object. This establishes a reliable spatial alignment basis for subsequent geometric feature extraction.

After executing MSGA, we employ 1×1 convolution for channel projection to map the features at four levels $\{\tilde{\mathbf{F}}_1, \tilde{\mathbf{F}}_2, \tilde{\mathbf{F}}_3, \tilde{\mathbf{F}}_4\}$ to 256, 512, 1024, and 1024 channels, respectively. Subsequently, different size adjustment strategies are adopted for different semantic levels to construct a coarse-to-fine spatial pyramid. Specifically, the modulated shallow feature $\tilde{\mathbf{F}}_1$ employs $4 \times$ transposed convolution for upsampling to preserve semantic detail information of the image. Similarly, the modulated middle-shallow feature $\tilde{\mathbf{F}}_2$ employs $2 \times$ transposed convolution for up-sampling to achieve a balance between middle-level semantics and spatial resolution. Unlike the operations of the previous two layers, the modulated middle-layer feature $\tilde{\mathbf{F}}_3$ maintains the original spatial resolution, while the modulated deep feature $\tilde{\mathbf{F}}_4$ performs downsampling through convolution with a stride of 2 to focus on global semantic information. Finally, all branches pass through normalization layers to unify the channel number to 256, forming spatially aligned and channel-consistent multi-scale feature representations $\{\mathbf{F}'_1, \mathbf{F}'_2, \mathbf{F}'_3, \mathbf{F}'_4\}$, providing standardized inputs for subsequent geometry-aware feature fusion.

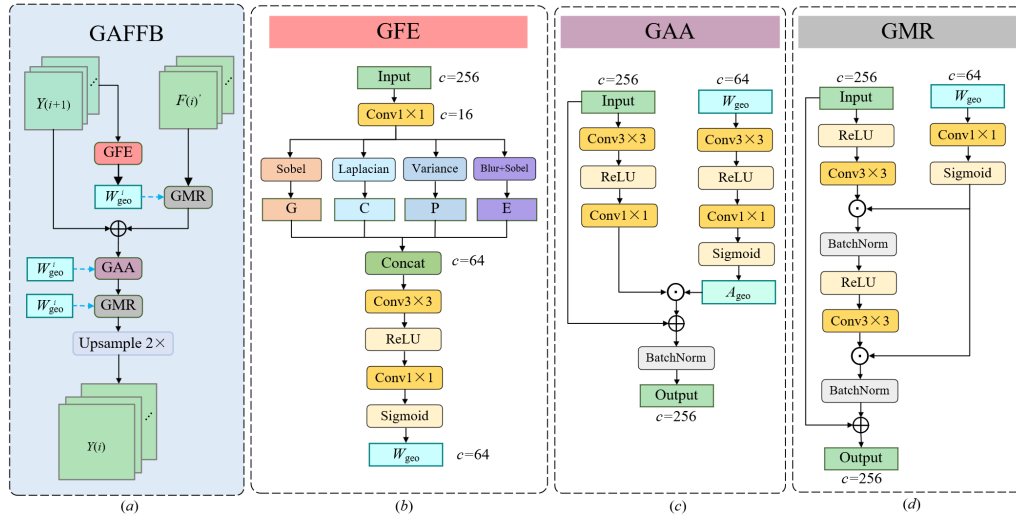


Figure 3. Feature Fusion Module. (a) Geometry-Aware Feature Fusion Block, (b) Geometric Feature Extractor, (c) Geometry-Aware Attention module, (d) Geometric Modulated Residual module.

3.3. Geometry-Aware Feature Fusion Module

Existing monocular building height estimation methods from remote sensing images primarily rely on implicit feature learning to capture geometric information. Through training with large amounts of data, networks can automatically learn feature representations. However, pure

implicit learning not only suffers from poor interpretability, but also makes it difficult to control the precision of feature learning [43]. In fact, ground objects in remote sensing height estimation tasks possess explicit geometric attributes and clear physical meanings. Specifically, gradient reflects height discontinuities in adjacent regions, curvature characterizes the degree of surface bending, planarity measures the flatness consistency of regions, and edges localize the contour boundaries of objects. This geometric prior knowledge is crucial for improving the accuracy of height estimation and the rationality of region segmentation. Therefore, we design a Geometry-Aware Feature Fusion Block (GAFFB) as shown in Figure 3(a).

Correspondingly, existing geometric attention mechanisms usually only utilize a single geometric cue and construct attention weights in an implicit manner [44]. This means that the existing geometric attention mechanism is difficult to fully characterize the geometric structure attributes of ground objects. Different from existing geometric attention mechanisms, GAFFB adopts a variety of operators to explicitly extract four types of complementary geometric features, including gradient, curvature, planarity and edge. Furthermore, as geometric priors of ground objects, these geometric features are simultaneously embedded into the two stages of attention modulation and feature propagation. Therefore, with the help of GAFFB, it is ensured that the constraints on the above four geometric features can run through the entire process of feature fusion. In GAFFB, the geometry-aware feature fusion process is shown in formula (4).

$$\begin{cases} \mathbf{W}_{\text{geo}}^i = \text{GFE}(\mathbf{F}_i'), & i = 4 \\ \mathbf{Y}_i = \text{Conv}_{1 \times 1}(\text{Upsample}(\text{GMR}(\text{GAA}(\text{GMR}(\mathbf{F}_i'; \mathbf{W}_{\text{geo}}^i); \mathbf{W}_{\text{geo}}^i); \mathbf{W}_{\text{geo}}^i), 2)), & i = 4 \\ \mathbf{W}_{\text{geo}}^i = \text{GFE}(\mathbf{Y}_{i+1}), & i \in \{3, 2, 1\} \\ \mathbf{Y}_i = \text{Conv}_{1 \times 1}(\text{Upsample}(\text{GMR}(\text{GAA}(\mathbf{Y}_{i+1} + \text{GMR}(\mathbf{F}_i'; \mathbf{W}_{\text{geo}}^i); \mathbf{W}_{\text{geo}}^i); \mathbf{W}_{\text{geo}}^i), 2)), & i \in \{3, 2, 1\} \end{cases} \quad (4)$$

where GFE is the Geometric Feature Extractor, GAA is the Geometry-Aware Attention module, GMR is the Geometric Modulated Residual module, Upsample($\cdot$) denotes bilinear upsampling, $\mathbf{W}_{\text{geo}}^i$ is the geometric weight of the i -th layer, and $\mathbf{Y}_i$ is the fused output feature of the i -th layer. The 4th layer serves as the starting point of decoding, where GFE directly extracts geometric features from the input feature $\mathbf{F}_4'$ of the 4th layer and generates comprehensive geometric weights $\mathbf{W}_{\text{geo}}^4$. For the i -th layer ($i \in \{3, 2, 1\}$), GFE generates comprehensive geometric weights $\mathbf{W}_{\text{geo}}^i$ from the fused output feature $\mathbf{Y}_{i+1}$ of the previous layer.

In the feature fusion process of the 4th layer, the input feature $\mathbf{F}_4'$ is sequentially modulated by the first GMR, GAA, and the second GMR, followed by $2 \times$ upsampling and 1×1 convolution operations to obtain the fused output $\mathbf{Y}_4$. For the i -th layer ($i \in \{3, 2, 1\}$), GMR first processes the input feature $\mathbf{F}_i'$, which is then element-wise added to the fused output feature $\mathbf{Y}_{i+1}$ from the previous layer. Subsequently, the added feature is sequentially processed by GAA and GMR. Finally, the fused output $\mathbf{Y}_i$ is obtained after sequential $2 \times$ upsampling and 1×1 convolution operations. In the above fusion process, GMR achieves channel-level geometric modulation guided by geometric weights $\mathbf{W}_{\text{geo}}^i$, while GAA further transforms the geometric weights $\mathbf{W}_{\text{geo}}^i$ into joint spatial-channel attention maps to achieve adaptive feature gating.

