remote sensing

MDPI

Article

GeoAdapt: Fine-Grained Keypoint Localization via Deformable Feature Refinement for Ground-Based Optical Remote Sensing

Yingwei Xia [1], Tian Yu [1,2], Wang Xi [1,2], Fan Wang [1], Yong Liu [1], Nanhao Liang [1,2] and Wen Zhang [1,3,*]

[1] Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China; xiayw@aiofm.ac.cn (Y.X.); tian_yu@mail.ustc.edu.cn (T.Y.); xw_cs@mail.ustc.edu.cn (W.X.); wangf@aiofm.ac.cn (F.W.); liuyong@aiofm.ac.cn (Y.L.); lnh@mail.ustc.edu.cn (N.L.)
[2] Science Island Branch of Graduate School, University of Science and Technology of China, Hefei 230026, China
[3] Wanjiang Center for Development of Emerging Industrial Technology, Tongling 244000, China
* Correspondence: zhangwen@aiofm.ac.cn

Highlights

What are the main findings?

- A lightweight deformable keypoint detector achieves a percentage of correct keypoints (PCK) of 89.3% on a synthetic ground-based optical remote sensing benchmark and the highest zero-shot PCK among the evaluated models on manually annotated real long-range optical imagery in six scenarios.
- Deformable convolution at the feature-pyramid level, combined with a rigidity-aware keypoint loss replacing the standard human-pose formulation, accounts for over 80% of the accuracy gain under the evaluated conditions, demonstrating the value of spatially adaptive refinement for fine-grained target characterization.

What are the implications of the main findings?

- Ground-based optical remote sensing at kilometer-scale standoff distances can benefit from compact models with geometric adaptation mechanisms that respond to haze, blur, and viewpoint-dependent appearance variation; explicit turbulence compensation remains outside the scope of this study.
- The physically based synthetic-data pipeline provides a reproducible approach to training deep learning models for remote sensing applications in which annotated real imagery is scarce or costly.

Abstract

Ground-based optical remote sensing of aerial targets at kilometer-scale standoff distances requires accurate keypoint localization for six-degree-of-freedom (6-DoF) pose recovery under variable illumination, motion blur, and atmospheric degradation. Many lightweight detectors use fixed-kernel convolutions, whose spatially invariant sampling may limit adaptation to heterogeneous target geometries and spatially varying image degradation. We introduce GeoAdapt, a compact keypoint detection framework that inserts deformable convolution v2 (DCNv2) modules between the feature pyramid network (FPN) neck and the detection head. GeoAdapt also replaces the standard object keypoint similarity (OKS) loss with a combination of Wing Loss and Bone Loss. The complete model contains 5.95 M parameters, 47.9% fewer than the You Only Look Once version 8 small pose model (YOLOv8s-pose). On a synthetic ground-based optical remote sensing benchmark, GeoAdapt achieved a percentage of correct keypoints (PCK) at a threshold of 0.05 times the bounding-box diagonal (PCK@0.05D) of 89.3% and a rotation error of 11.6°, improving PCK by 11.7 percentage points over YOLOv8s-pose. Zero-shot evaluation on manually

Check for updates

Academic Editor: Zhenwei Shi

Received: 7 July 2026
Revised: 24 July 2026
Accepted: 30 July 2026
Published: 3 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

annotated real ScanEagle and Matrice 200 imagery showed consistent advantages over YOLOv8s-pose and YOLO11s-pose in all six test scenarios. A factorial ablation indicated a positive interaction between DCNv2 and the Wing+Bone loss.

Keywords: ground-based optical remote sensing; lightweight deep learning; keypoint detection; deformable convolution; six-degree-of-freedom pose estimation

1. Introduction

Ground-based optical remote sensing of aircraft and unmanned aerial vehicles (UAVs) supports airspace monitoring, target reacquisition, and sensor pointing at kilometer-scale standoff distances [1,2]. Unlike nadir-viewing satellite or airborne systems, a ground-based electro-optical system observes targets through a long, near-horizontal atmospheric path. Haze, variable illumination, blur, and viewpoint-dependent appearance can therefore obscure fine structural details [3]. Bounding boxes describe target extent but do not provide the orientation and relative translation required by downstream pose-aware applications. We therefore study camera-frame six-degree-of-freedom (6-DoF) pose recovery from structural keypoints, including the nose, tail, wing tips, and propeller centers. The estimated pose is defined relative to the optical camera and can be georeferenced only when external camera calibration and platform geolocation are available.

The task combines two constraints. First, accurate 6-DoF recovery requires keypoint localization rather than box detection alone. Under the keypoint-to-Perspective-n-Point (PnP) paradigm [4–6], a network predicts the image coordinates of $N = 8$ predefined three-dimensional (3D) keypoints, and a PnP solver [7,8] recovers the rigid pose $(R, t) \in SE(3)$. Second, deployment favors compact models because many high-capacity remote sensing networks are impractical for resource-constrained platforms [9–13]. The challenge is therefore to preserve fine spatial accuracy within limited parameter and computational budgets.

Existing approaches present complementary strengths and limitations. Transformer-based DroneKey and DroneKey++ use dedicated keypoint tokens and achieve high accuracy on a constrained UAV benchmark, but their approximately 76 M parameters limit their suitability for compact deployment [14,15]. Models in the You Only Look Once (YOLO)-Pose family provide efficient single-stage box and keypoint prediction [16–18]; however, their standard fixed-grid convolutions provide no position-dependent sampling before the prediction head. In addition, the default object keypoint similarity (OKS) loss [19] uses sensitivity constants calibrated from human annotations. No canonical constants exist for rigid UAV landmarks, and OKS imposes no explicit constraint on inter-keypoint geometry. These observations motivate spatially adaptive feature refinement and a loss function tailored to rigid aerial-target geometry.

We use YOLOv8s-pose as a reproducible engineering reference rather than as a uniquely suitable or universally superior architecture. Its open implementation, integrated box–keypoint prediction, and modular neck–head interface support controlled modification and component-level ablation. GeoAdapt therefore retains the single-stage prediction framework while modifying the feature-refinement and keypoint-loss components that are most directly related to the identified task mismatch. Section 2 discusses recent one-stage, Transformer-based, and state-space alternatives and establishes the scope and comparability criteria for the empirical evaluation in Section 4.

GeoAdapt inserts deformable convolution v2 (DCNv2) modules [20] between the feature pyramid network (FPN) neck and the detection head. It also replaces the OKS training loss with normalized Wing Loss [21] and Bone Loss [22,23]. The complete lightweight

model contains 5.95 M parameters. Figure 1 summarizes the pipeline. The main contributions are as follows:

1. **Task-specific feature and loss design.** Feature-level DCNv2 provides position-dependent sampling at three FPN scales, whereas normalized Wing Loss and Bone Loss emphasize small localization errors and pairwise geometric consistency. Controlled ablations quantify the contributions and interaction of these components.
2. **Compact architecture.** Partial convolution and inference-time reparameterization allocate capacity to keypoint-sensitive features while producing a complete 5.95 M-parameter model, 47.9% smaller than YOLOv8s-pose. We report the parameters, floating-point operations (FLOPs), and measured throughput separately rather than treating model size as a proxy for latency.
3. **Multilevel evaluation.** A Blender Cycles pipeline with five-dimensional domain randomization [24,25] generates 10,000 frames for two UAV geometries, with annotations defined by model-embedded keypoints. The evaluation covers keypoint accuracy, 6-DoF pose, component ablation, computational cost, transfer to DroneKey, and zero-shot performance on manually annotated real imagery.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the materials and methods, including GeoAdapt, data generation, and pose recovery. Section 4 reports the results. Section 5 discusses the findings and limitations, and Section 6 concludes the paper.

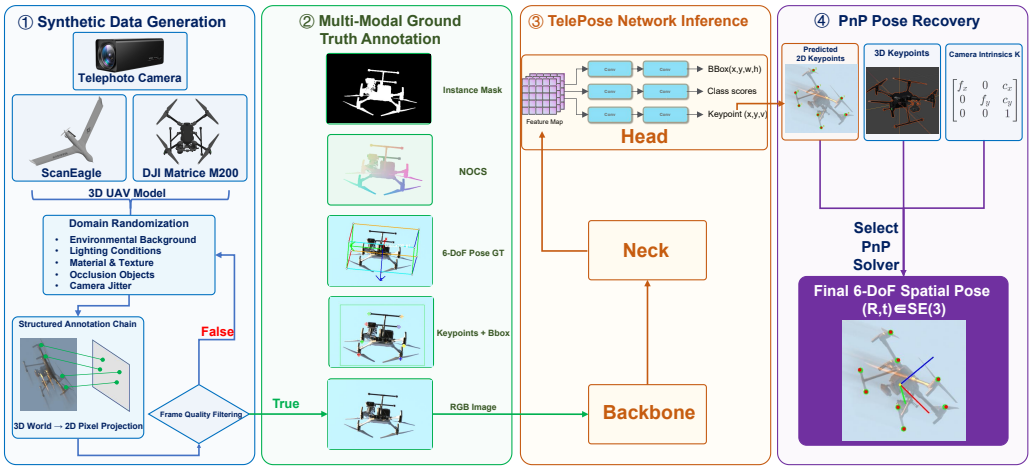


Figure 1. End-to-end GeoAdapt pipeline comprising synthetic-data generation with domain randomization, multimodal ground-truth anchoring, deformable keypoint detection, and Perspective-*n*-Point (PnP)-based six-degree-of-freedom (6-DoF) pose recovery for ground-based optical remote sensing. UAV, unmanned aerial vehicle; 2D, two-dimensional; 3D, three-dimensional; RGB, red–green–blue; GT, ground truth; Bbox, bounding box; NOCS, normalized object coordinate space; Conv, convolution; and SE(3), special Euclidean group in three dimensions.

2. Related Work

2.1. Lightweight Object Detection in Optical Remote Sensing

Object detection in optical remote sensing (RS) imagery has shifted from handcrafted features to deep convolutional networks over the past decade [9–11]. The YOLO family [18,26] has been widely adopted because of its speed–accuracy trade-off. Lightweight YOLOv8-based aircraft detectors [1], cross-scale information-mixing blocks [27], and attention-enhanced feature fusion [28] illustrate a broader trend toward jointly designing detection heads and feature aggregation for RS-specific challenges. Lightweight network design has also become important for RS edge deployment. Common compression strategies, including pruning [29], quantization, and knowledge distillation [13,30,31],

have been adapted to RS. In parallel, lightweight backbones such as MobileNetV3 [32], ShuffleNet V2 [33], and GhostNet [34] replace computationally expensive convolutions with efficient alternatives. Partial convolution in FasterNet [35] further reduces floating-point operations (FLOPs) by convolving only a subset of channels, whereas structural reparameterization [36,37] fuses multibranch training structures into single-path inference models. However, these approaches primarily address box-level detection or classification. Fine-grained keypoint localization for 6-DoF pose recovery imposes spatial-accuracy requirements that horizontal-box detectors do not need to satisfy.

2.2. Cross-Scale and Multimodal Feature Fusion

Recent optical remote sensing studies have examined feature fusion under low contrast, scale variation, and background clutter. PCFNet combines phase-congruency cues with cross-scale contextual fusion to preserve weak boundaries in salient-object detection for optical remote sensing imagery [38]. CMNFNet progressively integrates nested cross-model features to suppress background interference [39]. In related multimodal vision research, UMINet uses asymmetric red–green–blue (RGB) and depth or thermal branches with difference-aware fusion to reconcile complementary modalities [40]. These studies demonstrate the broader value of cross-scale context, feature selection, and complementary representations. However, their outputs are pixel-level saliency maps rather than category-specific rigid keypoints. They therefore provide relevant feature-fusion context but are not direct baselines for GeoAdapt's keypoint-to-PnP task.

