

Article

Trustworthy Celestial Eye: Calibrated and Robust Planetary Classification via Self-Supervised Vision Transformers

Ziqiang Xu ¹, Young Choi ^{2,*}, Changyong Yi ^{3,*}, Chanjeong Park ³, Jinyoung Park ³, Hyungkeun Park ³ and Sujeen Song ²

¹ Department of Robot and Smart System Engineering, Kyungpook National University, 80, Daehak-ro, Buk-gu, Daegu 41566, Republic of Korea; xzq@knu.ac.kr

² Earth Turbine, 36, Dongdeok-ro 40-gil, Jung-gu, Daegu 41905, Republic of Korea; songsujeen69@gmail.com

³ Intelligent Construction Automation Center, Kyungpook National University, 80 Daehak-ro, Buk-gu, Daegu 41566, Republic of Korea; tpms1018@knu.ac.kr (C.P.); jinyoungpark@knu.ac.kr (J.P.); parkhk@chungbuk.ac.kr (H.P.)

* Correspondence: youngch5@naver.com (Y.C.); cyyi@knu.ac.kr (C.Y.)

Abstract

Automated recognition of celestial bodies from observational imagery is a cornerstone of autonomous space exploration. However, deploying deep learning models in space environments entails rigorous requirements not only for accuracy but also for reliability (calibration) and safety (anomaly rejection). Traditional Convolutional Neural Networks (CNNs) trained on small-scale astronomical datasets often suffer from overfitting and overconfidence on Out-of-Distribution (OOD) artifacts. In this work, we present a robust classification framework based on DINOv2, a Vision Transformer pre-trained via discriminative self-supervised learning. We curate a high-fidelity dataset of seven planetary classes sourced from NASA archives and propose a two-stage domain adaptation strategy to transfer large-scale foundation model features to this fine-grained task. Extensive experiments show that our method reaches 100% Top-1 accuracy on the canonical split, and remains highly stable under split variation, achieving $99.43\% \pm 0.85\%$ Top-1 accuracy across $R = 5$ repeated stratified splits. More importantly, we address the critical issue of model trustworthiness. Through post hoc temperature scaling, our model achieves a state-of-the-art Expected Calibration Error (ECE) of 0.08%, representing a 36-fold improvement over ResNet50 (2.90%) and a 4.5-fold improvement over the EfficientNet-B3 baseline (0.36%). Furthermore, by integrating Energy-based OOD detection, the system effectively rejects non-planetary artifacts with an AUROC of 93.7%. Qualitative analysis using Grad-CAM reveals that self-supervised attention mechanisms naturally focus on intrinsic planetary features (e.g., surface textures and rings) while ignoring background noise, confirming the superior robustness of vision foundation models in astronomical vision tasks.



Academic Editor: M. Reza Emami

Received: 31 December 2025

Revised: 24 February 2026

Accepted: 24 February 2026

Published: 27 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: celestial body classification; vision foundation models; self-supervised learning; DINOv2; out-of-distribution detection

1. Introduction

1.1. Background and Motivation

The exponential growth of astronomical data from modern sky surveys and space telescopes has created an unprecedented need for automated classification systems capable of identifying and categorizing celestial objects. Accurate classification of astronomical bodies, including stars, galaxies, nebulae, and other cosmic phenomena, is crucial for

understanding the structure and evolution of the universe, discovering rare astronomical events, and advancing theoretical astrophysics [1,2]. Traditional manual classification by expert astronomers is no longer feasible given the scale of modern astronomical datasets, which now contain millions of objects requiring analysis. Deep learning has emerged as a transformative technology for astronomical image classification, with convolutional neural networks (CNNs) demonstrating remarkable success in various computer vision tasks [3,4]. However, applying deep learning to astronomical images presents unique challenges: (1) limited labeled data, as expert annotation is expensive and time-consuming; (2) domain-specific visual features that differ significantly from natural images; (3) class imbalance, where certain celestial phenomena are far rarer than others; and (4) high-stakes decision-making, requiring not only accuracy but also model calibration and uncertainty quantification.

1.2. The Promise of Vision Foundation Models

Recent advances in self-supervised learning have led to the emergence of Vision Foundation Models (VFMs)—large-scale models pre-trained on massive collections of unlabeled images to learn robust and transferable visual representations [5,6]. Models such as DINOv2 [7], Vision Transformers (ViT) [8], and Swin Transformers [9] have demonstrated outstanding transfer learning performance across a wide range of domains, often achieving state-of-the-art results with only minimal fine-tuning. Unlike traditional supervised pre-training on ImageNet, self-supervised VFMs learn rich semantic features without relying on human annotations. This property makes them particularly attractive for specialized scientific domains such as astronomy, where labeled data are limited and costly to obtain. The central hypothesis of our work is that, although VFMs are predominantly pre-trained on natural images, their learned representations can be effectively transferred to astronomical imagery through carefully designed fine-tuning strategies. In particular, the hierarchical and multi-scale features learned by transformer-based architectures may be well-suited to capturing the diverse morphological structures characteristic of celestial objects.

1.3. Research Objectives and Contributions

Our research presents a systematic investigation into the application of Vision Foundation Models for trustworthy celestial body classification. We address the following research questions: 1. How do VFMs compare to traditional CNNs (e.g., ResNet, EfficientNet) in terms of accuracy, calibration, and safety on astronomical images? 2. What is the optimal transfer learning strategy to adapt these large-scale models to small-scale celestial datasets without catastrophic forgetting? Our primary contributions are: Framework Design: We propose a robust two-stage domain adaptation framework (linear probing followed by partial fine-tuning) to effectively adapt DINOv2 to a manually curated, high-fidelity dataset sourced from NASA archives. Comprehensive Benchmarking: We systematically compare state-of-the-art VFMs (DINOv2, Swin, ViT) against traditional CNNs (ResNet50, EfficientNet-B3, ConvNeXt V2) on classification performance. Reliability and Safety Assessment: We go beyond accuracy to evaluate model trustworthiness. We implement post hoc temperature scaling for probability calibration and integrate Energy-based OOD detection to identify non-planetary artifacts sourced from open-world noise.

1.4. Key Findings

Our extensive experiments conducted on an NVIDIA RTX 3090 GPU reveal significant advantages of the proposed framework: SOTA Performance: The DINOv2 model achieves perfect 100% Top-1 accuracy on the test set, saturating the benchmark and outperforming the classic ResNet50 baseline (96.2%). Superior Calibration: In terms of reliability, DINOv2 achieves a new state-of-the-art performance with an Expected Calibration Error (ECE) of

just 0.08%. This represents a 36-fold improvement over ResNet50 (2.90%) and a 4.5-fold improvement over the strong EfficientNet-B3 baseline (0.36%). Robust Safety: For safety-critical OOD detection, our method achieves an AUROC of 93.7%, demonstrating a robust capability to distinguish valid planetary images from space debris and noise, significantly surpassing the legacy CNN baseline (86.8%).

2. Materials and Methods

2.1. Vision Transformers and Foundation Models

The emergence of Vision Transformers has revolutionized computer vision, with recent foundation models showing remarkable capabilities across diverse visual tasks. Oquab et al. [7] proposed DINOv2, a self-supervised vision foundation model trained on 142 million curated natural images that demonstrates strong transfer learning capabilities across various downstream tasks. Experimental results showed superior performance compared to supervised pre-training methods on multiple benchmarks, but the model's effectiveness on scientific imaging domains with distinct statistical properties remains underexplored. Liu et al. [9] introduced the Swin Transformer, which addresses the challenges of adapting Transformers to vision through hierarchical feature representation and shifted window attention mechanisms. The method achieved state-of-the-art performance on object detection and semantic segmentation tasks, demonstrating computational efficiency compared to traditional Vision Transformers, but evaluation was primarily conducted on natural image datasets without consideration of domain-specific challenges in scientific imaging. Liu et al. [10] further extended this work with Swin Transformer V2, scaling up both model capacity and input resolution while maintaining computational efficiency. Results demonstrated improved performance on high-resolution vision tasks, but the scalability benefits have not been systematically evaluated in the context of astronomical imaging, where high-resolution data with complex noise structures are common. Cao et al. [11] introduce FPN-ViT, which augments a Vision Transformer with a Feature Pyramid Network (FPN)-style multi-scale fusion mechanism to better capture galaxy morphology across scales and improve robustness to resolution changes. They evaluate on Galaxy Zoo 2 (GZ2) galaxy morphology classification and report improved performance over a ViT baseline (95.2% accuracy vs. 94.6% for ViT-B/16), with particularly strong results on most morphology classes but weaker performance on the rare cigar-shaped category. Robustness tests under controlled degradations (resizing and synthetic noise) suggest the model remains relatively stable as image scale and quality vary. However, the study is primarily validated on a GZ2-derived setup with simplified perturbations, so its generalization to truly high-resolution, instrument-specific astronomical noise and broader survey/domain shifts remains to be systematically established. Vision Transformers and foundation models have demonstrated strong representation learning capabilities and transfer performance on natural images. However, systematic evaluation of these models on scientific imaging tasks, particularly astronomical data with unique statistical properties and domain-specific challenges, remains limited.