As shown in Figure 3(b), the Geometric Feature Extractor (GFE) adopts a parallel architecture. To reduce computational complexity, for a given input feature $\mathbf{F} \in \mathbb{R}^{H' \times W' \times 256}$, GFE uses a 1×1 convolutional layer to reduce the channel number to 16, obtaining the dimension-reduced feature $\mathbf{F}_{\text{red}} \in \mathbb{R}^{H' \times W' \times 16}$. Subsequently, as shown in formula (5), four types of geometric information are extracted in parallel from $\mathbf{F}_{\text{red}}$, namely gradient features $\mathbf{G}$, curvature features $\mathbf{C}$, planarity features $\mathbf{P}$, and edge features $\mathbf{E}$.

$$\begin{aligned}
 \mathbf{G} &= \sqrt{(\mathbf{G}_x * \mathbf{F}_{red})^2 + (\mathbf{G}_y * \mathbf{F}_{red})^2 + \varepsilon} \\
 \mathbf{C} &= |\mathbf{K}_{lap} * \mathbf{F}_{red}| \\
 \text{Var} &= \text{AvgPool}_{3 \times 3}((\mathbf{F}_{red} - \text{AvgPool}_{3 \times 3}(\mathbf{F}_{red}))^2), \quad \mathbf{P} = \frac{1}{1 + \text{Var} + \varepsilon} \\
 \mathbf{E} &= \sqrt{(\mathbf{G}_x * \text{AvgPool}_{3 \times 3}(\mathbf{F}_{red}))^2 + (\mathbf{G}_y * \text{AvgPool}_{3 \times 3}(\mathbf{F}_{red}))^2 + \varepsilon}
 \end{aligned} \tag{5}$$

In formula (5), $\mathbf{G}_x$ and $\mathbf{G}_y$ represent the Sobel operators in the horizontal and vertical directions, respectively, $*$ denotes the convolution operation, $\varepsilon = 10^{-8}$ is the numerical stability constant, $\mathbf{K}_{lap}$ is the Laplacian operator, and AvgPool $_{3 \times 3}$ represents average pooling with a 3×3 window. As shown in formula (5), GFE explicitly uses mathematical operators that actively capture the geometric properties of input features. This means that geometric prior knowledge in the input features can be directly applied to the network's inference process. Specifically, gradient feature $\mathbf{G}$ is the feature representation obtained by computing the first-order derivative of $\mathbf{F}_{red}$ using the Sobel operator. Thus, gradient features can effectively characterize the direction and strength of edges. This means that gradient feature responses become more pronounced at building contours and terrain discontinuities. Curvature feature $\mathbf{C}$ is the feature representation obtained by computing the second-order derivative of $\mathbf{F}_{red}$ using the Laplacian operator. Thus, curvature features can effectively capture the degree of surface bending. This means that high-value regions of curvature features correspond to areas with roof structural changes or terrain undulations. Similarly, planarity feature $\mathbf{P}$ is the reciprocal of the local normalized variance of $\mathbf{F}_{red}$, measuring the degree of flatness in specific regions. Obviously, the higher the planarity feature value, the more gradual the height variation in the corresponding region. Furthermore, edge feature $\mathbf{E}$ is the feature representation obtained after $\mathbf{F}_{red}$ undergoes average pooling and gradient computation. Thus, edge features can more robustly localize true boundaries and effectively suppress interference from texture noise.

Subsequently, as shown in formula (6), the above four types of geometric features are concatenated along the channel dimension and processed through a fusion network to generate comprehensive geometric weights $\mathbf{W}_{geo}$.

$$\mathbf{W}_{geo} = \sigma(\text{Conv}_{1 \times 1}(\text{ReLU}(\text{Conv}_{3 \times 3}([\mathbf{G}, \mathbf{C}, \mathbf{P}, \mathbf{E}])))) \tag{6}$$

where $[\cdot]$ denotes the channel concatenation operation, σ is the Sigmoid activation function for normalizing weights to the $[0, 1]$ interval. Furthermore, the 3×3 convolution in formula (6) aggregates geometric features in the spatial domain, while the 1×1 convolution integrates features in the channel domain. It can be seen that $\mathbf{W}_{geo} \in \mathbb{R}^{H' \times W' \times 64}$ integrates complementary information from four dimensions: gradient, curvature, planarity, and edges, comprehensively characterizing the geometric saliency of each spatial position.

As shown in Figure 3(c), the Geometry-Aware Attention module (GAA) transforms the geometric priors extracted by GFE into joint spatial-channel attention maps, enabling the network to adaptively focus on regions with salient geometric features. As shown in formula (7), for a given input feature $\mathbf{F} \in \mathbb{R}^{H' \times W' \times 256}$ and comprehensive geometric weights $\mathbf{W}_{geo} \in \mathbb{R}^{H' \times W' \times 64}$, GAA first maps the geometric weights into attention maps aligned with the feature channels.

$$\mathbf{A}_{geo} = \sigma(\text{Conv}_{1 \times 1}(\text{ReLU}(\text{Conv}_{3 \times 3}(\mathbf{W}_{geo})))) \tag{7}$$

where ReLU is the activation function, and σ is the Sigmoid function for normalizing attention values to the $[0, 1]$ interval. Furthermore, the 1×1 convolution adjusts the channel number from

64 to 256 to match the feature dimensions, while the 3×3 convolution aggregates local spatial information. It can be seen that the attention map $\mathbf{A}_{\text{geo}} \in \mathbb{R}^{H' \times W' \times 256}$ assigns importance scores to each channel at each spatial position, achieving joint spatial-channel attention modulation. Subsequently, as shown in formula (8), GAA modulates features through a convolution with geometric gating.

$$\mathbf{Y} = \text{BN}(\text{Conv}_{1 \times 1}(\text{ReLU}(\text{Conv}_{3 \times 3}(\mathbf{F}))) \odot \mathbf{A}_{\text{geo}} + \mathbf{F}) \quad (8)$$

where BN denotes batch normalization, ReLU is the activation function, $\odot$ represents element-wise multiplication, and $+\mathbf{F}$ is the residual connection. Firstly, local spatial contextual information is extracted from the input feature $\mathbf{F}$ through 3×3 convolution, followed by ReLU activation and 1×1 convolution for feature transformation in the channel domain to obtain a joint representation of local geometric and semantic information. Then, geometric gating is applied through element-wise multiplication to weight the transformed features using the attention map $\mathbf{A}_{\text{geo}}$. Thus, for geometrically complex regions such as building edges and corners, the value of $\mathbf{A}_{\text{geo}}$ is high, and the features in corresponding regions are enhanced. Conversely, for background or flat regions, the value of $\mathbf{A}_{\text{geo}}$ is low, and the features in corresponding regions are suppressed. Of course, in cases where attention learning is insufficient or geometric information is ambiguous, the residual connection can still ensure that the original features are preserved. Finally, batch normalization stabilizes the feature distribution to ensure that the output feature $\mathbf{Y} \in \mathbb{R}^{H' \times W' \times 256}$ possesses both sensitivity to geometric features and stability of feature representation.

As shown in Figure 3(d), the Geometric Modulated Residual module (GMR) introduces channel-level geometric modulation based on the standard residual unit. This enables the residual path to dynamically adjust the strength of feature propagation according to local geometric structures. For a given input feature $\mathbf{F} \in \mathbb{R}^{H' \times W' \times 256}$ and geometric weights $\mathbf{W}_{\text{geo}} \in \mathbb{R}^{H' \times W' \times 64}$, the feature propagation process through two geometric modulations is shown in formula (9):

$$\begin{aligned} \mathbf{U}'_1 &= \text{Conv}_{3 \times 3}(\text{ReLU}(\mathbf{F})) \odot \sigma(\text{Conv}_{1 \times 1}(\mathbf{W}_{\text{geo}})) \\ \mathbf{U}'_2 &= \text{Conv}_{3 \times 3}(\text{ReLU}(\text{BN}(\mathbf{U}'_1))) \odot \sigma(\text{Conv}_{1 \times 1}(\mathbf{W}_{\text{geo}})) \\ \mathbf{Y} &= \text{BN}(\mathbf{U}'_2) + \mathbf{F} \end{aligned} \quad (9)$$

where $\mathbf{U}'_1$ and $\mathbf{U}'_2$ are the intermediate features after two geometric modulations, respectively. During channel-level modulation, 1×1 convolution maps $\mathbf{W}_{\text{geo}}$ from 64 channels to 256 channels. As shown in formula (9), the output feature $\mathbf{Y}$ is obtained after applying batch normalization and residual connection to the second intermediate feature $\mathbf{U}'_2$. Notably, GMR applies geometric modulation during both convolutions on the input feature. This ensures that geometric constraints are incorporated in each convolutional layer of the residual path, enabling the network to maintain continuous perception of properties such as edges, curvature, and planarity when processing complex geometric structures.