2.3. Ground-Based Optical Remote Sensing and Long-Range Imaging

Ground-based optical remote sensing encompasses established applications beyond satellite and airborne platforms. Terrestrial laser scanning and ground-based light detection and ranging (LiDAR) are routinely used for topographic mapping, vegetation-structure analysis, and deformation monitoring [9]. Ground-based all-sky cameras and horizon-scanning radiometers provide continuous atmospheric observations at meteorological stations. In long-range imaging, ground-based telescopes equipped with adaptive optics resolve astronomical and space-surveillance targets through kilometer-scale atmospheric paths while contending with scintillation, anisoplanatic blur, and turbulence-induced beam wander. The configuration studied here uses a fixed, narrow-field-of-view (FOV) electro-optical sensor to characterize aerial targets at ranges of 0.8–1.2 km. It shares three characteristics with these established modalities: noncontact measurement over a substantial standoff distance, atmospheric radiative transfer between the target and sensor, and computational postprocessing to estimate target state from degraded optical measurements. Despite these similarities, ground-based optical remote sensing of aerial targets remains less studied than satellite and airborne RS, particularly for keypoint-level characterization, in which atmospheric degradation can strongly affect localization.

2.4. Keypoint-Based 6-DoF Pose Estimation

The keypoint-to-PnP paradigm for 6-DoF pose estimation was introduced by BB8 [4] and extended by PVNet [41], HybridPose [42], GDR-Net [43], and differentiable PnP [44]. Within the YOLO family, YOLO-Pose [16] adapted the detection head to multiperson two-dimensional (2D) keypoints, and Yolo-Key-6D [45] extended this design to 6-DoF object pose. In RS applications, Fotiadis et al. [3] estimated aircraft pose from keypoints and structures, whereas Wang et al. [46] used keypoints for fine-grained aircraft recognition.

Recent alternatives extend beyond conventional YOLO-Pose designs. RTMO introduces dynamic coordinate classification for high-performance one-stage pose estimation [47]; ED-Pose uses explicit box and keypoint decoding in an end-to-end Transformer framework [48]; and MambaPose integrates Mamba-based feature modeling into a YOLO-

Pose-style human-keypoint detection framework [49]. These methods provide important alternative design directions, but their original interfaces target human keypoints. Applying them to rigid UAV landmarks requires reconfiguring category-specific keypoint heads, topology, dataset metadata, or optimization settings. We therefore distinguish complete-framework comparisons from evaluation-matched representation probes and do not treat published results obtained with task-incompatible settings as directly comparable evidence.

For spatial adaptability, DCNv2 [20] augments convolution with learnable offsets and modulation weights and has been effective in RS object detection [50–52]. Attention mechanisms, including squeeze-and-excitation (SE) [53], the convolutional block attention module (CBAM) [54], and the simple attention module (SimAM) [55], as well as large-kernel convolutions [56–58], provide complementary forms of spatial recalibration. Existing RS applications generally place deformable modules within the backbone or detection head and do not jointly examine feature-level placement and a rigid-keypoint loss. GeoAdapt focuses on this specific combination; it does not claim that deformable convolution, lightweight parameterization, or FPN interfaces are unique to its YOLOv8 reference implementation.

2.5. Keypoint Losses and Synthetic Data Generation

Keypoint-regression accuracy depends jointly on the architecture and loss function. OKS [19] uses per-keypoint standard deviations calibrated to human-annotation variability; these values are not defined for nonhuman keypoints [59]. Wing Loss [21] increases the influence of small and moderate errors relative to conventional L_1 and L_2 objectives, whereas skeleton-aware losses [22,23,60] encode constraints between linked keypoints. Annotation characteristics in RS keypoint detection differ from those in human-pose benchmarks. Keypoints on rigid aerial targets obey strong geometric constraints but have small inter-keypoint spacing at long imaging distances, creating an error regime in which the Gaussian decay of OKS may be poorly matched to the task. Synthetic-data generation addresses the RS annotation bottleneck: domain randomization [24,25] supports synthetic-to-real transfer, and photorealistic pipelines such as BlenderProc [61] and AirSim [62] have been validated for RS tasks [63]. Diffusion-based RS data generators, including AeroGen and TerraGen [64,65], can also create annotated imagery on demand. However, existing synthetic UAV-pose datasets primarily randomize background appearance rather than the physical imaging factors that dominate long-range optical degradation, including atmospheric scattering, sensor noise, and motion blur. They also lack the aligned normalized object coordinate space (NOCS), mask, and keypoint annotations required by an end-to-end synthetic-to-real keypoint-detection pipeline.

2.6. Research Gaps and Motivation for GeoAdapt

In summary, three gaps motivate this study. **(1) Architecture gap:** Lightweight RS detectors have rarely been evaluated with deformable refinement at the FPN–head interface for category-specific keypoints, and the interaction between this placement and keypoint-loss design remains unclear. **(2) Loss gap:** OKS is commonly inherited from human-pose pipelines, although its sensitivity constants are not calibrated for rigid UAV landmarks and it does not encode inter-keypoint geometry. Wing Loss and geometry-aware penalties address complementary aspects of this mismatch, but their combination has received limited attention in RS keypoint localization. **(3) Evaluation gap:** Existing UAV keypoint benchmarks, such as DroneKey, cover a restricted range of target geometries and imaging conditions. To our knowledge, public evaluations of long-range ground-based optical imagery with heterogeneous airframes, physically motivated image degradation, and joint keypoint–pose metrics remain limited. GeoAdapt addresses these gaps through

feature-level DCNv2 refinement, a combined Wing+Bone loss, and a five-dimensional domain-randomized synthetic dataset containing two UAV classes.

3. Materials and Methods

This section links each design choice to an observed task characteristic without assuming that any mechanism is unique to the YOLO family. Long-range imaging can produce spatially nonuniform appearance degradation, whereas fixed-grid convolution applies the same sampling pattern at every location. Feature-level DCNv2 (Section 3.2.1) is therefore evaluated as a learned, image-driven refinement mechanism rather than as a physical correction for atmospheric turbulence. UAV landmarks are sparse and geometrically constrained, unlike the human keypoints used to calibrate standard OKS constants. The Wing+Bone loss (Section 3.3) consequently combines sensitivity to small errors with pairwise distance consistency. Because annotated real long-range imagery is scarce, the synthetic pipeline (Section 3.4) provides model-embedded keypoints and five-dimensional domain randomization. A fixed 2D–3D correspondence order then supports PnP-based pose recovery (Section 3.5). The following subsections present the problem formulation, network architecture, loss design, data-generation pipeline, and pose-recovery procedure.

3.1. Problem Formulation

Given a single RGB image $I \in \mathbb{R}^{H\times W\times 3}$ captured by a calibrated long-focal-length pinhole camera with intrinsic matrix

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3}, \tag{1}$$

and $N = 8$ known 3D keypoints $\{X_n\}_{n=1}^{N} \subset \mathbb{R}^3$ rigidly attached to an aerial target of class $c \in \{1, 2\}$ (ScanEagle or DJI Matrice 200), the model predicts a 2D bounding box $\mathbf{b} = (x, y, w, h)$, a class label $\hat{c}$, and eight 2D keypoint locations with visibility states $\{(\hat{u}_n, \hat{v}_n, \hat{v}_n)\}_{n=1}^{N}$. The rigid 6-DoF pose $(R, t) \in SE(3)$ is then recovered such that

$$\pi_K(RX_n + t) \approx (\hat{u}_n, \hat{v}_n)^\top, \qquad n = 1, \ldots, N, \tag{2}$$

where $\pi_K(\mathbf{X}) = (f_x X_1/X_3 + c_x,\ f_y X_2/X_3 + c_y)^\top$. GeoAdapt produces the bounding box, class label, and keypoints in a single forward pass; a standard PnP solver then recovers the pose (Section 3.5). The geometric scale follows from the camera model. For a sensor width $w_s = 36$ mm, focal length $f = 11{,}250$ mm, and image width $w_p = 1024$ px, the per-axis FOV is

$$\theta_{\text{FOV}} = 2\arctan(w_s/2f) \approx 3.20 \text{ mrad}. \tag{3}$$

At a depth of $z = 1000$ m, a ScanEagle occupies nearly the entire image width, whereas a Matrice 200 occupies approximately 19% of the width. Through the Jacobian of π_K, subpixel accuracy in image coordinates corresponds to the following angular accuracy:

$$\Delta\theta = \Delta u/f_x \approx 1 \text{ px}/320{,}000 \text{ px} \approx 0.0031 \text{ mrad}, \tag{4}$$

This corresponds to ≈3.1 mm of geometric sampling per axis at 1 km under the idealized pinhole model. The focal length $f = 11{,}250$ mm represents an asymptotic optical configuration; operational ground-based systems commonly use shorter focal lengths and therefore have coarser per-pixel ground sampling. We used this extreme configuration to reduce pixel-quantization effects and stress fine-grained keypoint localization. The rendering pipeline varied atmospheric scattering, contrast, and blur but did not reproduce

turbulence-induced phase distortions. Consequently, the performance ranking at shorter focal lengths or under physical turbulence requires separate verification.

The target geometry fixes the coordinate frames and 2D–3D correspondences. The 3D keypoints $\{X_n\}$ are defined in the object-local coordinate frame of each UAV model after mesh merging and uniform scaling; the origin is retained from the source FBX model. The target structure defines the coordinate axes: the $+Y$ axis follows the longitudinal fuselage direction from tail to nose, the $+X$ axis points toward the target's right side (the right wing for the fixed-wing target and the right rotor pair for the quadrotor), and the $+Z$ axis points upward, orthogonal to the wing or rotor plane. For ScanEagle, the head and tail keypoints define the longitudinal fuselage axis ($+Y$), and the left- and right-wing keypoint pairs define the transverse axis ($+X$). For the DJI Matrice 200, the front rotor pair (rotor_1/rotor_2) relative to the rear rotor pair (rotor_3/rotor_4) defines $+Y$; the right-side rotor and landing-gear keypoint pairs define $+X$; and the rotor plane relative to the landing-gear direction defines $+Z$. All 3D keypoints are expressed in this object coordinate frame. The pose (R, t) in Equation (2) represents the rigid transformation from the object frame to the OpenCV camera coordinate system ($+X$ right, $+Y$ down, and $+Z$ forward). We obtained this transformation by reversing the Blender camera Y and Z axes during data export. A fixed class-specific output order establishes the correspondence between each predicted keypoint $(\hat{u}_n, \hat{v}_n)$ and its 3D counterpart X_n. The network's eight keypoint channels follow a predefined sequence (e.g., for Matrice 200: $\{$1fan, 1foot, . . . , 4fan, 4foot$\}$; for ScanEagle: $\{$head, tail, left_front_high_wing, . . . , right_behind_low_wing$\}$). The same order indexes the 3D template points during PnP, eliminating the need for post hoc matching [14].

A 6-DoF pose comprises three translational and three rotational degrees of freedom. Four noncoplanar, nondegenerate 2D–3D correspondences are theoretically sufficient for a PnP solution [7,8]. We used eight keypoints to provide redundancy against occlusion and detection failure. These points span the head, tail, wing tips, propeller centers, and landing gear of each airframe and therefore cover the principal geometric extents of the target. The keypoint count represents an engineering trade-off. Additional keypoints could reduce sensitivity to individual errors, but they would also increase annotation complexity and the likelihood of occlusion among closely spaced keypoints on small targets. We do not claim that a set of eight keypoints is optimal; evaluating the accuracy–robustness trade-off as a function of keypoint count remains future work.

The two target classes also differ in keypoint topology. ScanEagle keypoints are distributed along an elongated fuselage and wings, whereas Matrice 200 keypoints are concentrated near four propellers in an approximately square arrangement. A single shared backbone must therefore accommodate heterogeneous geometric distributions, motivating the adaptive sampling of DCNv2 (Section 3.2.1) and the directional aggregation of Strip Pooling (Section 3.2.2).

3.2. GeoAdapt Network Architecture

Figure 2 illustrates the architecture. GeoAdapt retains the YOLOv8s-pose single-stage framework and introduces deformable feature refinement at each FPN scale (Section 3.2.1) and a redesigned keypoint loss (Section 3.3), together with lightweight backbone and neck modifications (Sections 3.2.2 and 3.2.3).

