G-MAD: A Game-Based Data Generation Framework for Multi-View RGB-T Aerial Object Detection

Yechan Kim†
JongHyun Park‡
†LIG Defense&Aerospace
Republic of Korea
‡GIST
Republic of Korea

Dongho Yoon‡
Namhoon Jung†
Moongu Jeon‡
{yechan.kim26,namhoon.jung}@ligdna.com
{citizen135,gidong76}@gm.gist.ac.kr,mgjeon@gist.ac.kr

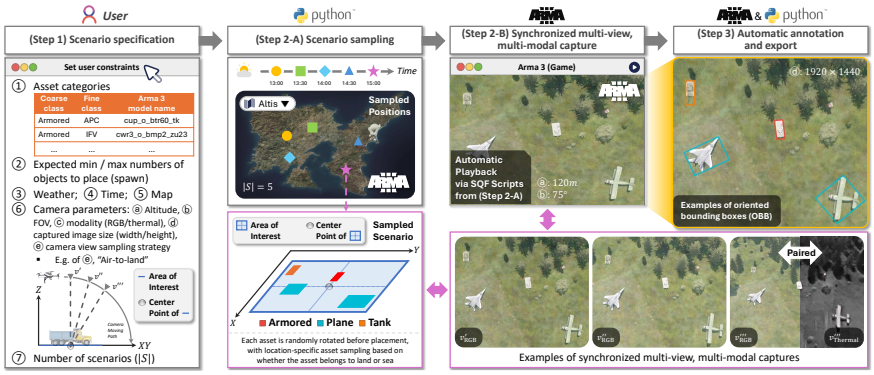


Figure 1: The overall architecture of G-MAD.

Abstract

This work introduces G-MAD, an open-source framework that uses Arma3 to generate synchronized multi-view RGB-T data for aerial object detection. G-MAD addresses key limitations of real-world aerial dataset construction, including limited viewpoint control, imperfect RGB-T alignment and high annotation cost. The framework supports structured scenario specification, controllable multi-view camera placement, simultaneous visible/thermal capture, and automatic bounding box annotation using engine-level geometric metadata. These capabilities enable controlled studies of viewpoint variation, multi-modal fusion, and synthetic-to-real transfer in aerial object detection. Besides, using G-MAD, we construct and release AMOD, a new large-scale multi-view aerial RGB-T object detection benchmark. The source code and the dataset are available at https://unique-chan.github.io/G-MAD-Project.

CCS Concepts

• **Computing methodologies** → **Object detection**.

Keywords

Synthetic Data Generation, Military Target, Object Detection, RGB, Thermal, Open-Source Toolkit, Benchmark, Arma3

ACM Reference Format:
Yechan Kim, JongHyun Park, Dongho Yoon, Namhoon Jung, and Moongu Jeon. 2026. G-MAD: A Game-Based Data Generation Framework for Multi-View RGB-T Aerial Object Detection. In *Proceedings of 34th ACM International Conference on Multimedia, November 10–November 14, 2026, Rio de Janeiro, Brazil (MM '26).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Aerial object detection has become an important problem in remote sensing, surveillance, disaster response, and autonomous driving. Compared with ground-level perception, aerial imagery involves large viewpoint changes, scale variation, and complex backgrounds, making robust object detection challenging. These challenges become more pronounced in RGB-thermal (RGB-T) settings, where models must exploit complementary visible and thermal cues while maintaining accurate cross-modal correspondence. However, constructing real-world aerial RGB-T datasets is costly and difficult because it requires synchronized sensor acquisition, repeated flights under controlled conditions, and labor-intensive annotation. As a result, existing datasets often provide limited control over viewpoints, environmental factors, and modality alignment.

Synthetic data generation provides a practical way to address these limitations [18, 52]. While recent approaches span a broad spectrum—from classical 3D graphics and simulation pipelines to neural rendering and generative models—not all methods provide the structured supervision and controllability required for perception tasks. In particular, simulation-based pipelines remain well

MM '26, Rio de Janeiro, Brazil

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

suited for such settings, as they provide direct access to scene geometry, annotations, and sensor configurations [20].

In this context, commercial games have emerged as practical platforms for synthetic data generation [53]. While not originally designed for this purpose, they are widely adopted due to their rich pre-built assets, which eliminate the substantial cost of asset creation required in general-purpose simulation platforms such as Unity [6] and Unreal [7] Engine [9, 12, 68]. Among these, GTA V-based pipelines established a prominent line of work demonstrating large-scale automatic annotation for visual perception. Prior studies leveraged GTA V [5] and similar environments to construct datasets spanning multiple tasks like segmentation [53], crowd counting [61], anomaly detection [41], showing that game-based pipelines can support scalable and diverse supervision. More recent efforts further extend this paradigm to specialized tasks such as scene flow estimation [29] and HDR reconstruction [11].

However, existing game-based data generation pipelines remain insufficient for RGB-T aerial object detection. For instance, GTA V has mainly been used for visible-domain data generation, as it does not natively provide thermal observations. Arma3 is a large-scale tactical simulation game that provides open-world outdoor environments, controllable entities, and scriptable in-game cameras. Besides, Arma3 enables RGB-T paired data generation. Nevertheless, Arma3 has not been fully explored as a structured data-generation platform for multi-view RGB-T aerial object detection.

Hence, we present ***G-MAD***, an open-source framework that transforms Arma3 [3] into a structured data-generation pipeline for multi-view RGB-T aerial object detection. G-MAD[1] supports scenario specification, synchronized RGB-T multi-view capture, and automatic object annotation using engine-level geometric information. Using G-MAD, we further construct ***AMOD***, a new multi-view RGB-T aerial detection benchmark for studying viewpoint robustness, RGB-T learning, and synthetic-to-real transfer.

2 Architecture of G-MAD

G-MAD is designed as an end-to-end data generation pipeline that transforms a compact set of user-specified parameters into structured multi-modal (RGB-T) data for aerial detection. As illustrated in Fig. 1, the proposed framework decomposes the data generation process into three stages: (1) scenario specification, (2) scenario sampling and synchronized multi-view, multi-modal capture, and (3) automatic annotation and export. The overall pipeline is implemented in SQF, the native scripting language of Arma3, allowing direct procedural control over scene composition, environmental configuration, in-game sensor control, and data export within the simulator. In this work, we use a term ***scenario*** to denote a parameterized specification of data generation conditions, including object arrangement and environmental factors such as time, weather, and background. A scenario defines a fixed underlying configuration shared across multiple observations. Meanwhile, we use a term ***scene*** to denote a concrete capture instance generated from a scenario, corresponding to a specific viewpoint and sensor configuration (e.g., camera pose and sensing modality like RGB). Under this formulation, a single scenario produces multiple scenes by varying viewpoints and sensing modalities while keeping the object arrangement and environment unchanged; see Fig. 1.

[1]It stands for "a data Generator for Multi-view Aerial object Detection using Arma3."

Table 1: Comparison of Arma3-based open source synthetic data generation frameworks in terms of aerial detection.

	[1]	[2]	G-MAD (Proposed)
Scene configuration and sampling			
Multi-map support	×	×	○
Max objects per image	≤ 1	≤ 1	**≥ 1**
Region of interest selection	Manual	Manual	**Automatic (Constraint-aware)**
Camera viewpoint sampling	Predefined orbit	Unconstrained random	**User-defined sampling model**
Occlusion check between camera and object	×	○	○
Annotation and output			
Annotation method	Post-hoc image differencing	Post-hoc image differencing	**Engine-native ground truth**
Bounding box support	×	HBB	**HBB, OBB**
Resolution (Image size)	Fixed	Fixed	**Configurable**
Resolution scaling (supersampling)	×	×	○
Execution and automation			
MATLAB dependency	○	× (Only Python)	**× (Only Python)**
Execution scheme	GUI-dependent control	GUI-dependent control	**GUI-independent launch**
Temporary file cleanup	Manual	Manual	**Automatic**
Workstation availability during generation	Unavailable	Unavailable	**Available**

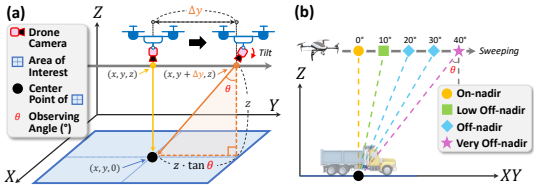


Figure 2: Illustration of one of the default camera viewpoint sampling methods in G-MAD: "air-to-air". (a) Geometric parameterization of camera motion around the area of interest, defined by the observing angle and lateral displacement. (b) Viewpoint sweep from nadir to off-nadir.

2.1 Scenario specification

Rather than requiring users to manually construct individual scenarios, the proposed framework generates scenarios automatically from a compact set of high-level constraints provided by the user. These constraints include ① asset categories[2], ② the expected minimum and maximum numbers of objects to place, ③ weather, ④ time (day/night), ⑤ map, ⑥ camera sensor configuration (⑥-a. Altitude, ⑥-b. FOV, ⑥-c. RGB-thermal modality[3], ⑥-d. captured image size (width/height), ⑥-e. camera view sampling strategy per each scene), ⑦ number of scenarios. Under these constraints, G-MAD automatically samples each scenario by randomly determining asset category, object arrangement, camera target point, and time, while keeping other factors like weather and sensor settings fixed.