It can be seen that the four types of complementary geometric attributes extracted by GFE characterize the 3D structural properties of ground objects from different dimensions. GAA transforms geometric priors into joint spatial-channel attention, achieving adaptive focus on geometrically salient regions. GMR applies channel-level geometric modulation at both side branch processing and feature refinement stages, ensuring that geometric constraints are incorporated in every stage of feature propagation. Therefore, through the tight collaboration of GFE, GAA, and GMR, GAFFB achieves cross-scale cumulative propagation of geometric priors.

3.4. Multi-Component Geometric Constraint Loss Function

Traditional height estimation methods typically use standard loss functions such as Mean Squared Error (MSE) or smooth L1. However, these losses only focus on pixel-level numerical errors and lack explicit constraints on real-world geometric laws. In fact, both ground objects and terrains possess explicit geometric properties. When estimating building heights, the planarity of building roofs needs to be considered, terrain slopes should be controlled within reasonable ranges, and height variations in adjacent regions should generally be continuous. Therefore, to ensure both numerical accuracy and geometric rationality of the network's predictions, we design a Multi-component Geometric Constraint Loss (MGCL) function. Since constraints such as gradients and normal vectors are beneficial to improving depth estimation performance [37][38][39][40]. To this end, we design a supervised learning framework with the help of the MGCL loss function that comprehensively utilizes the above geometric constraints to ensure the overall optimization of the height estimation model at three levels, including pixel accuracy, geometric consistency and physical rationality. In addition, MGCL has specially designed key parameters such as slope threshold and planarity constraint range based on the geometric characteristics of remote sensing ground objects. The computation of MGCL is shown in formula (10):

$$\mathcal{L}_{total} = \mathcal{L}_{base} + \lambda_{geo}\mathcal{L}_{geo} + \lambda_{phy}\mathcal{L}_{phy} \quad (10)$$

where $\mathcal{L}_{base}$, $\mathcal{L}_{geo}$ and $\mathcal{L}_{phy}$ are the basic regression loss, geometric consistency loss, and physical rationality loss, respectively. λ_{geo} and λ_{phy} are the weight parameters for geometric consistency and physical rationality, respectively. Thus, with the constraints of $\mathcal{L}_{geo}$ and $\mathcal{L}_{phy}$, the total loss $\mathcal{L}_{total}$ ensures pixel-level numerical fitting accuracy while avoiding height predictions that violate geometric features and physical structures.

$\mathcal{L}_{base}$ focuses on pixel-level errors between predicted heights and ground truth heights to ensure that predictions are numerically close to the true labels. $\mathcal{L}_{base}$ is computed as shown in formula (11):

$$\mathcal{L}_{base} = \frac{1}{N} \sum_{i=1}^N \text{SmoothL1}(\mathbf{H}_{pred}(i) - \mathbf{H}_{gt}(i)) \quad (11)$$

where $N = H \times W$ is the total number of pixels, and $\mathbf{H}_{pred}, \mathbf{H}_{gt} \in \mathbb{R}^{H \times W}$ are the predicted height and ground truth, respectively. When the absolute error is less than the hyperparameter, the L2-like term is used to make the loss function smooth with continuous gradients. When the absolute error is greater than or equal to the hyperparameter, the L1-like term is used to reduce the loss function's sensitivity to outliers. Thus, $\mathcal{L}_{base}$ enables the network to enhance robustness to noise while focusing on overall fitting accuracy.

$\mathcal{L}_{geo}$ constrains the predictions from multiple complementary geometric dimensions to ensure that the height map conforms to real-world geometric structures. As shown in formula (12), $\mathcal{L}_{geo}$ consists of three sub-losses, including smoothness loss $\mathcal{L}_{smooth}$, gradient magnitude consistency loss $\mathcal{L}_{grad}$, and surface normal loss $\mathcal{L}_{normal}$:

$$\begin{aligned} \mathcal{L}_{geo} &= \lambda_s \mathcal{L}_{smooth} + \lambda_g \mathcal{L}_{grad} + \lambda_n \mathcal{L}_{normal} \\ \mathcal{L}_{smooth} &= \frac{1}{2N} \sum_{i=1}^N (|\nabla_x^{pred}(i) - \nabla_x^{gt}(i)| + |\nabla_y^{pred}(i) - \nabla_y^{gt}(i)|) \\ \mathcal{L}_{grad} &= \frac{1}{N} \sum_{i=1}^N \left| \sqrt{(\nabla_x^{pred})^2 + (\nabla_y^{pred})^2 + \varepsilon} - \sqrt{(\nabla_x^{gt})^2 + (\nabla_y^{gt})^2 + \varepsilon} \right| \\ \mathcal{L}_{normal} &= \frac{1}{N} \sum_{i=1}^N (1 - \mathbf{n}_{pred}(i) \cdot \mathbf{n}_{gt}(i)), \quad \mathbf{n} = \frac{(-\nabla_x, -\nabla_y, 1)}{\sqrt{\nabla_x^2 + \nabla_y^2 + 1 + \varepsilon}} \end{aligned} \quad (12)$$

where the values of the weights λ_s , λ_g , and λ_n for the three sub-losses are 0.5, 1.0, and 0.5, respectively, and $\varepsilon = 10^{-8}$ is the numerical stability constant. Furthermore, ∇_x and ∇_y are the gradients in the horizontal and vertical directions computed based on the Sobel operator, respectively. $\mathbf{n}_{pred}(i)$ and $\mathbf{n}_{gt}(i)$ are the normal vectors at position i of the predicted height map and ground truth height map, respectively.

$\mathcal{L}_{smooth}$ constrains the gradient consistency between the predicted height map and ground truth height map by computing the L1 distance of gradients in the horizontal and vertical directions. Obviously, $\mathcal{L}_{smooth}$ encourages height variations of adjacent pixels to exhibit similar gradient patterns. Therefore, when predicting building roofs and flat terrain regions, $\mathcal{L}_{smooth}$ can effectively suppress high-frequency noise and artifacts. $\mathcal{L}_{grad}$ promotes the predicted height map and ground truth height map to have similar gradient magnitude distributions by focusing on the rate and strength of height variations. Since $\mathcal{L}_{grad}$ can more sensitively capture the structural features of objects, it significantly improves the localization accuracy and sharpness at key boundaries such as building contours and terrain fracture zones. $\mathcal{L}_{normal}$ is used to constrain the consistency between the predicted surface and ground truth surface in 3D shape. The normal vector $\mathbf{n}$ is obtained by computing the gradient of the height map and normalizing it, describing the orientation of each point on the ground object surface. $\mathcal{L}_{normal}$ uses cosine distance to measure the angle between the predicted normal vector $\mathbf{n}_{pred}(i)$ and ground truth normal vector $\mathbf{n}_{gt}(i)$. When the directions of $\mathbf{n}_{pred}(i)$ and $\mathbf{n}_{gt}(i)$ are consistent, $\mathcal{L}_{normal}$ approaches 0. Conversely, when the directional deviation between $\mathbf{n}_{pred}(i)$ and $\mathbf{n}_{gt}(i)$ increases, the value of $\mathcal{L}_{normal}$ also increases. Therefore, in regions with significant 3D shape features such as sloped building roofs and terrain slopes, $\mathcal{L}_{normal}$ encourages the overall orientation and curvature of the predicted surface to be consistent with the ground truth surface.

$\mathcal{L}_{phy}$ introduces real-world physical laws as soft constraints to avoid physically unreasonable predictions by the network. The computation of $\mathcal{L}_{phy}$ is shown in formula (13):

$$\begin{aligned}\mathcal{L}_{phy} &= \mathcal{L}_{slope} + \lambda_{planar} \mathcal{L}_{planar} \\ \mathcal{L}_{slope} &= \frac{1}{N} \sum_{i=1}^N (\text{ReLU}(\arctan(|\nabla_x^{pred}|) - \theta_{max}) + \text{ReLU}(\arctan(|\nabla_y^{pred}|) - \theta_{max})) \\ \mathcal{L}_{planar} &= \frac{1}{N} \sum_{i=1}^N \text{AvgPool}_{5 \times 5}((\mathbf{H}_{pred} - \text{AvgPool}_{5 \times 5}(\mathbf{H}_{pred}))^2)\end{aligned}\quad (13)$$

where $\mathcal{L}_{slope}$ and $\mathcal{L}_{planar}$ are the slope constraint loss and planarity constraint loss, respectively, λ_{planar} is the planarity constraint weight, and θ_{max} is the maximum allowable slope angle. In formula (13), the value of λ_{planar} is 0.1, and θ_{max} is 45° .