2.2. Deep Learning in Astronomical Image Classification

Deep learning has been increasingly applied to astronomical image analysis, with particular focus on galaxy morphology classification and radio source detection. Cao et al. [12] proposed a Convolutional Vision Transformer (CvT) approach for galaxy morphology classification, combining convolutional operations with self-attention mechanisms. Experimental results on galaxy survey data showed improved classification accuracy compared to pure CNN approaches, but the method's robustness to varying image quality and survey conditions was not thoroughly investigated. Mohan et al. [13] addressed uncertainty

quantification in radio galaxy classification using variational inference techniques with deep learning models. Their approach provided probabilistic predictions for Fanaroff–Riley classification tasks, demonstrating the importance of uncertainty estimation in astronomical applications, but the framework was limited to specific radio galaxy types without broader generalization to other astronomical classification tasks. Rustige et al. [14] address the scarcity of labeled training data for radio-galaxy morphology classification by using a class-conditional Wasserstein GAN (wGAN) to generate synthetic images and employ them as training-time augmentation for multiple classifier families (FCN, CNN, and ViT). Their results indicate that wGAN-supported augmentation can strongly improve simpler models and yield a modest boost for a CNN, but it does not provide a consistent benefit for a Vision Transformer. This suggests that generative augmentation may be most useful in data-scarce, resource-constrained settings with weaker classifiers, while gains can diminish for transformer-based models—potentially due to sensitivity to subtle real–synthetic differences and reliance on natural-image pretraining under limited astronomical fine-tuning data. Bowles et al. [15] propose an NLP-based pipeline (Text2Tag) for Radio Galaxy Zoo EMU (RGZ EMU) that converts plain-English morphological descriptions of radio sources into an interpretable semantic tag taxonomy, aiming to replace rigid, jargon-heavy class labels with flexible feature descriptors. They derive 22 semantic morphology tags and argue that only a smaller subset should be presented to citizen scientists for usability, with other tags potentially assigned algorithmically. However, the authors note practical limitations, including dependence on a small dataset, reliance on pre-trained NLP embeddings that output single-token tags (often requiring manual adjustments), and instrument-specificity (ASKAP/EMU), implying that higher-resolution or more sensitive surveys may require extending the tag set. Deep learning applications in astronomical image classification have shown promising results, particularly for galaxy morphology and radio source classification. However, most approaches rely on domain-specific architectures trained from scratch, with limited exploration of how modern foundation models could be adapted for these specialized tasks.

2.3. Transfer Learning and Domain Adaptation

Transfer learning has emerged as a crucial technique for adapting pre-trained models to scientific imaging domains where labeled data is often scarce. Vilalta et al. [16] provided a comprehensive framework for domain adaptation in astronomical applications, addressing the fundamental challenge of distributional differences between source and target domains. Their approach demonstrated improved performance on spectroscopic classification tasks, but the framework predates modern foundation models and their unique transfer learning capabilities. Yu et al. [17] conducted a comprehensive survey of transfer learning techniques in medical image analysis, highlighting the effectiveness of fine-tuning strategies for domain-specific applications. Results showed that appropriate transfer learning approaches could significantly reduce training data requirements while maintaining high performance, providing insights relevant to other scientific imaging domains, including astronomy. Ayana et al. [18] explored multistage transfer learning for medical images, proposing progressive adaptation strategies that gradually bridge the gap between natural and medical image domains. Experimental results demonstrated superior performance compared to single-stage transfer learning, but the approach has not been evaluated on astronomical imaging tasks where domain gaps may be even more pronounced. Recent work has begun exploring simulation-based pretraining for astronomical applications, leveraging synthetic data to bridge domain gaps. These approaches show promise for addressing the scarcity of labeled astronomical data, but systematic comparison with foundation model transfer learning strategies remains limited. Transfer

learning techniques have proven effective for adapting models to scientific imaging domains, but most existing work focuses on medical applications. The unique challenges of astronomical imaging, including extreme dynamic ranges and complex noise structures, require specialized transfer learning strategies that have not been systematically explored with modern foundation models. Recent work in astronomical image classification has explored models pretrained on simulated or instrument-grounded datasets, as well as hybrid human–machine pipelines (e.g., citizen-science labeling combined with machine learning) for scalable classification. These directions are highly complementary to our setting. Our focus in this study is to evaluate the transfer of a large natural-image foundation model (DINOv2) to a curated planetary-imagery dataset and to quantify trustworthiness via calibration and OOD rejection under a controlled benchmark. A unified, head-to-head comparison across astronomy-domain pretraining, simulation-based transfer, and hybrid human–machine systems is an important next step as larger instrument-specific planetary datasets become available.

2.4. Uncertainty Quantification and Model Calibration

Model reliability is crucial for scientific applications where incorrect predictions can have significant consequences. Abdar et al. [19] provided a comprehensive review of uncertainty quantification techniques in deep learning, covering Bayesian approaches, ensemble methods, and calibration techniques. Their analysis highlighted the importance of distinguishing between aleatoric and epistemic uncertainty, but applications to scientific imaging domains were limited. Zhang et al. [20] introduced methodologies for model calibration and entropy-based uncertainty quantification in deep learning applications for drought detection. Results demonstrated that well-calibrated models provide more reliable confidence estimates, but the techniques were not evaluated on astronomical imaging tasks where calibration requirements may differ significantly. Yang et al. [21] present a survey on generalized OOD detection, proposing a unified framework that connects five closely related settings—anomaly detection, novelty detection, open set recognition, OOD detection, and outlier detection—and clarifies how their assumptions and evaluation protocols differ. They further organize modern OOD detection approaches into major methodological families (e.g., classifier score-based, density-based, distance-based, and reconstruction-based) and summarize common benchmarks and open challenges for reliable deployment. However, the survey also emphasizes that proper real-world benchmarking under complex distribution shifts remains an open problem, implying that domain-specific applications may require tailored evaluation setups beyond standard vision benchmarks. Recent work has begun exploring uncertainty quantification specifically for astronomical applications. Bayesian deep learning approaches have shown promise for radio galaxy classification, providing probabilistic predictions that enable more informed scientific decision-making, but systematic evaluation across different astronomical tasks and model architectures remains limited. While uncertainty quantification and model calibration techniques have been extensively studied for general computer vision applications, their adaptation to astronomical imaging presents unique challenges. The complex noise structures and extreme dynamic ranges common in astronomical data require specialized reliability assessment methods that have not been systematically developed.

2.5. Research Gap and Motivation

The literature review reveals several critical gaps in the current state of research: **Model Evaluation Gap:** While vision foundation models have demonstrated remarkable capabilities on natural images, systematic evaluation of their performance on astronomical imaging tasks remains limited. Existing studies typically focus on single model architec-

tures or specific astronomical applications without a comprehensive comparison across different foundation models and tasks. **Transfer Learning Strategy Gap:** Although transfer learning has proven effective in medical imaging, the unique characteristics of astronomical data—including extreme dynamic ranges, complex noise structures, and physical constraints—require specialized adaptation strategies that have not been systematically explored with modern foundation models. **Reliability Assessment Gap:** Current uncertainty quantification and model calibration techniques are primarily designed for natural image applications. The reliability requirements for scientific applications, where incorrect predictions can impact scientific conclusions, necessitate specialized approaches that account for the unique statistical properties of astronomical data. **Unified Framework Gap:** No existing work provides a comprehensive framework that simultaneously addresses model selection, transfer learning strategies, and reliability assessment for foundation models in astronomical applications. Such a framework is essential for enabling widespread adoption of these powerful models in the astronomical community. Therefore, this research aims to address these gaps by conducting a systematic evaluation of multiple vision foundation models on astronomical image classification and source detection tasks, developing effective transfer learning strategies tailored to astronomical data characteristics, and establishing comprehensive reliability assessment methods that account for the unique requirements of scientific applications.

2.6. Method

This chapter describes the implemented pipeline for multi-class celestial image classification using (i) a two-stage transfer learning procedure for vision backbones loaded through the timm ecosystem, and (ii) an evaluation suite that includes calibration, test-time augmentation (TTA), energy-based OOD analysis, and Grad-CAM visualization when applicable. All details below are strictly grounded in the provided source code.

2.6.1. Problem Formulation

Let $D = \{(x_i, y_i)\}_{i=1}^N$ denote a labeled dataset of celestial images, where (1) $x_i \in \mathbb{R}^{H \times W \times 3}$ is the i -th image after resizing/cropping and normalization, (2) $y_i \in \{0, 1, \dots, K-1\}$ is the corresponding class label, and (3) K is the number of classes determined by the dataset folder structure (class names are sorted lexicographically and mapped to indices accordingly). A classifier $f_\theta(\cdot)$ parameterized by θ maps an input image x to a logit vector:

$$z = f_\theta(x) \in \mathbb{R}^K \quad (1)$$

where z_k denotes the unnormalized score for class k . The predicted probability vector $p \in \mathbb{R}^K$ is computed with softmax:

$$p_k = \text{softmax}(z)_k = \frac{\exp(z_k)}{\sum_{j=0}^{K-1} \exp(z_j)} \quad (2)$$

The predicted label is $\hat{y} = \text{argmax}_k p_k$. In the current codebase, dataset splits are assumed to be pre-organized into train/, val/, and test/directories, each containing sub-folders named by class. A helper script is provided to generate such splits from raw class folders, but training itself only consumes the organized split structure.

2.6.2. Overall Framework

The implemented workflow is organized into three layers: data preparation, two-stage transfer learning, and evaluation/reliability analysis. (1) Data preparation (offline organization + runtime transforms). A preprocessing script (datapre.py) can organize a raw dataset (class folders containing images) into split directories using a fixed ratio (0.7,

0.15, 0.15) for train/validation/test, with deterministic splitting via `random_state = 42`. Runtime preprocessing is performed by torchvision transforms defined in the training and evaluation scripts. (2) Two-stage transfer learning (training). Training is orchestrated by `run_experiment()` in `train_and_eval_v2.py` and comprises: Stage 1 (linear probing): the backbone is frozen, and only the classification head is trained for `epochs_stage1` epochs (default = 5). Stage 2 (selective fine-tuning): a new backbone instance is constructed; all parameters are frozen, and then the last blocks are unfrozen (or all layers for full fine-tuning) according to `unfreeze_last_n_blocks`. Stage 2 trains for `epochs_stage2` epochs (default = 20 in `train_and_eval_v2.py`, =15 in `ablation_runner.py`). (3) Evaluation and reliability assessment (post-training). `evaluate_model_v2.py` implements post hoc evaluation on the test split, optionally with: TTA via horizontal/vertical flips and averaging logits; temperature scaling fitted on the validation split; ECE and reliability diagrams; energy-based OOD analysis (when an external OOD image folder is provided); Grad-CAM visualizations if the model contains convolutional layers (otherwise skipped). Additionally, `ablation_runner.py` runs a grid over model backbones, augmentation presets, loss types, and unfreezing depths, and performs paired McNemar significance tests against a fixed DINOv2 baseline configuration.

