Survival of the Fittest Detectors: A Decentralized Framework for Evolving Deepfake Detection

BitMind

Abstract

The swift advancement of generative artificial intelligence technologies demands robust, scalable, and adaptable solutions for deepfake detection. This paper presents the BitMind Subnet, a decentralized framework that leverages economic incentives to foster innovation and competition among researchers. This system effectively translates academic research into practical, deployable tools that proficiently handle a wide array of real-world media challenges. We performed an extensive evaluation using diverse datasets, which included 46,000 real images from Google Images and ImageNet, along with 125,032 synthetic images from a variety of generators. The results reveal that models incentivized within the subnet achieve high classification accuracies, peaking at 98.53%, and exhibit strong detection capabilities on real-world datasets with accuracies up to 91.95%. The framework displays variability in detecting synthetic content from non-incentivized sources like MidJourney and DiffusionDB, where accuracies were notably lower. These findings highlight critical areas for further refinement and underscore the importance of a dynamic, evolving approach. By continuously adapting its focus to address relevant generative models, the subnet remains effective in mitigating emerging threats in the dynamic domain of synthetic media. This ongoing adaptability highlights the BitMind Subnet's capacity to act as a robust safeguard amidst the rapid advancements in generative AI technologies.

1. Introduction

The advent of generative AI has transformed digital media, enabling the creation of increasingly sophisticated synthetic audiovisual content. Today, the term "deepfake" no longer refers merely to manipulated facial imagery but encompasses a spectrum of AI-generated media, from subtle modifications to entirely fabricated scenes indistinguishable from reality. Our collective ability to identify synthetic content has deteriorated rapidly, undermining public trust and reasoned discourse. Widely accessible generative tools enable malicious actors to produce hyper-realistic synthetic content for misinformation campaigns or social engineering. AI-generated imagery fosters pervasive doubt even when created without harmful intent,fostering an environment where all digital information faces inherent skepticism.

These issues are further compounded by fundamental limitations in both academic and commercial deepfake detection development. Academic research focuses on narrow domains—specifically facial manipulation—and remains disconnected from practical, public-facing applications. Meanwhile, commercial detection systems are developed in siloed environments by centralized entities, limiting transparency and accessibility. Conventional academic approaches rely heavily on static datasets that quickly become outdated as new generative techniques emerge. Meanwhile, sophisticated proprietary detection methods remain inaccessible and inscrutable to the public due to their centralized development and closed-source nature. Consequently, there is an emerging need for solutions that bridge the gap between academic innovation and practical implementation, while benchmarking performance against dynamic, internet-sourced data that reflects real-world conditions.

Decentralized AI offers a compelling alternative for deepfake detection by fostering an open, competitive ecosystem by design. Instead of relying on a single entity to develop and maintain detection algorithms, a decentralized framework incentivizes a network of independent teams to iteratively refine detection models. Incentive-driven approaches to accelerating AI progress have historically proven successful: for instance, the Netflix Prize [2] competition galva-

nized a global community of researchers to improve collaborative filtering techniques for movie recommendations; the recent Frodobot framework [**?**] illustrated how a decentralized network can produce research data for embodied AI by rewarding those who participate in a gamified robotics experience; and perhaps most relevant, Meta's DeepFake Detection Challenge [12], which awarded a bounty of $1 million USD in aggregate to participants who submitted the top 5 models, spurred thedevelopment of novel detection architectures. These precedents highlight the potential of competitive frameworks and provide inspiration for our incentivized, decentralized deepfake detection system.

The solution presented in this paper leverages Bittensor [51], a foundational blockchain network designed for decentralized machine learning. Bittensor provides an infrastructure where specialized subnetworks can be established, enabling custom tasks such as deepfake detection to be carried out by incentivized participants. A Bittensor *subnet* is a dedicated collection of nodes within this broader network, where *miners* develop and deploy machine learning models and *validators* continually assess their performance according to protocols established by the *subnet owners*. By leveraging open-source principles, economic incentives, and blockchain technology, our approach creates a permissionless environment where detection models continuously evolve based on real-world performance metrics, driving long-term innovation in AI-driven security and media authentication.

In this paper, we present the BitMind Subnet: a functioning decentralized network for deepfake detection. Our key contributions include: (1) a novel decentralized framework that leverages blockchain incentives to continuously improve detection capabilities through competitive collaboration; (2) innovative video and image detection approaches developed by a leading miner on the network; (3) comprehensive benchmarking results from diverse datasets, including both established open-source collections and newly curated internet-scraped content, which validate our system's effectiveness across varied real-world scenarios; and (4) freely accessible consumer applications[1] built on this subnet, including browser extensions and messaging platform integrations, which deliver immediate public utility. This work not only advances the technical capabilities of deepfake detection but also establishes a sustainable, adaptable system that bridges the gap between research innovation and practical deployment in the ongoing challenge of synthetic media authentication.