$\mathcal{L}_{slope}$ is used to limit the slope of terrain and building surfaces to a reasonable range. In the real world, the slopes of natural terrain and man-made buildings are generally not excessively steep. $\mathcal{L}_{slope}$ uses the arctangent function of gradients to compute slope angles, applying ReLU penalty to regions that exceed the threshold. This soft constraint prevents predictions of physically impossible steep structures, thereby making the predictions more consistent with the physical characteristics of real terrain and buildings. Since slopes exceeding 45° in natural terrain are extremely rare, man-made building roof slopes usually range from 15° to 35° . It can be seen that when θ_{max} is set to 45° , it can effectively cover the true slope range of the ground objects in the three data sets. $\mathcal{L}_{planar}$ is used to encourage regions such as building roofs to maintain local flatness. Building roofs are typically planar or regular sloped surfaces, rather than randomly undulating surfaces. By minimizing the local variance within a 5×5 neighborhood, $\mathcal{L}_{planar}$ promotes regions such as building roofs to tend toward flatness. During height predic-

tion, this reduces unreasonable locally abrupt height predictions and improves the visual quality and geometric rationality of the prediction results.

In summary, MGCL comprehensively considers losses from three aspects: basic regression, geometric consistency, and physical constraints. The basic regression loss provides pixel-level fitting accuracy. The geometric consistency loss constrains the rationality of local geometric structures from three perspectives: smoothness, gradient, and normal vectors. The physical constraint loss prevents unreasonable predictions by introducing slope and planarity regularities. These three loss terms jointly guide the network to learn a height estimation model that conforms to physical laws and geometric constraints. This ensures comprehensive improvement of the prediction results in terms of numerical accuracy, geometric structure, physical characteristics, and visual quality.

4. EXPERIMENTS

4.1. Experimental Datasets

To evaluate the effectiveness of GA-DPNet, we conduct experimental validation on public datasets including International Society for Photogrammetry and Remote Sensing (ISPRS) Vaihingen, ISPRS Potsdam, and DFC2019. Figure 4 shows examples of images and labels from the above three datasets.

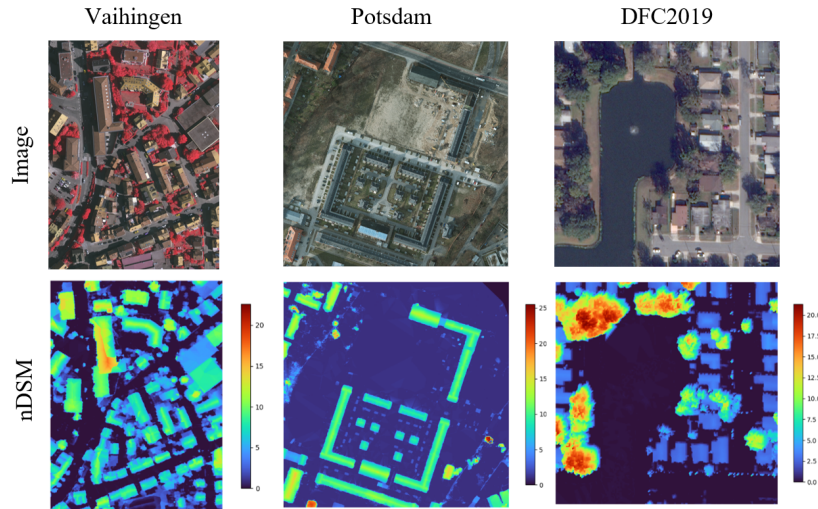


Figure 4. Dataset examples.

(1) **ISPRS Vaihingen:** The Vaihingen dataset consists of 33 high-resolution aerial images of varying sizes, with an average size of 2494×2064 pixels. Each image's orthoimage is composed of three bands: near-infrared, red, and green, and the Ground Sampling Distance (GSD) of the normalized Digital Surface Model (nDSM) is 9 cm. Following the official dataset split, we use 16 images for model training and the remaining 17 images for model testing.

(2) **ISPRS Potsdam:** The Potsdam dataset consists of 38 high spatial resolution aerial images. The image size is 6000×6000 pixels, and each image is composed of four spectral bands:

red, green, blue, and near-infrared. In our experiments, we only use the red, green, and blue bands, and the Ground Sampling Distance (GSD) of the normalized Digital Surface Model (nDSM) is 5 cm. Following the official dataset split, we use 24 images for model training and the remaining 14 images for model testing.

(3) **DFC2019**: The DFC2019 dataset consists of 2783 high spatial resolution aerial images. The image size is 1024×1024 pixels. The Ground Sampling Distance (GSD) of the normalized Digital Surface Model (nDSM) is 35 cm. We randomly split 2332 images for model training and the remaining 451 images for model testing.

For experimental convenience, we crop all images to 512×512 size, and all ground truth of nDSM is used as height maps during training and evaluation.

4.2. Experimental Details and Evaluation Metrics

Our experiments are implemented in the PyTorch framework, using a single NVIDIA RTX 4080 GPU (16GB memory) for training and testing. We use the AdamW optimizer with an initial learning rate of 1×10^{-6} and a weight decay coefficient of 5×10^{-4} . Furthermore, to accelerate the optimization of the decoder, the learning rate of the decoder is set to 1×10^{-5} . Meanwhile, we adopt a warmup strategy where the learning rate increases linearly for the first 5 epochs, followed by cosine annealing scheduling. In order to verify the rationality of the selection of backbone parameter scale, we also listed the experimental results using DINOv2-ViT-L and DINOv2-ViT-B as the backbone network. Unless otherwise specified, GA-DPNet adopts DINOv2-ViT-L as the complete version of the backbone network by default. Due to the limitation of single GPU memory, the batch size of GA-DPNet (ViT-L) is set to 2, the batch size of GA-DPNet (ViT-B) is set to 8, and the total number of training epochs is 100.

To verify the performance of GA-DPNet, we adopt Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Structural Similarity Index Measure (SSIM) as evaluation metrics, which are defined in Equation (14):

$$\begin{aligned} \text{MAE} &= \frac{1}{N} \sum_{i=1}^N |\mathbf{H}_{\text{pred}}^{(i)} - \mathbf{H}_{\text{gt}}^{(i)}| \\ \text{RMSE} &= \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{H}_{\text{pred}}^{(i)} - \mathbf{H}_{\text{gt}}^{(i)})^2} \\ \text{SSIM}(x, y) &= \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \end{aligned} \quad (14)$$

where N is the total number of pixels, $\mathbf{H}_{\text{pred}}^{(i)}$ and $\mathbf{H}_{\text{gt}}^{(i)}$ represent the predicted height value and ground truth height value of the i -th pixel, respectively. Furthermore, μ_x and μ_y are the mean values of the predicted and true height maps within the local window, σ_x and σ_y are the standard deviations, σ_{xy} is the covariance, and C_1 and C_2 are stability constants to avoid division by zero. MAE directly reflects pixel-level accuracy, assigns equal weight to all errors, and has good robustness and intuitive physical meaning. The unit of MAE is meters (m), and a smaller value indicates better performance. RMSE applies higher penalty to large errors through the squaring operation, is more sensitive to difficult regions such as building edges, and can reflect the dispersion degree of the error distribution. The unit of RMSE is meters (m), and a smaller value indicates better performance. SSIM comprehensively evaluates similarity from three dimensions: luminance, contrast, and structure. Since RMSE considers the spatial relationships between pixels, the value of RMSE can effectively reflect the fidelity of geometric structures such as

building contours and terrain undulations. The value range of SSIM is $[-1, 1]$. When the value of SSIM approaches 1, it means the model performs better.

4.3. Comparison of Experimental Results

Table 1 lists the experimental results of GA-DPNet and recent representative models on the ISPRS Vaihingen, ISPRS Potsdam, and DFC2019 datasets. The best MAE, RMSE, and SSIM values on each dataset are shown in bold.