2.2 Scenario sampling and multi-view, multi-modal capture

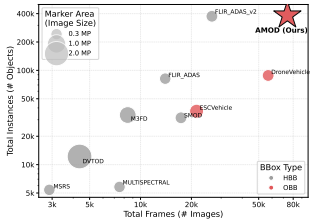
Given the user-specified constraints described in Sec. 2.1, G-MAD formulates data generation as a two-stage sampling process over scenarios and viewpoints. Let $\mathcal{S}$ denote the space of valid scenarios defined by user constraints, and $\mathcal{V}$ denote the space of possible viewpoints and sensing configurations. The data generation process can be expressed as $s \sim P(\mathcal{S} \mid \text{constraints})$, $v \sim P(\mathcal{V} \mid s)$, where each sampled pair (s, v) produces a single annotated scene.

[2]We additionally provide a ***pre-defined table that assigns each of the 513 unique Arma3 asset models*** to one of the 12 classes described in Sec. 4, enabling users to immediately test our code; kindly refer to "classes/CLASSES.csv". By referring to this table, users can create data for their desired assets. Moreover, users are ***not limited to military assets***; they can also import civilian equipment into Arma3. For further details, please refer to the "README.md" in our code and the official Arma3 website.

[3]Basically, ***white-hot thermal images*** are generated using the Arma3 TI engine. If specific spectrum (e.g., MWIR) is required, you may use image-to-image translation techniques (e.g., [17]), to convert the generated images into the target spectrum.

Table 2: Comparison of the AMOD dataset with existing RGB-T aerial object detection benchmarks.

Dataset	Scenario	Image Size	Total Frames	Bounding Box Type	Total Instances	Total Categories	Location	ID	Simultaneous Viewpoints
MULTISPECTRAL (MM-17) [58]	Driving (DV)	640x480	7,512	HBB	5.8k	5	Asia	N	Single
FLIR_ADAS (FLIR-18) [4]	DV	480x640	14,000	HBB	81.7k	4	North America	N	Single
DroneVehicle (TCSVT-22) [57]	Drone (DR)	640x512	56,878	OBB	88.3k	5	Asia	N	Single
FLIR_ADAS_v2 (FLIR-22) [4]	DV	640x512	26,442	HBB	375k	15	North America	N	Single
MSRS (IF-22) [59]	DV	480x640	2,888	HBB	5.4k	8	Asia	N	Single
M3FD (CVPR-22) [42]	DR + DV	1024x768	8,400	HBB	33.6k	6	Asia	N	Single
DVTOD (TIV-24) [55]	DR	1920x1080	4,358	HBB	12.2k	3	Asia	N	Single
SMOD (TMM-25) [14]	DV	640x512	17,352	HBB	31.4k	4	Asia	N	Single
ESCVehicle (TGRS-26) [54]	DR	704x704	21,454	OBB	36.9k	7	Asia	N	Single
AMOD (Ours)	**DR**	**1920x1440**	**73,920**	**OBB**	**383.2k**	**12**	**Multiple**	**Y**	**Multiple**



This formulation decouples scenario-level variables (object arrangement, environment) from scene-level variables (camera viewpoint and sensing modality), enabling structured data generation. A single scenario can produce multiple consistent observations by varying viewpoints and sensing modalities while preserving the underlying layout. Specifically, for each sampled scenario, G-MAD generates multiple scenes by sampling camera poses around a target region and assigning sensing modalities such as RGB and thermal. This yields a structured dataset of size $|\mathcal{S}| \times |\mathcal{V}|$, where scenes derived from the same scenario share identical object configurations but differ in viewpoint and modality.

Camera viewpoints are defined through a user-configurable sampling model, allowing *controlled multi-view data generation*. The current G-MAD framework offers two default motion modes, "air-to-air" and "air-to-land" for users who prefer not to design a custom sampling model. In "air-to-air", the camera moves laterally while maintaining a relatively stable altitude, producing sweeping aerial observations; see Fig. 2. In "air-to-land", both altitude and lateral position vary following a parabolic trajectory, resulting in oblique views directed toward the ground; see Fig. 1.

Importantly, the viewpoint sampler is designed to be modular and extensible[4]. Rather than relying on a fixed trajectory, G-MAD defines camera poses through a parameterized sampling routine, allowing users to incorporate custom motion patterns with minimal modification, e.g., circular orbits or task-specific reconnaissance paths. Such flexibility enables users to tailor viewpoint distributions to specific downstream tasks.

This structured generation process provides two key benefits for learning. First, it enables consistent supervision across viewpoints, allowing models to learn viewpoint-invariant representations from multiple observations of the same underlying scene. Second, it naturally supports aligned multi-modal data generation, where RGB and thermal images share identical annotations, facilitating supervised cross-modal and fusion-based learning.

2.3 Automatic annotation for object detection

G-MAD implements a direct annotation pipeline by querying object-level geometric information from the Arma3 engine and projecting it into the image plane. Unlike prior approaches that rely on foreground-background image differencing, our framework retrieves precise 3D bounding box information from the Arma3's engine and transforms it into 2D annotations aligned with each captured scene, ensuring high-quality labels.

Formally, let an object in a scenario be associated with a set of 3D bounding box vertices. These vertices are projected onto the image plane using the camera parameters corresponding to each sampled viewpoint. The resulting 2D coordinates are used to generate both horizontal and oriented bounding boxes (HBB/OBB), providing richer supervision for detection tasks.

Furthermore, G-MAD incorporates automatic filtering of invalid annotations, such as objects fully occluded by other scene elements, as determined by the absence of a direct line-of-sight between the camera and the object surface. This reduces label noise of ghost bounding boxes compared to pipelines such as [1] that do not explicitly handle occlusion.

2.4 Comparison with existing Arma3-based open source data generation frameworks

To clarify the contribution of G-MAD, we compare it with two representative open-source Arma3-based data generation frameworks in Tab. 1. Prior approaches [1, 2] are limited in both data diversity and systematic generation: they assume at most one object per image, rely on manual or inflexible camera setup, and provide little support for research-oriented, controlled dataset construction. Another key limitation is execution control. Existing methods depend on GUI-level automation via click-based actions to launch and operate Arma3, making the pipeline brittle to UI state changes and user interference. In addition, their annotations are obtained through image differencing[5], which is cumbersome and error-prone for multi objects. Furthermore, this inherently limits its extensibility to data generation frameworks for video-level tasks, such as tracking.

In contrast, G-MAD enables constraint-aware multi-object generation, GUI-independent execution, and Arma3 engine-native annotation of both HBBs and OBBs. Unlike [1, 2], our work further supports multi-map backgrounds, configurable resolution with supersampling, and automatic cache cleanup, establishing Arma3 as a structured and scalable data generation framework for object detection in overhead imagery.

3 Application of G-MAD

To demonstrate the research utility of G-MAD, we highlight three representative application settings enabled by our framework: 1) multi-view learning, 2) cross-modal RGB-T learning and fusion, and 3) pretraining for real-world object detection. Because G-MAD generates multiple synchronized observations from a shared underlying scenario while allowing controlled variation of viewpoint, modality, and environmental factors, it provides a practical foundation for studying data-centric learning problems.

[4]Users can implement their own camera viewpoint sampling model in "src/my_scene_generator/util.py". Specifically, please carefully update "get_all_cam_and_target_points()" and "get_all_cam_and_target_points_diff()". For more details, please see "README.md" of our code.

[5]In [1, 2], the object localization is inferred indirectly from image differencing between two distinct renders, one with the object present and the other with the object absent.

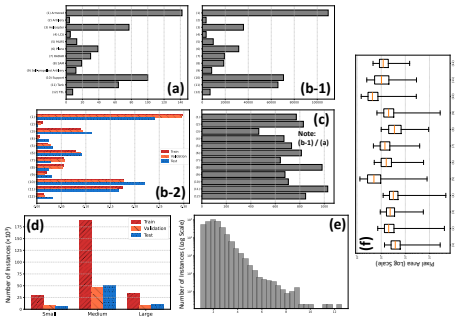


Figure 3: Statistics of the AMOD dataset. (a) Object models per category used in Arma3. (b-1) Instance count per category. (b-2) Instance distribution per category in train/dev/test splits. (c) Normalized instance count by Arma3 models across categories. (d) Size distribution (Small/Medium/Large) in train/dev/test splits. (e) Instance size of OBBs per category. (f) Aspect ratio distribution of OBBs per category.

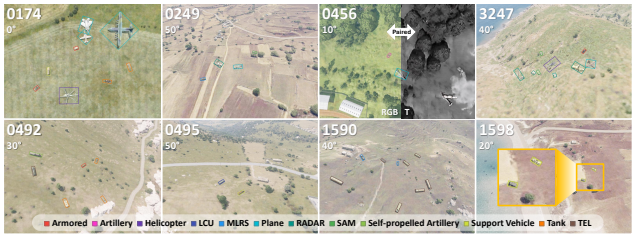


Figure 4: Annotated examples from the AMOD dataset featuring oriented bounding boxes (OBB). The dataset includes paired RGB-T imagery (see region '0456/10°' for instance).

Figure 5: Qualitative analysis of an AMOD-trained detector on unseen real-world RGB imagery containing military equipment on real-world RGB satellite photographs released through public news coverage of the Russia–Ukraine war.