2. Related Work

2.1. Deepfake Detection Methods

Modern approaches typically frame deepfake detection as a supervised learning problem, training binary classifiers to distinguish between real and synthetic visual data. Image-based models often rely on convolutional neural networks (CNNs) to extract spatial features [36, 43], with some studies extending into frequency or wavelet domain analysis to capture generative artifacts [25, 28] or taking a hybrid approach that processes both spatial and frequency-based features in separate branches [18].

Several studies have explored detection systems that make use of foundation models like CLIP [34]. Notably, [31] was the first to leverage CLIP alongside linear and nearest-neighbor classifiers, observing that features from foundation models *not specifically trained for deepfake detection* exhibit surprisingly strong generalization capabilities. Subsequent works expanded on this by investigating data augmentation techniques [39] and incorporating additional CLIP-family models with various classifier layers for video-based detection [4].

Another branch of research approaches deepfake detection through the lens of disentanglement learning, a machine learning paradigm that aims to decompose complex feature spaces into distinct, well-defined components [1]. This method has been applied to separate forgery-related features from irrelevant ones to improve classifier performance; however, these studies predominantly focus on human face datasets [16, 52]. More recently, Yan et al. extended this approach by further disentangling forgery-related features into common and architecture-specific components to improve generalization [48]. The widely-cited DeepfakeBench, also introduced by Yan et al., offers a comparative analysis of 15 state-of-the-art detection models along with standardized, publicly available training and evaluation code, some of which we adapted to provide a starting point for our subnet participants [49].

[1] https://bitmind.ai/apps

Beyond detection methods, industry leaders have established a watermarking paradigm that embeds markers directly into media content at the point of creation. Notable examples include Stable Signature [13] by the Fundamental AI Research (FAIR) at Meta, and SynthID by Google DeepMind [9]. But while this strategy and the broader industry trend it reflects ensures authenticity from the outset, the glaring fallback is that users can simply avoid AI generators applying watermarks and opt for any number of proliferating open-source alternatives.

2.2. The Emergence of Decentralized AI Networks

While traditional deepfake detection methods have made significant progress, they face fundamental limitations in practical deployment scenarios due to their centralized development approach. These limitations include: (1) reliance on static training datasets that quickly become outdated as new deepfake techniques emerge; (2) vulnerability to adversarial attacks targeting a single model; and (3) inability to rapidly incorporate emerging detection techniques from the global research community.

Beyond these methods, decentralized AI has emerged as a compelling paradigm for collective model training, partially reflected in federated learning approaches and competitive platforms like Kaggle. However, few have explored blockchain-based economic incentives for continual model improvement. Unlike traditional centralized approaches where a single entity controls the entire model development process, decentralized AI leverages distributed computing and economic incentives to create more adaptive and resilient systems.

Bittensor [51] provides one of the first functional frameworks for decentralized machine learning, offering an infrastructure for hosting specialized "subnets" that define custom tasks, rewarding participants based on performance. This framework employs a blockchain-based incentive structure where computational contributions that enhance the network's overall intelligence are rewarded with the blockchain's native cryptocurrency tokens (TAO). This design creates a dynamic ecosystem where models are continually tested and ranked by validators.

2.3. The Gap: Keeping Pace with Generative AI Innovation

By design, static detection frameworks inevitably fall behind the evolving capabilities of generative AI, despite significant advances in deepfake detection methods. Open-source AI plays a critical role in addressing this challenge, providing the transparency, agility, and collaborative environment necessary to swiftly adapt to these rapidly evolving generative techniques. However, innovation alone isn't sufficient; economic incentives are essential to motivate continuous improvement and proactive engagement from researchers and developers. In a market-driven environment, aligning incentives with performance encourages participants to rapidly iterate and improve detection models. The BitMind Subnet directly addresses this gap by combining the transparency and agility of open-source AI with the economic incentives, real-time competitive ecosystem of Bittensor uniquely positioned to keep pace with cutting-edge generative AI.

3. Method

3.1. Decentralized AI Framework for Deepfake Detection

The BitMind Subnet is a specialized network on the Bittensor blockchain that incentivizes deepfake detection research. Participants (miners) deploy detection models, while validators score these models on curated real and synthetic media. High-performing miners earn cryptocurrency rewards, creating a competitive environment that drives continuous improvement in model generalization and robustness. As generative AI evolves, so do the evaluation challenges - models that fail to generalize to data from emerging techniques receive diminishing rewards and are ultimately deregistered from the network, creating selective pressure for adaptable solutions.

The BitMind Subnet provides immediate real-world utility by processing classification requests from multiple downstream applications, including a browser extension, drag-and-drop web application, and various messaging platform bots (Discord, X, Telegram). This organic traffic is handled by a proprietary cloud system, featuring an API server and load balancer that distributes requests among validators based on geographic location and current validator load. This user-generated traffic naturally lacks ground truth labels and therefore generates no rewards for miners. Instead, these requests are deliberately designed to be indistinguishable from the scored challenges, ensuring that miners

respond to them despite the lack of incentive to do so.

