A. Experimental Setup

Dataset As we described in §4.1, we construct a custom dataset using drones equipped with EO cameras. The dataset contains videos with synchronized images and metadata collected from altitudes between 5 m and 550 m. Each video is recorded at 30 frames per second (fps) with 1280×720 resolution. We allocate 240,045 images (73%) for training, 46,818 images (14%) for validation, and 40,160 images (12%) for testing. Frames from the same video remain in the same subset to prevent data leakage. From the training set, we sample 1 out of every 10 frames to reduce redundancy among adjacent frames. Table 6 summarizes the dataset statistics.

Experimental Environment We evaluate performance with the standard COCO metrics [18]. These metrics include Average Precision (AP) and its variants. $\mathbf{AP}^{\text{val}}_{50:95}$ and $\mathbf{AP}^{\text{test}}_{50:95}$ measure average precision across IoU thresholds from 0.50 to 0.95 in steps of 0.05 on both the validation and test datasets, respectively. We also report $\mathbf{AP}^{\text{test}}_{50}$ and $\mathbf{AP}^{\text{test}}_{75}$ as IoU-specific scores, and $\mathbf{AP}^{\text{test}}_{S}$, $\mathbf{AP}^{\text{test}}_{M}$, and $\mathbf{AP}^{\text{test}}_{L}$ as scale-specific scores on the test dataset.

Implementation Details We start from YOLOX [9] pretrained on the COCO dataset [18], and fine-tune it on our dataset. We train META-YOLO with the SGD optimizer using eight NVIDIA A100 GPUs (40GB) with a batch size of 64 per GPU. We set the basic learning rate to 0.01 and the weight decay to 0.0005. We apply the cosine learning rate schedule of [9] and the exponential moving average (EMA) with ema_decay of 0.999. During training, we apply HSV augmentation and random flip operations. The main hyperparameters of META-YOLO are listed in Table 5 (refer to META-YOLO-Tiny for detailed configuration).

Comparison Model Settings For all baselines, the input resolution is fixed to 1280×768. To satisfy the 2^n resolution constraint of modern detectors, we apply minimal padding along the height dimension. All comparison models are trained using the MMYOLO framework [4] and initialized from the COCO-pretrained weights provided by the corresponding implementations. We adopt the default training strategy of MMYOLO, including the number of epochs, learning rate schedule, and data augmentations.

Item	Value
input size	(1280, 720)
activation function	silu
depth	0.33
width	0.375
scheduler	SGD
basic learning rate	0.01
weight decay	0.0005
cosine learning rate schedule	True
momentum	0.9
ema decay	0.999
flip probability	0.5
maximum epoch	100
test confidence	0.001
nms threshold	0.65
number of metadata	7
metadata strength	2

Table 5. Main Hyperparameters of META-YOLO-Tiny

B. Comparison with High-Capacity Models

To evaluate the scalability of our approach beyond the lightweight regime, we extend our experiments to large-capacity models. Specifically, we compare META-YOLO-L and META-YOLO-X with their YOLOX counterparts, other state-of-the-art YOLO variants, and prominent two-stage detectors such as Faster R-CNN and Sparse R-CNN. The results are summarized in Table 7.

META-YOLO maintains consistent improvements over the YOLOX baselines, even at a large scale. For instance, META-YOLO-L achieves 68.4 $\text{AP}^{Test}_{50:95}$, a notable +2.2 point gain over YOLOX-L. Similarly, META-YOLO-X reaches 68.9 $\text{AP}^{Test}_{50:95}$, outperforming YOLOX-X by +1.8 points. These improvements also extend to finer metrics; for example, Meta-YOLO-L shows enhanced performance in AP_{75} (79.5 vs. 77.4) and AP_{S} (50.8 vs. 47.1), indicating that our approach continues to benefit localization precision and small-object detection in high-capacity settings.

In comparison with other advanced detectors, META-YOLO demonstrates strong generalization and competitive performance on the test set. While models like YOLOv8-L show higher accuracy on the validation set, META-YOLO-L ultimately achieves superior performance on the test set with 68.4$\text{AP}^{Test}_{50:95}$ compared to YOLOv8-L's 67.9, and does so with slightly fewer GFLOPs (191.6 vs. 198.0). Furthermore, our model significantly outperforms other efficient detectors like PP-YOLOE, with META-YOLO-L surpassing PP-YOLOE-L by +1.6 points. A similar trend is observed in the XLarge scale, where META-YOLO-X substantially exceeds PP-YOLOE-X by +3.4 points. While YOLOv8-X holds a slight edge in overall $\text{AP}^{Test}_{50:95}$, our META-YOLO-X excels in crucial metrics, achieving a state-of-the-art 88.1 AP^{Test}_{50} and demonstrating better performance on small objects AP^{Test}_{S} When compared against two-stage detectors, both META-YOLO-L and META-YOLO-X provide substantially higher accuracy than models like Sparse R-CNN while maintaining the efficiency inherent in one-stage designs.