4 Benchmarking test of G-MAD

To validate the effectiveness of G-MAD, we conduct a set of benchmarking experiments focusing on two aspects, 1) and 3), aligned with Sec. 3. We construct a dataset using G-MAD, referred to as Aerial Military Object Detection (AMOD), which consists of synchronized RGB-T multi-view aerial observations with OBB annotations of following categories: *Armored*, *Artillery*, *Helicopter*, *LCU*, *MLRS*, *Plane*, *RADAR*, *SAM*, *Self-propelled Artillery*, *Support Vehicle*, *Tank*, and *TEL*; see Figs. 3, 4, Tab. 2, and Supp. Material for details. As a baseline detector, we adopt Oriented R-CNN [67] with a Swin-S [46] backbone, following prior work [31, 37, 63, 72] on aerial detection, using MMRotate [73].

Multi-view learning. We first evaluate the effect of multi-view supervision using synchronized observations generated from the same scenarios. As shown in Tab. 3, while single-view models degrade under unseen viewing angles, *multi-view training* significantly improves generalization, achieving a mean $AP_{50:95}$ of 50.96 compared to 44.57 for the best single-view model (+6.39).

Table 3: Cross-angular evaluation results ($AP_{50:95}$) on AMOD.

Train	Test 0°	10°	20°	30°	40°	50°	Mean
0°	51.16	49.62	47.44	43.07	36.88	26.60	42.46
10°	50.92	50.01	48.04	44.79	38.76	28.39	43.49
20°	50.51	49.88	48.57	45.64	40.13	32.71	44.57
30°	48.67	48.27	47.42	45.69	41.46	33.66	44.20
40°	46.56	46.67	46.41	44.80	42.24	37.07	43.96
50°	38.23	38.20	39.04	39.29	37.95	36.49	38.20
All	**56.18** (+5.02)	**54.75** (+4.74)	**53.93** (+5.36)	**51.03** (+5.34)	**47.67** (+5.43)	**42.20** (+5.13)	**50.96** (+6.39)

Table 4: Investigating the effectiveness of pretraining with AMOD on two real-world benchmarks, DIOR-R (Visible) and HIT-UAV (Thermal) (AP_{50}).

Pretrained from	Finetuned for DIOR-R Airplane	Ship	Vehicle	Windmill	Mean	Finetuned for HIT-UAV Person	Car	Others	Mean
ImageNet	71.88	81.02	42.59	57.07	63.14	85.30	81.29	57.12	74.57
AMOD	**79.71**	**89.04**	**53.21**	**57.36**	**69.83** (+6.69)	**87.68**	**82.15**	**61.40**	**77.08** (+2.51)

Pretraining for real-world detection. We further evaluate a synthetic-to-real transfer setting, where models are pretrained on RGB and thermal variants of AMOD and fine-tuned on real-world RGB and thermal benchmarks, respectively. As shown in Tab. 4, AMOD pretraining consistently outperforms ImageNet initialization, improving mean AP_{50} from 63.14 to 69.83 on DIOR-R (+6.69) and from 74.57 to 77.08 on HIT-UAV (+2.51). This suggests that AMOD provides transferable pretraining signals for both visible and thermal real-world aerial detection[6].

Unseen real-world inference of military targets. Since authoritative real-world datasets for military target detection are difficult to obtain, we further perform an unseen real-world inference test from public news coverage, strictly without any fine-tuning. As shown in Fig. 5, the AMOD-trained detector[7] still identifies plausible military targets on previously unseen real images.

Limitations of AMOD. Although AMOD provides a large-scale, synchronized multi-view RGB-T benchmark for aerial object detection, it is primarily specialized for military targets. This specialization stems from the use of Arma3, whose built-in assets and terrains are particularly well suited for military scenarios. However, since Arma3 supports user-defined asset integration, G-MAD can also be used to construct datasets for non-military general equipment.

5 Conclusions

In this work, we presented G-MAD, a scalable and configurable framework for generating synthetic RGB-T aerial detection data with synchronized multi-view capture and automatic annotation. Using G-MAD, we also constructed and released AMOD, an RGB-T aerial multi-view object detection dataset, and preliminary experiments with AMOD suggest that the proposed framework supports both controlled benchmarking and transfer to real-world settings.

Licensing and ethical considerations. G-MAD relies on Arma3 as an off-the-shelf rendering and scenario-execution environment. Accordingly, the framework is designed to avoid redistributing Bohemia Interactive intellectual property including Arma3 executable game files, original asset packages, or modified copies of the game.

[6]For pretraining, we perform a one-time syn-to-real style transfer of our AMOD toward the target RGB/thermal domain, respectively; the detailed pipeline is in Supp. Material.
[7]During AMOD training, only the RGB split is leveraged, incorporating DOTA [66]-style transfer; kindly refer to Supp. Material for details.

G-MAD only provides external scripts that users may run with their own legitimately obtained local installation of Arma3. The framework is released solely for non-commercial academic or prototype research purposes, and users are responsible for ensuring that their use complies with the Arma3 End User License Agreement.

Acknowledgments

Note that for the development of commercial software, LIG Defense&Aerospace (LIG D&A) only uses duly licensed subscriptions to VBS4, a professional military simulation platform developed by Bohemia Interactive Simulations. Arma3, developed by Bohemia Interactive, was used in this study solely for non-commercial academic purposes to facilitate the publication and dissemination of the paper and related research outcomes to the broader research community. It was not used in the development of the company's commercial software. Moreover, the views, analyses, and conclusions presented in this paper are solely those of the authors and do not represent the official position, policies, or endorsement of LIG D&A. This work was partly supported by the National Research Foundation of Korea (NRF) Grant by the Korean Government through Ministry of Science and ICT (MSIT) under Grant No. RS-2026-25475324. The authors would like to thank SooYeon Kim (KRIT), Sung Heon Kim (GIST), Sihyun Kim (YU), Junggyun Oh (HDC), and Prof. Yeongmin Ko (KNU) for their assistance.