The subnet's architecture includes the following key components:

- **Miners** deploy detection models and compete to achieve optimal classification accuracy across the evolving challenge set
- **Validators** execute standardized evaluation protocols to rank miners based on their performance
- The **incentive mechanism** defines the criteria for evaluating and rewarding miner outputs
- The **subnet owner** maintains the subnet implementation, specifying both the computational tasks for miners and the evaluation criteria for validators

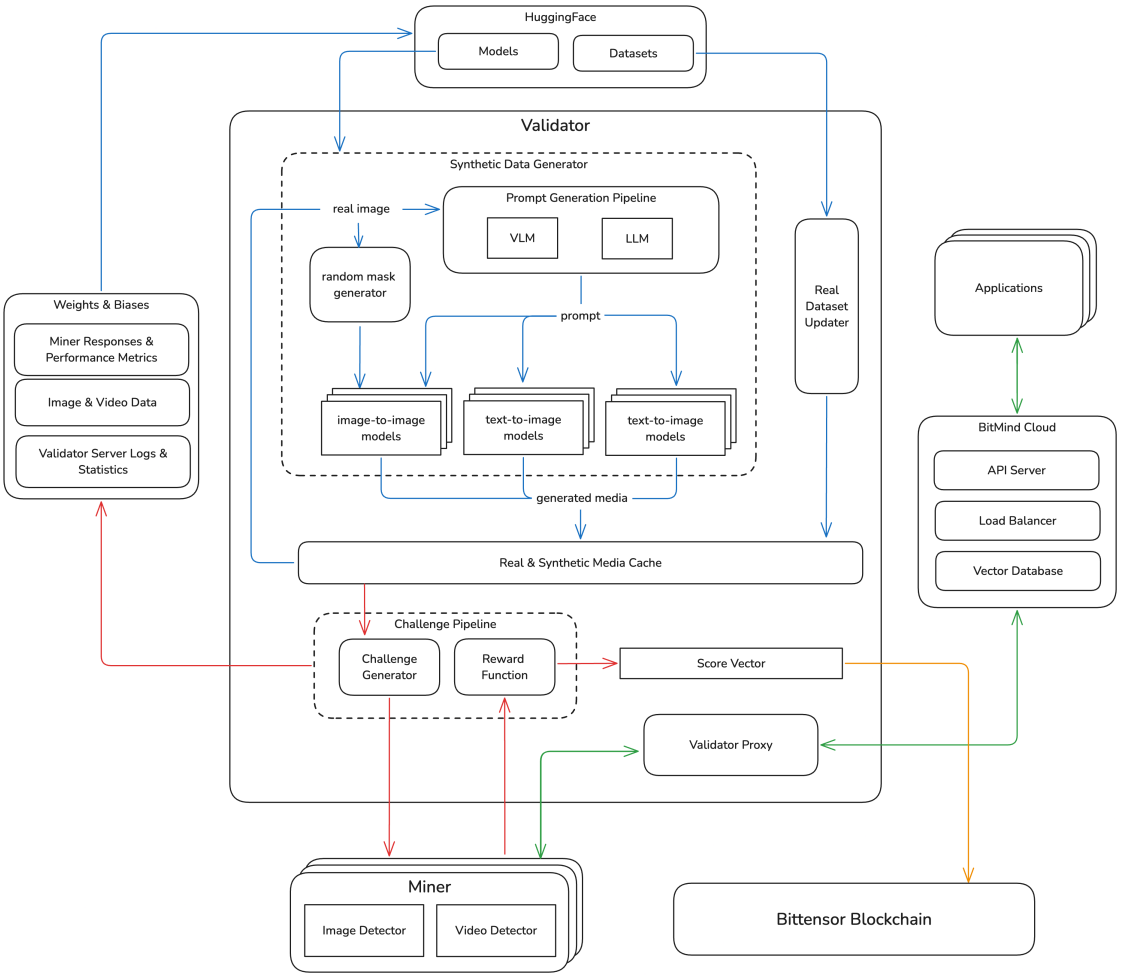

Figure 1. Comprehensive subnet architecture overview. Blue arrows show how validators maintain a cache of real and synthetic data for challenging miners. Pink arrows show how validators issue challenges and score miner responses. Green arrows show how our applications interact with the subnet for image and video classification. The core components shown in this diagram are covered extensively in the remainder of section 3.

3.2. Incentive Mechanism Design

The incentive mechanism relies on a dynamic dataset infrastructure that combines curated open-source data with continuously generated synthetic media. At the core of this infrastructure is a high-capacity cache maintained by validators, who use it to issue classification challenges to miners at one-minute intervals. These challenges test miners' ability to distinguish authentic and AI-generated content across both image and video modalities, with each challenge

subjected to random augmentations that further increase semantic diversity.

Our data strategy operates along two parallel tracks. The first track sources from a diverse collection of open-source datasets, carefully selected to represent the broad spectrum of real-world media content. The second track uses our data generation pipeline, orchestrating two key components: a prompt generation submodule, and a suite of fifteen state-of-the-art generative image and video models.