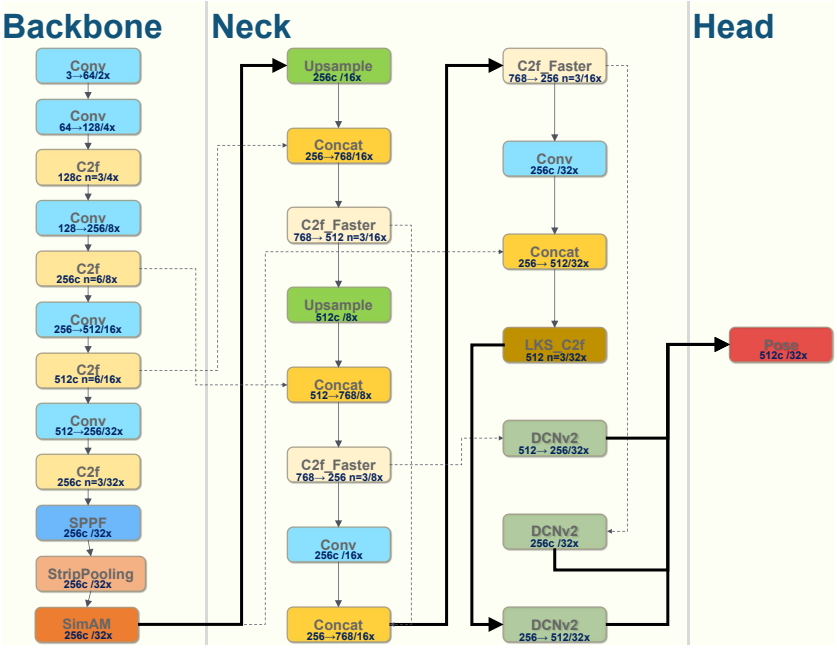


Figure 2. GeoAdapt network architecture. The model comprises a C2f backbone with a triple-enhanced P5 stage, a hybrid C2f_Faster + LKS_C2f neck, and deformable convolution v2 (DCNv2) feature-refinement modules inserted between the neck and the standard pose head at each feature pyramid network (FPN) scale.

3.2.1. Deformable Feature Refinement at FPN Scales

The standard YOLO-Pose architecture [16] feeds FPN-neck outputs directly into a detection head with three parallel 3×3 convolutional branches and therefore provides no position-dependent sampling before the task branches. GeoAdapt inserts one DCNv2 module [20] at each FPN scale *between the neck and the standard head* (Figure 2). This placement has three motivations: (i) it makes the refined representation available to all downstream branches; (ii) the learned offsets Δp_k and modulation weights allow the sampling pattern to respond to local target geometry and image appearance, unlike fixed-grid convolution; and (iii) one DCNv2 module per scale adds only approximately 0.3 M parameters in total, whereas equipping all three head branches incurs a greater cost. DCNv2 provides image-driven adaptation; it is not an explicit physical model of atmospheric distortion.

Each DCNv2 module generalizes a standard 3×3 convolution by displacing the sampling positions with learnable offsets and reweighting them with modulation coefficients:

$$y(p) = \sum_{k=1}^{K} w_k \cdot m_k(p) \cdot x(p + p_k + \Delta p_k(p)), \tag{5}$$

where $y(p)$ is the output feature at spatial position p; $K = 9$ indexes the 3×3 kernel positions; w_k is the learnable convolutional weight at the kth position; $p_k \in \{(-1,-1),(-1,0),\ldots,(1,1)\}$ is the predefined regular-grid offset for the kth sampling location; $\Delta p_k(p) \in \mathbb{R}^2$ is a position-dependent offset predicted by an auxiliary 3×3 convolution applied to the input feature map; $m_k(p) \in [0,1]$ is a modulation weight predicted by a parallel convolution with sigmoid activation; and $x(\cdot)$ denotes the input feature map sampled through bilinear interpolation at fractional coordinates when required.

One DCNv2 module is inserted at each of the three FPN output scales: P3, P4, and P5. Each module preserves the input channel count and spatial resolution, and the refined features are passed to the unchanged YOLO-Pose head. The placement ablation in Section 4.6 compares this feature-level design with symmetric DCNv2 placement in all three head branches.

3.2.2. Backbone with Triple-Enhanced P5

The backbone follows YOLOv8 in stages P1–P4 and uses C2f blocks [66]. At P5, we increase the channel width from 256 to 512 and introduce a triple-enhancement sequence comprising spatial pyramid pooling–fast (SPPF), Strip Pooling [67], and SimAM [55]. The three modules serve complementary functions: SPPF provides isotropic multiscale context, Strip Pooling provides anisotropic directional aggregation for elongated structures such as the ScanEagle fuselage, and SimAM provides pixel-level saliency. The triple stack adds less than 1% to the backbone FLOPs. SPPF applies three max-pooling operations with progressively dilated 5×5 stencils, concatenates their outputs, and projects them through a 1×1 convolution. Strip Pooling averages the feature map horizontally and vertically, broadcasts both summaries, and adds the residual with $O(CHW)$ complexity. SimAM derives a parameter-free spatial-attention mask from an energy-minimization formulation [55].

3.2.3. Hybrid Faster + LKS Neck

The neck follows the bidirectional feature pyramid network/path aggregation network (FPN/PAN) pattern [68,69]. The top-down blocks use C2f_Faster, which is bottlenecked by partial convolution (PConv) [35] and convolves a channel fraction of $r = 0.25$, reducing the convolutional FLOPs by a factor of 16. The bottom-up P5 block uses LKS_C2f with reparameterizable asymmetric depthwise convolution. During training, three depthwise branches (7×7, 1×7, and 7×1) operate in parallel. At deployment, zero-padded kernel addition fuses them into a single 7×7 depthwise kernel [36]. At stride 32, the 7×7 kernel has a 224-pixel receptive field in the input image, sufficient to cover the complete UAV airframe.

3.2.4. Inference-Time Reparameterization

After training, we applied two fusion operations. The first was standard convolution and batch-normalization (Conv+BN) fusion:

$$W' = \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} W, \qquad b' = \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} (b - \mu) + \beta, \tag{6}$$

The second operation used the custom `RepAsym_DWConv.fuse()` function to collapse the three-branch asymmetric depthwise convolution into a single 7×7 kernel. These transformations removed the training-time branches before the unified throughput benchmark reported in Section 4.8.

3.3. Loss Function Design

YOLOv8s-pose decomposes the pose loss as $L_{\text{pose}} = \text{OKS_loss} + L_{\text{kobj}}$. We retained the keypoint-objectness term L_{kobj} and replaced the OKS term with a combination of Wing Loss and Bone Loss.

3.3.1. Why Remove OKS

For a predicted keypoint $\hat{p}_n$ and its ground-truth counterpart p_n, Lin et al. [19] defined OKS as

$$\text{OKS} = \frac{\sum_n \exp\left(-d_n^2 / (2s^2\kappa_n^2)\right) \cdot \mathbf{1}[v_n > 0]}{\sum_n \mathbf{1}[v_n > 0]}, \tag{7}$$

where $d_n = \|\hat{p}_n - p_n\|_2$ is the Euclidean pixel error for keypoint n; $s = \sqrt{\text{area}}$ is the scale factor, computed as the square root of the ground-truth bounding-box area in pixels; κ_n denotes a per-keypoint constant derived from the variability of human-body annotations in the Common Objects in Context (COCO) dataset [19]; $\mathbf{1}[\cdot]$ is the indicator function; and $v_n \in \{0, 1\}$ is the ground-truth visibility flag. The OKS loss is $L_{\text{OKS}} = 1 - \text{OKS}$. This

formulation raises three issues for UAV keypoints. **Sigma miscalibration.** No canonical κ values exist for UAV keypoints. Existing implementations commonly apply a uniform value of $\kappa_n = 0.054$, which does not reflect the near-zero annotation uncertainty of rendered UAV keypoints. **Curvature mismatch near zero error.** Differentiation yields

$$\frac{\partial L_{\text{OKS}}}{\partial d_n} = \frac{d_n}{N s^2 \kappa_n^2} \exp\left(-d_n^2/(2s^2\kappa_n^2)\right), \tag{8}$$

which approaches zero as $d_n \to 0$, precisely the regime relevant to strict PCK thresholds. The weak gradient near the origin is analogous to the behavior that motivated Feng et al. [21] to replace L2 loss with Wing Loss. **Independent-keypoint assumption.** OKS treats each keypoint independently and does not penalize implausible global keypoint geometry.

Wing Loss addresses the sigma miscalibration and curvature mismatch, whereas Bone Loss addresses the independent-keypoint assumption.

3.3.2. Normalized Wing Loss

For each predicted keypoint with Euclidean error $d = \|\hat{p} - p\|_2$, we normalized the error as follows:

$$d_{\text{norm}} = \frac{d}{\sqrt{\text{area} + \varepsilon_a}} \cdot 100, \qquad \varepsilon_a = 10^{-9}. \tag{9}$$

Dividing by $\sqrt{\text{area}}$ makes the error scale-invariant across targets of different sizes. The factor of 100 expresses the normalized error as a percentage, ensuring that the Wing Loss thresholds $w = 20.0$ and $\epsilon = 2.0$, originally calibrated by Feng et al. [21] on the 300 W facial-landmark dataset, operate over the same numerical range as in the original formulation. This linear rescaling contains no learned parameters. It standardizes the loss magnitude across datasets with different image resolutions and target scales without changing the gradient profile of the Wing function.

The Wing function uses $w = 20.0$, $\epsilon = 2.0$, and $C = w - w\log(1 + w/\epsilon)$:

$$\text{wing}(d_{\text{norm}}) = \begin{cases} w \cdot \log(1 + d_{\text{norm}}/\epsilon), & d_{\text{norm}} < w, \\ d_{\text{norm}} - C, & d_{\text{norm}} \geq w. \end{cases} \tag{10}$$

Differentiation gives

$$\frac{\partial \text{wing}}{\partial d_{\text{norm}}} = \begin{cases} \frac{w}{\epsilon + d_{\text{norm}}}, & d_{\text{norm}} < w, \\ 1, & d_{\text{norm}} \geq w. \end{cases} \tag{11}$$

As $d_{\text{norm}} \to 0$, the gradient approaches $w/\epsilon = 10$, which is ten times the L1 gradient magnitude, whereas the OKS gradient approaches zero. The linear regime limits the growth of the loss for outliers. Let $m_{i,n} \in \{0, 1\}$ denote the ground-truth visibility mask for keypoint n of instance i, and let $\alpha = N_{\text{instances}} / \sum_{i,n} \mathbf{1}[m_{i,n} \neq 0] + \varepsilon_a$ denote the per-batch instance-count normalization factor used to prevent large instances from dominating. The final Wing Loss is

$$L_{\text{wing}} = \mathbb{E}_{i,n}\left[\alpha \cdot \text{wing}(d_{\text{norm}}^{(i,n)}) \cdot m_{i,n}\right]. \tag{12}$$

3.3.3. Bone Loss

To prevent "rubber-band" drift, in which individual keypoints satisfy the error threshold but the overall geometry is distorted, we introduced Bone Loss over a predefined set of edges $\mathcal{E}$. These edges cover geometrically informative pairs among the eight keypoints (e.g., adjacent rotors and rotor-to-foot pairs for Matrice 200, and the head–tail and wing-tip pairs for ScanEagle). In Equation (13), $m_j^{(i)} \in \{0, 1\}$ is the ground-truth visibility mask for keypoint j of instance i (1 if visible and 0 otherwise); $\alpha = N_{\text{instances}} / \sum_{i,n} \mathbf{1}[m_{i,n} \neq 0] + \varepsilon_a$

is the per-batch instance-count normalization factor, identical to that used for Wing Loss; and $\|\hat{p}_j^{(i)} - \hat{p}_k^{(i)}\|$ and $\|p_j^{(i)} - p_k^{(i)}\|$ are the predicted and ground-truth Euclidean distances, respectively, for keypoint pair (j,k):

$$L_{\text{bone}} = \mathbb{E}_i\left[\alpha \cdot \frac{1}{|\mathcal{E}|}\sum_{(j,k)\in\mathcal{E}} \frac{\left|\,\|\hat{p}_j^{(i)} - \hat{p}_k^{(i)}\| - \|p_j^{(i)} - p_k^{(i)}\|\,\right|}{\sqrt{\text{area}_i + \varepsilon_a} + \varepsilon_a} \cdot m_j^{(i)} m_k^{(i)}\right]. \tag{13}$$

For a rigid object, pairwise inter-keypoint distances are invariants of (R,t). Bone Loss penalizes deviations from these invariants and thereby acts as a soft rigidity prior. In the long-range regime considered here ($z \approx 1000$ m and depth variation $\ll z$), perspective foreshortening is approximately uniform, and image-space bone-length ratios closely approximate the corresponding 3D ratios. The ablation results in Section 4.6 show that Bone Loss provides an additional accuracy gain under the evaluated protocol.