References

[1] [Website]. https://github.com/cloftus96/Synthetic-Data-Generation.
[2] [Website]. https://github.com/ttsiapras/Arma3DatasetGen.
[3] [Website]. Arma3. https://arma3.com.
[4] [Website]. Free FLIR Thermal Dataset for Algorithm Training. https://oem.flir.com/solutions/automotive/adas-dataset-form/.
[5] [Website]. Grand Theft Auto V. https://www.rockstargames.com/gta-v.
[6] [Website]. Unity. https://unity.com.
[7] [Website]. Unreal. https://www.unrealengine.com.
[8] Eunsu Baek, Keondo Park, Jiyoon Kim, and Hyung-Sin Kim. 2024. Unexplored faces of robustness and out-of-distribution: Covariate shifts in environment and sensor domains. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 22294–22303.
[9] Maciej Bala, Yin Cui, Yifan Ding, Yunhao Ge, Zekun Hao, Jon Hasselgren, Jacob Huffman, Jingyi Jin, JP Lewis, Zhaoshuo Li, et al. 2024. Edify 3d: Scalable high-quality 3d asset generation. *arXiv preprint arXiv:2411.07135* (2024).
[10] Mahroosh Banday and Brejesh Lall. 2025. Multi spectral visible-thermal IR image translation using improved u-net & conditional diffusion. *Neurocomputing* (2025), 131006.
[11] Hrishav Bakul Barua, Kalin Stefanov, KokSheik Wong, Abhinav Dhall, and Ganesh Krishnasamy. 2025. GTA-HDR: A large-scale synthetic dataset for HDR image reconstruction. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 7876–7886.
[12] Octavian Blaga, AI Synetic, and David Scott. 2025. Breaking the Bottleneck: Synthetic Data as the New Foundation for Vision AI. (2025).
[13] Lijing Cai, Xiangyu Dong, Kailai Zhou, and Xun Cao. 2024. Exploring video denoising in thermal infrared imaging: Physics-inspired noise generator, dataset, and model. *IEEE Trans. Image Process.* 33 (2024), 3839–3854.
[14] Zizhao Chen, Yeqiang Qian, Xiaoxiao Yang, Chunxiang Wang, and Ming Yang. 2025. AMFD: Distillation via adaptive multimodal fusion for multispectral pedestrian detection. *IEEE Trans. Multimedia* (2025).
[15] Gong Cheng, Junwei Han, Peicheng Zhou, and Lei Guo. 2014. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS J. Photogramm. Remote Sens.* 98 (2014), 119–132.
[16] Gong Cheng, Jiabao Wang, Ke Li, Xingxing Xie, Chunbo Lang, Yanqing Yao, and Junwei Han. 2022. Anchor-free oriented proposal generator for object detection. *IEEE Trans. Geosci. Remote Sens.* 60 (2022), 1–11.
[17] Sungjin Cheong, Wonho Jung, Yoon Seop Lim, and Yong-Hwa Park. 2024. Thermal-infrared remote-target detection system for maritime rescue using 3-D game-based data augmentation with GAN. *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024), 1–13.
[18] Rita Delussu, Lorenzo Putzu, and Giorgio Fumera. 2024. Synthetic data for video surveillance applications of computer vision: A review. *International Journal of Computer Vision* 132, 10 (2024), 4473–4509.
[19] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Ying Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, et al. 2021. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 11 (2021), 7778–7796.
[20] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on Robot Learning (CoRL)*. PMLR, 1–16.
[21] Aritra Dutta, Srijan Das, Jacob Nielsen, Rajatsubhra Chakraborty, and Mubarak Shah. 2024. Multiview aerial visual recognition (mavrec): Can multi-view improve aerial visual perception?. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 22678–22690.
[22] Ronald L. Graham. 1972. An efficient algorithm for determining the convex hull of a finite planar set. *Info. Proc. Lett.* 1 (1972).
[23] Zonghao Han, Shun Zhang, Yuru Su, Xiaoning Chen, and Shaohui Mei. 2024. DR-AVIT: Toward diverse and realistic aerial visible-to-infrared image translation. *IEEE Trans. Geosci. Remote Sens.* 62 (2024), 1–13.
[24] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H Hsu. 2017. Drone-based object counting by spatially regularized regional proposal network. In *IEEE Int. Conf. Comput. Vis. (ICCV)*. 4145–4153.
[25] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. 2015. Multispectral pedestrian detection: Benchmark dataset and baseline. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 1037–1045.
[26] Yuxiang Ji, Boyong He, Zhuoyue Tan, and Liaoni Wu. 2025. Game4loc: A uav geo-localization benchmark from game data. In *AAAI Conf. Artif. Intell. (AAAI)*, Vol. 39. 3913–3921.
[27] Yuxiang Ji, Boyong He, Zhuoyue Tan, and Liaoni Wu. 2025. MMGeo: Multimodal Compositional Geo-Localization for UAVs. In *IEEE Int. Conf. Comput. Vis. (ICCV)*. 25165–25175.
[28] Chenchen Jiang, Huazhong Ren, Fengguang Li, Zhonghua Hong, Hongtao Huo, Junqiang Zhang, and Jiuyuan Xin. 2025. Object detection from aerial multi-angle thermal infrared remote sensing images: Dataset and method. *ISPRS J. Photogramm. Remote Sens.* 228 (2025), 438–452.
[29] Zhao Jin, Yinjie Lei, Naveed Akhtar, Haifeng Li, and Munawar Hayat. 2022. Deformation and correspondence aware unsupervised synthetic-to-real scene flow estimation for point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7233–7243.
[30] Beomsu Kim, Gihyun Kwon, Kwanyoung Kim, and Jong Chul Ye. 2024. Unpaired image-to-image translation via neural schr\" odinger bridge. In *Int. Conf. Learn. Represent. (ICLR)*.
[31] Yechan Kim, SooYeon Kim, and Moongu Jeon. 2025. Nbbox: Noisy bounding box improves remote sensing object detection. *IEEE Geoscience and Remote Sensing Letters* 22 (2025), 1–5.
[32] Yechan Kim, Younkwan Lee, and Moongu Jeon. 2021. Imbalanced image classification with complement cross entropy. *Pattern Recognit. Lett.* 151 (2021), 33–40.
[33] Alexander Kirillov et al. 2023. Segment anything. In *IEEE Int. Conf. Comput. Vis. (ICCV)*.
[34] Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. 2018. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856* (2018).
[35] Yunwei Lan, Zhigao Cui, Xin Luo, Chang Liu, Nian Wang, Menglin Zhang, Yanzhao Su, and Dong Liu. 2025. When Schrodinger Bridge Meets Real-World Image Dehazing with Unpaired Training. In *IEEE Int. Conf. Comput. Vis. (ICCV)*. 8756–8765.
[36] Dong-Guw Lee, Myung-Hwan Jeon, Younggun Cho, and Ayoung Kim. 2023. Edge-guided multi-domain rgb-to-tir image translation for training vision tasks with challenging labels. In *IEEE Int. Conf. Robot. Autom. (ICRA)*.
[37] Yujie Lei, Jie Zhang, Wenjie Sun, Wei He, Jiasong Zhu, and Qingquan Li. 2025. MGNet: A Remote Sensing Oriented Object Detector Based on Multi-Cascaded Feature Selection and Geometric Constraints. *IEEE Transactions on Geoscience and Remote Sensing* (2025).
[38] Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. 2023. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 1952–1961.
[39] Ke Li et al. 2020. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* 159 (2020).
[40] Na Li, Haining Wang, Huijie Zhao, and Wen Ou. 2025. Cross-Modal Visible-to-Infrared Image Translation in Remote Sensing Guided by Thermal Features. *IEEE Trans. Geosci. Remote Sens.* (2025).
[41] Wei Lin, Junyu Gao, Qi Wang, and Xuelong Li. 2021. Learning to detect anomaly events in crowd scenes from synthetic data. *Neurocomputing* 436 (2021), 248–259.
[42] Jinyuan Liu, Xin Fan, Zhanbo Huang, Guanyao Wu, Risheng Liu, Wei Zhong, and Zhongxuan Luo. 2022. Target-aware dual adversarial learning and a multi-scenario multi-modality benchmark to fuse infrared and visible for object detection. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 5802–5811.

[43] Jiahang Liu, Xiaozhen Wang, Mao Guo, Ruilei Feng, and Yue Wang. 2023. Shadow detection in remote sensing images based on spectral radiance separability enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.* 46, 5 (2023), 3438–3449.
[44] Kang Liu and Gellert Mattyus. 2015. Fast multiclass vehicle detection on aerial images. *IEEE Geosci. Remote Sens. Lett.* 12, 9 (2015), 1938–1942.
[45] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised image-to-image translation networks. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*. 1–9.
[46] Ze Liu et al. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE International Conference on Computer Vision (ICCV)*.
[47] Zikun Liu, Hongzhen Wang, Lubin Weng, and Yiping Yang. 2016. Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE Geosci. Remote Sens. Lett.* 13, 8 (2016), 1074–1078.
[48] Yang Long, Yiping Gong, Zhifeng Xiao, and Qing Liu. 2017. Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* 55, 5 (2017), 2486–2498.
[49] Decao Ma, Juan Su, Bing Li, Yong Xian, Shaopeng Li, and Yao Ding. 2025. Self cycle strategy for unpaired visible-to-infrared image translation. *Pattern Recognit.* (2025), 112253.
[50] Mehmet Akif Özkanoğlu and Sedat Ozer. 2022. InfraGAN: A GAN architecture to transfer visible images to infrared domain. *Pattern Recognit. Lett.* 155 (2022), 69–76.
[51] Jiancheng Pan, Yanxing Liu, Yuqian Fu, Muyuan Ma, Jiahao Li, Danda Pani Paudel, Luc Van Gool, and Xiaomeng Huang. 2025. Locate anything on earth: Advancing open-vocabulary object detection for remote sensing community. In *AAAI Conf. Artif. Intell. (AAAI)*, Vol. 39. 6281–6289.
[52] Goran Paulin and Marina Ivasic-Kos. 2023. Review and analysis of synthetic dataset generation methods and techniques for application in computer vision. *Artificial Intelligence Review* 56, 9 (2023), 9221–9265.
[53] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. 2016. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision (ECCV)*. Springer, 102–118.
[54] Jiamin Song, Nan Zhang, Zhenhao Wang, and Tian Tian. 2026. ESCVehicle: A Drone-based Visible-Infrared Vehicle Benchmark with Extensive Scene Coverage. *IEEE Trans. Geosci. Remote Sens.* (2026).
[55] Kechen Song, Xiaotong Xue, Hongwei Wen, Yingying Ji, Yunhui Yan, and Qinggang Meng. 2024. Misaligned visible-thermal object detection: A drone-based benchmark and baseline. *IEEE Trans. Intell. Veh.* (2024).
[56] Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, Wenhui Diao, Jin Chen, Jihao Li, Yingchao Feng, Tao Xu, et al. 2022. FAIR1M: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS J. Photogramm. Remote Sens.* 184 (2022), 116–130.
[57] Yiming Sun, Bing Cao, Pengfei Zhu, and Qinghua Hu. 2022. Drone-based RGB-infrared cross-modality vehicle detection via uncertainty-aware learning. *IEEE Trans. Circuits Syst. Video Technol.* 32, 10 (2022), 6700–6713.
[58] Karasawa Takumi, Kohei Watanabe, Qishen Ha, Antonio Tejero-De-Pablos, Yoshitaka Ushiku, and Tatsuya Harada. 2017. Multispectral object detection for autonomous vehicles. In *ACM Int. Conf. Multimedia (MM) Worksh.* 35–43.
[59] Linfeng Tang, Jiteng Yuan, Hao Zhang, Xingyu Jiang, and Jiayi Ma. 2022. PIAFusion: A progressive infrared and visible image fusion network based on illumination aware. *Inf. Fusion* 83 (2022), 79–92.
[60] Godfried T Toussaint. 1983. Solving geometric problems with the rotating calipers. In *IEEE Melecon*, Vol. 83.
[61] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. 2019. Learning from synthetic data for crowd counting in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8198–8207.
[62] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. 2021. Pixel-wise crowd understanding via synthetic data. *International Journal of Computer Vision* 129, 1 (2021), 225–245.
[63] Tao Wang, Chenyu Lin, Chenwei Tang, Jizhe Zhou, Deng Xiong, Jianan Li, Jian Zhao, and Jiancheng Lv. 2026. Adaptive image zoom-in with bounding box transformation for UAV object detection. *ISPRS Journal of Photogrammetry and Remote Sensing* 233 (2026), 452–466.
[64] Nicholas Weir, David Lindenbaum, Alexei Bastidas, Adam Van Etten, Sean McPherson, Jacob Shermeyer, Varun Kumar, and Hanlin Tang. 2019. Spacenet mvoi: A multi-view overhead imagery dataset. In *IEEE Int. Conf. Comput. Vis. (ICCV)*. 992–1001.
[65] Ancong Wu, Wei-Shi Zheng, Hong-Xing Yu, Shaogang Gong, and Jianhuang Lai. 2017. RGB-infrared cross-modality person re-identification. In *IEEE Int. Conf. Comput. Vis. (ICCV)*. 5380–5389.
[66] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. 2018. DOTA: A large-scale dataset for object detection in aerial images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3974–3983.
[67] Xingxing Xie, Gong Cheng, Jiabao Wang, Ke Li, Xiwen Yao, and Junwei Han. 2024. Oriented R-CNN and beyond. *International Journal of Computer Vision* 132, 7 (2024), 2420–2442.
[68] Ze Yang, Jingkang Wang, Haowei Zhang, Sivabalan Manivasagam, Yun Chen, and Raquel Urtasun. 2025. Genassets: Generating in-the-wild 3d assets in latent space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 22392–22403.
[69] Haijun Zhang, Mingshan Sun, Qun Li, Linlin Liu, Ming Liu, and Yuzhu Ji. 2021. An empirical study of multi-scale object detection in high resolution UAV images. *Neurocomputing* 421 (2021), 173–182.
[70] Lei Zhao, Mengwei Li, Bo Li, and Xingxing Wei. 2025. Diverse Visible-to-Thermal Image Translation via Controllable Temperature Encoding. *IEEE Trans. Multimedia* (2025).
[71] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. 2022. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*. 3609–3623.
[72] Shangdong Zheng, Yang Xu, Peng Zheng, Zhihui Wei, and Zebin Wu. 2026. See Hidden Insight From Transposition: Multi-Axis Feature Aggregation for Aerial Object Detection. *IEEE Transactions on Geoscience and Remote Sensing* (2026).
[73] Yue Zhou, Xue Yang, Gefan Zhang, Jiabao Wang, Yanyi Liu, Liping Hou, Xue Jiang, Xingzhao Liu, Junchi Yan, Chengqi Lyu, et al. 2022. Mmrotate: A rotated object detection benchmark using pytorch. In *ACM International Conference on Multimedia*. 7331–7334.
[74] Haigang Zhu, Xiaogang Chen, Weiqun Dai, Kun Fu, Qixiang Ye, and Jianbin Jiao. 2015. Orientation robust object detection in aerial images using deep convolutional neural network. In *IEEE Int. Conf. Image Process. (ICIP)*. IEEE, 3735–3739.