Overall, these results confirm that the proposed metadata-guided modulation scales robustly to high-capacity models. It is worth noting, however, that the relative performance gains are more pronounced in the lightweight regime. This suggests that the benefits of metadata are most significant when a model's intrinsic representational capacity is constrained, offering a compelling direction for future research on efficient model design.

Split	# Video Sequence	# Image	# Car	# Bus	# Truck	# UV	# TV
Train	44	240,045 (73%)	992,518 (75%)	12,041 (77%)	371,162 (79%)	91,290 (77%)	91,348 (74%)
Valid	9	46,818 (14%)	145,488 (11%)	1,929 (12%)	49,514 (10%)	14,597 (12%)	16,145 (13%)
Test	9	40,160 (12%)	192,896 (14%)	1,672 (11%)	51,359 (11%)	12,784 (11%)	15,734 (13%)
Total	62	327,023	1,330,902	15,642	472,035	118,671	123,227

Table 6. Statistics of the dataset. The dataset consists of 62 video sequences, with 327,023 frames across 5 categories: car, bus, truck, utility vehicle, and transport vehicle. We split the dataset at the video-level and sample 10% of frames for training after the split.

Model	#Epochs	#Params (M)	GFLOPs	AP$^{Val}_{50:95}$	AP$^{Val}_{50}$	AP$^{Test}_{50:95}$	AP$^{Test}_{50}$	AP$^{Test}_{75}$	AP$^{Test}_{S}$	AP$^{Test}_{M}$	AP$^{Test}_{L}$
Two-stage Detectors											
Faster R-CNN (R50) [25]	24	41.4	191.0	57.2	79.9	57.1	79.5	67.3	36.3	65.0	85.5
Faster R-CNN (R101) [25]	24	60.4	261.0	55.7	78.4	52.3	73.4	62.0	32.1	60.1	85.8
Sparse R-CNN (R50) [29]	36	106.0	158.0	**58.1**	80.6	**60.2**	**84.7**	**70.7**	**39.8**	66.9	87.0
Sparse R-CNN (R101) [29]	36	125.0	227.0	57.9	**80.7**	60.0	84.5	70.4	37.3	**67.2**	**87.9**
Large-sized Models											
YOLOX-L [9]	300	54.2	186.0	64.9	84.6	66.2	84.7	77.4	47.1	72.3	85.1
YOLOv7-L [32]	300	36.5	124.0	38.0	66.0	44.1	71.7	49.8	25.9	50.0	56.7
YOLOv8-L [10]	500	43.6	198.0	**69.0**	**87.1**	67.9	85.3	79.4	46.5	**74.4**	80.4
PP-YOLOE-L [40]	80	51.3	129.0	65.9	85.5	66.8	87.4	78.5	43.0	73.7	89.2
Meta-YOLO-L	100	55.0	191.6	66.3	85.7	**68.4**	**87.6**	**79.5**	**50.8**	74.0	**90.3**
XLarge-sized Models											
YOLOX-X [9]	300	99.0	338.0	66.0	85.7	67.1	85.1	78.9	47.6	73.4	**90.3**
YOLOv7-X [32]	300	70.8	226.0	42.0	64.9	47.9	71.4	55.9	27.0	54.9	57.8
YOLOv8-X [10]	500	68.2	309.0	**69.2**	**86.8**	**69.2**	86.7	**80.4**	50.5	**74.9**	84.3
PP-YOLOE-X [40]	80	97.3	244.0	64.2	82.2	65.5	84.0	76.9	45.4	71.8	87.4
Meta-YOLO-X	100	100.4	346.3	67.7	86.3	68.9	**88.1**	80.2	**51.0**	74.4	88.6

Table 7. Comparison of our proposed META-YOLO with large-capacity detectors. We report results of META-YOLO-L and META-YOLO-X against YOLOX, YOLOv7, YOLOv8, PP-YOLOE, and two-stage counterparts.