3.2.1. Data Generation Pipeline

Our synthetic data generator, presented by Figure 2, operates as a continuous, asynchronous process on each validator node in the network. The pipeline employs a two-stage architecture: a prompt generation stage that combines vision-language (VLM) and large language models (LLM) to create diverse, semantically rich prompts, followed by a content generation stage that leverages these prompts across various generative models. For image-to-image models, the pipeline executes an additional masking phase that dynamically designates a region for generative filling. This region is inpainted using the original image's caption as a semantic guide in order to maintain visual coherence with the surrounding image context, resulting in modifications that are challenging to detect through casual observation. This continuous generation process provides validators with a fresh and diverse cache of synthetic content, which is essential for maintaining the challenge system's effectiveness in evaluating detection models deployed by miners.

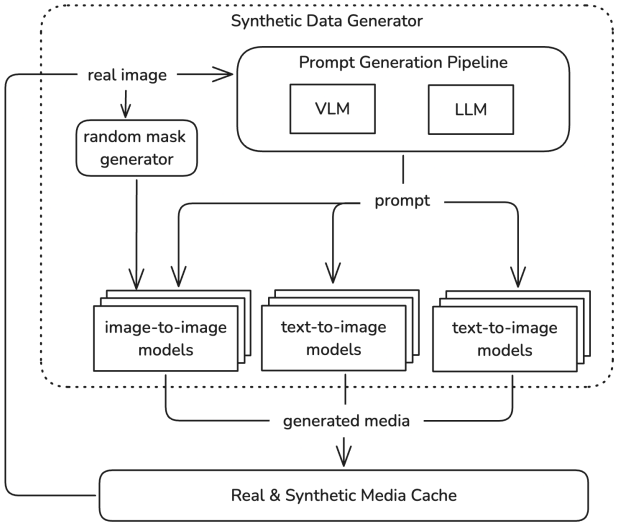

Figure 2. The synthetic data generation pipeline. The generated images and videos are stored in a cache, which serves as a sampling source for generating challenges for subnet miners.

The prompt generation stage uses BLIP2-OPT-6.7B-COCO [22] as its VLM and Llama-3.1-8B-Instruct [14] as its LLM, both optimized for efficiency through 4-bit quantization. The VLM first generates concise captions for images sampled from the validator cache. These initial captions are then refined by the LLM, which enriches them with additional details while ensuring semantic coherence and, for video generation, incorporating temporal and motion-related elements. The prompt generation stage leverages batching to minimize GPU I/O, producing several prompts before relinquishing the VRAM to downstream generative models. In the case of a validator cold start, this batch size is set to 1 in order to minimize the time it takes to initialize the cache with diverse synthetic data. While this pipeline architecture can scale horizontally with additional GPUs, our default configuration targets a single 80 GB VRAM GPU to maintain accessible validation costs.

Our data generation pipeline's design prioritizes extensibility to incorporate emerging generative AI techniques. This adaptability is implemented through a generalized class that can load and invoke any model from the Huggingface Diffusers package [45]. Adding new models to the challenge distribution requires only a Python dictionary entry specifying the model's Huggingface URL and any necessary loading or invocation parameters. Table 1 presents the

(a) Authentic image from the FFHQ dataset [19]

(b) Synthetic "mirror" generated by Flux.1-dev [3]

Figure 3. Example input and output of our synthetic data generation pipeline, where figure 3a is the sampled input image, and figure 3b is the output generated by Flux-1.dev [3]. The output of the prompt generation submodule for this image is "*A baby lies on a blue blanket in a sunny setting, surrounded by a blue background, in a portrait view.*"

complete suite of generative models currently deployed in our pipeline.

Model	Model Task
SDXL Base 1.0 [32] [42]	Text-to-Image
RealVisXL V4.0 [38]	Text-to-Image
Mobius [8]	Text-to-Image
FLUX.1 [3]	Text-to-Image
OpenJourney V4 [33]	Text-to-Image
Animagine XL 3.1 [20]	Text-to-Image
DeepFloyd IF [10] [11]	Text-to-Image
Janus Pro 7B [5]	Text-to-Image
Dreamshaper-8-Inpainting [29]	Image-to-Image
SDXL Inpainting [41]	Image-to-Image
SD1.5 Inpainting [35]	Image-to-Image
Hunyuan Video [47]	Text-to-Video
Mochi-1 [44]	Text-to-Video
CogVideoX-5b [50]	Text-to-Video
AnimateDiff Lightning [23]	Text-to-Video

Table 1. Generative models deployed in our synthetic data generation pipeline.

3.2.2. Open-Source Datasets

The system maintains a parallel data track alongside the synthetic generation pipeline, managing a dynamic cache of authentic images and videos. This process implements an efficient approach that downloads randomly selected compressed segments from large open-source datasets. Through selective partial extraction, the system accesses individual media files within compressed archives without requiring full decompression. This optimization enables the

challenge generation pipeline to be fed highly diverse authentic data from multiple sources while minimizing storage and bandwidth requirements. The approach is particularly valuable for incorporating large datasets like OpenVid1m, which would otherwise demand terabytes of storage capacity and substantially increase validator initialization times.