3.3.4. Total Loss

$$L_{\text{total}} = \lambda_{\text{box}} L_{\text{box}} + \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{dfl}} L_{\text{dfl}} + \lambda_{\text{kpt}}(L_{\text{wing}} + 0.5 \cdot L_{\text{bone}}) + \lambda_{\text{kobj}} L_{\text{kobj}}, \tag{14}$$

where $\lambda_{\text{box}} = 7.5$, $\lambda_{\text{cls}} = 0.5$, $\lambda_{\text{dfl}} = 1.5$, $\lambda_{\text{kpt}} = 12.0$, and $\lambda_{\text{kobj}} = 1.0$. These weights are identical to the YOLOv8 defaults, except that Wing $+0.5\cdot$Bone replaces the OKS term. We adopted $w = 20$ and $\epsilon = 2$ from Feng et al. [21] without modification and selected the Bone Loss coefficient of 0.5 through a coarse grid search.

3.4. Remote Sensing Data Generation Pipeline

Real long-range remote sensing imagery of aerial targets with annotated keypoints is scarce and costly to obtain. We therefore built a synthetic-to-real pipeline in Blender Cycles that produced 10,000 fully annotated samples. The pipeline addresses limited data availability, whereas the loss redesign (Section 3.3) addresses the optimization requirements of precise rigid-object keypoint localization.

3.4.1. Camera Model and Imaging Conditions

We used a pinhole camera with a 36 mm sensor width, an 11,250 mm focal length, and an image size of 1024×1024 pixels, yielding an FOV of approximately 3.2 mrad. UAV depth was sampled uniformly from 800 to 1200 m, and orientation was sampled uniformly from $SO(3)$ according to the Haar measure. Multitarget frames were generated with a probability of 50% and constrained by a 3D distance threshold. ScanEagle frames were rejected and resampled unless all eight keypoints were projected within the image.

3.4.2. Annotation by Construction with Model-Embedded Keypoints

Keypoints were annotated by construction rather than through post hoc labeling. Eight semantically meaningful keypoints were embedded as named Blender Empty objects in the source 3D mesh of each UAV class and propagated through the complete transformation chain. These auxiliary objects were created at design-defined locations in Autodesk 3ds Max and exported with the FBX mesh. During asset loading, the keypoint objects were matched to a predefined name list and converted to local-frame positions X_n^{local} through parent-relative coordinates or, when necessary, a world-space fallback computation. The auxiliary objects were then removed from the scene. For each frame, given the rigid transformation $M_{\text{inst}} \in SE(3)$ of an instance, Blender's camera model projected the keypoints into image space:

$$X_n^{\text{world}} = M_{\text{inst}} \cdot X_n^{\text{local}}, \qquad (u_n^{\text{nd}}, v_n^{\text{nd}}, z_n^{\text{cam}}) = \pi_K^{\text{Blender}}(X_n^{\text{world}}), \tag{15}$$

The pixel-space coordinates were $(u_n, v_n) = (u_n^{\mathrm{nd}} \cdot W_{\mathrm{px}},\ (1 - v_n^{\mathrm{nd}}) \cdot H_{\mathrm{px}})$. Visibility ν_n was set to one if and only if the keypoint was in front of the camera and within the image boundaries. Because this closed-form chain used the same M_{inst} as the rendering process, the resulting coordinates were accurate to single-precision floating-point error. The forward chain also served as a rejection sampler and ensured that every accepted frame contained fully annotated instances.

3.4.3. Five-Dimensional Domain Randomization

To reduce the synthetic-to-real domain gap, we randomized each frame over five dimensions derived from physical or phenomenological models of long-range optical degradation: atmosphere and illumination, sensor response, target motion, intervening occluders, and scene composition. **Atmosphere and illumination.** The Nishita sky model [70] simulated physically based atmospheric scattering by integrating the single-scattering radiative-transfer equation along the camera ray. For a ray parameterized by distance t from the camera, the radiance reaching the sensor is

$$L(\lambda, \theta) = \int_0^{\infty} \left(\beta_R(\lambda) P_R(\theta) + \beta_M(\lambda) P_M(\theta)\right) \cdot e^{-\tau(t)}\, dt, \tag{16}$$

where β_R and β_M are the Rayleigh and Mie scattering coefficients; P_R and P_M are their respective phase functions; θ is the scattering angle relative to the solar direction; and $\tau(t)$ is the optical depth along the path. Solar elevation, azimuth, and atmospheric turbidity were sampled uniformly. Turbidity ranged from 1 to 8, spanning clear to hazy conditions, to cover variations in the illumination direction and atmospheric attenuation encountered in operational remote sensing. **Sensor response.** The rendered image passed through Blender's compositor to model non-ideal properties of the imaging chain. Gaussian noise with normalized standard deviation $\sigma_n \in [0, 0.02]$ represented photon shot noise and readout noise. Defocus blur, implemented with a Gaussian kernel of radius $r_d \in [0, 3]$ pixels, represented slight misfocus and the finite aperture of long-focal-length optics. Contrast and gamma jitter were defined as $I' = \alpha \cdot I^{\gamma}$, with $\alpha \in [0.8, 1.2]$ and $\gamma \in [0.8, 1.2]$, to perturb the sensor's radiometric response and represent automatic-exposure variation and sensor nonlinearity. **Target motion.** With a probability of 0.5 per instance, the UAV was assigned a linear displacement $\mathbf{v}\Delta t$ between successive keyframes, and the built-in Cycles motion-blur model integrated radiance over the shutter interval. The displacement magnitude was sampled uniformly to represent target velocities of 5–30 m/s at the operational range. **Intervening occluders.** Bird silhouettes, cloud layers, and haze layers were inserted between the camera and the target. These elements were rendered only in the RGB pass and excluded from the NOCS, mask, and keypoint passes to preserve annotation integrity. The haze layers applied an additional extinction coefficient according to the Beer–Lambert law, $I = I_0 e^{-\beta_{\mathrm{haze}} \cdot d}$, where d is the layer thickness and β_{haze} was sampled uniformly. **Scene composition.** Each frame contained one or two UAVs at randomized 3D positions, producing natural partial occlusion of keypoints, which is an important failure mode in operational scenarios.

3.4.4. Annotation Generation and Quality Filtering

Each frame produced five aligned outputs: a postprocessed RGB image, a NOCS map [71], per-instance occlusion-aware binary masks, eight 2D keypoints with visibility flags, and the 6-DoF object pose. Tight 2D bounding boxes were recomputed from the rendered masks. A frame was accepted only if the NOCS pass contained sufficient non-background pixels, the RGB image was not severely clipped, and every instance mask

contained more than 200 visible pixels. After filtering, the 10,000 frames were divided into training and validation sets at a ratio of 80:20.

3.5. PnP-Based 6-DoF Pose Recovery

At test time, the pose $(R, t) \in SE(3)$ was recovered from the predicted 2D keypoints $\{(\hat{u}_n, \hat{v}_n)\}_{n=1}^{N}$, known 3D model points $\{X_n\}_{n=1}^{N}$, and camera intrinsic matrix K by minimizing the reprojection error:

$$(R^*, t^*) = \arg\min_{R,t} \sum_{n=1}^{N} \rho\big(\|\pi_K(RX_n + t) - (\hat{u}_n, \hat{v}_n)^\top\|\big), \tag{17}$$

We used the iterative Levenberg–Marquardt (LM) PnP solver in OpenCV (cv2.SOLVEPNP_ITERATIVE), initialized by efficient PnP (EPnP) [8]. At least four visible, nondegenerate 2D–3D correspondences are required for a well-constrained PnP estimate; we therefore excluded frames with fewer than four visible keypoints from the pose evaluation. In practice, all available visible keypoints, typically six to eight per target, were used to improve robustness. We report the rotation geodesic error $\theta = \arccos((\mathrm{tr}(R_{\mathrm{gt}}^\top R^*) - 1)/2)$ in degrees, translation error $\|t^* - t_{\mathrm{gt}}\|_2$ in meters, average distance (ADD) [72], and reprojection error. For deployment, we additionally used random sample consensus (RANSAC) through cv2.solvePnPRansac, with a reprojection threshold of $\tau = 8$ pixels. The ablation studies used the deterministic LM solver to maintain direct comparability between keypoint and pose accuracy.

4. Results

4.1. Evaluation Design Rationale

The evaluation addresses five questions. **Primary accuracy:** How accurately does GeoAdapt localize keypoints and recover camera-frame pose on the synthetic benchmark with complete ground truth? **Architectural transfer:** Does the design transfer to the independently developed DroneKey benchmark? **Component attribution:** Which modules account for the observed changes, and do DCNv2 and Wing+Bone interact? **Synthetic-to-real transfer:** Do models trained on synthetic data retain useful keypoint accuracy on manually annotated real ScanEagle and Matrice 200 imagery without fine-tuning? **Efficiency:** How do the parameter count, FLOPs, and measured throughput trade off against accuracy? The following comparisons, ablations, transfer experiments, and efficiency analyses address these questions.

4.2. Datasets

We constructed the synthetic long-range remote sensing dataset using the pipeline described in Section 3.4. It comprised 10,000 frames with a resolution of 1024 × 1024 pixels. Each frame contained one or two aerial targets (ScanEagle or DJI Matrice 200) at simulated depths of 800–1200 m and included a tight 2D bounding box, class label, eight keypoints with visibility flags, per-instance binary mask, NOCS map, and 6-DoF object pose. We used an 80:20 split, with 8000 training frames and 2000 validation frames. All metrics were computed on the complete 2000-frame validation set. We also evaluated the models on the public DroneKey dataset [14], which contains synthetically generated images of a single quadrotor with four propeller-center keypoints. For compatibility with our eight-keypoint head, we supplemented the four propeller labels with four auxiliary keypoints derived from the published 3D model. The original PCK protocol was retained for the four propeller keypoints to ensure a fair comparison. Table 1 summarizes the principal differences between the two datasets.

Table 1. Comparison of the proposed synthetic dataset and the DroneKey dataset. 3D denotes three-dimensional, FOV denotes field of view, px denotes pixels, and 6-DoF denotes six degrees of freedom.