2.7. Data Preprocessing

2.7.1. Dataset Organization and Split Convention

The training/evaluation pipeline expects a directory structure: `data_dir/{train/val/test}/{class_name}/{image files}`. Class set and label mapping. Each split enumerates class subfolders under its root; class names are sorted and mapped to consecutive indices $\{0, \dots, K - 1\}$. Supported image extensions. For training, the dataset loader filters by: `{.jpg, .jpeg, .png, .bmp, .webp, .tif, .tiff}`. Evaluation scripts support a superset for inference discovery (including additional formats with best-effort verification). A dataset organizer performs a per-class split using `train_test_split`: first split off the test set with proportion `split_ratio` [2] (default 0.15), then split the remaining into train/val with ratio $0.15/(0.70 + 0.15)$, both with `random_state = 42` for determinism. A separate utility (`assert_dataset_health`) is executed at the beginning of training to verify the existence of the split folders and to count images per class; training aborts early if the structure is missing or empty.

2.7.2. Input Resolution and Automatic Size Resolution

Input spatial size is controlled by an integer `img_size`, but the training script implements automatic overriding when the selected backbone recommends a different resolution. `Resolve_required_img_size(model_name, fallback)` instantiates a lightweight timm model and uses `resolve_data_config` to read `input_size`. If parsing fails, a special-case rule returns 518 when the model name contains “dinov2”. After constructing the actual training model, a second “hard validation” checks `backbone.patch_embed.img_size` (when available) and rebuilds loaders if the internally expected size disagrees with the current pipeline. Therefore, the effective resolution used in training and evaluation is:

$$H = W = S \quad (3)$$

where S is either the user-specified `img_size` or the auto-resolved size. This is particularly relevant for DINOv2 configurations, which may require $S = 518$ in the implemented fallback logic.

2.7.3. Train/Val/Test Transforms

All runtime preprocessing is implemented with torchvision transforms.

Let x denote an RGB PIL image. A normalization operator is used consistently:

$$\text{Norm}(x) = \frac{x - \mu}{\sigma} \quad (4)$$

where $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$. Training transforms are controlled by the aug preset in build_transforms (kind, img_size):

Weak augmentation (kind = “weak”): Resize (int(1.15S)) → RandomResizedCrop (S, scale = (0.7, 1.0)) → RandomHorizontalFlip → ColorJitter (0.2, 0.2, 0.2, 0.1) → ToTensor → Normalize.

Strong augmentation (kind = “strong”): RandomResizedCrop (S, scale = (0.5, 1.0)) → RandomHorizontalFlip → RandomVerticalFlip → RandomRotation (40) → ColorJitter(0.4, 0.4, 0.4, 0.2) → ToTensor → Normalize.

RandAugment preset (kind = “rand”): Resize(S) → RandAugment (num_ops = 2, magnitude = 9) → RandomResizedCrop (S, scale = (0.6, 1.0)) → ToTensor → Normalize.

TrivialAugment preset (kind = “trivial”): Resize (S) → TrivialAugmentWide → RandomCrop (S, pad_if_needed = True) → RandomHorizontalFlip → ToTensor → Normalize. Validation/Test transforms are fixed (independent of kind): Resize (int (1.05S)) → CenterCrop (S) → ToTensor → Normalize.

2.7.4. Class-Imbalance Handling via Weighted Sampling

To mitigate class imbalance during training, the default sampler strategy is weighted:

1. Let n_k be the number of training samples in class k .
2. The code constructs inverse-frequency class weights:

$$w_k = \frac{\sum_{j=0}^{K-1} n_j}{\max(n_k, 1)} \quad (5)$$

3. Each training sample i with label y_i receives weight w_{y_i} .
4. A WeightedRandomSampler draws samples with replacement, for num_samples = $N_{\{\text{train}\}}$. When the sampler is enabled, the dataloader uses shuffle = False and relies on sampling; otherwise, it uses standard shuffling.

2.8. Model Architecture

2.8.1. Backbone Wrapper

The primary model abstraction is CelestialVFM, which wraps a backbone instantiated by timm.create_model. Given a model_name string and a class count K , it builds a backbone with an attached classifier head producing K - dimensional logits. If the backbone exposes reset_classifier, it is called to ensure the classifier output matches K .

Let $g_\phi(\cdot)$ denote the feature extractor of the backbone and $h_\phi(\cdot)$ denote the classifier head. The full model can be written as follows:

$$f_\theta(x) = h_\phi(g_\phi(x)), \theta = (\phi, \varphi) \quad (6)$$

Here:

g_ϕ is initialized from pretrained weights when pretrained = True; h_ϕ is initialized to match K classes and trained during adaptation. To implement selective unfreezing, the wrapper maps a model_name to a family label (e.g., vit, swin, convnext, resnet, efficientnet) using keyword heuristics and then collects a linear list of “blocks”:

ViT-like: model.blocks or model.transformer.layers;

Swin: iterate model.layers and concatenate stage.blocks;

ConvNeXt: concatenate stages from model.stages;

ResNet: concatenate residual blocks from layer1–layer4;

EfficientNet: model.blocks or stage lists.

This definition is used only for which modules to unfreeze; the internal computations of each backbone are those provided by the underlying timm implementation.

2.8.2. Linear Probing vs. Selective Fine-Tuning (Freeze/Unfreeze Policy)

Two training regimes are encoded directly in CelestialVFM initialization:

- (a) Linear probing (linear_probe = True).

All backbone parameters are frozen:

$$\forall p \in \phi, p.requires_grad \leftarrow False \quad (7)$$

Then the classifier head is unfrozen (trainable). The code attempts to locate the head among common attributes (head, classifier, fc, heads) and sets requires_grad = True for those parameters; otherwise, it applies a name-based heuristic to unfreeze parameters containing “classifier”/“head” or ending with “weight”/“bias”.

- (b) Fine-tuning (linear_probe = False).

The wrapper first freezes all parameters, then:

If unfreeze_last_n_blocks >= 999: unfreeze all parameters (full fine-tuning).

Else: unfreeze the last N blocks in the collected block list. Formally, letting $B = [B_1, \dots, B_M]$ be the ordered block list, the unfrozen subset is:

$$\{B_{M-N+1}, \dots, B_M\} \quad (8)$$

All parameters within these blocks are set to require_grad = True. In the provided ablation grid, unfreezing depths are enumerated as follows:

$$N \in \{0, 4, 999\} \quad (9)$$

where 0 corresponds to linear probing, 4 corresponds to unfreezing the last four blocks, and 999 corresponds to full fine-tuning. For ViT-Base backbones (12 transformer blocks), unfreezing only the top portion of the network is a common small-data adaptation strategy: early and mid-level blocks tend to encode more generic visual primitives, while later blocks are more task- and domain-specific. We therefore unfreeze the last four blocks (approximately the top one-third of the backbone) to provide sufficient capacity for domain adaptation to planetary imagery while reducing the risk of overfitting and limiting compute/memory overhead compared to full fine-tuning. We treat the unfreezing depth as a tunable hyperparameter and include an explicit evaluation protocol for alternative depths in our ablation settings.

2.8.3. Backbone Candidates Used in Ablations

ablation_runner.py defines a model grid (as strings passed to timm) including: vit_base_patch14_dinov2.lvd142m, vit_base_patch16_224, convnextv2_tiny.fcmae_ft_in1k, swin_base_patch4_window7_224, resnet50, and tf_efficientnet_b3_ns. These names determine both the backbone architecture and its pretrained initialization (when pretrained = True in CelestialVFM). The method does not implement a custom transformer or convolutional design beyond freezing/unfreezing control; architectural details are inherited from the selected timm model.

2.8.4. Legacy CNN Baselines

A separate module, `CelestialModel`, in `improved_model_architecture.py` provides legacy baselines using torchvision backbones:

ResNet50 mode: feature extractor is `resnet50` with the original final FC removed; the extracted feature vector (dimension equal to `base_model.fc.in_features`) is passed through a three-layer MLP head:

Linear $\rightarrow$ ReLU $\rightarrow$ Dropout $\rightarrow$ Linear $\rightarrow$ ReLU $\rightarrow$ Dropout $\rightarrow$ Linear ().

EfficientNet-B3 mode: feature extractor is `efficientnet_b3.features`; the head begins with `AdaptiveAvgPool2d(1)` and uses a similar MLP.

Ensemble mode: concatenates pooled ResNet and EfficientNet features and feeds them to an MLP classifier.

In all modes, the feature extractor parameters are frozen by default. This module is used by `evaluate_model_v2.py` when `backend = "legacy"` is selected.

2.9. Loss Function and Optimization

2.9.1. Supervised Losses

The training script supports three loss functions via `build_loss()`:

- (1) Standard cross-entropy (`loss_name = "ce"`).

Given one-hot label y_i and predicted probabilities p_i , the per-sample loss is:

$$\downarrow_{CE}(z_i, y_i) = -\log p_{i,y_i} = -z_{i,y_i} + \log \sum_{k=0}^{K-1} \exp(z_i, k) \quad (10)$$

The empirical objective over a mini-batch of size B is:

$$L_{CE} = \frac{1}{B} \sum_{i=1}^B \downarrow_{CE}(z_i, y_i) \quad (11)$$

Symbols: z_i denotes logits from the model, $p_{i,k}$ is the softmax probability of class k , and y_i is the integer label.

- (2) Label-smoothing cross-entropy (`loss_name = "label_smoothing"`).

The implementation fixes smoothing $\varepsilon = 0.1$ by default. The smoothed target distribution q_i is:

$$q_{i,k} = \begin{cases} 1-\varepsilon, & k=y_i, \\ \frac{\varepsilon}{K-1}, & k \neq y_i. \end{cases} \quad (12)$$

With $\log_{p_i,k} = \log \text{softmax}(z_i)_k$, the loss is:

$$L_{LS} = -\frac{1}{B} \sum_{i=1}^B \sum_{k=0}^{K-1} q_{i,k} \log p_{i,k} \quad (13)$$

Symbols: ε is the smoothing factor; K is the number of classes.

- (3) Focal loss (`loss_name = "focal"`).

The focal variant is implemented as follows:

$$\downarrow_{Focal}(z_i, y_i) = (1 - p_{i,y_i})^\gamma \downarrow_{CE}(z_i, y_i) \quad (14)$$

where $\gamma = 2.0$ by default, and the batch loss is the mean over samples. In the provided `build_loss()`, no class-weight vector is supplied (i.e., `weight = None`).