Dataset	Description
Open Images V7 [27]	Large-scale dataset with ∼9M images across 19.8K classes
CelebA-HQ [26]	High-quality version of CelebFaces dataset with 30K celebrity face images at 1024×1024 resolution
FFHQ-256 [19]	70K high-quality face images at 256×256 resolution, diverse in age, ethnicity and image background
MS-COCO [24]	Large-scale object detection dataset with 330K images containing everyday objects in their natural context
AFHQ [6]	Animal Faces-HQ dataset with ∼15K high-quality images of cats, dogs and wildlife faces
LFW [17]	Labeled Faces in the Wild dataset with 13K images of faces collected from the web
Caltech-256 [15]	30K images across 256 object categories
Caltech-101 [21]	9K images across 101 object categories
DTD [7]	Describable Textures Dataset with 5640 images organized by 47 terms describing textures
OpenVid-1M [30]	Large-scale open source video dataset with 1M video clips
ImageNet-VidVRD [40]	Video visual relationship detection dataset based on ImageNet

Table 2. Open source datasets used for model evaluation

3.2.3. Data Augmentation

Data augmentation introduces essential semantic diversity to validator challenges by applying a broad set of transformations to sampled images and videos. Each challenge sample undergoes randomly selected transformations from one of four difficulty levels (0–3). These levels progressively increase in complexity and variety, spanning basic geometric changes (e.g., flips and rotations) to more severe distortions such as heavy JPEG compression, color manipulations, and additive noise.

Motivation. Malicious actors attempting to evade deepfake detection often exploit consistent patterns in training data, or apply post-processing (e.g., compression, color shifting) that weakens detection. By systematically varying how images and videos are augmented, we discourage miners (model providers) from overfitting to narrow artifacts. Models must instead learn features robust to real-world factors like random camera angles, lighting shifts, compression artifacts, and even partial occlusions.

Difficulty Levels.

- **Level 0 (Baseline):** Minimal transformations, applying only `CenterCrop` and `Resize` operations to standardize input dimensions. This ensures that **all** samples at least have consistent shapes for downstream processing, but no additional perturbations are applied.
- **Level 1 (Geometric Transformations):** Introduces random rotations up to 20°, random cropping (via `RandomResizedCrop`), and random flips (horizontal/vertical). These simulate typical real-world capture variations, such as different camera orientations or slight user-applied flips.
- **Level 2 (Color & Compression Adjustments):** Builds on the geometric transformations by further altering color saturation (`CS`), color contrast (`CC`), and JPEG compression quality. These perturbations reflect common post-processing or device-specific pipelines—e.g., a social media platform re-encoding an image at lower quality, or a user applying a mild filter.
- **Level 3 (Advanced Noise & Blur):** The most challenging tier, adding Gaussian noise (`GNC`) and Gaussian blur (`GB`) in addition to color transformations and compression. This replicates extreme cases where an image may be heavily compressed, intentionally blurred, or otherwise degraded. Such distortions test whether detection models

remain robust to low-quality media frequently encountered in real user submissions (e.g., older device cameras, multiple re-uploads).

Parameter Tuning. The intensity of each transformation is governed by the parameters in Table 3. These parameters gradually increase from Level 0 to Level 3. For instance, *JPEG Compression* might be applied lightly in Level 2 (`[0,1]` range), but more severely in Level 3 (`[0,2]` range). Similarly, *Gaussian Blur* (`GB`) introduces wider kernel sizes at higher levels, simulating more significant image degradation.

	Intensity Level			
Augmentation	**0**	**1**	**2**	**3**
CenterCrop	Yes	–	–	–
Resize	Yes	–	–	–
RandomRotation	–	20°	20°	20°
RandomResizedCrop	–	(0.2, 1.0)	(0.2, 1.0)	(0.2, 1.0)
RandomHorizontalFlip	–	Yes	Yes	Yes
RandomVerticalFlip	–	Yes	Yes	Yes
Color Saturation (CS)	–	–	[0, 1]	[0, 2]
Color Contrast (CC)	–	–	[0, 1]	[0, 2]
JPEG Compression	–	–	[0, 1]	[0, 2]
Gaussian Noise (GNC)	–	–	–	[0, 2]
Gaussian Blur (GB)	–	–	–	[0, 2]

Table 3. Image augmentation parameters across difficulty levels. "Yes" indicates the transform is applied, while numeric ranges (e.g., [0,2]) represent sampling intervals for intensities or severity factors.

Overall, this hierarchical augmentation scheme ensures an increasingly complex and realistic challenge set, encouraging miners to develop models that can handle diverse visual conditions and thwart emerging adversarial tactics.