6 Supplementary material

§A Bounding box extraction from Arma3

This section describes the procedure used to automatically extract 2D bounding boxes (BBoxes), both horizontal bounding boxes (HBBs) and oriented bounding boxes (OBBs), from 3D objects in Arma3. For convenience, we assume the BBox extraction for a single object.

§A.1 Obtaining the model-space BBox

For each target object, we first invoke the function ***boundingBoxReal***, which returns the lower and upper corners of the object's 3D BBox in the object's (local) model coordinate system. Note that Arma3 assumes that each object is located at the origin of its own model space (m). Selecting index 1 yields the upper-corner coordinate (x_m, y_m, z_m); see Fig. S-1(a). Given this upper-corner value, we enumerate all sign combinations of x_m, y_m, and z_m to generate the eight canonical model-space corner points $\{p_m^i\}_{i=1}^8$; see Fig. S-1(b).

§A.2 Transforming to world coordinates

Each model (m)-space corner point is then mapped into the (global) Arma3 world coordinate frame (w) using the transformation function ***modelToWorldVisual***, producing world-space coordinates p_w^i for $i \in \{1, \cdots, 8\}$; see Fig. S-1(c).

§A.3 Projecting into screen coordinates

To obtain 2D pixel-space coordinates, each world (w)-space point is projected onto the rendered image plane via the Arma3 camera projection function ***worldToScreen***; see Fig. S-1(d). Though Fig. S-1(d) does not explicitly depict it, an additional post-processing step is required to obtain the actual pixel coordinates from ***worldToScreen*** in Arma3. The function ***worldToScreen*** returns normalized screen coordinates expressed within the Arma3's *safe-zone* layout rather than the final pixel grid of the rendered image. Consequently, one must apply safe-zone compensation before obtaining valid pixel-space positions as follows.

Let $(q^i[x], q^i[y])$ denote the 2D coordinates returned by ***worldToScreen***. The final pixel coordinates $(p^i[x], p^i[y])$ are computed

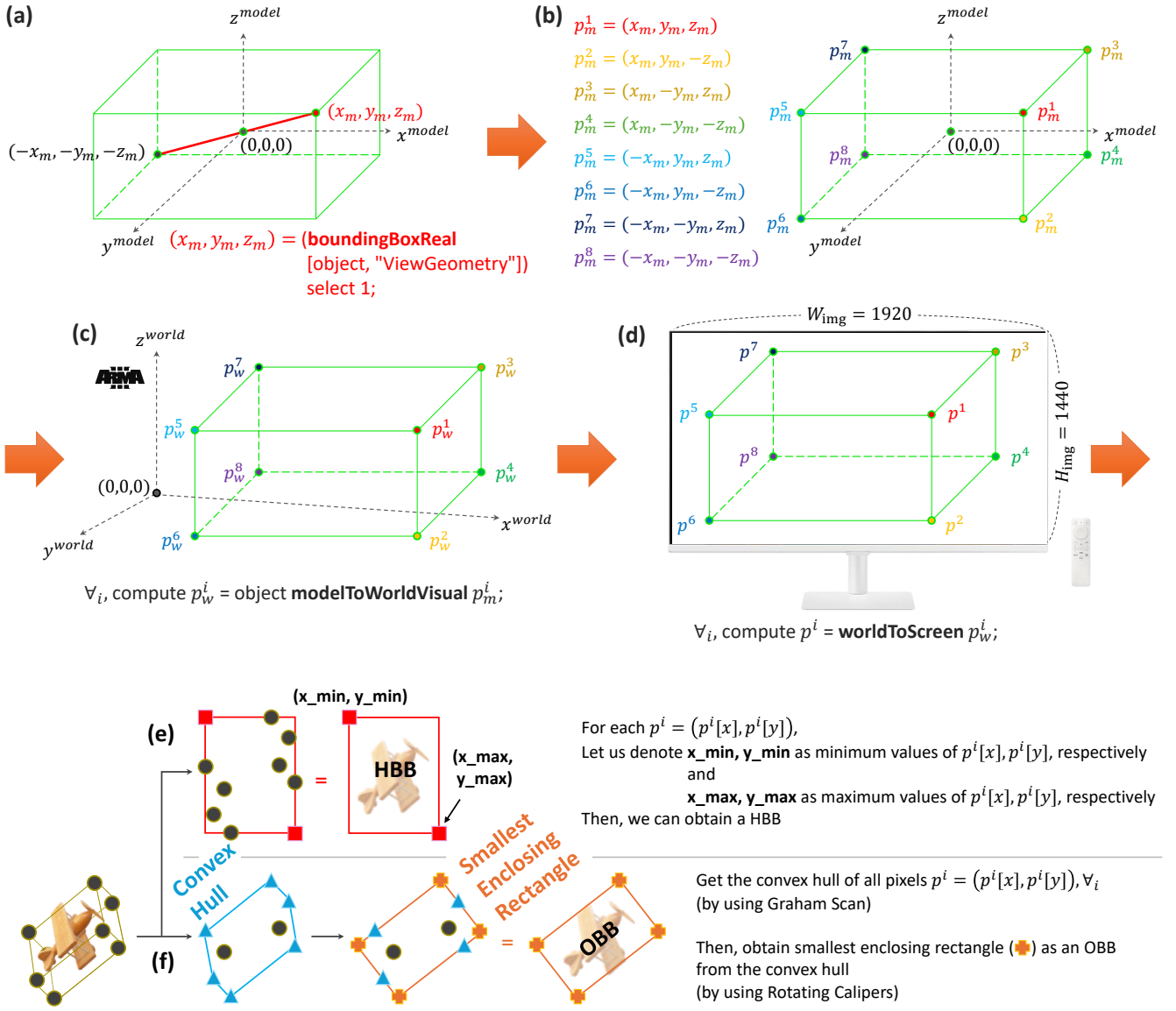


Figure S-1: Overview of the pipeline for computing both a 2D horizontal bounding box (HBB) and an oriented bounding box (OBB) for a certain 3D object in Arma3. (a) The 'boundingBoxReal' function is called in Arma3, which returns the lower and upper corners of the object's bounding box (BBox) in the object's own (local) model coordinate system (m), assuming the object itself is located at the origin; selecting index 1 yields the upper-corner coordinate (x_m, y_m, z_m). (b) Using (x_m, y_m, z_m), all eight m-space corner points p_m^i $(i = 1, \cdots, 8)$ are generated by enumerating all sign combinations of $\pm x_m$, $\pm y_m$, and $\pm z_m$. (c) Each model-space corner is transformed into the (global) Arma3 world coordinate frame (w) by invoking the 'modelToWorldVisual' function of Arma3, producing p_w^i. (d) The w-space points are then projected into the image plane frame via the Arma3's 'worldToScreen' function, yielding $p^i \in \mathrm{R}^2$. (e) From these projected points, the HBB is obtained in screen coordinates by taking the minimum and maximum from x and y values in $\{p^i = (p^i[x], p^i[y])\}_{i=1}^{8}$. (f) The convex hull of all projected points is computed using Graham Scan, and the smallest enclosing rectangle of this hull is found using the Rotating Calipers algorithm, resulting in the final 2D OBB in screen coordinates.

by removing the safe-zone offset (SafeZoneX, SafeZoneY) and scaling with respect to the safe-zone width (SafeZoneW) and height (SafeZoneH):

$$p^i[x] = \frac{q^i[x] - \text{SafeZoneX}}{\text{SafeZoneW}} \times W_{\text{img}};$$
$$p^i[y] = \frac{q^i[y] - \text{SafeZoneY}}{\text{SafeZoneH}} \times H_{\text{img}}, \quad (1)$$

where SafeZoneX, SafeZoneY, SafeZoneW, and SafeZoneH are constants in Arma3; see https://community.bistudio.com/wiki/safeZone for details. In this work, we set W_{img} and H_{img} to 1920 and 1440, respectively, thus generating images with a resolution of 1920×1440.