Table 1. Metrics on the Vaihingen, Potsdam and DFC2019 datasets.

Method	Vaihingen			Potsdam			DFC2019		
	MAE↓	RMSE↓	SSIM↑	MAE↓	RMSE↓	SSIM↑	MAE↓	RMSE↓	SSIM↑
IM2HEIGHT[10]	1.485	2.253	0.919	2.198	4.137	0.881	2.037	3.641	0.850
aerial[11]	1.487	2.197	0.923	1.926	3.506	0.907	1.991	3.310	0.838
IM2ELEVATION[12]	1.226	1.882	0.944	1.711	3.514	0.916	1.771	3.478	0.869
PLNet[13]	1.178	1.775	0.952	1.203	2.356	0.957	1.567	2.855	0.886
LUNet[21]	0.969	1.523	0.955	1.222	2.158	0.953	1.289	2.234	0.898
Heightformer[16]	0.873	1.433	0.957	0.974	2.193	0.959	1.051	2.057	0.903
CTMR-Net[22]	1.184	1.790	0.956	1.147	2.420	0.961	1.187	2.213	0.908
RAEPHR[23]	0.902	1.448	0.959	0.874	2.018	0.963	0.901	1.950	0.912
GA-DPNet(ViT-B)	0.831	1.377	0.961	1.126	2.231	0.952	0.875	1.937	0.910
GA-DPNet(ViT-L)	0.783	1.219	0.963	1.022	2.102	0.958	0.835	1.891	0.917

As shown in Table 1, GA-DPNet achieves comprehensively optimal performance on both the Vaihingen and DFC2019 datasets. On the Vaihingen dataset, compared to the second-best method Heightformer, GA-DPNet reduces MAE and RMSE by 10.3% and 14.9%, respectively, while improving SSIM to 0.963. The Vaihingen dataset represents a high-resolution dense urban scene with tightly arranged buildings and significant height variations. GA-DPNet achieves precise localization of building boundaries through explicit geometric feature extraction. Furthermore, the reduction in RMSE on the Vaihingen dataset is significantly larger than the reduction in MAE, which validates the effectiveness of the geometric constraint loss adopted by GA-DPNet in suppressing excessive errors in edge regions. Similarly, on the DFC2019 dataset, compared to the second-best method RAEPHR, GA-DPNet reduces MAE and RMSE by 7.3% and 3.0%, respectively, while improving SSIM to 0.917. In fact, the DFC2019 dataset has a larger ground sampling distance and relatively lower spatial resolution. The experimental results in Table 1 show that due to the specially designed geometry-aware mechanism, GA-DPNet can still effectively capture the main structural features of buildings even under such resolution-limited conditions. It can be seen that compared to existing representative methods, GA-DPNet not only achieves the highest prediction accuracy on the Vaihingen and DFC2019 datasets, but also demonstrates good scale adaptability and better generalization performance. It is worth noting that as a variant of GA-DPNet, when DINOv2-ViT-B replaces DINOv2-ViT-L as the backbone network, the performance of GA-DPNet (ViT-B) on the Vaihingen and DFC2019 data sets has declined, but it is still better than most existing methods. It can be seen that the effectiveness of GA-DPNet does not simply depend on the backbone of a specific scale, and the model has good scalability.

On the Potsdam dataset, although GA-DPNet is slightly lower than the best-performing RAEPHR method in all three metrics (MAE, RMSE, and SSIM), GA-DPNet still outperforms the other six comparison methods overall. In fact, GA-DPNet's performance on the Potsdam dataset being inferior to RAEPHR is related to the characteristics of this dataset. Compared to the Vaihingen and DFC2019 datasets, the remote sensing images in the Potsdam dataset have higher spatial resolution and an image size of 6000×6000 pixels. When images are uniformly cropped to 512×512 size during the training process, the actual geographical range covered by high-resolution images will be significantly reduced. Compared with the other two data sets, the spatial context information in the receptive field of the GA-DPNet network will also be greatly reduced at this time. However, effective geometric feature extraction often relies on larger spatial scales to effectively capture macro-level geometric relationships such as the overall structure and surrounding layout of ground objects. It can be seen that when high-resolution remote sensing images are cropped into smaller sizes, it will have an adverse impact on the geometric perception capabilities of GA-DPNet. In addition, the MAE of GA-DPNet (ViT-B) on the Potsdam dataset is 1.126, which is also lower than RAEPHR. This further illustrates that the performance gap of GA-DPNet on the Potsdam dataset mainly comes from the loss of spatial context information after high-resolution image cropping.

Table 2. Complexity comparison of representative methods.

Method	Backbone	Params(M)	FLOPs(G)	Inference Time(ms)	MAE↓ (Vaihingen)
PLNet	ResNet-50	45.3	38.2	42.3	1.178
HeightFormer	MiT-B4	82.1	94.7	78.6	0.873
CTME-Net	ResNet-101	67.4	76.3	91.2	1.184
DINOv2-ViT-L+DPT	DINOv2-ViT-L	319.0	479.3	241.3	0.956
GA-DPNet(ViT-B)	DINOv2-ViT-B	101.3	163.5	112.4	0.831
GA-DPNet(ViT-L)	DINOv2-ViT-L	322.1	491.8	243.8	0.783

In order to further evaluate the computational overhead and deployment feasibility of GA-DPNet, Table 2 lists the results of model complexity comparison of representative methods from multiple dimensions. The compared methods include CNN-based PLNet, Transformer-based HeightFormer, multi-task learning-based CTME-Net, and GA-DPNet (ViT-B) and GA-DPNet (ViT-L) adopting backbone networks of different sizes.

As shown in Table 2, the number of parameters and FLOPs of GA-DPNet (ViT-L) are significantly higher than other comparison methods, which is mainly due to the scale of the DINOv2-ViT-L backbone network itself. However, compared with Baseline using the same backbone network, GA-DPNet (ViT-L) only introduces about 3M additional parameters, but the MAE on the Vaihingen dataset drops from 0.956 to 0.783. This shows that the accuracy improvement of GA-DPNet mainly comes from the contributions of the three modules of MSGA, GAFFB and MGCL, rather than simply relying on the scale of the backbone network. In addition, the parameter amount of GA-DPNet (ViT-B) is approximately 31.4% of that of GA-DPNet (ViT-L), and the inference time is approximately 46%. With a slight decrease in accuracy, GA-DPNet (ViT-B) significantly reduces the computational overhead and can be used for deployment in scenarios with limited computing resources.

Compared to other methods, PLNet adopts a Progressive Residual Modification mechanism

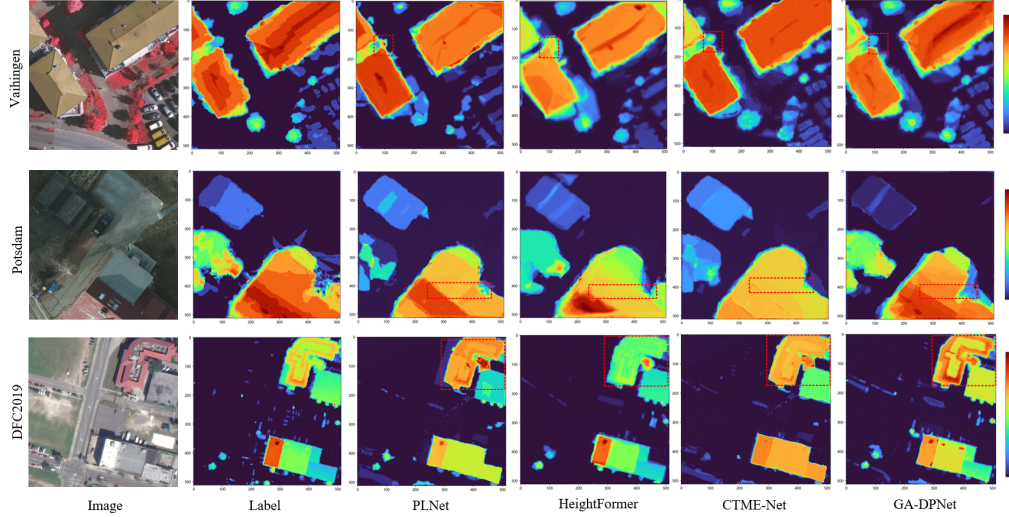


Figure 5. The visual comparison results of representative methods on the three datasets.