3.2.4. Mitigating Reverse Image Lookup Attacks

In addition to improving model robustness, the image augmentations described above also hinder adversarial attempts at matching transformed images back to their original versions via embedding-based searches. By introducing random rotations, crops, flips, and moderate color or compression distortions, each image in the dataset can take on multiple visually distinct forms. This expansion makes it much harder for an attacker to index or retrieve every possible variant of a given image.

Milvus Vector Search Testing. To quantify the impact of these augmentations on reverse lookups, we created a standalone Milvus vector database with 768-dimensional embeddings (from a ViT-based encoder) of 100,000 unaltered "raw" images. We then produced one augmented version per raw image—using the flips, rotations, crops, etc. outlined above—and queried each augmented embedding against the database. We checked whether the database returned the original unaltered image as the top result.

- **Without Additional Distortions:** Only about 42% of queries correctly matched their original. Thus, 58% of augmented images *evaded* top-1 recognition by embedding lookup.
- **With Mild Distortions:** When further layering mild color changes, JPEG compression, blur, or noise on top of geometric augmentations, the top-1 match rate dropped to around 17.55%.

These findings confirm that layering multiple transformations significantly complicates a naive embedding-based reverse lookup. To further explore how a deliberately adversarial miner might fare, we simulated a miner that calculates the embedding for each transformed image, compares it to all raw embeddings, and then applies a threshold-based decision:

1. Compute the distance to the nearest embedding.
2. Compute the distance to the second-closest embedding.
3. If the gap between them exceeds a learned threshold T, declare it a match; otherwise, label it a non-match.

Following the methodology in our vulnerability tests, we set T as in Eq. 1:

$$T = \max\Big(\text{median}_{\text{non\_match}} + c \times \text{IQR}_{\text{non\_match}},\ P_{25,\text{match}}\Big), \tag{1}$$

where $c = 0.5$. Under these transformations (without advanced distortions), the adversarial miner achieved only 36.75% accuracy in recovering originals; once mild distortions were added, performance dropped further, to 34.15%.

Storage and Scalability Barriers. Because each base image can spawn tens or hundreds of plausible variants, the total embedding count rapidly grows into the billions when dealing with large datasets. Storing or indexing these at scale (and searching them in real time) incurs prohibitively high costs, effectively deterring brute force or systematic embedding-based attacks.

Future Directions. Ongoing work seeks to:
- Extend these transformations with domain-specific distortions (e.g., block artifacts, inpainting).
- Investigate "proof of inference" methods that ensure miners' outputs arise from genuine model inference rather than pure database lookups.
- Experiment with alternative embedding types (e.g., local feature embeddings or geometric encoders) to anticipate more advanced retrieval systems.

By continually updating our augmentation strategies, we reduce the likelihood that any single embedding-based attack could reliably match transformed images to their unmodified originals.

3.3. Reward Distribution

Validators send challenges to miners on a one-minute interval and use a reward vector to track their perfomance over time. Every 72 minutes, (360 blocks, where a block is approximately 12 seconds), validators assign weights that determine how mining rewards are distributed through a normalized exponential moving average (EMA) of this reward vector.

A miner i's instantaneous reward C_i is calculated by combining their performance across both modalities (image and video) and scoring types (binary and multiclass).

$$C_i = \sum_{m \in \{image, video\}} p_m \sum_{k \in \{b,m\}} w_k MCC_{km} \tag{2}$$

For each modality m, weighted by configurable parameter p_m, the reward incorporates both binary (b) and multi-class (m) Matthews Correlation Coefficient (MCC) scores, weighted by w_k. The binary MCC measures accuracy in distinguishing synthetic from authentic content, while the multiclass MCC evaluates the more granular classification of fully- and semi-synthetic content. Through the parameters p and w, the relative importance of each modality and classification type can be adjusted.

To create strong incentives for excellence, we apply a rational transform that dramatically amplifies rewards for top-performing miners. This transform maps raw performance scores to final rewards through a hyperbolic function:

$$C_i^* = \mathcal{R}\left(C_i; p\right) \tag{3}$$

where we define the rational transform $\mathcal{R} : \mathbb{R} \times \mathbb{R}+ \to \mathbb{R}+$ is defined as

$$\mathcal{R}(x; p) = \frac{1}{p - x} \tag{4}$$

During each challenge period *t*, validators evaluate a randomly sampled pool of fifty miners, computing their rewards as described above. These instantaneous rewards are then incorporated into the validator's score vector *V* using an exponential moving average (EMA):

$$V_t = 0.05 \cdot C_t + 0.95 \cdot V_{t-1} \tag{5}$$

The low learning rate ($\alpha = 0.05$) in the EMA calculation ensures that scores reflect consistent performance rather than temporary fluctuations, promoting stability in the reward mechanism.