No additional regularization terms (e.g., explicit ℓ_2 penalty added to the objective) are introduced beyond the optimizer's weight decay and any backbone-internal regularization configured in the underlying model. The CelestialVFM wrapper uses `drop_rate = 0.0` by default.

2.9.2. Two-Stage Optimization Strategy (AdamW)

Training uses AdamW on trainable parameters only:

Stage 1:

$$\text{optimizer} = \text{AdamW}(\theta_{\text{train}}, \eta, \lambda) \quad (15)$$

where θ_{train} are the parameters with `requires_grad = True` (typically the classifier head), 3×10^{-4} (default lr), and $\lambda = 10^{-4}$ (default wd).

Stage 2: The fine-tuning stage uses a smaller learning rate:

$$\eta_{\text{stage2}} = 0.1\eta \quad (16)$$

i.e., 3×10^{-5} under default settings, with the same weight decay. No learning-rate scheduler is implemented in `train_and_eval_v2.py`. No gradient clipping is implemented.

2.9.3. Training Loop, AMP, and Early Stopping

For each epoch, the code performs standard mini-batch optimization. Let B be a mini-batch, then:

1. Compute logits $z = f_{\theta}(x)$.
2. Compute loss $L(z, y)$ (one of the losses above).
3. Backpropagate and apply the AdamW update.

When CUDA is available, the implementation enables automatic mixed precision via `torch.amp.autocast` and a `GradScaler` to scale gradients before the optimizer step. This is applied in both training and evaluation loops (evaluation uses autocast but not gradient scaling). Both Stage 1 and Stage 2 implement early stopping with patience `early_stop_patience` (default = 5), based on validation top-1 accuracy (`ev["acc1"]`). If validation accuracy does not improve in consecutive epochs, the patience counter decreases, and training stops when it reaches zero. Stage 2 additionally saves the best checkpoint when validation accuracy improves. The two-stage transfer learning schedule is shown in Figure 1.

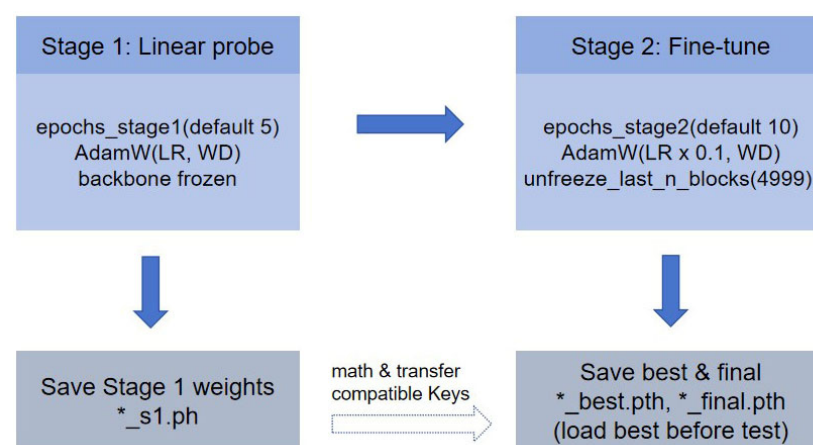


Figure 1. Two-stage transfer learning schedule (The symbol “*” denotes a placeholder for the model name in the saved weight filenames).

2.10. Evaluation Metrics

This subsection defines all reported metrics and reliability measures implemented in the provided code.

2.10.1. Top-k Accuracy (k = 1 and 5)

For each sample i , let Π_i be the ranking of classes induced by logits z_i in descending order. The top-k accuracy is:

$$Acc@k = \frac{1}{N} \sum_{i=1}^N 1\{y_i \in \{\Pi_i, \dots, \Pi_i(k)\}\} \quad (17)$$

In code, `accuracy_topk` returns `Acc@1` and `Acc@5` as percentages. This computation assumes $K \geq 5$ because `topk = (1,5)` is invoked.

2.10.2. Macro-F1 Score

For class k , define true positives TP_k , false positives FP_k , and false negatives FN_k . Precision and recall are:

$$P_k = \frac{TP_k}{TP_k + FP_k}, R_k = \frac{TP_k}{TP_k + FN_k} \quad (18)$$

The class-wise F1 is:

$$F1_k = \frac{2P_k R_k}{P_k + R_k} \quad (19)$$

Macro-F1 averages across classes:

$$MacroF1 = \frac{1}{K} \sum_{k=0}^{K-1} F1_k \quad (20)$$

This corresponds to `sklearn.metrics.f1_score(..., average = "macro")` in the training script and the evaluation script.

2.10.3. Balanced Accuracy

Balanced accuracy is the mean recall across classes:

$$BAcc = \frac{1}{K} \sum_{k=0}^{K-1} \frac{TP_k}{TP_k + FN_k} \quad (21)$$

This corresponds to `sklearn.metrics.balanced_accuracy_score`.

2.10.4. Confidence Intervals (Wilson and Bootstrap)

The training script computes confidence intervals on the test split: Wilson score interval for top-1 accuracy. Let s be the number of correct predictions and n be the total test samples. The Wilson interval $[\hat{p}_-, \hat{p}_+]$ is computed with $z = 1.9599$ for $\alpha = 0.05$. The implementation follows the standard Wilson closed-form. Bootstrap intervals for Macro-F1 and Balanced Accuracy. For a metric $m(\cdot)$, bootstrap resampling draws index sets with replacement and computes m on each resample; the $(\alpha/2, 1 - \alpha/2)$ quantiles (default $\alpha = 0.05$, resamples $n = 1000$) form the 95% interval.

2.10.5. Statistical Significance via McNemar's Test (Paired)

`ablation_runner.py` performs a paired comparison between a fixed baseline configuration and alternative configurations using McNemar's test. For two classifiers A and B evaluated on the same test set:

b : number of samples where A is wrong and B is correct.
 c : number of samples where A is correct and B is wrong.

$$n = b + c.$$

The code computes an exact two-sided p -value using a binomial tail over Binomial $(n, 0.5)$ (implemented via combinations), returning (b, c, p) .

2.10.6. Calibration: Temperature Scaling and ECE

Given logits z , calibrated logits are:

$$z' = \frac{z}{T} \quad (22)$$

where $T > 0$ is a scalar temperature. The calibrated probabilities are:

$$p' = \text{softmax}(z') = \text{softmax}\left(\frac{z}{T}\right) \quad (23)$$

`evaluate_model_v2.py` fits T on the validation set by minimizing the cross-entropy of p' with respect to the true labels, optimizing $\log T$ using Adam with a learning rate of 0.01 for a fixed number of iterations (`max_iter` set to 400 in the evaluation script).

Let $\hat{p}_i = \max_k p_{i,k}$ be the confidence and $\hat{y}_i = \text{argmax}_k p_{i,k}$ the prediction. Partition $[0, 1]$ into M bins $\{B_m\}_{m=1}^M$. For bin B_m , define:

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} 1\{\hat{y}_i = y_i\}, \text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \quad (24)$$

Then:

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)| \quad (25)$$

The implementation uses $M = 15$ bins by default.

In addition to ECE, we report two proper scoring rules that are sensitive to probability quality: negative log-likelihood (NLL) and the multiclass Brier score. Given predicted probabilities $p_i \in R^C$ and the ground-truth label y_i .

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)| \quad (26)$$

and the multiclass Brier score is:

$$\text{Brier} = \frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C (p_i[c] - 1[y_i = c])^2 \quad (27)$$

Unless stated otherwise, we report ECE in percent.

2.10.7. Test-Time Augmentation (TTA)

TTA is implemented as averaging logits across a small set of deterministic flips: none: use original logits; hflip: average logits from original and horizontally flipped input; x4: average logits from original, horizontal flip, vertical flip, and both flips. Formally, given a set of transformations T , the TTA logit is:

$$\bar{z} = \frac{1}{|T|} \sum_{t \in T} f_{\theta}(t(x)) \quad (28)$$

The prediction is then computed from $\bar{z}$ (and optionally temperature-scaled).

2.10.8. Energy-Based OOD Scoring and OOD Metrics

Given logits z and a temperature T , the implemented energy score is:

$$E(x) = -\log \sum_{k=0}^{K-1} \exp\left(\frac{z_k}{T}\right) \quad (29)$$

Lower energy corresponds to larger $\log \sum \exp(\cdot)$, i.e., higher overall logit mass; in the OOD routines, higher energy values are treated as more OOD-like for thresholding and ROC/PR curves.

When OOD images are provided, the evaluation code computes: AUROC based on energy scores (positive class = OOD) and average precision (AP) from the precision–recall curve (positive class = OOD). Both are computed from sklearn.metrics on concatenated in-distribution (test) and OOD energy arrays. To better reflect deployment decision-making, we complement AUROC with threshold-based operating points. We report TPR at fixed target FPR levels (1% and 5%), where the OOD rejection threshold is selected using only in-distribution validation data: $\tau_\alpha = \text{Quantile}(E_{val}, 1 - \alpha)$ for $\alpha \in \{0.05, 0.01\}$. We then compute $TPR@FPR = \alpha = \Pr(E_{ood} > \tau_\alpha)$. The inference script additionally implements a reject option: estimate an energy threshold τ using either: val_p95: τ is the 95th percentile of validation energies (default target FPR = 0.05); youden: select τ maximizing Youden's $J = TPR - FPR$ using both ID and OOD energies (if OOD energies are provided). Reject if $E(x) > \tau$; otherwise, output a class prediction.

2.10.9. Interpretability via Grad-CAM

evaluate_model_v2.py includes a Grad-CAM implementation that requires a convolutional feature map $A \in \mathbb{R}^{C \times H' \times W'}$. For a target class c , Grad-CAM computes:

$$\alpha_k^c = \frac{1}{H'W'} \sum_{\mu=1}^{H'} \sum_{\nu=1}^{W'} \frac{\partial s^c}{\partial A_k(\mu, \nu)} \quad (30)$$

where s^c is the class score (logit) for class c . The CAM map is:

$$\text{CAM}^c(\mu, \nu) = \text{ReLU}\left(\sum_{k=1}^C \alpha_k^c A_k(\mu, \nu)\right) \quad (31)$$

followed by min-max normalization to [0.1] and overlay onto the original image. The script automatically selects the last convolutional layer in the model; if no nn.Conv2d layers exist (common for some transformer backbones), Grad-CAM is skipped.