(PRM) to achieve high-quality preservation of ground object boundaries and contours. While Heightformer can effectively balance local details and global semantic information by introducing Transformers. As a representative method based on multi-task learning, CTME-Net has certain advantages in retaining semantic boundaries. Therefore, to validate GA-DPNet’s capability to improve ground object edge localization and prediction using global geometry awareness, Figure 5 shows visual comparison results of GA-DPNet, PLNet, Heightformer and CTME-Net on the ISPRS Vaihingen, ISPRS Potsdam, and DFC2019 datasets. The first set of red dashed boxes highlights a complex scene where buildings overlap with vegetation. In such regions, the height differences between different ground object categories are significant, and boundary delineation is particularly important. GA-DPNet predicts the clearest and most distinct boundary line between buildings and vegetation, with sharp building edges and obvious height discontinuities from the surrounding vegetation. In contrast, PLNet and Heightformer exhibit obvious height bleeding at the boundaries. This is because the predicted building edges diffuse into the vegetation areas, causing the building boundaries predicted by these two methods to become blurred. The experimental results show that GA-DPNet’s GAFFB module can effectively utilize these edge features and geometric information with significant gradient changes, achieving precise localization of building edges and accurate delineation between different ground object categories.

The second and third sets of boxes emphasize the geometric quality of the buildings’ interiors. In these regions, building roofs exhibit local depressions or height discontinuities, which are key test scenarios for evaluating the geometric modeling capability of methods. As shown in the experimental results in Figure 5, GA-DPNet produces the most reasonable and distinct predictions for these depression structures, with clear depth, shape, and boundaries of the depressions. PLNet often produces overly smooth predictions in these regions, where concave structures are not obvious or are even completely smoothed out. This is because the lack of explicit geometric constraints causes the network to tend to predict flat surfaces. Although Heightformer can identify the existence of depressions, the boundaries of the depressions are not sharp enough.

Although the semantic prior introduced by CTME-Net can constrain the height distribution between different categories, it lacks effective perception of local height changes within the same semantic target. On the one hand, GA-DPNet’s superior performance stems from the network’s ability to accurately perceive local geometric changes by extracting planarity and curvature features. On the other hand, it stems from the physical constraints introduced by the MGCL module, which ensure the geometric rationality of the depression structures.

4.4. Ablation Study

To quantitatively analyze the contribution of each component of GA-DPNet to model performance improvement, we conduct systematic ablation studies on the Vaihingen dataset. The baseline model adopts the standard DPT architecture with DINOv2-Large as the encoder but does not include any geometry-aware mechanisms. Table 3 shows the ablation study results of the main components of GA-DPNet on the Vaihingen dataset.

Table 3. Ablation study of main components on the Vaihingen dataset.

Method	Components			Metrics		
	MSGa	GAFFB	MGCL	MAE↓	RMSE↓	SSIM↑
Baseline	×	×	×	0.956	1.672	0.921
+ MSGa	✓	×	×	0.898	1.598	0.934
+ GAFFB	×	✓	×	0.851	1.526	0.943
+ MGCL	×	×	✓	0.915	1.641	0.928
+ MSGa+GAFFB	✓	✓	×	0.819	1.482	0.949
+ MSGa+MGCL	✓	×	✓	0.871	1.562	0.937
+ GAFFB+MGCL	×	✓	✓	0.829	1.498	0.946
GA-DPNet (Full)	✓	✓	✓	0.783	1.219	0.963

As shown in Table 3, MSGa, GAFFB, and MGCL all significantly contribute to the improvement of model performance. When the MSGa module is added alone to the baseline, the MAE value decreases from 0.956m to 0.898m. Compared to the baseline, the MAE value after adding the MSGa module is improved by 6.1%. This shows that mitigating the spatial inconsistency of features at different levels through multi-scale geometric alignment is beneficial to improving prediction accuracy. Similarly, when the GAFFB module is added alone to the baseline, the MAE value decreases to 0.851m. Compared to the baseline, the MAE value after adding the GAFFB module is improved by 11.0%. This indicates that the explicit introduction of geometric prior knowledge is more effective than purely data-driven implicit learning methods. Finally, when the MGCL loss constraint is added alone to the baseline, the MAE value decreases to 0.915m. Compared to the baseline, the introduction of MGCL improves the MAE value by 4.3%. This shows that when geometric consistency loss and physical constraint loss are used to guide the network learning from the level of supervision signals, the improvement in model prediction performance is also obvious.

In addition, Table 3 further demonstrates the synergy between components. As shown in Table 3, the performance of the combination of MSGa+GAFFB is significantly better than the use of these two modules alone, and the improvement exceeds the simple accumulation after introducing a single module. This shows that spatial alignment provides a more reliable feature

basis for geometric feature extraction, and there is an obvious mutual promotion between the two. When GAFFB+MGCL is combined, the model also shows good synergy. It can be seen that the cooperation of geometric modeling and geometric constraints at the architectural level can provide better help for improving the prediction performance of the model.

When GA-DPNet integrates the MSGA, GAFFB, and MGCL components, the MAE value reaches an optimal 0.783m. Compared to the baseline, the MAE is improved by 18.1%. Notably, GA-DPNet’s RMSE value drops significantly to 1.219m. Compared to the baseline, the RMSE decreases by 27.1%, far exceeding the decrease in MAE. This indicates that GA-DPNet is more effective in suppressing large errors and anomalies in prediction, resulting in better overall stability and reliability. Furthermore, GA-DPNet’s SSIM value increases to 0.963, an increase of 4.5%. This also verifies that GA-DPNet’s geometric perception mechanism has a significant advantage in preserving ground feature characteristics. In summary, the MSGA, GAFFB, and MGCL components constitute a complete geometric perception system at the three levels of feature processing, feature fusion, and loss optimization, ensuring the rationality and accuracy of GA-DPNet in building height prediction.

Table 4. Ablation study of geometric features in GFE on the Vaihingen dataset.

Method	MAE↓	RMSE↓	SSIM↑
w/o Gradient (G)	0.821	1.312	0.951
w/o Curvature (C)	0.806	1.274	0.957
w/o Planarity (P)	0.812	1.256	0.959
w/o Edge (E)	0.778	1.226	0.948
GFE (Full)	0.783	1.219	0.963

To further validate the individual contributions and complementarity of the four geometric features in GFE, we conduct ablation experiments on the Vaihingen dataset by removing each feature individually. As shown in Table 4, after removing the gradient, curvature, and planarity features, the MAE and RMSE of the model increase to varying degrees while SSIM decreases accordingly, demonstrating that all three features have a significant impact on model performance. Furthermore, after removing the edge feature, the model achieves the best MAE but exhibits the most significant decrease in SSIM, indicating that the edge feature is indispensable for maintaining the structural integrity of ground object contours. When all four geometric features are retained, the model achieves the best RMSE and SSIM with the best overall performance, verifying the necessity and synergistic complementarity of the four geometric features.

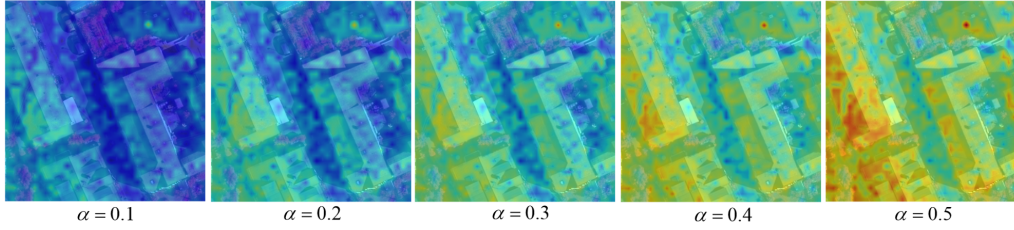
4.5. Parameter Sensitivity

In order to verify the rationality of the modulation intensity hyperparameter α in the MSGA module, we performed a sensitivity analysis on the Vaihingen data set for α in the range of 0.1, 0.2, 0.3, 0.4, 0.5. As shown in Table 5, when the value of α is too small, the spatial guidance signal is insufficient, shallow features cannot be fully guided by spatial alignment, and the model performance declines. When the value of α is too large, excessive feature modulation destroys the original distribution of pre-trained features, which also leads to performance degradation. When α is 0.3, the model achieves optimal results in all three indicators, and when α changes within the range of 0.2 to 0.4, the MAE change does not exceed 2%. This shows that GA-DPNet has better robustness to the selection of α .