§A.4 Computing HBB and OBB in screen space

Given the projected points $\{p^i\}_{i=1}^{8}$, we compute both forms of BBox as follows:

- **Horizontal Bounding Box (HBB).** The HBB is obtained by taking the minimum and maximum of the x- and y-coordinates across all projected points; see Fig. S-1(e).
- **Oriented Bounding Box (OBB).** The OBB is obtained by first computing the convex hull of the projected points using the Graham Scan algorithm. The smallest enclosing rectangle of this convex hull is then found using the Rotating Calipers method, yielding the final 2D OBB in screen coordinates; see Fig. S-1(f).

§A.5 Summary

This pipeline enables accurate extraction of both HBB and OBB annotations directly from Arma3's internal geometry and rendering functions. Because the pipeline relies solely on engine-grounded transformations and camera projections, the resulting annotations are geometrically consistent and suitableas ground-truth labels for training and evaluating aerial object detection models.

§B AMOD: a novel synthetic benchmark for RGB-T aerial military object detection

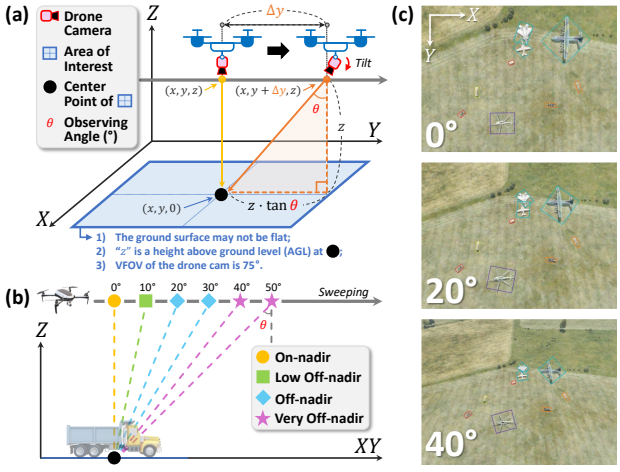


Figure S-2: Illustration of how to construct our AMOD benchmark. (a) Geometric definition of the observing angle θ, where the camera shifts laterally by Δy in Y-axis, producing an off-nadir view. (b) Drone camera sweeping setup simultaneously capturing the same area from nadir to very off-nadir viewpoints. (c) Examples from the RGB version of AMOD under different observation angles, showing how target appearance and perspective distortion vary with θ.

We introduce ***AMOD***[8], a new benchmark to provide synchronized multi-view aerial images using a game Arma3, enabling systematic analysis of view variation in aerial detection. We adopt the Arma3 [3] rather than general-purpose game engines such as Unity [6] and Unreal [7]. While those engines offer high-fidelity rendering, they often require custom asset modeling or significant licensing costs of domain-specific 3D models of equipment and terrain [9, 12, 20, 68], for large-scale data construction. In contrast, Arma3 already provides rich 3D object models with various terrain assets. This eliminates the need for heavy 3D modeling.

§B.1 Related work

Existing benchmark datasets for aerial object detection. Early benchmarks, such as NWPU VHR-10 [15], DLR 3K Vehicle [44], UCAS-AOD [74], HRSC [47], CARPK [24], and RSOD [48], primarily rely on static nadir-view imagery collected from Google Earth or airborne platforms. These datasets, typically consist of fewer than 2,000 images and limited to horizontal bounding boxes (HBB). They offer only coarse variations in viewpoint and scale, restricting their utility for robust generalization. Subsequent large-scale efforts, including DOTA-v2 [19], DIOR-R [16], FAIR1M [56], and DroneVehicle [57], expand coverage with tens of thousands of images and a greater number of categories (up to 20) with oriented bounding boxes (OBB) to better capture object rotations. To further expand data diversity, Pan et al. [51] propose LAE-1M, which unifies ten existing remote sensing (RS) datasets, such as xView [34], DOTA [66], and DIOR [39], into a single large-scale benchmark. Nevertheless, these datasets still largely center on single-view and nadir-oriented captures, lacking in multi-angle observations of the same scene (i.e. *homogeneous aerial perspective*).

A few efforts to bridge multiple viewpoints have also emerged. For instance, MAVREC [21] is the first attempt to integrate aerial and ground viewpoints in a temporally synchronized manner, allowing simultaneous observations of the same scene from dual perspectives. Though datasets, like MOHR [69], DroneVehicle [57], and DVTOD [55], also provide multiple viewpoints, they do not ensure viewpoint simultaneity[9]. We argue that multi-perspective alignment in data represents an important step toward modeling cross-view understanding.

Hence, our AMOD dataset is proposed as a synthetically constructed, multi-angular visible aerial object detection benchmark based on Arma3 [3]. While our dataset does not surpass recent large-scale datasets in terms of volume or category diversity, it offers a qualitatively distinct contribution: Unlike prior benchmarks focused on scale expansion, our dataset is captured from six distinct observation angles (i.e. *simultaneous multiple viewpoints*), enabling systematic analysis of viewpoint variation on aerial detection.

Multi-view aerial image understanding. Several lines of research in Earth vision leverage multi-view signals. For example, Weir et al. [64] introduces a benchmark entitled MVOI to reveal how

[8]It stands for "Aerial Military Object Detection in multi-view RGB-T scenarios."
[9]Recently, the dataset MATIR-OD [28] has been collected and released using a DJI Matrice 300 drone-based camera to investigate how different observation angles affect aerial object detection in the infrared domain. While MATIR-OD provides valuable insights into multi-angle thermal detection, it focuses on a single category and a modest-scale collection (3,000 images with 43,540 instances) without simultaneous multi-view captures.

off-nadir imagery degrades building extraction performance, emphasizing the geometric and radiometric challenges that arise with varying observation angles. The MVOI dataset provides multi-angle satellite imagery of the same scenes captured nearly simultaneously, but due to high annotation costs, only the nadir-view images are manually labeled; these annotations are then reused for off-nadir views, limiting the precision of view-specific supervision. In contrast, we generate our data using the Arma3 game, which allows us to provide far more precise bounding box labels for all viewing angles.

Another important direction concerns cross-view geo-localization and matching [27]. Ji et al. [26] introduces Game4Loc, a benchmark constructed using the GTA V [5] environment to facilitate this research line. These studies demonstrate the potential of simultaneous multi-view data for geometric reasoning and spatial correspondence, though their primary focus lies in geo-localization and retrieval rather than category-level detection. In comparison, AMOD explores how viewpoint diversity influences category-level detection performance, providing a complementary perspective on multi-view aerial understanding.

§B.2 General setup

Category design. As Arma3 is a military FPS game, most of its available 3D assets are related to military equipment. We adopt a total of 513 models in Arma3, 258 from the Eastern and 255 from the Western faction. Each model is then categorized into one of 12 classes: **Armored**, **Artillery**, **Helicopter**, **Landing Craft Utility (LCU)**, **Multiple Launch Rocket System (MLRS)**, **Plane**, **RADAR**, **Surface-to-air Missile (SAM)**, **Self-propelled Artillery**, **Support Vehicle**, **Tank**, and **Transporter Erector Launcher (TEL)**; the distribution of Arma3 assets assigned to each category is illustrated in Fig. 3 (Main).

Objects placement (spawn). Let $\mathcal{A}$ denote an area of interest (or target imaging area) for which simultaneous multi-view captures are provided as in Fig. 1 (Main). For each $\mathcal{A}$, our data generator[10] randomly selects and positions items as follows:

- **Step 1.** Sample k classes among the above 12 categories, and store them into a set S. Here, k is chosen between 1 and 8.
- **Step 2.** Randomly determine the number of items e to be placed in the $\mathcal{A}$ (e is between 8 and 14). The e items are selected from the 513 assets, restricted to the classes in S.

To further enhance data diversity, each object is assigned a unique rotation angle. Besides, we enforce that land assets must be located only on land and naval assets for sea, and that no objects overlap with each other.

Recording setup. As shown in Fig. S-2, we control the Arma3 to capture six simultaneous multi-view, RGB-T paired images for each $\mathcal{A}$, corresponding to observation angles of 0°, 10°, 20°, 30°, 40°, and 50°. Along with the images, the associated metadata is also extracted to generate bounding-box annotations for each view. Each image has a size of 1920×1440 and the cameras are configured such that the central pixel of the 0°-observing image is a ground sampling distance (GSD) of 0.1 m/pixel. For this nominal GSD, we set the camera altitude (z) to 120 m with Vertical FOV[11] 75° in Arma3. When z exceeds 120 m, shadows largely disappear, resulting in a notable loss of image detail in Arma3. Since the presence of shadows in aerial imagery is known to be of significant research importance [43], we do not consider z >120 m. Besides, we set the time range from 9 to 18 for capturing different lighting conditions[12]. Meanwhile, to enlarge the geographical coverage, we leverage several official Arma3 maps as follows[13]:

- For train/validation splits: Altis, Malden, Stratis, Tanoa, Weferlingen;
- For test split: Malden and Livonia.