At longer intervals (every 360 blocks), validators set miner weights by submitting normalized score vectors to the Bittensor blockchain:

$$w = \frac{\mathrm{V}}{\|\mathrm{V}\|_1} \tag{6}$$

These individual validator assessments are then aggregated through Bittensor's Yuma Consensus algorithm to determine final miner incentives. YC computes a rank *R* for each miner *k* by weighting the collective assessment matrix *W* with validator stake vector *S*. The collective weight matrix $W \in \mathbb{R}^{n \times m}$ comprises these normalized score vectors from all n validators across m miners, where each row $W_i = w_i$ represents an individual validator's assessments.

$$R_k = \sum_i S_i \cdot W_{ik} \tag{7}$$

Yuma Consensus enforces alignment among validator weights by penalizing those that deviate too far from consensus. This ensures uniformity in performance evaluation and prevents potential collusion between miners and validators.

Finally, each miner's incentive *I* is calculated as their proportional share of the sum of all miner ranks:

$$I_j = \frac{R_j}{\sum_k R_k} \tag{8}$$

4. Generalized Deepfake Detection by Miners

This section describes the practical workflow through which miners submit models, optimize methodologies, and receive rewards within the decentralized subnet. Miners actively develop and submit deepfake detection models, either building upon existing base models or introducing novel model architectures. Their primary goal is to improve data collection processes, refine training methodologies, and improve overall classification performance on subnet-specific content.

4.1. Baseline Deepfake Detection Models

To accelerate miner onboarding, we provide pre-trained state-of-the-art deepfake detection models as well as training and evaluation code. These baselines serve as starting points that miners can deploy or fine-tune, and were especially important in lowering the barrier to entry and bootstrapping the network when it was in its infancy.

UCF (CVPR 2023) Generalizing across forgery types remains a core challenge in deepfake detection, as many models overfit to dataset-specific artifacts. UCF [48] mitigates this by disentangling image information into forgery-irrelevant features, method-specific artifacts, and common forgery features. It employs a multi-task learning strategy with multi-class and binary classification, alongside contrastive regularization to separate common forgery signatures from generation artifacts. A conditional decoder reconstructs images using extracted features, reinforcing the model's ability to generalize.

By focusing on method-agnostic forgery cues, UCF improves cross-method generalization and outperforms prior baselines on unseen deepfake techniques. Its reliance on common forgery patterns rather than specific dataset biases makes it a strong foundation for miners' detection models.

NPR (CVPR 2024) Deepfake detection methods often overlook the architectural artifacts introduced by GAN and diffusion-based generators. NPR [43] (Neighboring Pixel Relationships) addresses this by analyzing pixel interdependencies caused by up-sampling operations, rather than relying solely on frequency-based artifacts. This enables more robust detection across generative techniques.

Tested on a dataset spanning 28 generative models, NPR outperforms state-of-the-art baselines by 11.6%. Its architecture-focused approach enhances resilience to style variations and method-specific distortions, making it a highly adaptable detection model.

CAMO: Modular Orchestration Framework Beyond individual models, we provide CAMO (Configurable Adaptive Model Orchestrator), a framework that dynamically routes images to specialized detection backends. For instance, CAMO directs face-containing images to a face-optimized version of UCF, while other content may be processed by NPR. This modular approach allows miners to integrate UCF, NPR, or custom models to optimize detection based on image characteristics.

The subsequent section presents empirical results achieved by top-performing miners, illustrating how effectively this approach translates into practical outcomes. For detailed methodologies, specific model architectures, and additional technical insights from leading miners, refer to the supplementary material provided.

5. Results

To assess the generalization capabilities of BitMind's decentralized detection framework beyond incentivized challenges, we conducted comprehensive benchmarking using a variety of data sources. This included open-source datasets, curated real and synthetic data from multiple internet sources, and custom-generated synthetic datasets tailored to mimic real-world imagery. Notably, our evaluation avoided data directly used within the subnet's incentive mechanism to prevent bias from miners optimizing solely for these cases.

5.1. Data Collection

For this evaluation, we collected 23,000 real images from Google Images and ImageNet, and generated three synthetic counterparts for each image using SDXL, Flux, and RealVis. Real images from Google Images were scraped using a custom search engine query generator, which leveraged the Llama-3.1-8B-Instruct-bnb-4bit model by Meta to produce unique search terms across 24 categories. These categories span tangible subjects (e.g., "country," "animal") and more abstract concepts (e.g., "culture," "time"), ensuring a broad representation of real-world imagery. To avoid contamination from AI-generated images, we applied a strict date filter, selecting only image candidates created before 2018, predating most consumer AI image generation solutions such as Midjourney (released July 12, 2022). In addition, our evaluation incorporated AI-generated images from DiffusionDB and included synthetic images scraped from Reddit and Flickr, further expanding the diversity of generative models under examination. Overall, this process yielded over 69,000 synthetic images that semantically mirror real images, enabling direct performance comparisons across different generative models.

