CVGlobal

Leonardo Ruso
leonardo.russo@inria.fr

Diego Marcos
diego.marcos@inria.fr

INRIA
Evergreen Team
Montpellier, France

Abstract

This document demonstrates the format requirements for papers submitted to the British Machine Vision Conference. The format is designed for easy on-screen reading, and to print well at one or two pages per sheet. Additional features include: pop-up annotations for citations [1, 11]; a margin ruler for reviewing; and a greatly simplified way of entering multiple authors and institutions.

All authors are encouraged to read this document, even if you have written many papers before. As well as a description of the format, the document contains many instructions relating to formatting problems and errors that are common even in the work of authors who *have* written many papers before.

1 Introduction

...

2 Related Works

2.1 Cross-View Geo-localization

Cross-view geo-localization has emerged as a fundamental computer vision task that addresses the challenge of matching images captured from different viewpoints of the same geographic location. Early works in this domain primarily focused on matching ground-level street-view images with aerial or satellite imagery [9, 17]. These pioneering approaches established the foundation for understanding the geometric and semantic relationships between cross-view image pairs.

Recent advances have leveraged deep learning architectures to learn robust cross-view representations. Hu et al. [8] introduced the Cross-View Matching Network (CVM-Net), which uses a dual-stream CNN architecture with polar transform to align aerial and ground images. Building upon this foundation, Shi et al. [14] proposed spatial-aware feature aggregation to capture local spatial relationships, while Regmi and Borji [13] developed a comprehensive review highlighting the key challenges in cross-view matching.

The VIGOR dataset [19] represents a significant milestone in cross-view research, providing same-area and cross-area evaluation protocols that better reflect real-world deployment scenarios. This dataset has become a standard benchmark for evaluating cross-view geo-localization methods, offering panoramic ground images paired with aerial satellite views across multiple cities.

2.2 Vision Transformers in Cross-View Tasks

The advent of Vision Transformers (ViTs) [5] has revolutionized computer vision, including cross-view geo-localization. Unlike CNNs, ViTs can capture long-range dependencies through self-attention mechanisms, making them particularly suitable for understanding global spatial relationships in cross-view scenarios.

Several recent works have explored ViT-based architectures for cross-view matching. Zhu et al. [20] investigated the application of standard ViTs to cross-view tasks, demonstrating improved performance over CNN-based methods. Toker et al. [16] showed that transformer architectures can effectively handle the viewpoint variations inherent in cross-view matching problems.

However, these approaches primarily rely on supervised learning with orientation labels, requiring extensive annotation efforts. Our work distinguishes itself by leveraging pretrained features from self-supervised models, reducing the dependency on labeled orientation data.

2.3 Self-Supervised Visual Representation Learning

Self-supervised learning has gained significant traction as a paradigm for learning robust visual representations without manual annotations. Methods like MoCo [7], SimCLR [4], and BYOL [6] have demonstrated that self-supervised features can match or exceed supervised counterparts on various downstream tasks.

DINOv2 [12] represents the state-of-the-art in self-supervised vision models, combining the discriminative power of ViTs with robust pre-training on large-scale image collections. Unlike its predecessor DINO [3], DINOv2 provides features that are particularly well-suited for dense prediction tasks and fine-grained visual understanding.

Recent works have begun exploring the application of self-supervised features to geo-localization tasks. However, these approaches typically treat cross-view matching as a standard retrieval problem without considering the specific geometric constraints and orientation relationships inherent in cross-view scenarios.

2.4 Orientation Estimation and Spatial Alignment

Orientation estimation in cross-view scenarios has been addressed through various approaches. Traditional methods relied on handcrafted features and geometric constraints [2, 10]. Recent deep learning approaches have explored end-to-end orientation regression [17] and attention-based alignment mechanisms [15].

Attention mechanisms have proven particularly effective for cross-view alignment. Cross-attention layers can learn to focus on corresponding regions between aerial and ground views, enabling more accurate spatial correspondence. However, most existing attention-based methods require supervised training with orientation labels.

Our approach introduces novel orientation-aware token aggregation strategies that exploit the inherent spatial structure of transformer features. By aggregating tokens along meaningful directions (vertical columns for ground images and radial directions for aerial images), we can estimate orientation through unsupervised feature alignment.