Annotation. While capturing each scene, our generator obtains the 3D bounding box using 'boundingBoxReal' function of Arma3; the detailed pipeline is presented in Sec. A (Supp. Material). It then projects its corner points onto the image plane, and derive both horizontal and oriented bounding box (HBB/OBB) annotations. The HBB is computed by taking the minimum and maximum projected coordinates, while the OBB is obtained by applying a convex hull [22] followed by rotating calipers [60] to find the minimum-area enclosing rectangle. Finally, as summarized in Tab. 2 (Main), in our benchmark, to facilitate future multi-view studies, the same object observed across different views is assigned a consistent object ID for each region of interest ($\mathcal{A}$).

Data balancing. Class imbalance is undesirable, as it may bias model training toward overrepresented categories and degrade generalization performance [32]. However, achieving perfectly uniform class distributions is infeasible in our data because the number of available 3D assets varies across categories as seen in Fig. 3(a) (Main). To address this, we post-process our initially generated data so that the ratio between the number of instances per class and the number of Arma3 items is made as uniform as possible; see Fig. 3(c) (Main).

§B.3 Statistics, challenges, and applications

Train, validation, and test splits. The dataset is partitioned into three subsets: a training set comprising 47,808 images (64.68%), a validation set with 11,988 images (16.21%), and a test set with 14,124 images (19.11%). We maintain similar class distributions across all splits as shown in Fig. 3(b-2) (Main).

Object size and aspect ratio distributions. Following [21], we categorize object sizes into three groups: small (≤ 32×32 pixels), medium (between 32×32 and 96×96), and large (≥ 96×96). The object size distribution of our dataset is illustrated in Fig. 3(d) (Main). Meanwhile, Figs. 3(e)-(f) (Main) illustrate the average size and aspect ratio of OBBs for each category, respectively.

Distinct characteristics of AMOD. Our dataset is designed with three notable properties as follows.

(1) It provides *simultaneous multi-view observations* of each region of interest ($\mathcal{A}$), enabling research on cross-view consistency in aerial image understanding. Owing to the complex

[10] The values of k and e are empirically chosen and can be changed.

[11] VFOV 75° is a broad viewing angle similar to that of a standard wide-angle camera. As our primary focus is on unprocessed UAV imagery, neither orthorectification nor lens distortion correction is applied. Nevertheless, pretraining with AMOD improves performance over the baseline when evaluated on real-world datasets like DIOR-R, composed of satellite images from Google Earth (Standard Orthophoto).

[12] In Arma3, visible images captured at night are usually rendered entirely black, making them unusable. Therefore, we do not collect RGB-T pairs under nighttime conditions.

[13] Note that all maps are utilized at a nearly equal frequency.

Figure S-3: Examples of (a) low inter-class and (b) high intra-class variances in Arma3 models we used for our dataset.

Table S-1: Class-wise Test AP on AMOD.

	Armored	Artillery	Helicopter	LCU	MLRS	Plane	RADAR	SAM	Self-propelled Artillery	Support	Tank	TEL	Mean
AP_{75}	40.30	74.20	50.40	78.30	65.20	79.30	25.70	30.60	37.40	40.60	47.40	53.70	51.93
$AP_{50:95}$	47.19	64.35	50.47	58.27	56.08	65.20	31.65	38.92	43.53	46.46	48.91	54.76	50.48

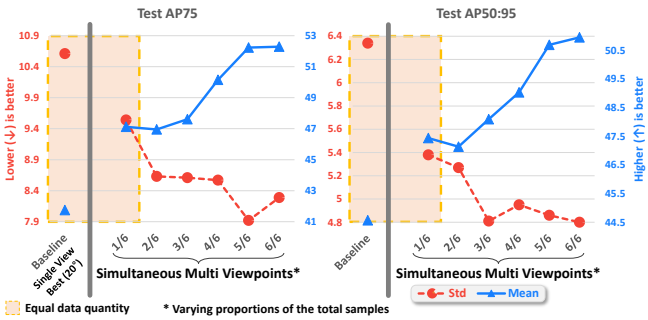


Figure S-4: Effect of angular diversity versus data quantity on AMOD (AP_{75}, $AP_{50:95}$). The blue and red dots denote the mean and standard deviation of AP values across all observation angles, respectively.

terrain in Arma3 environments, such as cliffs and mountains, certain objects are occluded in some viewpoints[14].

(2) It exhibits both *low inter-class and high intra-class variations*. For instance, visually similar vehicle types such as tanks and armored vehicles challenge fine-grained recognition, while a single class like RADAR includes highly diverse structures, as illustrated in Fig. S-3.

(3) It ensures *diverse geographical backgrounds* by leveraging diverse official Arma3 maps, thereby enhancing scene variability and model's generalization ability.

§B.4 Detailed experimental setup and more experimental results for our AMOD benchmark dataset

Unless otherwise stated, all experiments here are conducted with the same detector, dataset, and optimization configuration. We adopt Oriented R-CNN with a Swin-S backbone as implemented in MMRotate, initialized from ImageNet-pretrained weights.

Class-wise performance on AMOD We first report class-wise detection performance of AMOD in Tab. S-1, which serves as a reference breakdown of category-level performance under the default evaluation setting. Categories such as LCU (78.30 AP_{75}) and Plane (79.30 AP_{75}) achieve relatively high performance, likely because they exhibit distinctive global shapes with limited visual ambiguity. In contrast, categories such as RADAR (25.70 AP_{75}) and SAM (30.60 AP_{75}) show substantially lower performance. We hypothesize that this is partly attributable to high intra-class variation; as shown in Fig. S-3(b).

Multi-angle training. As shown in Tab. 3 (Main), the model trained with all view angles ("All") achieves the highest overall performance compared to single-angle training. However, one may argue that this improvement simply stems from the increased amount of training data, since each single-angle model only sees one-sixth of the total samples. To disentangle the effect of angular diversity from that of data quantity, we conduct an additional experiment as illustrated in Fig. S-4.

- ***Data quantity (or scene diversity) is the dominant factor for overall performance gain***. As the total data quantity decreases, the overall test mAP gradually drops, while the AP variance across look angles increases.
- ***Yet, when controlling for data quantity, we find that angular diversity itself plays a crucial role in enhancing generalization***, leading to both higher mAP and lower performance variance across look angles.

§C Real-world Domain Adaptation

§C.1 Preliminary Background

Though our synthetic data offers scalable and precisely annotated samples, the visual characteristics of synthetic images significantly differ from real-world data in aspects such as color tone, texture, and background complexity [62]. This discrepancy is generally referred to as *domain gap*. Even in real-world scenarios, domain gaps can still naturally arise due to varying environmental factors, sensors, and imaging modalities [8, 65]. Note that the thermal version of our AMOD utilizes the Arma 3 TI engine to generate white-hot thermal imagery. If a specific spectrum like MWIR is required, users may convert the output using image-to-image translation methods.

§C.2 Overall pipeline

To bridge this gap, we adopt a pipeline consisting of domain transfer, pretraining, and finetuning a detection model, as shown in Fig. S-9. We employ UNIT [45] to transform our synthetic AMOD images ($\mathcal{X}_S$) into photo-realistic ones that resemble the target domain ($\mathcal{X}_T$). Both domains $\mathcal{X}_S$ and $\mathcal{X}_T$ are embedded into a shared latent space ($\mathcal{Z}$), from which visually realistic and target-oriented images are augmented[15], as illustrated in Figs. S-5 and S-6.

After translating AMOD images, we pretrain the object detector on these samples to learn domain-invariant features that generalize well across synthetic and target real data. Subsequently, we finetune the pretrained detector using labeled real-world data to adapt to the final target distribution.

[14]We automatically filter objects as fully occluded if their direct line-of-sight intersects with other scene elements.

[15]The embedding in $\mathcal{Z}$ is fed into the generator $G_{S\to T}$ which transfers it from $\mathcal{X}_S$ to $\mathcal{X}_T$, producing a synthetic image in the target style. Both $G_{S\to T}$ and $G_{T\to S}$ are trained, yet we only utilize $G_{S\to T}$ of the UNIT.

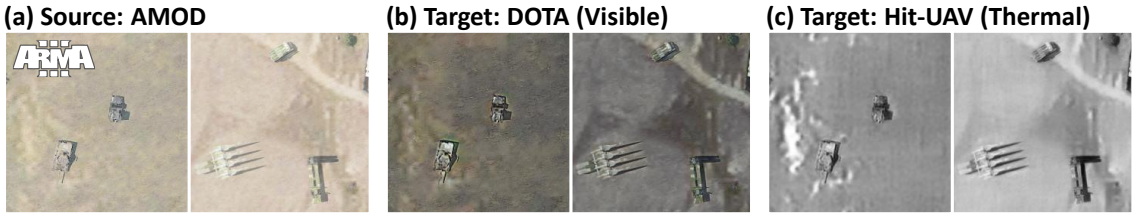


Figure S-5: Examples of translated synthetic image from AMOD using UNIT.