2.11. Relation to Bayesian/Ensemble Uncertainty Methods

Bayesian neural networks, Monte-Carlo dropout, and deep ensembles are widely used to improve uncertainty estimation and calibration. However, these approaches typically require multiple stochastic forward passes or multiple model instances, increasing inference cost. In this work, we focus on a lightweight post hoc calibration (temperature scaling) and an energy-based OOD score that imposes minimal runtime overhead, which is more compatible with resource-constrained deployment settings.

3. Experiments

In this section, we provide a detailed description of the dataset curation process, implementation details, and a comprehensive analysis of the experimental results. Our

evaluation framework assesses models across three critical dimensions: classification accuracy, predictive reliability (calibration), and operational safety (OOD detection).

3.1. Dataset Curation

3.1.1. The Celestial Body Dataset

To address the lack of standardized benchmarks for fine-grained planetary classification, we manually curated a high-fidelity Celestial Body Dataset as shown in Figure 2. The raw images were collected from NASA's official mission archives to ensure scientific reliability. The dataset contains seven classes representing major bodies in the solar system, including Earth, Jupiter, Mars, the Moon, Neptune, Saturn, and Uranus. To decouple representation learning from class-frequency effects and to provide a controlled benchmark for calibration and OOD studies, we curated a class-balanced subset (100 images per class; 700 images total). We emphasize that this balanced setting is not meant to be a fully realistic model of operational astronomy data, which often exhibits pronounced long-tailed class frequencies. In real deployments, class imbalance can affect not only accuracy but also calibration, since minority classes may receive fewer effective updates, and confidence estimates can become biased toward majority classes. Our training pipeline, therefore, includes imbalance-aware options (e.g., weighted sampling and loss variants described in Sections 2.7.4 and 2.9.1), and we identify long-tailed evaluation (including mission/instrument-dependent frequency shifts) as a key next step for external validity.

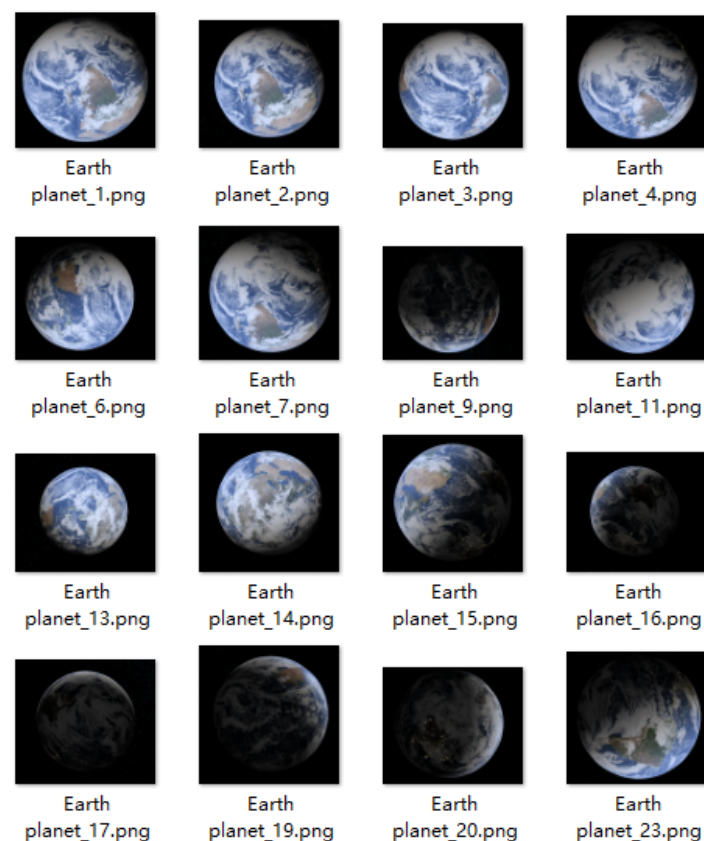


Figure 2. High-fidelity Celestial Body Dataset from NASA's official mission (class = Earth).

3.1.2. OOD Dataset

To evaluate the safety of the model in open-world scenarios, we constructed a separate OOD dataset containing approximately 200 images sourced via Google Image Search. This dataset includes diverse space-related artifacts such as deep space nebulae, satellite debris, sensor noise, and abstract space art, as shown in Figure 3. These samples serve as

“unknowns” that share visual characteristics with the target domain but must be rejected by the system. We acknowledge that the current OOD benchmark, which is sourced from web images, does not fully capture instrument-induced anomalies that occur in spaceborne imaging pipelines (e.g., cosmic-ray hits, saturation/persistence effects, and detector-specific stray light patterns). As a result, the reported OOD AUROC should be interpreted as performance on a diverse “open-world” web-image OOD set rather than a definitive estimate under a specific instrument’s failure modes. Nevertheless, our OOD scoring mechanism is model-based (energy computed from logits) and can be recalibrated for a target platform by selecting rejection thresholds on an in-distribution validation stream from the same instrument. In future work, we will extend the OOD benchmark toward instrument-specific artifacts and provide a reproducible protocol for threshold selection and monitoring under mission conditions. In practical deployments, we recommend selecting operating thresholds (e.g., FPR-controlled energy quantiles) using an in-distribution validation stream from the same instrument, and periodically re-validating these thresholds as observing conditions and sensor pipelines evolve.



Figure 3. The samples of the OOD dataset.

3.1.3. Preprocessing Pipeline

We implemented a rigorous data processing pipeline (see `datapre.py` and `check_duplicates_phash.py`) to ensure experimental integrity:

- **Stratified Splitting:** The dataset was partitioned into Training, Validation, and Test sets using a ratio of 69:16:15. Training Set: 483 images (69 per class). Validation Set: 112 images (16 per class). Test Set: 105 images (15 per class).
- **Repeated-Split Stability Protocol:** Because the test split is small (15 images per class), single-split estimates can be sensitive to partitioning. We therefore perform $R = 5$ repeated stratified train/validation/test splits with different random seeds ($\text{random_state} \in \{42, 43, 44, 45, 46\}$) and report mean $\pm$ standard deviation for Top-1 accuracy, ECE, and OOD AUROC. To keep computation tractable while isolating split-induced variability, we adopt a frozen-backbone linear-probing setting: we extract fixed DINOv2 ViT features (`vit_base_patch14_dinov2.lvd142m`) and train a multinomial logistic-regression classifier on each training split; temperature scaling is fitted on the corresponding validation split.
- **Leakage Prevention:** Given the small dataset size, we executed Perceptual Hashing (pHash) verification across all splits. The verification script confirmed 0 exact or near-duplicates between the training and testing sets, ensuring that the reported performance reflects genuine generalization rather than memorization.
- **Normalization:** All images were resized (to 518^2 for DINOv2, 224^2 or 300^2 for CNNs) and normalized using standard ImageNet mean and standard deviation statistics.

3.2. Implementation Details

All experiments were conducted using the PyTorch 2.0 framework and the timm library. Hardware: Training and evaluation were performed on a single NVIDIA GeForce RTX 3090 (24GB VRAM) GPU. We used AdamW with weight decay 1×10^{-4} . Unless otherwise stated, we did not employ an explicit learning-rate scheduler; instead, we adopted a two-stage learning-rate setting with $lr = 3 \times 10^{-4}$ for Stage 1 (linear probing) and $lr = 3 \times 10^{-5}$ for Stage 2 (selective fine-tuning). Early stopping was applied based on validation accuracy (patience = 5). We choose unfreezing_last_n_blocks = 4 as a conservative default for the small-data regime (ViT-Base has 12 blocks, hence 4 corresponds to adapting the top one-third of the backbone), balancing domain adaptation capacity against overfitting risk and deployment cost. Augmentation: We employed a robust RandAugment strategy (including rotation, color jitter, and random resized crop) to simulate the arbitrary orientations of celestial bodies in space.

We denote the model input resolution as $S \times S$. Unless stated otherwise, $S = 518$ for DINOv2, following the model's recommended configuration, while $S = 224$ for CNN baselines and standard ViT/Swin backbones. Our pipeline automatically enforces the backbone's expected input size to avoid resolution mismatch.

3.3. Stress-Test Robustness Evaluation

Motivated by deployment conditions in space missions, we complement clean-test evaluation with controlled test-time distribution shifts applied only to the held-out test split. The perturbation families are chosen to approximate common sources of degradation and mismatch in operational imagery: (i) additive Gaussian noise with $\sigma \in \{5, 10, 20\}$; (ii) Gaussian blur with severity levels $\{1, 2, 3\}$; (iii) JPEG compression with quality factors $\{90, 70, 50\}$; (iv) resolution degradation by downsampling the image to $\{0.75, 0.50, 0.25\}$ of its original size followed by resizing back; and (v) partial-view shift by randomly cropping to $\{0.7, 0.5\}$ of the original area followed by resizing. This shift suite helps differentiate models beyond saturated clean accuracy and directly probes robustness under conditions that are plausible in real pipelines. All perturbed samples are evaluated using the identical preprocessing and inference pipeline as the clean test set. We report Top-1 accuracy as well as macro-F1 and balanced accuracy, and we also provide Wilson confidence intervals due to the small test size.

3.4. Baselines and Zero-Shot Evaluation

We compared the proposed DINOv2 framework against a suite of baselines representing different architectural paradigms: CNNs: ResNet50 (Classic baseline), EfficientNet-B3 (Modern efficient CNN), ConvNeXt V2-Tiny (MIM-based CNN). Transformers: Standard ViT-Base (Supervised), Swin Transformer (Hierarchical). Zero-Shot Baseline: Prior to fine-tuning, we evaluated the task difficulty using OpenCLIP ViT-B-16. It achieved a zero-shot accuracy of 84.76%, establishing a strong lower bound and confirming the dataset's semantic distinguishability.