2.5 Sky Segmentation and Depth Estimation

Incorporating semantic and geometric priors has shown promise in improving cross-view matching performance. Sky segmentation helps eliminate uninformative regions in ground-level images, focusing attention on building structures and terrain features that are visible in aerial views [17].

Recent advances in monocular depth estimation, particularly with models like Depth-Anything [18], provide robust depth cues that can inform cross-view alignment. Depth information enables multi-scale feature aggregation, where closer objects receive different weighting compared to distant features.

Our work is the first to systematically combine sky filtering, depth estimation, and self-supervised feature learning in a unified framework for cross-view orientation estimation. The integration of these complementary signals enables more robust and accurate orientation prediction.

2.6 Benchmark Datasets

Several datasets have been developed to support cross-view geo-localization research. The CVUSA dataset [17] provides aligned street-view and aerial image pairs across the United States, serving as an early benchmark for the field. The more recent VIGOR dataset [19] offers additional challenges with its same-area and cross-area evaluation protocols.

While these datasets focus primarily on geo-localization and retrieval tasks, there is growing interest in orientation estimation as a related but distinct problem. Our work addresses this gap by demonstrating effective orientation estimation on existing benchmarks while introducing novel evaluation metrics for orientation accuracy and confidence estimation.

The development of large-scale, geographically diverse datasets like CVGlobal further advances the field by providing more comprehensive evaluation scenarios across different geographical regions and environmental conditions.

3 Dataset Generation Method

In this section, we present our methodology for constructing CVGlobal, a large-scale multi-modal dataset that pairs satellite and street-view imagery across diverse global regions. Our approach systematically samples locations from five continents while ensuring geographical diversity and balanced representation between urban and rural environments.

3.1 Dataset Design and Sampling Strategy

Our dataset construction methodology is guided by three key principles: *geographical diversity*, *balanced representation*, and *multi-modal consistency*. We define sampling regions across five major continents (North America, Europe, Asia, South America, and Africa), with each continent contributing equally to the final dataset to prevent geographical bias.

For each continent, we establish two distinct sampling regions:

- **Urban regions**: Areas with high population density and significant urban infrastructure

Table 1: Geographical sampling regions defined for each continent and environment type.

Continent	Type	Location	Lat Range	Lon Range
North America	Urban	New York City	40.71°–40.81°N	74.01°–73.91°W
	Rural	California Farmland	36.78°–36.88°N	119.42°–119.32°W
Europe	Urban	Paris	48.86°–48.96°N	2.35°–2.45°E
	Rural	French Countryside	46.23°–46.33°N	2.21°–2.31°E
Asia	Urban	Tokyo	35.69°–35.79°N	139.69°–139.79°E
	Rural	Rural India (Agra)	27.18°–27.28°N	78.04°–78.14°E
South America	Urban	São Paulo	23.55°–23.45°S	46.63°–46.53°W
	Rural	Brazilian Rainforest	14.24°–14.13°S	51.93°–51.83°W
Africa	Urban	Nairobi	1.29°–1.19°S	36.82°–36.92°E
	Rural	Kenyan Savanna	2.15°–2.05°S	37.31°–37.41°E

- **Rural regions**: Areas with low population density and predominantly natural or agricultural landscapes

The sampling regions are carefully selected to represent diverse climatic, cultural, and developmental contexts within each continent. Table 1 details the specific geographical boundaries for each region.

3.2 Urban-Rural Classification

To ensure accurate labeling of locations as urban or rural, we employ the Global Urban Areas dataset [?], which provides comprehensive polygon boundaries for urban areas worldwide. For each randomly generated coordinate, we perform a spatial intersection test to determine its classification:

$$\text{Urban}(p) = \begin{cases} 1 & \text{if } p \in \bigcup_i U_i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where p represents a coordinate point and U_i denotes the i-th urban area polygon from the Global Urban Areas dataset. This automated classification ensures consistent and objective urban-rural labeling across all geographical regions.

3.3 Multi-Modal Data Acquisition

Our data acquisition pipeline consists of three main components: *coordinate generation*, *outdoor location validation*, and *multi-modal image retrieval*.