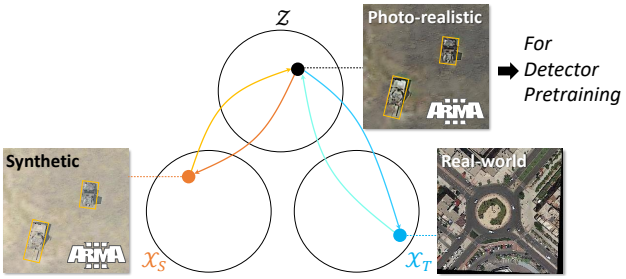


Figure S-6: Illustration of the shared latent space ($\mathcal{Z}$) for domain adaptation between our synthetic ($\mathcal{X}_S$) and target real-world ($\mathcal{X}_T$) aerial image domains, inspired by UNIT [45]. Both domains are mapped to $\mathcal{Z}$, from which Arma3-rendered photo-realistic images are generated as intermediate samples to bridge $\mathcal{X}_S$ and $\mathcal{X}_T$; see Fig. S-5 for translated synthetic image examples (from $\mathcal{Z}$) of AMOD.

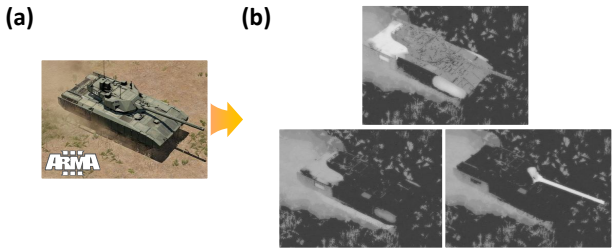


Figure S-7: Example of visible-to-thermal variations. (a) shows the same visible image, while (b) presents multiple thermal renderings produced under varying ambient and object heat conditions in Arma3, with the time condition kept unchanged. It illustrates that thermal-modality translation is inherently ill-posed, as an identical optical input can correspond to diverse thermal outcomes.

§C.3 Stochastic Boundary Smoothing (SBS): a thermal-specific augmentation trick for AMOD-pretraining

As illustrated in Fig. S-7, the process of translation into thermal-spectrum is inherently *ill-posed*. Consequently, this process is fundamentally a one-to-many translation, which no deterministic model can fully capture. For instance, though UNIT [45] we adopt produces visually coherent outputs, its deterministic nature results in limited representations that cannot reflect the diverse thermal variations encountered in real-world sensing. Unfortunately, existing studies [10, 23, 36, 40, 49, 50, 70] largely treat such translation as a deterministic mapping. A straightforward remedy might be stochastic image translation [30, 35, 38, 71]. However, this direction often introduces cumbersome network architectures or complicated training procedures like multi-step sampling.

To address this challenge, we avoid modifying the translator itself and introduce stochasticity through augmentation instead. Specifically, we apply perturbations to the pseudo-thermal translated images during detector pretraining. This allows us to rely on simpler deterministic translators while still expanding thermal variability.

To design a meaningful perturbation strategy, we take into account that thermal sensors suppress inner textures while preserving sharp boundary gradients, yet those boundaries can still become blurred due to weak thermal contrast or sensor smoothing [13, 25]. Motivated by this fact, we introduce ***stochastic boundary smoothing*** (SBS), a simple but effective augmentation strategy during AMOD-pretraining for thermal-infrared detection, applied to each object at each iteration. Consequently, selectively smoothing contour mitigates the overly crisp edges in synthetic sources, effectively shrinking the domain gap, while improving thermal variability.

Let f denote the transform applied to the image in each iteration during pretraining. In Fig. S-9, f is basically an identity mapping. However, for SBS, we replace f with:

$$f(\mathbf{X}) = \begin{cases} \mathbf{X}, & z = 0, \\ \mathbf{X} \odot (1 - \mathbf{C}) + G_\sigma(\mathbf{X}) \odot \mathbf{C}, & z = 1, \end{cases} \tag{2}$$

where $\mathbf{X}$ is a translated image (size: $w \times h$) and z is drawn from a Bernoulli distribution with a probability of p. $\odot$ denotes the Hadamard product. G_σ represents a 2D Gaussian blur operator parameterized by the standard deviation σ. $\mathbf{C}$ is the aggregated mask of contours over all instances as:

$$\mathbf{C} = \bigvee_i \Gamma(\mathbf{M}_i; \tau_i), \tag{3}$$

where $\bigvee$ denotes a pixel-wise logical OR operator applied over all contour masks. $\mathbf{M}_i \in \{0,1\}^{w \times h}$ is a mask of i-th object in $\mathbf{X}$. For $\mathbf{M}_i$, τ_i is a contour thickness randomly drawn from a uniform distribution $\mathcal{U}(\tau_{\min}, \tau_{\max})$. Γ is a contour-band operator that produces a binary band (shape: $\{0,1\}^{w \times h}$) that lies in the boundary of $\mathbf{M}_i$ with τ_i as:

$$\Gamma(\mathbf{M}_i; \tau_i)(x) = \mathbf{1}\left[\left|\mathrm{dist}(x, \partial\mathbf{M}_i)\right| \le \tau_i\right], \tag{4}$$

where x denotes each pixel of $\mathbf{X}$ and $\mathrm{dist}(x, \partial\mathbf{M}_i) = \min_{y \in \partial\mathbf{M}_i} ||x - y||_2$. $\partial\mathbf{M}_i$ is a contour of i-th object.

For SBS, each instance-wise mask $\mathbf{M}_i$ is automatically extracted by Segment Anything Model (SAM) [33] with bounding box prompts. Note that while SBS is designed for AMOD-based pretraining, it could potentially be extended to other datasets as well, as shown in Fig. S-8. SBS can be seamlessly incorporated into other datasets as well, as can be seen in Fig. S-8. We begin by extracting object masks from a DIOR-R visible (see Fig. S-8(a)) image using SAM, translate

Table S-2: Evaluation results (AP_{50}) on HIT-UAV. Effect of the SBS perturbation (f: SBS) under AMOD pretraining.

Pretrained from		Finetuned for HIT-UAV				
Source Data	Pretraining-specific perturbation	Person	Car	Others	Mean	Δ
ImageNet	–	85.30	81.29	57.12	74.57	–
AMOD	✗	86.59	82.00	58.51	75.70	+1.13
	✓ (f: SBS)	87.68	82.15	61.40	77.08	+2.51

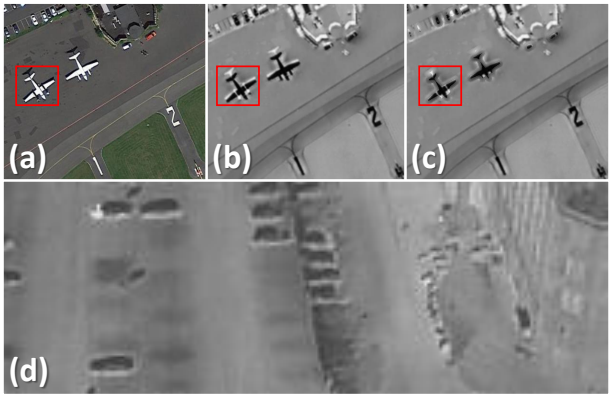


Figure S-8: (a) Example visible image from DIOR-R. (b) thermal-style translation of (a) using a pretrained UNIT model. (c) Result after applying the proposed Stochastic Boundary Smoothing (SBS), where the object boundaries become intentionally blurred to better emulate the characteristic edge attenuation observed in real airborne thermal sensors. (d) Example real thermal image from HIT-UAV.

the image into a thermal-like domain with a pretrained UNIT model (see Fig. S-8(b)), and then apply SBS (see Fig. S-8(c)). The difference between (b) and (c) shows that SBS deliberately smooths object boundaries to replicate the characteristic edge attenuation observed in real airborne thermal sensors (see Fig. S-8(d)). This observation confirms that SBS can serve as a general-purpose augmentation module by boundary smoothing for enhancing thermal-style realism across diverse datasets. We defer this to future work, as the primary focus of this paper is on the AMOD dataset.

Adopting our ***stochastic boundary smoothing*** (SBS) (see Sec. §C.3) during pretraining consistently leads to additional performance gains. For all experiments, $p = 0.5$, $\sigma = 11$, $\tau_{\min} = 2$, $\tau_{\max} = 5$ are used in SBS. The performance reported on the HIT-UAV thermal benchmark in Tab. 4 (Main) is obtained after applying AMOD pretraining, where our SBS method is incorporated during the pretraining stage. A comparison of performance before and after applying SBS is provided in Tab. S-2.

§C.4 Detailed experimental setup for real-world generalization

Pretraining setup. We follow the same training configuration described in Sec. §B.4, except for the training epochs. The detector is pretrained for 5 epochs. Before pretraining, AMOD is translated into the corresponding target-domain style.

Finetuning setup. After pretraining, only the backbone weights are transferred to the downstream detector. The detection head is **re-initialized** to match the number of classes of the target dataset

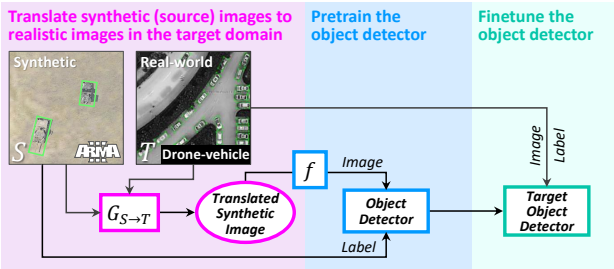


Figure S-9: Overall domain adaptation pipeline used in this work. Here, $\mathcal{X}_T$ is a proxy target-style domain used for translation during pretraining, which may differ from the final downstream benchmark used for finetuning/evaluation.

(e.g., DIOR-R, DroneVehicle, HIT-UAV). Finetuning is conducted for 25 epochs, with all optimizer parameters (learning rate, momentum, weight decay, LR schedule, and warm-up) based on the conventional training protocol used for each target dataset.