Table 5. Sensitivity analysis of super parameter α .

α	MAE↓	RMSE↓	SSIM↑
0.1	0.809	1.276	0.957
0.2	0.795	1.248	0.960
0.3	0.783	1.219	0.963
0.4	0.798	1.257	0.959
0.5	0.816	1.289	0.955

Figure 6 shows the heat map of the MSGA spatial guidance map $\mathbf{G}_4$ under different α values, which is used to qualitatively reflect the influence trend of modulation intensity on the characteristic response under different α values. When α is small, the spatial saliency distribution is relatively uniform, and the response difference between the ground object target area and the background area is not obvious. When α is 0.3, the overall contour area of the ground object gets a clearer and continuous response. This shows that when the modulation intensity increases, it can effectively guide features to focus on key feature areas. However, when α is too large, a strong response also appears in the background area. It can be seen that over-modulation will introduce background interference, thereby affecting the original distribution of shallow features.

**Figure 6.** Heatmap of $\mathbf{G}_4$ under different α values.

To investigate the impact of loss function weight configuration on GA-DPNet performance, we conduct a systematic sensitivity analysis on the Vaihingen dataset. As described in Section 3.4, MGCL consists of three parts: the basic regression loss $\mathcal{L}_{base}$, the geometric consistency loss $\mathcal{L}_{geo}$, and the physical constraint loss $\mathcal{L}_{phy}$, with corresponding weight coefficients λ_{geo} and λ_{phy} . Furthermore, $\mathcal{L}_{geo}$ includes three sub-losses: smoothness $\mathcal{L}_{smooth}$, gradient magnitude consistency $\mathcal{L}_{grad}$, and normal vector $\mathcal{L}_{normal}$, with corresponding weight coefficients λ_s , λ_g and λ_n , respectively. Similarly, $\mathcal{L}_{phy}$ includes two sub-losses: slope constraint $\mathcal{L}_{slope}$ and planar constraint $\mathcal{L}_{planar}$, with corresponding weight coefficients λ_{slope} and λ_{planar} , respectively. Table 6 shows the performance corresponding to the above weights under different configurations.

The experimental results shown in Table 3 demonstrate that the weighting of λ_{geo} and λ_{phy} has a significant impact on model performance. We fix the weight of the basic regression loss, making it dominant, to ensure basic numerical fitting accuracy. As shown in row 1, when λ_{geo} and λ_{phy} are too small, the values of MAE, RMSE, and SSIM are 0.841m, 1.523m, and 0.943, respectively. As shown in row 3, as the weights increase to the optimal configuration, MAE and RMSE decrease to 0.783m and 1.219m, respectively, while SSIM increases to 0.962. The

Table 6. Sensitivity analysis of loss function weights on the Vaihingen dataset.

<i>Number</i>	λ_{geo}	λ_{phy}	$\lambda_s:\lambda_g:\lambda_n$	$\lambda_{slope}:\lambda_{planar}$	MAE↓	RMSE↓	SSIM↑
1	0.01	0.01	0.5:1.0:0.5	1.0:0.1	0.841	1.523	0.943
2	0.05	0.03	0.5:1.0:0.5	1.0:0.1	0.809	1.434	0.951
3	0.10	0.05	0.5:1.0:0.5	1.0:0.1	0.783	1.219	0.963
4	0.15	0.05	0.5:1.0:0.5	1.0:0.1	0.796	1.268	0.958
5	0.20	0.05	0.5:1.0:0.5	1.0:0.1	0.815	1.354	0.951
6	0.10	0.10	0.5:1.0:0.5	1.0:0.1	0.798	1.312	0.956
7	0.10	0.05	0.5:0.5:0.5	1.0:0.1	0.812	1.476	0.948
8	0.10	0.05	0.7:1.0:0.5	1.0:0.1	0.801	1.458	0.951
9	0.10	0.05	0.5:1.0:0.5	1.0:0.2	0.806	1.461	0.950

decreases in RMSE and MAE are 20.0% and 6.9%, respectively. Clearly, the decrease in RMSE is significantly greater than that of MAE. This further verifies the effectiveness of geometric constraints in suppressing large errors. Conversely, as shown in rows 4 and 5, when λ_{geo} increases from 0.10 to 0.15 and 0.20 respectively, MAE rises to 0.796m and 0.815m respectively, and correspondingly, RMSE increases to 1.268m and 1.354m. Clearly, the excessively strong geometric constraints at this point interfere with the fundamental regression. Similarly, as shown in line 6, when λ_{phy} increases to 0.10, excessive physical constraints actually lead to an increase in the MAE value. Experimental results indicate that, to ensure a good balance between constraint strength and fitting accuracy, the optimal configuration values for λ_{geo} and λ_{phy} are 0.10 and 0.05, respectively.

Further observation of the experimental results in Table 3 shows that the weight configuration of the geometric consistency sub-loss and the physical constraint sub-loss also affects the model's performance. When the weight λ_g of the gradient magnitude consistency loss decreases from 1.0 to 0.5, the RMSE value increases significantly to 1.476m. For the weights λ_s of the smoothness loss and λ_n of the normal vector loss, a value of 0.5 yields the best performance. Furthermore, for the physical constraint sub-loss, the model performs best when λ_{slope} and λ_{planar} are set to 1.0 and 0.1, respectively. However, when λ_{planar} is increased to 0.2, the excessively strong planarity constraint actually suppresses the true details of ground objects.

It can be seen that the optimal parameter configuration of GA-DPNet is shown in line 3 of Table 3. In addition, when λ_{geo} and λ_{phy} change between 0.05 to 0.15 and 0.03 to 0.10 respectively, the performance fluctuation of GA-DPNet is relatively flat. At this point, the change in MAE is less than 4%. This shows that GA-DPNet is robust to weight allocation.

5. CONCLUSION

Aiming at the performance defects caused by existing methods in monocular height estimation from remote sensing images, which only rely on semantic features while neglecting geometric structural information, we propose a Geometry-Aware Dense Prediction Network for Monocular Height Estimation from Remote Sensing Images (GA-DPNet). At the feature processing stage, we design a Multi-Scale Geometric Alignment module (MSGa) to reduce the spatial cognitive inconsistency among multi-scale features extracted by DINOv2. At the feature fusion stage, we design a Geometry-Aware Feature Fusion Block (GAFFB), which achieves deep

fusion of geometric information with semantic features by explicitly extracting four types of geometric features including gradient, curvature, planarity, and edge. At the loss function design stage, we construct a Multi-component Geometric Constraint Loss (MGCL) to constrain the predictions from three dimensions of pixel accuracy, geometric consistency, and physical rationality. Experimental results on three public datasets, Vaihingen, Potsdam, and DFC2019, show that GA-DPNet achieves comprehensively superior performance on the Vaihingen and DFC2019 datasets. Moreover, GA-DPNet remains competitive on the Potsdam dataset. These results demonstrate the effectiveness and generalization ability of the proposed method.

However, the proposed method has certain limitations. The publicly available datasets used for training and testing GA-DPNet, including Vaihingen, Potsdam, and DFC2019, were all collected under clear weather conditions, lacking systematic coverage of adverse weather environments such as fog, shadows, and cloud cover. Future research could focus on constructing datasets and optimizing the model under various weather conditions to further improve the model's robustness in complex environments.

Author Contributions

Fan Yang: Developed the geometry-aware framework and proposed the core methodology. Led the overall design and implementation of the GA-DPNet network including MSGA, GAFFB, and MGCL modules. Conducted all experiments, performed formal analysis, and prepared the original manuscript with visualizations.

Yuecheng Yu: Provided overall guidance and supervision for the research. Assisted in methodology development and network implementation. Participated in experimental validation and parameter optimization.

Lizhou Huang & Yuanju Lu: Provided computational resources and managed data collection and preprocessing. Assisted in experimental validation and result verification.

Jiasong Shi & Jiemin Zhang: Supervised the research project and provided strategic guidance. Made critical decisions on research direction and technical approaches.