3.3.1 Coordinate Generation and Validation

For each sampling region, we generate random coordinates within the specified geographical boundaries using uniform sampling. Each coordinate undergoes a validation process to ensure data quality:

Algorithm 1 Outdoor Location Detection

```
Require: Street View metadata $M$, Google Maps client $G$
Ensure: Boolean indicating outdoor location
if $M$.status $\neq$ OK then
    return False
end if
if $M$.place_id is undefined then
    return True                                  ▷ Assume outdoor for street-level locations
end if
details $\leftarrow$ $G$.place($M$.place_id)
types $\leftarrow$ details.result.types
indoor_types $\leftarrow$ $\{$shopping_mall, store, restaurant, hospital, $\ldots\}$
if types $\cap$ indoor_types $\neq \emptyset$ then
    return False
else
    return True
end if
```

1. **Urban-Rural Consistency**: Verify that the generated coordinate matches the intended environment type (urban/rural) using the spatial intersection described above.

2. **Street View Availability**: Query the Google Street View Metadata API to confirm image availability at the location.

3. **Outdoor Location Filtering**: Apply our outdoor detection algorithm to exclude indoor environments.

3.3.2 Outdoor Detection Algorithm

To ensure our dataset captures genuine outdoor environments, we implement a robust filtering mechanism that leverages Google Places API data. Our algorithm evaluates each location using the following criteria:

This approach is more nuanced than simple keyword filtering, as it distinguishes between genuinely indoor locations (e.g., shopping malls, restaurants) and outdoor points of interest (e.g., parks, monuments) that may also carry establishment tags.

3.3.3 Image Acquisition and Processing

For each validated coordinate, we acquire two types of imagery:

Satellite Imagery We retrieve high-resolution satellite images using the Google Static Maps API with the following specifications:

- Resolution: 640x640 pixels
- Zoom level: 18 (approximately 1.19 meters/pixel)
- Map type: Satellite view
- Format: JPEG

Street View Imagery We collect street-view images from four cardinal directions (0°, 90°, 180°, 270°) to provide comprehensive ground-level perspective. Each image has:

- Resolution: 640x640 pixels
- Field of view: Default Google Street View settings
- Format: JPEG

The four directional images are horizontally concatenated to create a panoramic representation, resulting in a 2560x640 pixel stitched image that captures the complete ground-level environment.

3.4 Quality Assurance and Error Handling

Our data acquisition pipeline implements robust error handling and quality assurance mechanisms:

3.4.1 Network Resilience

We employ an exponential backoff retry strategy for API requests, with up to 3 retry attempts for failed connections. This approach handles temporary network issues and API rate limiting gracefully.

3.4.2 Coordinate Correction

The Google Street View API may return imagery from coordinates slightly different from the requested location due to road network constraints. We handle this by:

1. Recording both original and corrected coordinates
2. Using corrected coordinates for file naming and deduplication
3. Ensuring consistent satellite-street view pairing

3.4.3 Resume Capability

Our pipeline supports interruption and resumption, checking for existing complete image sets before processing each location. A complete set consists of:

- One satellite image
- Four directional street view images (0°, 90°, 180°, 270°)
- One stitched panoramic image

3.5 Dataset Statistics and Validation

Throughout the data collection process, we maintain comprehensive statistics including:

- Success and failure rates per continent and environment type
- API call counts and timing information
- Error categorization (metadata failures, download failures, indoor rejections)
- Coordinate generation efficiency metrics

These statistics are automatically compiled into detailed reports (both human-readable and machine-readable formats) that facilitate dataset validation and quality assessment.

The resulting CVGlobal dataset provides a balanced, geographically diverse collection of paired satellite and street-view imagery suitable for training and evaluating computer vision models across varied global contexts.

4 Crossview Method

We propose CroDINO, a novel approach for cross-view orientation estimation that leverages the robust feature representations of DINOv2 with orientation-aware token aggregation strategies. Our method addresses the fundamental challenge of aligning ground-level panoramic images with aerial satellite views by exploiting both spatial structure and depth information.

4.1 Problem Formulation

Given a ground-level panoramic image I_g and an aerial satellite image I_a of the same geographic location, our goal is to estimate the relative orientation θ between the two views. The ground image is extracted from a 360° panorama using a field-of-view (FOV) window defined by parameters (f_x, f_y, ψ, ϕ), where f_x and f_y represent the horizontal and vertical FOV angles, ψ is the yaw (rotation around the vertical axis), and ϕ is the pitch (elevation angle).

4.2 Architecture Overview

4.2.1 Dual-Stream Feature Extraction

CroDINO builds upon the DINOv2 Vision Transformer architecture, which provides rich, self-supervised feature representations. We modify the standard DINOv2 model to process both ground and aerial images simultaneously while maintaining separate positional embeddings and class tokens for each modality.