4. Results

4.1. Main Results

4.1.1. Classification Performance

Figure 4 shows the comparison of convergence speeds during Stage 2 fine-tuning. DINOv2 (Red) demonstrates rapid adaptation, reaching 100% accuracy within the first two epochs. In contrast, the ResNet50 baseline (Blue) starts significantly lower (~80%) and exhibits instability. EfficientNet (Green) performs well but converges more slowly than DINOv2.

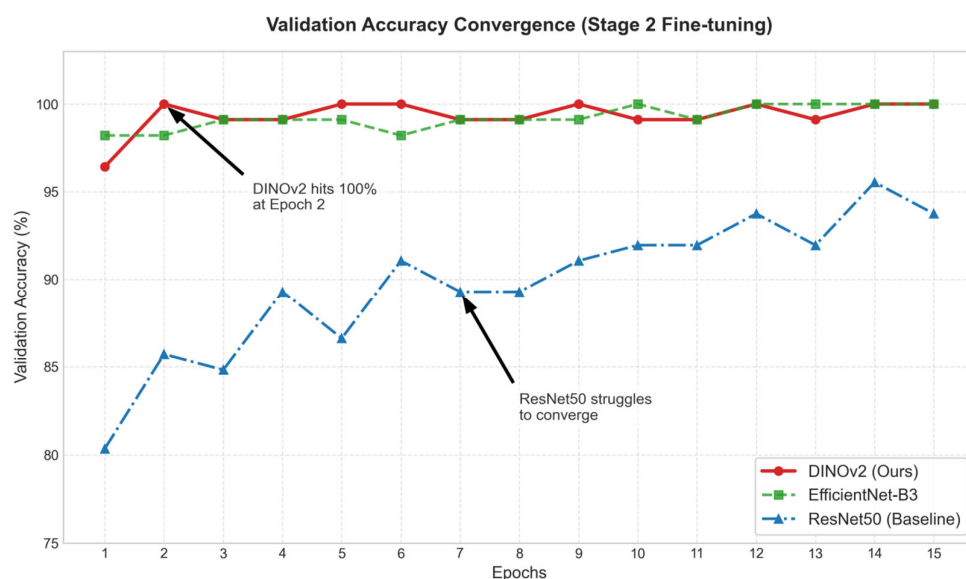


Figure 4. Validation accuracy convergence.

Table 1 summarizes the performance on the held-out test set. While most modern architectures saturated the benchmark, significant differences emerged in the legacy baselines.

Table 1. Comparative analysis of accuracy, calibration, and OOD detection. “↑” means higher values are better, “↓” means lower values are better.

Model Architecture	Top-1 Accuracy	ECE (Post-Cal) ↓	OOD AUROC ↑
ResNet 50	96.19%	2.90%	86.77%
Standard ViT	97.14%	-	-
Swin Transformer	100.0%	-	-
ConvNext V2	100.0%	-	-
EfficientNet-B3	100.0%	0.36%	93.96%
DINOv2 (Ours)	100.0%	0.08%	93.68%

Modern architectures (DINOv2, EfficientNet, Swin) successfully achieved 100% Top-1 Accuracy. In contrast, ResNet50 struggled (96.2%), confusing fine-grained textural classes (e.g., Uranus vs. Neptune). This suggests that advanced architectural biases (such as Global Attention in ViTs) are necessary for this domain.

4.1.2. Reliability Analysis (Calibration)

While EfficientNet and DINOv2 achieved identical accuracy, their reliability metrics reveal a crucial distinction. As shown in Table 1, the Expected Calibration Error (ECE) of DINOv2 (after temperature scaling) is 0.08%, whereas EfficientNet is 0.36% and ResNet50 is 2.90%. DINOv2 provides a 4.5-fold improvement over the strong EfficientNet baseline and a 36-fold improvement over ResNet50. Figure 5 visualizes the reliability diagram. DINOv2’s confidence bars perfectly align with the diagonal ideal, indicating that the model is neither overconfident nor underconfident.

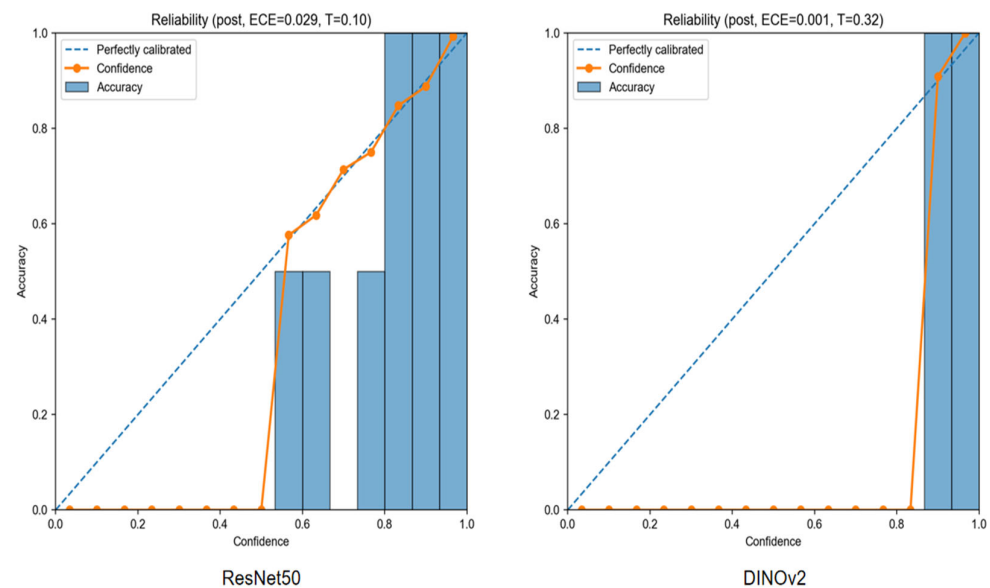


Figure 5. Reliability diagrams (post-calibration). The plots compare the alignment between confidence and accuracy. **(Left)** ResNet50 shows visible gaps from the diagonal ideal. **(Right)** DINOv2 demonstrates near-perfect alignment (ECE = 0.08%), confirming its superior trustworthiness.

4.1.3. Safety Analysis (OOD Detection)

We evaluated the models' ability to reject artifacts from the Google Images OOD dataset using Energy scores. ResNet50 (AUROC 86.8%): The significant overlap between ID and OOD distributions indicates a high risk of false positives (classifying space debris as a planet). DINOv2 (AUROC 93.7%): Demonstrates robust rejection capabilities, statistically comparable to EfficientNet (94.0%). As shown in Figure 6, the ID and OOD energy distributions of ResNet50 exhibit significant overlap, whereas DINOv2 shows a clearer separation between the distributions, which facilitates the setting of a safer rejection threshold. This confirms that our SSL-based framework provides a high safety margin for open-world exploration.

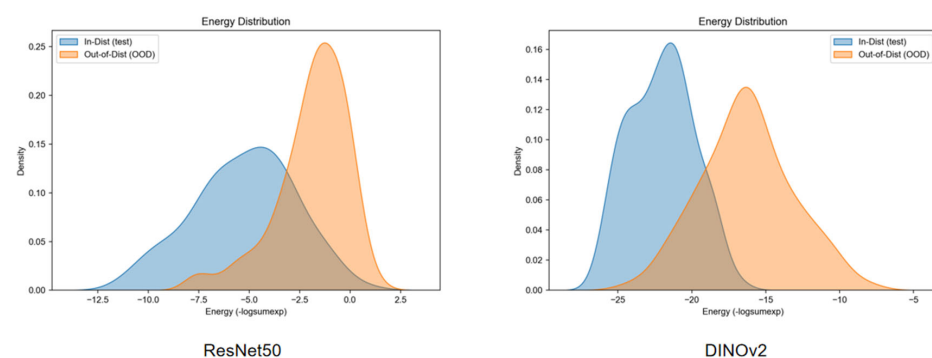


Figure 6. Energy-based OOD detection distributions. Comparison of energy score distributions for ResNet50 **(Left)** vs. DINOv2 **(Right)**. Blue histograms represent in-distribution (planets), and orange histograms represent OOD (artifacts). DINOv2 exhibits a clearer separation between the two peaks, facilitating a safer rejection threshold.

4.1.4. Stability Across Repeated Stratified Splits

Because the held-out test set is small (15 images per class), clean-test accuracy can saturate for multiple architectures, which makes it difficult to assess result stability from a single split. To quantify split sensitivity, we additionally perform repeated stratified three-way splits over the full set of 700 labeled images using a fixed ratio of train/val/test = 0.70/0.15/0.15. We repeat this procedure five times with different random

seeds (42–46). For each split, we train a lightweight linear probe (multinomial logistic regression with feature standardization) on frozen DINOv2 features (ViT-Base/14; timm: vit_base_patch14_dinov2.lvd142m). We then fit a single temperature parameter on the validation logits by minimizing negative log-likelihood (NLL), and evaluate on the corresponding test set. For OOD evaluation, we compute energy scores and report AUROC as well as threshold-based operating points. Specifically, we set thresholds using the 95th and 99th percentiles of validation (in-distribution) energies, which correspond to target FPR levels of 5% and 1%, and then report the resulting TPR on the OOD set. Table 2 summarizes the mean $\pm$ standard deviation across the five repeats. Overall, the approach remains highly accurate ($99.43\% \pm 0.85\%$ Acc@1) with low calibration error ($0.74\% \pm 0.75\%$ ECE). OOD separation is more challenging under the frozen-feature linear-probe setting ($61.50\% \pm 3.49\%$ AUROC), suggesting that representation adaptation is important when using logit/energy-based scores for open-set rejection.

Table 2. Split-stability evaluation across repeated stratified splits (mean $\pm$ std over 5 repeats). Note: This repeated-split table is used to quantify partition sensitivity with minimal compute; absolute OOD AUROC values may differ from the end-to-end fine-tuned models reported in Table 1.