Property	Ours	DroneKey [14]
Source	Blender Cycles, physically based path tracing	3D render composited onto 360° backgrounds
Image size	1024 × 1024 px	Various
Frame count	10,000	∼10,000
Aerial target models	2	1
Keypoint count	8	4
Background diversity	Five-dimensional domain randomization	5 real-world 360° backgrounds
Imaging conditions	Long-range optical, FOV ≈ 3.2 mrad	Standard pinhole
6-DoF pose ground truth	Yes	Yes

4.3. Implementation Details

GeoAdapt and the primary YOLO baselines were trained for 300 epochs on a single NVIDIA GeForce RTX 3090 GPU (NVIDIA Corporation, Santa Clara, CA, USA) using AdamW with an initial learning rate of 0.001, cosine annealing to 0.0001, a weight decay of 0.01, a momentum of 0.9, and a one-epoch warm-up period. The batch size was 8, and the input size was 640 × 640 pixels. Mosaic augmentation was disabled; multiscale training was enabled; and the dropout and erasing-augmentation probabilities were set to 0.1 and 0.5, respectively. Horizontal and vertical flips were not applied because they would alter the target heading. The loss weights followed Section 3.3.4, and all training runs used random seed 42. The alternative-backbone and state-space probes used the same training and validation split, 640 × 640 input resolution, eight-keypoint definition, and bounding-box-diagonal PCK evaluation. Only converged runs were retained in the main comparison. However, their training procedures and task heads remained architecture-specific; these experiments are therefore evaluation-matched representation probes rather than strictly controlled backbone swaps or complete detector-level comparisons. For GeoAdapt, `RepAsym_DWConv.fuse()` was applied before benchmarking.

The unified frames-per-second (FPS) benchmark used the same RTX 3090, a batch size of 1, FP32 precision, 10 warm-up iterations, 100 timed iterations, and CUDA synchronization immediately before and after timing. The common timing boundary was the raw model forward pass; image preprocessing, nonmaximum suppression (NMS), coordinate decoding, metric computation, and other postprocessing operations were excluded. The executed forward graph was architecture-specific. GeoAdapt and the YOLO baselines used `DetectionModel.forward()`, including the bounding-box, classification, and keypoint branches. The MobileViT, EdgeNeXt, ConvNeXt, and custom state-space model (SSM) probes used `backbone.forward_features()` followed by their keypoint-regression heads, whereas RTMO used `extract_feat()` followed by its head. Thus, the protocol standardized the measurement boundary and runtime conditions but did not require models with different task interfaces to execute identical heads. Because the native task heads differed, comparisons between the complete-framework and probe blocks describe the tested implementations rather than isolated backbone speed. We used PyTorch 2.5.1, CUDA 12.1, and Ultralytics 8.3.x.

4.4. Evaluation Metrics

We report the keypoint, detection, 6-DoF pose, and efficiency metrics. The keypoint metrics are normalized mean error (NME), defined as the Euclidean pixel error divided by D, the diagonal of the ground-truth bounding box; PCK@0.05D, defined as the fraction of keypoints with an error below $0.05 \times D$; the area under the PCK curve (AUC) over $[0, 0.5D]$; and OKS average precision (OKS-AP), AP50, AP75, and average recall (AR), following the COCO protocol [19]. The detection metrics are mean average precision (mAP) at an intersection-over-union (IoU) threshold of 0.5, mAP averaged over IoU thresholds from 0.5 to 0.95, precision, recall, F1 score, and mean IoU. The pose metrics are MAE_R, the mean geodesic rotation error in degrees; MAE_t, the mean translation error in meters; and Proj.2D, the mean reprojection error in pixels. All pose metrics used iterative LM initialized by EPnP without RANSAC to maintain direct comparability between keypoint and pose accuracy in the ablations. The efficiency metrics are the number of parameters in millions, FLOPs in billions at an input resolution of 640×640 pixels, and FPS. Differences of less than one PCK percentage point between adjacent ablation rows should be interpreted as indicative rather than definitive.

We used OKS-AP only as an evaluation metric. Section 3.3.1 critiques OKS as a training loss because its near-zero gradient can impede optimization in the subpixel regime. Nevertheless, OKS-AP remains useful for evaluation because it is the standard COCO protocol for keypoint assessment and facilitates comparison with previous work, including DroneKey [14]. We therefore report OKS-AP together with PCK and NME for completeness. The critique in Section 3.3.1 concerns the gradient properties of the loss during training, not the validity of OKS-AP for ranking predictions. PCK@0.05D and NME remain the primary accuracy metrics.

4.5. Comparison with Baselines

We compared GeoAdapt with converged YOLOv8s-pose [18], YOLO11s-pose [17], and DroneKey-Transformer [14] baselines. The two YOLO baselines used the same training procedure. On the DroneKey dataset, we evaluated only YOLOv8s-pose and GeoAdapt because the baseline PCK of 99.5% indicated that the benchmark was close to its discriminative ceiling for these models. Figure 3 summarizes the comparison setup, and Table 2 reports the primary synthetic-benchmark results.

Table 2. Main results on the proposed synthetic dataset using the 2000-frame validation set. NME denotes normalized mean error; PCK, percentage of correct keypoints; AUC, area under the PCK curve; OKS-AP, object keypoint similarity average precision; and mAP, mean average precision. The best result in each column is shown in bold.

Model	NME ↓	PCK@ 0.05D ↑	AUC ↑	OKS-AP ↑	OKS-AP50 ↑	OKS-AP75 ↑	mAP50 (Box)	Params (M) ↓
YOLOv8s-pose [18]	0.131	77.6	81.3	0.651	0.955	0.545	0.867	11.42
YOLO11s-pose [17]	0.101	77.5	85.3	0.859	0.956	0.887	0.857	9.72
DroneKey-Transformer [14]	0.212	24.9	68.4	—	—	—	—	76.02
GeoAdapt (ours)	**0.076**	**89.3**	**90.3**	**0.951**	**0.981**	**0.962**	**0.875**	**5.95**

GeoAdapt achieved the highest value for every keypoint metric while using 47.9% fewer parameters than YOLOv8s-pose. The 11.7-percentage-point improvement in PCK@0.05D was primarily associated with DCNv2 feature refinement and Wing+Bone loss (Section 4.6). Its 5.95 M-parameter footprint provided a parameter-efficient accuracy–complexity trade-off; throughput is analyzed separately in Section 4.8. DroneKey-Transformer achieved a PCK of 24.9% after transfer to our benchmark, indicating a substan-

tial difference between its original constrained setting and the long-range remote sensing conditions evaluated here.

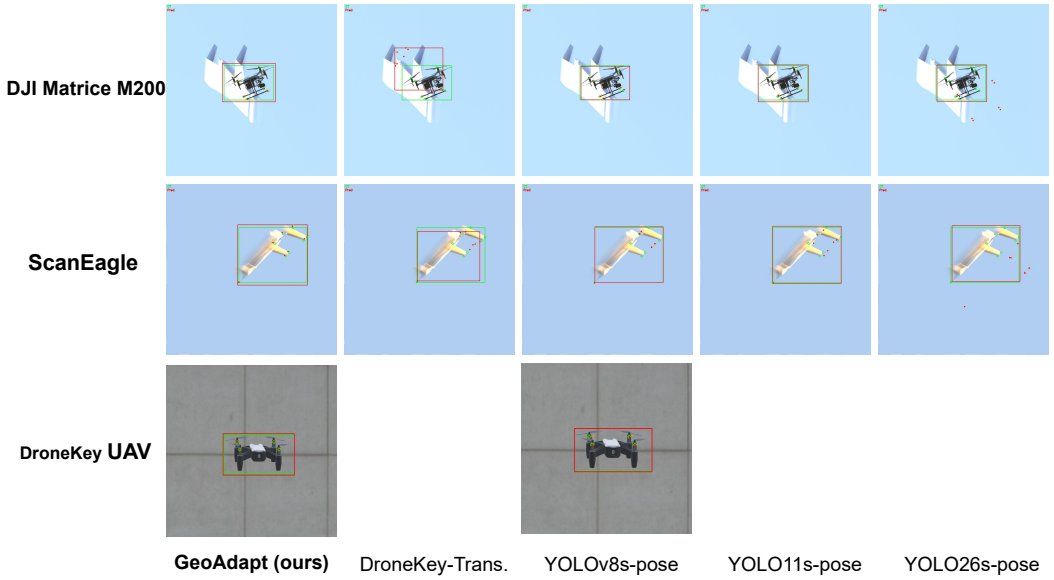


Figure 3. Experimental comparison setup, including the evaluated aerial-target platforms (Matrice 200, ScanEagle, and the DroneKey UAV) and algorithms (YOLO-family baselines, DroneKey-Transformer, and GeoAdapt). Green boxes and keypoint markers denote ground truth (GT), whereas red boxes and markers denote model predictions (Pred).

Fotiadis et al. [3] proposed a keypoint-and-structure framework for 6-DoF aircraft pose estimation from monocular RS images and thus addressed a closely related problem. A direct quantitative comparison was not feasible for three reasons: (i) Fotiadis et al. evaluated satellite imagery with nadir-viewing geometry and aircraft targets, whereas our benchmark used ground-based long-range optical imagery of UAVs; (ii) no public implementation or trained model was available; and (iii) the keypoint definitions differed. We include this discussion to position GeoAdapt within the literature on keypoint-based pose estimation for RS and to motivate the development of future benchmarks that support direct cross-method comparison. Table 3 reports the independently trained DroneKey transfer experiment.

Table 3. Architecture transferability: models independently trained from scratch on the DroneKey training set and evaluated on its validation split.

Model	PCK@ 0.05D ↑	OKS-AP ↑	Params (M) ↓
YOLOv8s-pose	99.5	0.995	11.42
DroneKey-Trans. [14]	—	0.997	76.02
GeoAdapt (ours)	100.0	0.997	5.95

GeoAdapt and YOLOv8s-pose were trained from scratch on DroneKey; the DroneKey-Transformer values are from [14]. PCK was computed on the propeller-only subset. OKS-AP uses different keypoint conventions across methods and is therefore not directly comparable.

Table 4 reports the corresponding camera-frame 6-DoF pose-estimation errors on the synthetic validation set.

Table 4. 6-DoF pose-estimation accuracy on the synthetic validation set.

Model	MAE_R (°) ↓	MAE_t (m) ↓	Proj.2D (px) ↓
YOLOv8s-pose	26.2	1417	83.0
YOLO11s-pose	18.5	134	50.3
GeoAdapt (ours)	11.6	95	34.0

Pose recovery uses EPnP + LM without RANSAC for ablation consistency; the RANSAC-protected variant (Section 3.5) is more robust to PnP degeneracy on extreme-pose frames.

4.6. Ablation Study

We performed four groups of quantitative ablations: incremental construction from the baseline to GeoAdapt, one-component removal from the full model, loss-function comparison, and a 2×2 factorial analysis of DCNv2 and Wing+Bone loss. A qualitative component-level comparison using matched challenging examples provided an additional visual illustration. Table 5 reports the incremental construction results.

Table 5. Incremental ablation study.

Configuration	PCK@0.05D ↑	OKS-AP ↑
YOLOv8s-pose (baseline)	77.6	0.651
+ DCNv2 feature refinement	84.4	0.815
+ Wing+Bone loss	87.2	0.898
+ P5 triple enhancement	87.8	0.920
+ C2f_Faster (neck)	87.6	0.923
+ LKS_C2f (P5 BU)	88.4	0.940
+ width scaling (GeoAdapt)	89.3	0.951

DCNv2 feature refinement and Wing+Bone loss were the dominant contributors and together accounted for more than 80% of the total improvement. The lightweight backbone and neck modifications provided smaller incremental gains, whereas the final width-scaling step produced a complete 5.95 M-parameter model, 47.9% smaller than YOLOv8s-pose, while further improving accuracy. Table 6 complements this construction sequence with one-component removal from the complete model.

Table 6. One-component removal ablation for the complete GeoAdapt model.

Variant	Operation	PCK@ 0.05D	Δ from Full
GeoAdapt (complete)	Complete reference configuration	89.3	—
Without DCNv2	Replace DCNv2 with standard 3×3 neck convolutions	82.7	−6.6
Without Wing Loss	Revert to OKS loss	84.3	−5.0
Without Bone Loss	Remove L_{bone}	88.0	−1.3
Without Strip Pooling	Remove Strip Pooling at P5	88.4	−0.9
Without SimAM	Remove SimAM at P5	88.8	−0.5
Without LKS_C2f	Use standard C2f at bottom-up P5	88.1	−1.2
Symmetric DCN †	Apply DCNv2 to all three head branches	89.0	−0.3

† For the symmetric DCN head, box mAP50 decreased from 0.875 to 0.808, indicating that DCNv2 placement in the cv2/cv3 branches reduced detection accuracy while leaving PCK nearly unchanged.