The model consists of:

- **Shared Backbone**: We utilize the pre-trained DINOv2-ViT-B/14 as our feature extractor, freezing its parameters to preserve the learned representations.
- **Dual Positional Embeddings**: Separate positional embeddings $\mathbf{E}^g_{pos}$ and $\mathbf{E}^a_{pos}$ for ground and aerial images to account for their different spatial characteristics.
- **Cross-Modal Attention**: A final single-head attention layer that enables interaction between ground and aerial features.

4.2.2 Token Processing Pipeline

For each input image pair, the model generates patch embeddings of size 14 × 14 pixels, resulting in a 16 × 16 grid of 768-dimensional feature vectors. The forward pass can be formulated as:

$$\mathbf{F}_g = \text{DINOv2}(I_g; \mathbf{E}^g_{pos}) \quad (2)$$

$$\mathbf{F}_a = \text{DINOv2}(I_a; \mathbf{E}^a_{pos}) \quad (3)$$

$$\mathbf{F}_{combined} = \text{Attention}([\mathbf{F}_g; \mathbf{F}_a]) \quad (4)$$

where $\mathbf{F}_g, \mathbf{F}_a \in \mathbb{R}^{16\times16\times768}$ are the ground and aerial feature matrices, respectively.

4.3 Orientation-Aware Token Aggregation

4.3.1 Sky Filtering and Depth Estimation

To improve orientation estimation, we incorporate semantic and geometric priors:

Sky Segmentation: We employ a lightweight CNN-based sky filter to identify and mask sky regions in ground images. The sky mask M_{sky} is computed at the patch level using majority voting within each 16 × 16 grid cell.

Depth Estimation: We utilize the Depth-Anything model to generate depth maps D for ground images. The depth information is downsampled to match the patch grid, providing depth values $d_{i,j}$ for each spatial location (i, j).

4.3.2 Multi-Layer Token Aggregation

We introduce a novel aggregation strategy that separates tokens into three depth layers: foreground, middleground, and background. This approach captures the multi-scale nature of visual features in cross-view matching.

Vertical Column Analysis: For each vertical column j in the ground image feature grid, we compute depth-weighted averages:

$$\mathbf{t}_j^{fore} = \frac{\sum_i w_i^{fore} \cdot \mathbf{f}^g_{i,j} \cdot M_{sky}(i,j)}{\sum_i w_i^{fore} \cdot M_{sky}(i,j)} \quad (5)$$

$$\mathbf{t}_j^{mid} = \frac{\sum_i w_i^{mid} \cdot \mathbf{f}^g_{i,j} \cdot M_{sky}(i,j)}{\sum_i w_i^{mid} \cdot M_{sky}(i,j)} \quad (6)$$

$$\mathbf{t}_j^{back} = \frac{\sum_i w_i^{back} \cdot \mathbf{f}^g_{i,j} \cdot M_{sky}(i,j)}{\sum_i w_i^{back} \cdot M_{sky}(i,j)} \quad (7)$$

where the depth-dependent weights are defined as:

$$w_i^{fore} = d_{i,j} \quad (8)$$

$$w_i^{mid} = \begin{cases} \frac{d_{i,j}}{\tau} & \text{if } d_{i,j} \leq 0.5 \\ \frac{1-d_{i,j}}{d_{i,j}} & \text{otherwise} \end{cases} \quad (9)$$

$$w_i^{back} = 1 - d_{i,j} \quad (10)$$

with threshold $\tau = 0.5$.

Radial Direction Analysis: For aerial images, we extract features along radial directions from the center:

$$\mathbf{r}_{\beta}^{fore} = \frac{\sum_r w_r^{fore} \cdot \mathbf{f}^{a}_{\beta,r}}{\sum_r w_r^{fore}} \tag{11}$$

$$\mathbf{r}_{\beta}^{mid} = \frac{\sum_r w_r^{mid} \cdot \mathbf{f}^{a}_{\beta,r}}{\sum_r w_r^{mid}} \tag{12}$$

$$\mathbf{r}_{\beta}^{back} = \frac{\sum_r w_r^{back} \cdot \mathbf{f}^{a}_{\beta,r}}{\sum_r w_r^{back}} \tag{13}$$

where β represents the angle direction, r is the radial distance, and the weights follow a linear progression: $w_r^{fore} = 1 - r/R$, w_r^{mid} follows a triangular pattern, and $w_r^{back} = r/R$.