Acknowledgements

This work was supported by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. SJCX25_2523). The authors gratefully acknowledge the ISPRS and IEEE GRSS for providing the Vaihingen, Potsdam, and DFC2019 datasets. We also thank the anonymous reviewers for their valuable comments and suggestions that helped improve this manuscript.

References

- [1] C. Beumier and M. Idrissa, "Digital terrain models derived from digital surface model uniform regions in urban areas," *Int. J. Remote Sens.*, vol. 37, no. 15, pp. 3477–3493, 2016.
- [2] J. Tu, H. Sui, W. Feng, and Z. Song, "Automatic building damage detection method using high-resolution remote sensing images and 3D GIS model," in *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. III-8, 2016, pp. 43–50.
- [3] J. Mahmud, T. Price, A. Bapat, and J.-M. Frahm, "Boundary-aware 3D building reconstruction from a single overhead image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 441–451.
- [4] C. Zhang, Y. Cui, Z. Zhu, S. Jiang, and W. Jiang, "Building height extraction from GF-7 satellite images based on roof contour constrained stereo matching," *Remote Sens.*, vol. 14, no. 7, Art. no. 1566, 2022.

- [5] J. Ten Harkel, H. Bartholomeus, and L. Kooistra, "Biomass and crop height estimation of different crops using UAV-based LiDAR," *Remote Sens.*, vol. 12, no. 1, Art. no. 17, 2019.
- [6] W. Zhao, C. Persello, and A. Stein, "Semantic-aware unsupervised domain adaptation for height estimation from single-view aerial images," *ISPRS J. Photogramm. Remote Sens.*, vol. 196, pp. 372–385, 2023.
- [7] G. Liasis and S. Stavrou, "Satellite images analysis for shadow detection and building height estimation," *ISPRS J. Photogramm. Remote Sens.*, vol. 119, pp. 437–450, 2016.
- [8] M. Izadi and P. Saeedi, "Three-dimensional polygonal building model estimation from single satellite images," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 6, pp. 2254–2272, Jun. 2012.
- [9] A. Khan *et al.*, "A survey of the vision transformers and their CNN-transformer based variants," *Artif. Intell. Rev.*, vol. 56, no. S3, pp. 2917–2970, 2023.
- [10] L. Mou and X. X. Zhu, "IM2HEIGHT: Height estimation from single monocular imagery via fully residual convolutional-deconvolutional network," arXiv:1802.10249, Feb. 2018.
- [11] H. A. Amirkolaee and H. Arefi, "Height estimation from single aerial images using a deep convolutional encoder-decoder network," *ISPRS J. Photogramm. Remote Sens.*, vol. 149, pp. 50–66, Mar. 2019.
- [12] C.-J. Liu, V. A. Krylov, P. Kane, G. Kavanagh, and R. Dahyot, "IM2ELEVATION: Building height estimation from single-view aerial imagery," *Remote Sens.*, vol. 12, no. 17, Art. no. 2719, 2020.
- [13] S. Xing, Q. Dong, and Z. Hu, "Gated feature aggregation for height estimation from single aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Art. no. 9469035, 2021.
- [14] S. Chen, Y. Shi, Z. Xiong, and X. X. Zhu, "HTC-DC Net: Monocular height estimation from single remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Art. no. 5623018, 2023.
- [15] Z. Chen *et al.*, "HeightFormer: A multilevel interaction and image-adaptive classification–regression network for monocular height estimation with aerial images," *Remote Sens.*, vol. 16, no. 2, Art. no. 295, 2024.
- [16] J. Wu *et al.*, "HeightFormer: Single imagery height estimation transformer with bilateral feature pyramid fusion," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, Art. no. 10681105, 2024, doi: 10.1109/LGRS.2024.3461791.
- [17] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 10012–10022.
- [18] M. Carvalho, B. Le Saux, P. Trouvé-Peloux, A. Almansa, and F. Champagnat, "Multitask learning of height and semantics from aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1391–1395, Aug. 2020.
- [19] S. Xing, Q. Dong, and Z. Hu, "SCE-Net: Self- and cross-enhancement network for single-view height estimation and semantic segmentation," *Remote Sens.*, vol. 14, no. 9, Art. no. 2252, 2022.
- [20] Z. Gao *et al.*, "Joint learning of semantic segmentation and height estimation for remote sensing image leveraging contrastive learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Art. no. 5611015, 2023.
- [21] S. Du *et al.*, "LUMNet: Land use knowledge guided multiscale network for height estimation from single remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, Art. no. 6007505, 2024.
- [22] X. Peng *et al.*, "Synergistic semantic segmentation and height estimation for monocular remote sensing images via cross-task interaction," *Remote Sens.*, vol. 17, no. 9, Art. no. 1637, May 2025.
- [23] Y. Lei and W. Jiang, "A relation aware and edge preserving height refinement network for single-view height estimation from remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 18, pp. 12988–13002, 2025.
- [24] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv:2010.11929, Oct. 2020.
- [25] A. A. Aleissae *et al.*, "Transformers in remote sensing: A survey," *Remote Sens.*, vol. 15, no. 7, Art. no. 1860, 2023.
- [26] M. Caron *et al.*, "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 9650–9660.
- [27] M. Oquab *et al.*, "DINOv2: Learning robust visual features without supervision," arXiv:2304.07193, Apr. 2023.
- [28] J. Song *et al.*, "SynRS3D: A synthetic dataset for global 3D semantic understanding from monocular remote sensing imagery," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 37, 2024, pp. 117388–117425.
- [29] M. Luo *et al.*, "Domain generalization for semantic segmentation of remote sensing images via vision foundation model fine-tuning," *ISPRS J. Photogramm. Remote Sens.*, vol. 230, pp. 126–146, 2025.
- [30] I. Ray and A. Skurikhin, "Deep clustering of remote sensing scenes through heterogeneous transfer learning," *Int. J. Remote Sens.*, vol. 46, no. 8, pp. 3089–3109, 2025.
- [31] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention (MICCAI)*, Cham, Switzerland: Springer, 2015, pp. 234–241.
- [32] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 2117–2125.
- [33] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 8759–8768.

- [34] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 12179–12188.
- [35] J. Song and S. J. Lee, "Knowledge distillation of multi-scale dense prediction transformer for self-supervised depth estimation," *Sci. Rep.*, vol. 13, no. 1, Art. no. 18939, Nov. 2023.
- [36] J. Gąsienica-Józkowy *et al.*, "Deep learning-based monocular estimation of distance and height for edge devices," *Information*, vol. 15, no. 8, Art. no. 474, 2024.
- [37] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 270–279.
- [38] M. Ramamonjisoa and V. Lepetit, "SharpNet: Fast and accurate recovery of occluding contours in monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, 2019, pp. 2109–2118.
- [39] V. Patil *et al.*, "P3Depth: Monocular depth estimation with a piecewise planarity prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 1610–1621.
- [40] G. Yang *et al.*, "Transformer-based attention networks for continuous pixel-wise prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 16249–16259.
- [41] Y. Huang *et al.*, "Comparative analysis of ImageNet pre-trained deep learning models and DINOv2 in medical imaging classification," in *Proc. IEEE 48th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, 2024, pp. 1–6.
- [42] C. Xu *et al.*, "Hybrid attention-aware transformer network collaborative multiscale feature alignment for building change detection," *IEEE Trans. Instrum. Meas.*, vol. 73, Art. no. 5023314, 2024.
- [43] L. Scorzato, "Reliability and interpretability in science and deep learning," *Minds Mach.*, vol. 34, no. 3, Art. no. 27, 2024.
- [44] T. Naderi, A. Sadovnik, J. Hayward, and H. Qi, "Monocular depth estimation with adaptive geometric attention," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, 2022, pp. 944–954.

Author Biography

Fan Yang: M.S. student. Research interests include computer vision and image processing.

Yuecheng Yu: (corresponding author) Professor and Master's supervisor. Research interests include computer vision and image processing.

Lizhou Huang: Mainly engaged in research on nuclear, biological, and chemical (NBC) informatization.

Yuanju Lu: M.S. student. Research interests include computer vision and image processing.

Jiasong Shi: Mainly engaged in research on nuclear, biological, and chemical (NBC) informatization.

Jiemin Zhang: Mainly engaged in research on nuclear, biological, and chemical (NBC) informatization.