The one-component removal results were consistent with the incremental study: DCNv2 and Wing+Bone were the dominant contributors. SimAM added no learnable

parameters, and the symmetric DCN-head variant supported the use of feature-level rather than head-level placement. We compared five loss configurations using the complete GeoAdapt architecture (Table 7).

Table 7. Loss ablation on the full GeoAdapt architecture (300 epochs).

Loss	PCK@ 0.05D ↑	OKS-AP ↑	NME ↓
OKS only (default COCO sigmas)	84.3	0.890	0.103
OKS only (recalibrated σ)	85.8	0.906	0.098
OKS + 0.5·Bone (default σ)	85.7	0.901	0.099
Wing only	88.3	0.942	0.082
Wing + 0.5·Bone (ours)	89.3	0.951	0.076

All rows use the complete GeoAdapt architecture. Recalibrated κ_n values were computed from the positional variance of the training-set keypoints.

The loss ablation produced three findings. Recalibrating the OKS sensitivity constants yielded a modest improvement, indicating that sigma mismatch was a secondary factor. Even after recalibration, OKS underperformed Wing Loss, consistent with the weak near-zero gradient of its Gaussian decay. Adding Bone Loss improved OKS, but the combination remained less accurate than Wing Loss alone, suggesting that the curvature mismatch was the principal limitation under the evaluated protocol.

We further evaluated the interaction between DCNv2 and the keypoint loss in a 2×2 factorial experiment (Table 8).

Table 8. Factorial experiment: DCNv2 × Keypoint Loss.

Configuration	PCK@ 0.05D ↑	OKS-AP ↑	NME ↓
Without DCNv2 + OKS (supporting architecture only)	80.1	0.745	0.116
Without DCNv2 + Wing+Bone	82.7	0.852	0.097
DCNv2 + OKS	84.3	0.890	0.103
DCNv2 + Wing+Bone (GeoAdapt)	89.3	0.951	0.076

The Without DCNv2 + Wing+Bone and DCNv2 + OKS rows correspond to the Without DCNv2 and Without Wing Loss rows in Table 6, respectively. The Without DCNv2 + OKS row quantifies the supporting architecture alone.

The interaction term was +2.4 PCK percentage points, indicating a positive interaction: each component produced a larger gain when the other component was present.

In the selected examples, the baseline produced longer error segments for several challenging keypoints (Figure 4). The effects of DCNv2 and Wing+Bone varied across images and keypoints, suggesting that they addressed different error modes. The complete model generally maintained close alignment between the ground truth and predictions and avoided several large baseline errors. This visualization should be interpreted together with the quantitative ablations in Tables 5–8; it is included for interpretability rather than as independent statistical evidence.

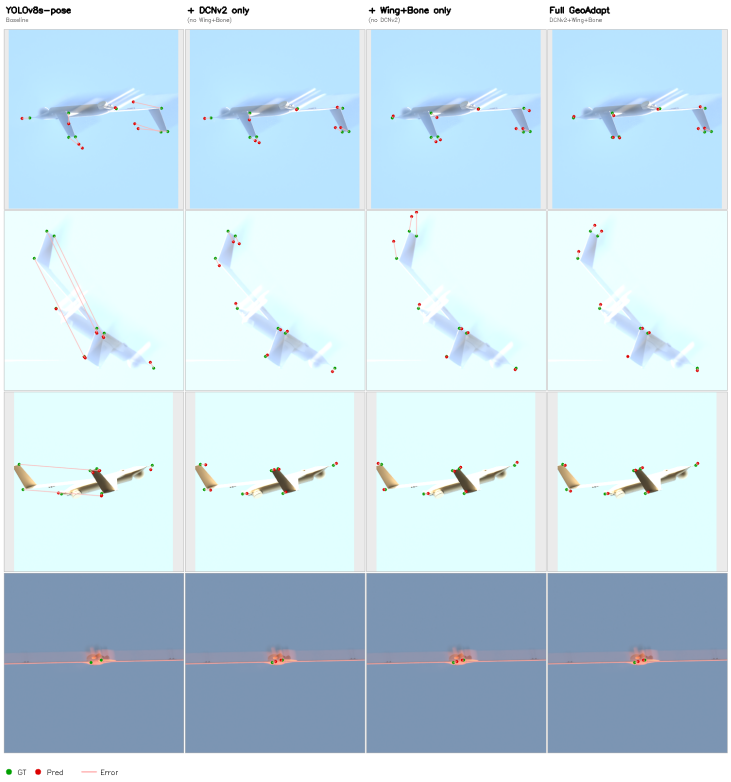


Figure 4. Qualitative component-level comparison using matched representative challenging examples. From left to right, the columns show the YOLOv8s-pose baseline, the DCNv2-only configuration, the Wing+Bone-only configuration, and the complete GeoAdapt model with DCNv2 and Wing+Bone. Each row shows the same image and crop for all four configurations. Green circles denote ground-truth keypoints, red circles denote predicted keypoints, and pink line segments represent localization errors. These examples are illustrative and do not correspond one-to-one with the quantitative configurations in Table 8.

4.7. Saturation-Gap Analysis

YOLOv8s-pose achieved 99.5% PCK on DroneKey and 77.6% on our benchmark, a difference of 21.9 percentage points. DroneKey remains a valuable public benchmark, but its near-ceiling score provided limited resolution for distinguishing the evaluated detectors. The long-range optical remote sensing setting used here, which includes heterogeneous airframes and five-dimensional domain randomization, produced greater separation among the evaluated methods. Part of this difference may be attributable to the extreme focal length of 11,250 mm rather than to benchmark design alone; evaluation at moderate focal lengths remains future work.

4.8. Computational Efficiency

Table 9 presents two complementary forms of evidence. The first block compares complete single-stage detection–pose frameworks, whereas the second reports converged alternative-architecture probes under matched evaluation conditions. The probes used the same data split, input resolution, keypoint definition, and PCK calculation but retained architecture-specific training implementations and task heads. They should therefore be interpreted as representation stress tests rather than detector-level state-of-the-art comparisons or strictly controlled backbone substitutions.

Table 9. Accuracy–efficiency comparison of converged models at an input resolution of 640 × 640 pixels. PCK is normalized by the ground-truth bounding-box diagonal D.

Model	Params (M)	FLOPs (G)	FPS	PCK@0.05D (%)
Complete single-stage detection–pose frameworks				
GeoAdapt (ours)	5.95	22.7	68.6	**89.3**
YOLOv8s-pose	11.42	29.6	102.2	77.6
YOLO11s-pose	9.72	22.5	77.6	77.5
Evaluation-matched alternative-architecture probes				
MobileViTv2-1.0	5.84	2.98	119.4	21.37
MobileViT-XXS	1.98	0.62	118.6	18.80
MobileViT-S	6.70	3.03	118.5	17.31
EdgeNeXt-Small	6.28	2.06	119.3	14.98
EdgeNeXt-XXS	1.84	0.47	149.5	14.31
ConvNeXt-Femto	6.01	1.71	225.9	10.66
Custom selective-scan SSM †	4.76	—	133.8	9.10

All FPS measurements used a batch size of 1, FP32 precision, 10 warm-up iterations, 100 timed iterations, CUDA synchronization, and a raw-forward timing boundary that excluded preprocessing, nonmaximum suppression (NMS), decoding, metric computation, and postprocessing. As detailed in Section 4.3, the executed graph remained architecture-specific; cross-block FPS comparisons are therefore descriptive. † This custom one-directional selective-scan implementation is not an official implementation of a published visual SSM architecture. Standard FLOP counters omit its selective-scan CUDA operator; therefore, the total FLOPs are not reported.

GeoAdapt used 47.9% fewer parameters than YOLOv8s-pose and improved PCK by 11.7 percentage points, but its throughput of 68.6 FPS was lower than the 102.2 FPS achieved by YOLOv8s-pose. Its 22.7 G FLOPs were also similar to the 22.5 G of YOLO11s-pose. GeoAdapt was therefore parameter-efficient but not uniformly superior across all computational measures. Under the tested implementations, the evaluation-matched alternative-architecture probes achieved PCK values of 9.10–21.37% and did not reproduce the accuracy of GeoAdapt. Because their training procedures and task heads were architecture-specific, these results neither isolated the effect of the backbone family nor established the intrinsic inferiority of any architecture family.

RTMO-t, PVTv2-B0, MambaOut-Femto, and YOLO26s-pose were also adapted and evaluated, but the corresponding runs did not yield stable, valid keypoint localization under the tested configurations. We therefore excluded them from the primary comparisons because reporting their low endpoint scores as ordinary baseline results would conflate optimization or adaptation failure with architectural capability. Supplementary Table S1 reports their configurations, endpoint metrics, and diagnostic status for transparency.

4.9. Keypoint Assessment on Real Long-Range Optical Imagery

To assess synthetic-to-real transfer, we evaluated GeoAdapt, YOLOv8s-pose, and YOLO11s-pose on two independently collected real long-range optical datasets. All three models were trained under the same protocol on our synthetic benchmark (Section 3.4). Before zero-shot inference, each image was cropped to 640 × 640 pixels around the tracking center.

The ScanEagle dataset comprised 510 visible-light frames of a fixed-wing ScanEagle UAV (Insitu Inc., Bingen, WA, USA) under diverse atmospheric conditions. The Matrice 200 dataset comprised 553 frames of a DJI Matrice 200 quadrotor (SZ DJI Technology Co., Ltd., Shenzhen, China) in five scenarios. Neither dataset was originally collected with keypoint-level ground truth. For quantitative comparison, we manually annotated every frame in both datasets. PCK@0.05D was computed using Hungarian matching between the annotations and predictions [14]. Figure 5 shows representative inference results for all models.

Table 10 reports PCK@0.05D for manually annotated keypoints in one ScanEagle scenario and five Matrice 200 scenarios. GeoAdapt outperformed both baselines in all six scenarios, with improvements of 4.3–8.1 percentage points. For every model, PCK was lower on ScanEagle than on Matrice 200, indicating that the elongated fixed-wing geometry presented a greater synthetic-to-real transfer challenge. The higher PCK of GeoAdapt was

consistent with the intended effects of DCNv2-augmented features and Wing + Bone loss: stronger gradients in the small-error regime and greater adaptation to the geometry of real target appearances than the standard OKS-based keypoint heads.

Table 10. Zero-shot PCK@0.05D on manually annotated real long-range imagery. The best result for each dataset is shown in bold.

Dataset	GeoAdapt (Ours)	YOLOv8s-pose	YOLO11s-pose
ScanEagle	**41.2%**	36.9%	33.1%
Matrice 200 (sky)	**52.4%**	46.1%	43.5%
Matrice 200 (city)	**54.8%**	49.3%	46.0%
Matrice 200 (water)	**48.5%**	42.7%	40.2%
Matrice 200 (mountain)	**51.7%**	45.5%	42.8%
Matrice 200 (cloud)	**53.1%**	47.8%	44.6%

Figure 5 presents representative predictions. For Matrice 200, GeoAdapt produced compact keypoint clusters around the propeller centers and geometrically consistent configurations. ScanEagle was more challenging because of its elongated fixed-wing geometry; nevertheless, GeoAdapt produced structurally plausible keypoint configurations and achieved the highest PCK among the compared models (Table 10). Its 4.3-percentage-point improvement over YOLOv8s-pose on ScanEagle and its consistent lead across all five Matrice 200 subsets suggest that the observed benefit was not limited to a single target geometry. These results are consistent with the hypothesis that deformable feature refinement can improve synthetic-to-real keypoint transfer relative to standard convolutional keypoint heads. Quantitative 6-DoF pose evaluation on real sensor data still requires instrumented pose ground truth and remains future work (Section 5.7).