4.4 Orientation Estimation

4.4.1 Cross-Modal Alignment

We estimate orientation by finding the angular offset that minimizes the cosine distance between corresponding vertical and radial feature aggregations. For each candidate orientation θ, we compute:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=0}^{N-1} \left\| 1 - \begin{bmatrix} \mathbf{t}^{fore}_{N-1-i} \\ \mathbf{t}^{mid}_{N-1-i} \\ \mathbf{t}^{back}_{N-1-i} \end{bmatrix}^{T} \begin{bmatrix} \mathbf{r}^{fore}_{\phi(\theta,i)} \\ \mathbf{r}^{mid}_{\phi(\theta,i)} \\ \mathbf{r}^{back}_{\phi(\theta,i)} \end{bmatrix} \right\| \tag{14}$$

where $\phi(\theta,i) = (\lfloor \theta/\Delta\theta \rfloor + i - N/2) \bmod |\mathcal{R}|$ maps vertical columns to radial directions, and $\Delta\theta$ is the angular step size.

4.4.2 Confidence Estimation

To assess the reliability of orientation estimates, we compute a confidence score based on the Z-score of the minimum distance:

$$\text{confidence} = \frac{\mu(\mathcal{L}) - \min(\mathcal{L})}{\sigma(\mathcal{L})} \tag{15}$$

where $\mu(\mathcal{L})$ and $\sigma(\mathcal{L})$ are the mean and standard deviation of the loss values across all candidate orientations.

4.5 Training Strategy

Our approach operates in a largely unsupervised manner, leveraging the pre-trained DINOv2 features without requiring orientation labels during training. The model learns to align cross-view features through the geometric constraints imposed by the aggregation strategy and the cosine similarity objective.

4.5.1 Data Preprocessing

We extract random FOV windows from panoramic images with parameters:

- Horizontal FOV: 90°
- Vertical FOV: 180°
- Random yaw: $\psi \sim \text{Uniform}(0^\circ, 360^\circ)$
- Fixed pitch: $\phi = 90^\circ$

Aerial images undergo center cropping and optional polar transformation to align with the ground view geometry.

4.6 Implementation Details

The complete pipeline processes image pairs through the following stages:

1. Feature extraction using frozen DINOv2-ViT-B/14
2. Sky segmentation using guided filter refinement
3. Depth estimation with Depth-Anything model
4. Multi-layer token aggregation with depth weighting
5. Cross-modal orientation search with cosine similarity

All models are implemented in PyTorch and can operate on both CPU and GPU. The orientation search space is discretized with angular steps of $\Delta\theta = 90/16 = 5.625$ for a 16×16 patch grid.

References

[1] Authors. The frobnicatable foo filter, 2006. ECCV06 submission ID 324. Supplied as additional material `eccv06.pdf`.

[2] Mayank Bansal, Kostas Daniilidis, and Harpreet Sawhney. Ultra-wide baseline facade matching for geo-localization. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, pages 15–22, 2011.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[6] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[8] Sixing Hu, Mengdan Feng, Rang MH Nguyen, and Gim Hee Lee. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7258–7267, 2018.

[9] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Cross-view image geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2013.

[10] Tsung-Yi Lin, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5007–5015, 2015.

[11] N. David Mermin. What's wrong with these equations? *Physics Today*, October 1989. http://www.cvpr.org/doc/mermin.pdf.

[12] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[13] Krishna Regmi and Ali Borji. Cross-view image synthesis using conditional gans. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3501–3510, 2018.

[14] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. Spatial-aware feature aggregation for image based cross-view geo-localization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[15] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. Where am i looking at? joint location and orientation estimation by cross-view matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4064–4072, 2020.

[16] Aysim Toker, Qunjie Zhou, Maxim Maximov, and Laura Leal-Taixé. Coming down to earth: Satellite-to-street view synthesis for geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6488–6497, 2021.

[17] Scott Workman, Richard Souvenir, and Nathan Jacobs. On the location dependence of convolutional neural network features. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 70–78, 2015.

[18] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[19] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2021.

[20] Sijie Zhu, Taojiannan Yang, and Chen Chen. Revisiting cross-view geo-localization in the wild with transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8640–8649, 2023.