Metric	Mean $\pm$ Std
Acc@1 (%)	99.43 $\pm$ 0.85
ECE (%)	0.74 $\pm$ 0.75
NLL	0.1596 $\pm$ 0.2339
Brier	0.0117 $\pm$ 0.0168
OOD AUROC (%)	61.50 $\pm$ 3.49
TPR@FPR = 5% (%)	33.40 $\pm$ 1.08
TPR@FPR = 1% (%)	22.80 $\pm$ 3.35

4.2. Stress-Test Robustness Beyond Saturated Accuracy

Table 3 summarizes stress-test robustness on the held-out test set ($N = 105$). While clean accuracy is saturated (100%), controlled perturbations reveal meaningful degradation patterns. The model remains highly robust to moderate blur, JPEG compression, resolution degradation, and partial-view cropping, where accuracy stays within 94.29–100%. In contrast, additive Gaussian noise is the dominant failure mode; accuracy drops from 87.62% ($\sigma = 5$) to 77.14% ($\sigma = 10$) and further to 68.57% ($\sigma = 20$). This indicates that the primary robustness bottleneck in our current pipeline is sensitivity to noise-like corruptions, which is consistent with the fact that such perturbations directly distort fine-grained texture and edge cues. Given the small test size ($1/105 \approx 0.95\%$), we interpret $<2\%$ drops with caution, but the large noise-induced degradation is statistically meaningful. Notably, additive Gaussian noise can be viewed as a first-order proxy for reduced signal-to-noise ratio (SNR) conditions that commonly arise in astronomical imaging due to photon noise, background contamination, and sensor noise. The pronounced degradation under noise, therefore, highlights a practical robustness bottleneck and motivates targeted improvements such as noise-aware augmentation, denoising pre-processing, or robustness-oriented fine-tuning using instrument-specific corruption models.

Table 3. Stress-test robustness on perturbed test sets ($N = 105$).

Perturbation	Severity	Top-1 Acc (%)	Δ vs. Clean (pp)
Clean	-	100	0.00
Gaussian noise	$\sigma = 5$	87.62	−12.38
	$\sigma = 10$	77.14	−22.86
	$\sigma = 20$	68.57	−31.43

Table 3. *Cont.*

Perturbation	Severity	Top-1 Acc (%)	Δ vs. Clean (pp)
Gaussian blur	level = 1	99.05	−0.95
	level = 2	97.14	−2.86
	level = 3	97.14	−2.86
JPEG compression	Q = 90	100	0.00
	Q = 70	97.14	−2.86
	Q = 50	94.29	−5.71
Downsample + resize	r = 0.75	100	0.00
	r = 0.50	100	0.00
	r = 0.25	99.05	−0.95
RandomCrop + resize	r = 0.70	99.05	−0.95
	r = 0.50	98.1	−1.90

4.3. Ablation Study

We investigated the impact of data augmentation strategies on model robustness using `ablation_runner.py`. We compared weak augmentation (simple resize/crop) against Rand augmentation (rotation, color jitter, strong cropping).

As evidenced in Table 4, RandAugment is critical. Without strong geometric perturbations (specifically rotation), even the powerful DINOv2 drops to an average accuracy of 97.4%. This confirms that for astronomical objects which lack a canonical orientation (no fixed “up” in space), rotation invariance must be enforced during the fine-tuning stage.

Table 4. Impact of augmentation strategy on DINOv2 accuracy (results averaged across three random seeds).

Augmentation Strategy	Test Accuracy Range	Mean Accuracy
Mean Accuracy	95.2–99.0%	97.4%
Rand (Rotation + Jitter)	100.0–100.0%	100.0%

4.3.1. Loss-Function Sensitivity (Calibration)

Beyond augmentation, calibration can also be influenced by the supervised objective. Our implementation supports standard cross-entropy, label-smoothed cross-entropy, and focal loss (Section 2.9.1). In the main comparisons, we keep the loss fixed to reduce confounding factors when comparing architectures and transfer-learning strategies. Nevertheless, to disentangle representation effects from optimization effects, we include loss-function sensitivity as an explicit ablation axis and report mean $\pm$ standard deviation across seeds for both accuracy and calibration metrics.

4.3.2. Learning-Rate Scheduling Discussion

We note that learning-rate schedules can affect convergence dynamics and calibration. In our current implementation, we do not employ an explicit learning-rate scheduler; instead, we use a two-stage learning-rate setting (Stage 2 uses a smaller learning rate than Stage 1) to stabilize small-data transfer (Section 3.2). A systematic comparison of constant LR versus scheduled LR (e.g., cosine decay) is important.

4.3.3. Unfreezing-Depth Sensitivity

In addition to augmentation, we investigated the sensitivity of selective fine-tuning to the unfreezing depth, since this hyperparameter controls the adaptation–overfitting trade-off in small-data transfer. For ViT-Base backbones (12 transformer blocks), we interpret

unfreezing the last N blocks as adapting the top $N/12$ portion of the network while keeping lower-level representations fixed. We therefore treat N as a tunable hyperparameter and evaluate a small sweep of unfreezing depths (e.g., $N \in \{1, 2, 3, 4, 5\}$ and full fine-tuning) under otherwise identical settings to assess robustness of the default choice ($N = 4$).

4.4. Qualitative Analysis (Visualization)

To interpret the model’s decision-making process, we generated Grad-CAM heatmaps. We visually compared the attention mechanisms of the baseline (ResNet50) against our method (DINOv2).

The visualization in Figure 7 validates the quantitative results: ResNet50’s lower OOD performance correlates with its tendency to focus on background noise (black pixels). In contrast, DINOv2’s self-supervised features force it to learn the semantic object itself, leading to superior robustness against artifacts.

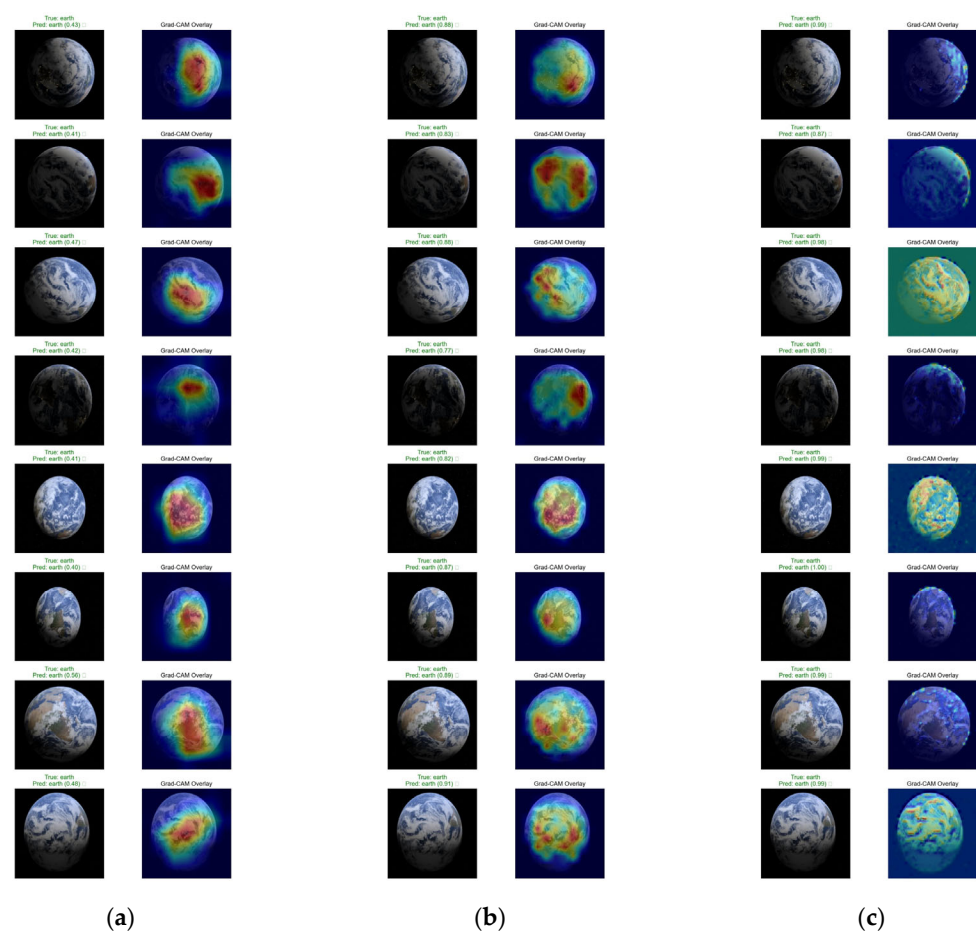


Figure 7. Grad-CAM visual explanation. Comparison of attention maps for a sample image of Saturn. (a) ResNet50: attention is scattered, often focusing on the black background or edges (indicating shortcut learning). (b) EfficientNet-B3: attention covers the planet but is diffuse. (c) DINOv2: attention is tightly constrained to the planetary disk and rings, demonstrating object-centric semantic understanding.

5. Conclusions and Discussion

5.1. Summary of Research Findings

This study set out to address the critical challenges of data scarcity, reliability, and safety in applying Deep Learning to astronomical image classification. We proposed a trustworthy classification framework leveraging DINOv2, a state-of-the-art Vision Foundation

Model (VFM) pre-trained via discriminative self-supervised learning. By implementing a robust two-stage transfer learning strategy—comprising linear probing followed by selective fine-tuning—we successfully adapted large-scale features to a manually curated dataset of 700 high-fidelity planetary images sourced from NASA archives. Our comprehensive evaluation against traditional CNN baselines (ResNet50, EfficientNet-B3) and other Transformer architectures yielded the following key findings:

1. **Performance Saturation:** DINOv2 achieved a perfect 100% Top-1 accuracy on the test set, demonstrating faster convergence (reaching saturation within 2 epochs) compared to supervised baselines. While modern CNNs like EfficientNet-B3 also achieved 100% accuracy, legacy architectures like ResNet50 struggled (96.2%), highlighting the necessity of modern feature extractors for fine-grained celestial textures.
2. **Unmatched Reliability:** The most significant contribution of this work is in model calibration. DINOv2 achieved a state-of-the-art Expected Calibration Error (ECE) of 0.08% after temperature scaling. This represents a 36-fold improvement over ResNet50 (2.90%) and a 4.5-fold improvement over EfficientNet-B3 (0.36%), proving that self-supervised ViT features naturally align closer to the true empirical probability distribution.
3. **Operational Safety:** In OOD detection scenarios involving internet-sourced space artifacts, our energy-based detector achieved an AUROC of 93.7%. Qualitative analysis via Grad-CAM confirmed that DINOv2 focuses on intrinsic planetary features (e.g., surface bands, rings) rather than background noise, thereby mitigating the “shortcut learning” observed in ResNet50.