Figure 5. Zero-shot keypoint detection on real long-range optical images. Green denotes the predicted bounding box; cyan and yellow denote the predicted eight-keypoint configuration; and red squares and lines denote the manually annotated ground truth. Rows 1–2 show the Matrice 200 quadrotor, and Rows 3–4 show the fixed-wing ScanEagle.

5. Discussion

Like satellite and airborne systems, ground-based optical systems measure distant targets without contact. However, their near-horizontal viewing geometry and operational

constraints produce a distinct image distribution. We therefore evaluated GeoAdapt specifically on long-range ground-based imagery and do not present it as a platform-independent solution. Its 5.95 M-parameter architecture and synthetic-data pipeline address the need for compact models and the limited availability of annotated data. The manually annotated real-image assessment (Section 4.9) provides initial evidence of synthetic-to-real transfer: without fine-tuning, GeoAdapt achieved higher PCK than the evaluated YOLOv8s-pose and YOLO11s-pose baselines in all six scenarios. This evidence is limited to 2D keypoints and does not establish 6-DoF pose accuracy on real images.

5.1. Compact Design and Deployment Considerations

GeoAdapt contains 5.95 M parameters, 47.9% fewer than YOLOv8s-pose, and improved PCK@0.05D by 11.7 percentage points on the synthetic benchmark. This result supports targeted capacity allocation: a small parameter budget is assigned to geometrically adaptive refinement, whereas lightweight operations are used elsewhere. However, parameter count alone does not determine deployment speed. The irregular memory-access pattern of DCNv2 reduced throughput relative to YOLOv8s-pose despite the smaller model. GeoAdapt is therefore parameter-efficient, but deployment on a specific edge processor requires device-level measurements of latency, memory use, and energy consumption.

5.2. Evidence Across Alternative Architecture Paradigms

We interpret the expanded evidence at three levels. First, we compared GeoAdapt with the complete YOLOv8s-pose and YOLO11s-pose frameworks using the same bounding-box-normalized PCK metric. Second, converged MobileViT, EdgeNeXt, ConvNeXt, and custom selective-scan SSM probes evaluated alternative representations using a common data split, input resolution, keypoint definition, and PCK calculation. Because their training procedures and task heads remained architecture-specific, they documented the behavior of the tested implementations rather than providing strictly controlled comparisons among architecture families. Third, we disclose the nonconvergent RTMO-t, PVTv2-B0, MambaOut-Femto, and YOLO26s-pose adaptations only as diagnostics in Supplementary Table S1. This separation avoids interpreting optimization failure as evidence against one-stage, Transformer, gated-convolution, or newer YOLO paradigms. Accordingly, the results support the suitability of GeoAdapt for the present rigid-UAV keypoint task without claiming exhaustive superiority over all recent pose-estimation methods.

5.3. Why GeoAdapt Works: The Interplay of Architecture and Loss Design

GeoAdapt achieved PCK@0.05D of 89.3% on the synthetic ground-based long-range optical benchmark with 5.95 M parameters. The ablation results associate this performance with task-specific capacity allocation and a loss curvature matched to the precision of the annotations.

Feature-level DCNv2 provides geometric flexibility, whereas Wing+Bone provides a gradient signal tailored to precise keypoint localization. Placing DCNv2 between the neck and head allows all downstream branches to access geometrically refined features, which is relevant when keypoint appearance varies with pose under a fixed ground-based viewing geometry. The symmetric DCN-head variant reduced box mAP50 from 0.875 to 0.808 (Table 6), indicating that feature-level placement better preserved features suitable for detection while providing geometric adaptation to all task branches. The complementarity of DCNv2 and Wing+Bone may arise from their coupling during optimization. The learnable DCNv2 offsets Δp_k are optimized through gradients from the keypoint loss. If the loss provides weak gradients in the small-error regime, as OKS does, the offset-prediction network receives limited information for refining the sampling positions. Wing Loss addresses this limitation with a gradient magnitude of $w/\epsilon = 10$ at the origin, which can

guide the offsets toward local support for each keypoint. The 2 × 2 factorial experiment (Table 8) quantified this interaction: DCNv2 improved PCK by 4.2 percentage points under OKS and by 6.6 percentage points under Wing+Bone, whereas Wing+Bone improved PCK by 2.6 percentage points without DCNv2 and by 5.0 percentage points with DCNv2. The interaction term of 2.4 percentage points indicated that the two components produced greater gains when used together. The annotation-by-construction pipeline produced deterministic ground truth at renderer precision, which may favor Wing+Bone over OKS more than manually annotated data would. However, the analytical concerns regarding OKS—sensitivity-constant miscalibration for rigid-object keypoints, weak gradient curvature in the subpixel regime, and the independent-keypoint assumption—do not depend solely on annotation uncertainty. Controlled experiments with synthetic annotation noise would provide a more direct test of this interpretation.

Lightweight co-design supported the gains but was not their primary source. Partial convolution and reparameterized large-kernel convolution reduced computational cost, whereas their individual contributions to accuracy were modest. Their role was to maintain a complete model with fewer than 6 M parameters while preserving the gains from feature refinement and loss design. Width scaling reduced the parameter count and coincided with a 0.9-percentage-point PCK improvement in Table 5. This change may reflect regularization or optimization effects on the two-class synthetic dataset, but the present experiment did not isolate the underlying mechanism.

5.4. Synthetic Data for Remote Sensing

The synthetic-data pipeline addresses the difficulty of obtaining annotated imagery at operational ranges. Long-range optical remote sensing requires domain expertise for annotation, and manual keypoint annotation at 800–1200 m is costly and time-consuming. The model-embedded keypoint mechanism generated annotations by construction: each frame contained eight keypoint coordinates defined at renderer precision. The five-dimensional domain-randomization strategy varied atmosphere and illumination, sensor response, target motion, intervening occluders, and scene composition. Each dimension represented a nuisance factor encountered in real imaging. The Nishita sky model [70] provided physically based atmospheric scattering relevant to remote sensing. The principal limitation of the synthetic pipeline is the absence of long-range atmospheric effects such as scintillation, anisoplanatic blur, and chromatic aberration, whose spatial and temporal correlations are not captured by the current randomization. Integrating a differentiable turbulence simulator into the training pipeline is therefore an important direction for future work. The annotation-by-construction principle can be extended to any rigid object with a textured 3D model, including aircraft, ships, ground vehicles, and infrastructure.

5.5. Atmospheric Degradation and Deformable Convolution

Long horizontal optical paths can introduce attenuation, blur, scintillation, and spatially varying distortion. Fixed-grid convolution cannot change its sampling locations in response to local image content, whereas DCNv2 predicts position-dependent offsets $\Delta p_k(p)$ and modulation weights $m_k(p)$ (Equation (5)). This difference provides a plausible image-domain explanation for the improved localization observed in the ablations: deformable sampling can follow local target structure and downweight less informative sampling locations.

This interpretation should not be confused with physical turbulence compensation. DCNv2 was not derived from atmospheric-propagation theory, and the training renderer included single-scattering attenuation and image-domain blur rather than phase-screen turbulence. The learned offsets therefore cannot be interpreted as estimates or inverses

of a physical point-spread function. Testing such a hypothesis would require a controlled turbulence simulator and comparisons with explicit mitigation methods, including phase-screen restoration, adaptive optics, or multiframe deconvolution.

5.6. Benchmark Implications

The difference in YOLOv8s-pose performance—99.5% PCK on DroneKey and 77.6% on our dataset—indicates that the two benchmarks differ substantially in difficulty. A near-ceiling baseline provides limited resolution for distinguishing incremental improvements [73], whereas the present benchmark retains headroom for the evaluated models. The DroneKey authors described their data as "small-scale, limited to single models, and collected under constrained environments" [15]. Nevertheless, the 21.9-percentage-point difference cannot be attributed to benchmark design alone because the target geometry, rendering process, training configuration, and extreme-focal-length protocol also differ. Evaluation at moderate focal lengths and on additional public datasets is required before broader conclusions can be drawn about benchmark representativeness.

5.7. Limitations

We acknowledge the following limitations:

- **Predominantly synthetic pose validation.** The primary benchmark results, including NME, OKS-AP, and 6-DoF pose errors, were computed on the synthetic validation set. We additionally report real-image PCK@0.05D for manually annotated ScanEagle and Matrice 200 frames (Section 4.9, Table 10), but quantitative 6-DoF pose accuracy on real sensor data has not yet been established.
- **Limited class diversity.** The evaluation covered two UAV classes. Other rotorcraft configurations and non-UAV targets, such as ships, aircraft, and ground vehicles, were not evaluated.
- **Atmospheric turbulence not modeled during training.** Scintillation, anisoplanatic blur, and beam wander were absent from the training distribution. The Nishita sky model represented single-scattering attenuation but not turbulence-induced phase distortion.

Expected failure modes include severe occlusion, under which DCNv2 offsets may shift toward background edges; extreme viewing angles (more than 80° from broadside), at which the apparent wingspan decreases to only a few pixels; systematic motion blur affecting multiple keypoints; very small targets occupying less than 5% of the image width, for which the P5 feature-map resolution may be insufficient; and previously unseen target geometries that require retraining with model-specific synthetic data.

5.8. Future Work

Future work will extend the validation to real remote sensing imagery from ground-based, aerial, and satellite platforms, with particular emphasis on instrumented 6-DoF pose ground truth. We will also extend the synthesis pipeline to additional target classes, integrate EPro-PnP [44] for differentiable 6-DoF supervision, and examine temporal modeling for trajectory-aware pose smoothing. Further priorities include incorporating atmospheric-turbulence models into the synthetic-data pipeline, developing a joint detection–pose loss using ADD as a differentiable proxy, conducting controlled experiments with synthetic annotation noise, and quantitatively analyzing the failure modes identified in Section 5.7.

6. Conclusions

This paper presented GeoAdapt, a lightweight deformable keypoint detector for camera-frame 6-DoF pose estimation of aerial targets in ground-based optical remote sensing. On the synthetic long-range benchmark, GeoAdapt achieved PCK@0.05D of

89.3% with 5.95 M parameters, 47.9% fewer than YOLOv8s-pose. On the DroneKey benchmark, GeoAdapt achieved 100.0% PCK. On manually annotated real ScanEagle and Matrice 200 imagery, it achieved the highest zero-shot PCK among the compared YOLO-Pose models in all six scenarios. Evaluation-matched convolutional neural network (CNN), CNN–Transformer, and selective-scan SSM probes did not reproduce this accuracy under their tested architecture-specific implementations. This observation was not used to infer the superiority or inferiority of any architecture family.

The ablation results associated these gains mainly with three design choices. First, feature-level DCNv2 refinement between the FPN neck and detection head provided learnable geometric adaptation with limited parameter overhead. Second, an OKS-free keypoint loss combining Wing Loss and Bone Loss addressed sensitivity-constant miscalibration, weak near-zero gradients, and the independent-keypoint assumption of OKS in the present task. Third, lightweight backbone–neck co-design preserved these accuracy gains within a compact parameter footprint. The factorial experiment indicated a positive interaction between DCNv2 and Wing+Bone loss.

The main limitations are the predominantly synthetic pose validation, despite the real-image PCK evaluation reported in Section 4.9; the two-class evaluation scope; and the absence of atmospheric turbulence during training. Future work will prioritize 6-DoF pose evaluation with real-sensor ground truth, expanded edge-platform benchmarking, and turbulence-aware synthetic-data generation. The design principles examined here—feature-level deformable refinement, loss functions matched to remote sensing precision requirements, and capacity allocation informed by gradient analysis—may guide the development of lightweight deep-learning methods for optical remote sensing.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/rs18152545/s1: Table S1: Diagnostic results for nonconvergent adaptations at an input resolution of 640 × 640 pixels. PCK is normalized by the ground-truth bounding-box diagonal D.

Author Contributions: Conceptualization, T.Y. and W.X.; methodology, T.Y., W.X., and W.Z.; software, T.Y. and W.X.; validation, T.Y., W.X., Y.X., and N.L.; formal analysis, T.Y., N.L., and F.W.; investigation, T.Y., W.X., F.W., and Y.L.; resources, W.Z. and Y.X.; data curation, T.Y. and W.X.; writing—original draft preparation, T.Y. and W.X.; writing—review and editing, Y.X., W.Z., F.W., and Y.L.; visualization, T.Y. and W.X.; supervision, Y.X. and W.Z.; project administration, W.Z.; funding acquisition, W.Z. and Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Central Guidance for Regional Science and Technology Development Program, grant number 202407a12020013.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request, subject to restrictions imposed by the supporting research program. The Supplementary Material provides the diagnostic results described in the manuscript. The Blender-based data-generation pipeline and trained model weights are not publicly archived at the time of revision and are intended for release after article acceptance.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, J.; Bai, Z.; Zhang, X.; Qiu, Y. A lightweight remote sensing aircraft object detection network. *Remote Sens.* **2024**, *16*, 857. [CrossRef]
2. Shen, X.; Wang, H.; Wei, B.; Cao, J. Real-time scene classification of unmanned aerial vehicles remote sensing image based on Modified GhostNet. *PLoS ONE* **2023**, *18*, e0286873. [CrossRef] [PubMed]
3. Fotiadis, E.P.; Vretos, N.; Nikolaidis, N.; Pitas, I. Estimating 6D aircraft pose from keypoints and structures. *Remote Sens.* **2021**, *13*, 663. [CrossRef]

4. Rad, M.; Lepetit, V. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
5. Tekin, B.; Sinha, S.N.; Fua, P. Real-time seamless single shot 6D object pose prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
6. Hu, Y.; Hugonot, J.; Fua, P.; Salzmann, M. Segmentation-driven 6D object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
7. Quan, L.; Lan, Z. Linear N-point camera pose determination. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 774–780. [CrossRef]
8. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An accurate O(n) solution to the PnP problem. *Int. J. Comput. Vis.* **2009**, *81*, 155–166.
9. Cheng, G.; Han, J. A survey on object detection in optical remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28. [CrossRef]
10. Tang, Y.; Qin, R.; Li, R. Remote sensing object detection in the deep learning era—A review. *Remote Sens.* **2024**, *16*, 327. [CrossRef]
11. Li, Z.; Wang, Y.; Zhang, N.; Zhang, Y.; Zhao, Z.; Xu, D.; Ben, S.; Gao, Y. Deep learning-based object detection techniques for remote sensing images: A survey. *Remote Sens.* **2022**, *14*, 2385. [CrossRef]
12. Liu, S.; Cao, L.; Li, Y. Lightweight pedestrian detection network for UAV remote sensing images based on strideless pooling. *Remote Sens.* **2024**, *16*, 2331. [CrossRef]
13. Gu, L.; Fang, Q.; Wang, Z.; Popov, E.; Dong, G. Learning lightweight and superior detectors with feature distillation for onboard remote sensing object detection. *Remote Sens.* **2023**, *15*, 370. [CrossRef]
14. Hwang, S.B.; Cho, Y.J. DroneKey: Drone 3D pose estimation in image sequences using gated key-representation and pose-adaptive learning. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hangzhou, China, 19–25 October 2025.
15. Hwang, S.B.; Cho, Y.J. DroneKey++: A size prior-free method and new benchmark for drone 3D pose estimation from sequential images. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Vienna, Austria, 1–5 June 2026.
16. Maji, D.; Nagori, S.; Mathew, M.; Poddar, D. YOLO-Pose: Enhancing YOLO for multi-person pose estimation using object keypoint similarity loss. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA, 19–20 June 2022.
17. Khanam, R.; Hussain, M. YOLOv11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:2410.17725.
18. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 29 July 2026).
19. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
20. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable ConvNets v2: More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
21. Feng, Z.H.; Kittler, J.; Awais, M.; Huber, P.; Wu, X.J. Wing loss for robust facial landmark localisation with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
22. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016.
23. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 172–186.
24. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017.
25. Tremblay, J.; Prakash, A.; Acuna, D.; Brophy, M.; Jampani, V.; Anil, C.; To, T.; Cameracci, E.; Boochoon, S.; Birchfield, S. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018.
26. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023.
27. Yu, R.; Zhang, Y.; Liu, S. CIMB-YOLOv8: A lightweight remote sensing object detection model. *Electronics* **2025**, *14*, 2657. [CrossRef]
28. Liu, S.; Shao, F.; Chu, W.; Dai, J.; Zhang, H. An improved YOLOv8-based lightweight attention mechanism for cross-scale feature fusion. *Remote Sens.* **2025**, *17*, 1044. [CrossRef]

29. Guo, X.; Hou, B.; Ren, B.; Ren, Z.; Jiao, L. Network pruning for remote sensing images classification based on interpretable CNNs. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5402615. [CrossRef]
30. Yang, X.; Zhang, S.; Yang, W. Two-way assistant: A knowledge distillation object detection method for remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5614015. [CrossRef]
31. Liu, D.; Zhang, J.; Liang, X.; Qi, Y.; Song, Y.; Xi, Y.; Jin, J. RS-LLIC: A lightweight learned image compression model with knowledge distillation for onboard remote sensing. *IEEE Trans. Geosci. Remote Sens.* **2026**, *64*, 5610213. [CrossRef]
32. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
33. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
34. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
35. Chen, J.; Kao, S.; He, H.; Zhuo, W.; Wen, S.; Lee, C.H.; Chan, S.H.G. Run, don't walk: Chasing higher FLOPs efficiency for fast neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023.
36. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style ConvNets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
37. Ding, X.; Zhang, X.; Han, J.; Ding, G. Diverse branch block: Building a convolution as an inception-like unit. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
38. Jiang, J.; Wang, W.; Lin, X.; Miao, P.; Gao, L.; Xu, M. Phase Congruency-Guided Cross-Scale Contextual Fusion Network for Salient Object Detection in Optical Remote Sensing Images. *Remote Sens.* **2026**, *18*, 1847. [CrossRef]
39. Xu, M.; Wang, S.; Hu, Y.; Tang, H.; Cong, R.; Nie, L. Cross-Model Nested Fusion Network for Salient Object Detection in Optical Remote Sensing Images. *IEEE Trans. Cybern.* **2025**, *55*, 5332–5345. [CrossRef] [PubMed]
40. Gao, L.; Fu, P.; Xu, M.; Wang, T.; Liu, B. UMINet: A Unified Multi-Modality Interaction Network for RGB-D and RGB-T Salient Object Detection. *Vis. Comput.* **2024**, *40*, 1565–1582. [CrossRef]
41. Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; Bao, H. PVNet: Pixel-wise voting network for 6DoF pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
42. Song, C.; Song, J.; Huang, Q. HybridPose: 6D object pose estimation under hybrid representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
43. Wang, G.; Manhardt, F.; Tombari, F.; Ji, X. GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
44. Chen, H.; Wang, P.; Wang, F.; Tian, W.; Xiong, L.; Li, H. EPro-PnP: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022.
45. Çetiner, K.A.; Ekenel, H.K. Yolo-Key-6D: Single stage monocular 6D pose estimation with keypoint enhancements. *arXiv* **2026**, arXiv:2603.03879.
46. Qiantong, W.; Xiurui, G.; Li, P.; Zhang, L.; Niu, B.; Wang, F.; Guangyao, Z.; Hu, Y. Keypoint detection-based aircraft fine-grained recognition for high-resolution optical remote sensing images. *Remote Sens.* **2025**, *17*, 3577. [CrossRef]
47. Lu, P.; Jiang, T.; Li, Y.; Li, X.; Chen, K.; Yang, W. RTMO: Towards High-Performance One-Stage Real-Time Multi-Person Pose Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 1491–1500.
48. Yang, J.; Zeng, A.; Liu, S.; Li, F.; Zhang, R.; Zhang, L. Explicit Box Detection Unifies End-to-End Multi-Person Pose Estimation. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
49. Zhang, J.; Hou, J.; He, Q.; Yuan, Z.; Xue, H. MambaPose: A Human Pose Estimation Based on Gated Feedforward Network and Mamba. *Sensors* **2024**, *24*, 8158. [CrossRef] [PubMed]
50. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Lei, L.; Zou, H. Multi-scale object detection in remote sensing imagery with deformable convolution networks. *Remote Sens.* **2018**, *10*, 1483.
51. Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Lei, L.; Zou, H. Fusing DCN and BBAV for remote sensing image object detection. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *118*, 103293. [CrossRef]
52. Li, Q.B.; Fan, Z.C.; Zhao, X.Y. An advanced adaptive detector for oriented objects in remote sensing imagery (AdaR-YOLOv8 with C2f-DCN). *Sci. Rep.* **2025**, *15*, 25902. [CrossRef] [PubMed]
53. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.

54. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
55. Yang, L.; Zhang, R.Y.; Li, L.; Xie, X. SimAM: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the International Conference on Machine Learning (ICML), Virtual Event, 18–24 July 2021.
56. Ding, X.; Zhang, X.; Han, J.; Ding, G. Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022.
57. Ding, X.; Chen, Y.; Ma, N.; Han, J.; Ding, G. UniRepLKNet: A universal perception large-kernel ConvNet for audio, video, point cloud, time-series and image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024.
58. Liu, S.; Chen, T.; Chen, X.; Chen, X.; Xiao, Q.; Wu, B.; Pechenizkiy, M.; Mocanu, D.; Wang, Z. More ConvNets in the 2020s: Scaling up kernels beyond 51×51 using sparsity. *arXiv* **2022**, arXiv:2207.03620.
59. Bulat, A.; Tzimiropoulos, G. How far are we from solving the 2D & 3D face alignment problem? (And a dataset of 230,000 3D facial landmarks). In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
60. Zhang, F.; Zhu, X.; Dai, H.; Ye, M.; Zhu, C. Distribution-aware coordinate representation for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
61. Denninger, M.; Sundermeyer, M.; Winkelbauer, D.; Zidan, Y.; Olefir, D.; Elbadrawy, M.; Lodhi, A.; Katam, H. BlenderProc. *arXiv* **2019**, arXiv:1911.01911.
62. Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*; Springer: Cham, Switzerland, 2018.
63. Liu, W.; Liu, J.; Luo, B. Can synthetic data improve object detection results for remote sensing images? *arXiv* **2020**, arXiv:2006.05015.
64. Tang, D.; Cao, X.; Wu, X.; Li, J.; Yao, J.; Bai, X.; Jiang, D.; Li, Y.; Meng, D. AeroGen: Enhancing remote sensing object detection with diffusion-driven data generation. *arXiv* **2024**, arXiv:2411.15497.
65. Tang, D.; Wang, H.; Xin, Y.; Qiao, H.; Jiang, D.; Li, Y.; Yu, Z.; Cao, X. TerraGen: A unified multi-task layout generation framework for remote sensing data augmentation. *arXiv* **2025**, arXiv:2510.21391.
66. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020.
67. Hou, Q.; Zhang, L.; Cheng, M.M.; Feng, J. Strip pooling: Rethinking spatial pooling for scene parsing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
68. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
69. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
70. Nishita, T.; Sirai, T.; Tadamura, K.; Nakamae, E. Display of the Earth taking into account atmospheric scattering. In Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93), Anaheim, CA, USA, 2–6 August 1993.
71. Wang, H.; Sridhar, S.; Huang, J.; Valentin, J.; Song, S.; Guibas, L.J. Normalized Object Coordinate Space for category-level 6D object pose and size estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
72. Hinterstoisser, S.; Lepetit, V.; Ilic, S.; Holzer, S.; Bradski, G.; Konolige, K.; Navab, N. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In Proceedings of the Asian Conference on Computer Vision (ACCV), Daejeon, Republic of Korea, 5–9 November 2012.
73. Recht, B.; Roelofs, R.; Schmidt, L.; Shankar, V. Do ImageNet classifiers generalize to ImageNet? In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.