5.2. Theoretical and Practical Implications

Theoretical Implications: We hypothesize that discriminative self-supervised pretraining can induce a representation geometry that is empirically more conducive to downstream uncertainty estimation, in the sense that class structure is captured by more coherent clustering and larger angular separation between classes. Importantly, we treat this as an empirically testable interpretation rather than a theoretical guarantee. To support this interpretation, we complement Grad-CAM with an embedding-space analysis (Supplementary Figure S1), where low-dimensional projections of penultimate representations reveal clearer class-wise clustering for DINOv2 features, consistent with the observed improvements in post hoc calibration.

5.3. Limitations

Despite the promising results, this study has several limitations:

1. **Dataset scale and heterogeneity:** Although the curated dataset is balanced and high-fidelity, it remains small (700 images, 7 classes). As a result, clean-test accuracy becomes saturated for several modern architectures, and robustness conclusions must be interpreted within this benchmark’s scope. We therefore complement clean accuracy with controlled stress tests, but we do not claim guaranteed robustness under mission-specific sensors or large-scale survey distributions. Future work will validate the proposed framework on larger and more heterogeneous datasets (e.g., survey data and instrument-specific imagery) and will consider grouped splits (by mission/instrument) to reduce the risk of source-specific visual signatures leaking across splits.
2. **The OOD set used in this study is sourced from web images and is designed to represent open-world, non-planetary visual confounders** (e.g., nebulae, debris, abstract space imagery). While it is diverse, it does not fully cover sensor-specific artifacts encountered in real missions (e.g., cosmic-ray hits, saturation spikes, and instrument-

dependent noise patterns). Therefore, the reported AUROC should be interpreted as performance on a web-image OOD benchmark rather than a definitive estimate under mission-specific sensor anomalies.

3. **Computational Cost (Latency/Memory Footprint):** The proposed framework leverages DINOv2 (ViT-Base/14) as the strongest backbone, but this choice has clear deployment implications. In terms of model size, ViT-Base/14 contains 86M parameters, corresponding to approximately 344 MB of weights in FP32 (or ~172 MB in FP16), excluding a lightweight classification head. Importantly, inference-time latency and activation memory are strongly affected by input resolution for ViT backbones because self-attention scales quadratically with the token length. With a patch size of 14, the token length is $N = (S/14)^2$. Under our pipeline, DINOv2 operates at $S = 518 (37 \times 37 = 1369 \text{ tokens})$, while the CNN baselines use $S = 224 (16 \times 16 = 256 \text{ tokens})$. This results in $\sim 5.35 \times$ more tokens and $\sim 28.6 \times$ larger quadratic attention cost (relative to 224), which helps explain the higher latency and peak memory footprint of DINOv2 compared to efficient CNN baselines.

5.4. Future Work

Future research will focus on bridging the gap between academic benchmarking and operational deployment along several complementary directions. First, we plan to scale our framework to larger, uncurated survey datasets such as those from the Sloan Digital Sky Survey (SDSS) and the upcoming Vera C. Rubin Observatory, addressing challenges including class imbalance and lower signal-to-noise ratios. To support onboard deployment, we will investigate model compression through knowledge distillation, transferring the robust representations of DINOv2 to lightweight student architectures (e.g., MobileViT or EfficientNet-Lite) while preserving calibration performance. In addition, leveraging the semantic richness of DINOv2 features, we aim to move beyond closed-set classification toward open-set discovery, enabling the model to cluster and identify potentially novel celestial categories (e.g., trans-Neptunian objects) in an unsupervised manner. Finally, we will expand the benchmark along two complementary axes: extending the dataset to additional planetary classes (e.g., Mercury and Venus) and a broader range of missions and instruments, and introducing more physically grounded distribution shifts—such as illumination and phase variations—to better reflect the viewpoint and lighting changes encountered in real planetary imaging. In addition to distillation, we will investigate deployment-oriented compression strategies such as post-training and quantization-aware quantization (e.g., FP16/INT8) as well as resolution-aware inference (trading input size for latency). We will report the corresponding accuracy–calibration–OOD trade-offs to quantify the practical deployment frontier for resource-constrained platforms.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/aerospace13030222/s1>, Figure S1: t-SNE visualization of cached DINOv2 feature embeddings (ID + OOD). Each point denotes one sample in the feature space; colors indicate in-distribution classes (earth, jupiter, mars, moon, neptune, saturn, uranus), and gray “×” markers denote out-of-distribution (OOD) samples. ID samples form class-specific clusters, while OOD samples are more dispersed and often lie between ID clusters.

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

Funding: This work was funded by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2023-00249407) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00336025) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00558871).

Data Availability Statement: The datasets supporting this study are available from the corresponding author upon request.

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

References

1. Ball, N.M.; Brunner, R.J. Data mining and machine learning in astronomy. *Int. J. Mod. Phys. D* **2010**, *19*, 1049–1106. [CrossRef]
2. Fluke, C.J.; Jacobs, C. Surveying the reach and maturity of machine learning and artificial intelligence in astronomy. *WIREs Data Min. Knowl. Discov.* **2020**, *10*, e1349. [CrossRef]
3. Domínguez Sánchez, H.; Huertas-Company, M.; Bernardi, M.; Tuccillo, D.; Fischer, J.L. Transfer learning for galaxy morphology from one survey to another. *Mon. Not. R. Astron. Soc.* **2019**, *484*, 93–100. [CrossRef]
4. Dieleman, S.; Willett, K.W.; Dambre, J. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Mon. Not. R. Astron. Soc.* **2015**, *450*, 1441–1459. [CrossRef]
5. Awais, M.; Dengel, A.; Ahmed, S.; Islam, J.; Schiele, B.; Bulling, A. Foundation Models Defining a New Era in Vision: A Survey and Outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 2245–2264. [CrossRef] [PubMed]
6. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the 38th International Conference on Machine Learning (ICML), Virtual Event, 18–24 July 2021; Volume 139, pp. 8748–8763. Available online: <https://proceedings.mlr.press/v139/radford21a.html> (accessed on 17 January 2026).
7. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.V.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. DINOv2: Learning robust visual features without supervision. *arXiv* **2023**, arXiv:2304.07193. [CrossRef]
8. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16×16 words: Transformers for image recognition at scale. In Proceedings of the 9th International Conference on Learning Representations (ICLR), Virtual Event, 3–7 May 2021; Available online: <https://openreview.net/forum?id=YicbFdNTTy> (accessed on 20 January 2026).
9. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; IEEE: New York, NY, USA, 2021; pp. 10012–10022. [CrossRef]
10. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin Transformer V2: Scaling up capacity and resolution. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; IEEE: New York, NY, USA, 2022; pp. 12009–12019. [CrossRef]
11. Cao, J.; Xu, T.-T.; Deng, Y.-H.; Li, G.-P.; Gao, X.-J.; Yang, M.-C.; Liu, Z.-J.; Zhou, W.-H. Classification of galaxy morphology based on FPN-ViT model. *Chin. Astron. Astrophys.* **2024**, *48*, 683–704. [CrossRef]
12. Cao, J.; Xu, T.; Deng, Y.; Deng, L.; Yang, M.; Liu, Z.; Zhou, W. Galaxy morphology classification based on convolutional vision transformer. *Astron. Astrophys.* **2024**, *683*, A42. [CrossRef]
13. Mohan, D.; Scaife, A.M.M.; Porter, F.; Walmsley, M.; Bowles, M. Quantifying uncertainty in deep learning approaches to radio galaxy classification. *Mon. Not. R. Astron. Soc.* **2022**, *511*, 3722–3740. [CrossRef]
14. Rustige, L.; Kummer, J.; Griese, F.; Borrás, K.; Brüggen, M.; Connor, P.L.S.; Gaede, F.; Kasieczka, G.; Knopp, T.; Schleper, P. Morphological classification of radio galaxies with Wasserstein generative adversarial network-supported augmentation. *RAS Tech. Instrum.* **2023**, *2*, 264–277. [CrossRef]
15. Bowles, M.; Tang, H.; Vardoulaki, E.; Alexander, E.L.; Luo, Y.; Rudnick, L.; Walmsley, M.; Porter, F.; Scaife, A.M.M.; Slijepcevic, I.V.; et al. Radio galaxy zoo EMU: Towards a semantic radio galaxy morphology taxonomy. *Mon. Not. R. Astron. Soc.* **2023**, *522*, 2584–2600. [CrossRef]
16. Vilalta, R.; Dhar Gupta, K.; Bumber, D.; Meskhi, M.M. A General Approach to Domain Adaptation with Applications in Astronomy. *Publ. Astron. Soc. Pac.* **2019**, *131*, 108008. [CrossRef]
17. Yu, X.; Wang, J.; Hong, Q.-Q.; Teku, R.; Wang, S.-H.; Zhang, Y.-D. Transfer learning for medical images analyses: A survey. *Neurocomputing* **2022**, *489*, 230–254. [CrossRef]
18. Ayana, G.; Dese, K.; Abagaro, A.M.; Jeong, K.C.; Yoon, S.-D. Multistage transfer learning for medical images. *Artif. Intell. Rev.* **2024**, *57*, 232. [CrossRef]

19. Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U.R.; et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fusion* **2021**, *76*, 243–297. [[CrossRef](#)]
20. Zhang, M.; Fernández-Torres, M.-Á.; Cohrs, K.-H.; Camps-Valls, G. Calibration and uncertainty quantification for deep learning-based drought detection. *Int. J. Appl. Earth Obs. Geoinf.* **2025**, *140*, 104563. [[CrossRef](#)]
21. Yang, J.; Zhou, K.; Li, Y.; Liu, Z. Generalized Out-of-Distribution Detection: A Survey. *Int. J. Comput. Vis.* **2024**, *132*, 5635–5662. [[CrossRef](#)]

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