5.2. Classification Strategies

Each image was classified by multiple miners, and we applied three different aggregation strategies to derive a final label per sample:

1. **Mean:** Average of miner outputs $p_i \in [0, 1]$ (where $p_i = 1$ suggests "synthetic" and $p_i = 0$ suggests "real"); a sample is labeled synthetic if its mean score is ≥ 0.5.
2. **Majority Vote:** Each output is binarized at 0.5, with the majority class determining the label.
3. **Capped Mean:** Similar to Mean but limits consecutive identical values to a maximum of five to mitigate outlier influence.

These methods were rigorously applied to ensure robust classification across our datasets.
Table 4 presents the accuracy results (in percentages) for each dataset under these three aggregator strategies.

5.3. Observations

The following insights emerge from our data analysis, substantiated by the accuracy metrics in Table 4:

- **Real-Image Accuracy:** Google Images and ImageNet showed robust classification accuracies, achieving up to 90.97% and 91.95%, respectively, across different aggregation methods. This indicates a high capability of the system to recognize authentic content.
- **Synthetic Data Variation:** Synthetic image mirrors of real images generated by incentivized models like SDXL achieved very high classification accuracies (up to 98.53%), demonstrating effective miner training. However, non-incentivized datasets like Midjourney and DiffusionDB presented significant challenges, with the best accuracies reaching only 42.62% and 50.61%, respectively, indicating difficulty in generalizing to these out-of-distribution models.
- **Out-of-Distribution Challenges:** The DiffusionDB and Midjourney datasets present greater challenges: DiffusionDB exhibits near-random chance classification performance, with Mean (48.65%), Majority Vote (48.42%), and

Dataset	Type	Sample Size	Mean (%)	Majority Vote (%)	Capped Mean (%), $n = 5$
Google Images	Real	23,000	90.84	**90.97**	90.17
ImageNet [37]	Real	23,000	91.74	91.72	**91.95**
Flux Mirrors	Synthetic	23,000	87.23	86.93	**88.16**
RealVis Mirrors	Synthetic	23,000	95.60	95.55	**95.81**
SDXL Mirrors	Synthetic	23,000	98.27	98.23	**98.53**
DiffusionDB [46]	Synthetic	23,000	48.65	48.42	**50.61**
Midjourney	Synthetic	23,000	39.16	37.98	**42.62**
Flickr	Synthetic	9,728	81.32	81.31	**82.06**
Reddit	Synthetic	1,304	75.84	75.08	**77.91**

Table 4. Accuracy results comparing Mean, Majority Vote, and Capped Mean aggregator strategies for each dataset. Highest accuracy per dataset is **bolded**.

Capped Mean (50.61%) struggling to produce reliable results. Midjourney is also less performant, reaching a maximum accuracy of just 44.62%. These findings highlight that miners excel on incentivized data but struggle with out-of-distribution cases in contrast to datasets generated by in-distribution models, demonstrating the challenge system's effectiveness in guiding training efforts toward subnet-relevant generative models.

However, it is not optimal to keep every model in the incentivization pool indefinitely, as models inevitably fall out of favor in the wild or become obsolete. Continuing to allocate incentives toward outdated or rarely used models can lead to inefficient training and misaligned subnet priorities. Instead, the subnet must dynamically update its incentivized challenges to prioritize generative models that remain relevant in active use, ensuring that classification capabilities are always aligned with real-world distributions.

5.4. Miner Prediction Distributions

To further elucidate the differentiation in detection capabilities between in-distribution and out-of-distribution synthetic images, we analyze miner prediction distributions from various datasets. Figure 4 displays the prediction density across two categories: real image datasets and synthetic image datasets.

- **In-Distribution Synthetic Images:** The SDXL, Flux, and RealVis mirror datasets, which are part of the incentivized training data, show a distinct peak near the prediction value of 1. This indicates a high confidence among miners in classifying these images as synthetic. The tight clustering around high prediction values suggests effective training and adaptation to the characteristics of these generative models.
- **Out-of-Distribution Synthetic Images:** The datasets such as MidJourney and DiffusionDB, which are not part of the incentivized training, exhibit significantly broader and lower peaks in the distribution, particularly around the classification threshold. This indicates a higher level of uncertainty and lower confidence in classification decisions.

5.5. Miner Consensus

To provide insights into the consensus among miners on the classification of real versus synthetic images, we analyzed miner agreement across various datasets. The results are visualized in Figure 5, which shows the percentage of miners that agree on the classification of each dataset.

5.5.1. Observations from Miner Agreement Visualization

The visualization (refer to Figure 5) indicates:

- High agreement among miners in classifying images from in-distribution synthetic datasets such as SDXL Mirrors, Flux Mirrors, and RealVis Mirrors, where agreement percentages are remarkably high (above 99%).
- Conversely, out-of-distribution datasets such as DiffusionDB and MidJourney show lower agreement rates (96.0% and 98.7% respectively). This disparity underscores the challenge in achieving consensus among decentralized miners when faced with synthetic images not covered by the training incentives.
- The lower agreement rates for out-of-distribution datasets highlight the difficulty in maintaining consistent classification accuracy, reflecting the need for adaptive training approaches to include a wider variety of synthetic generation models.

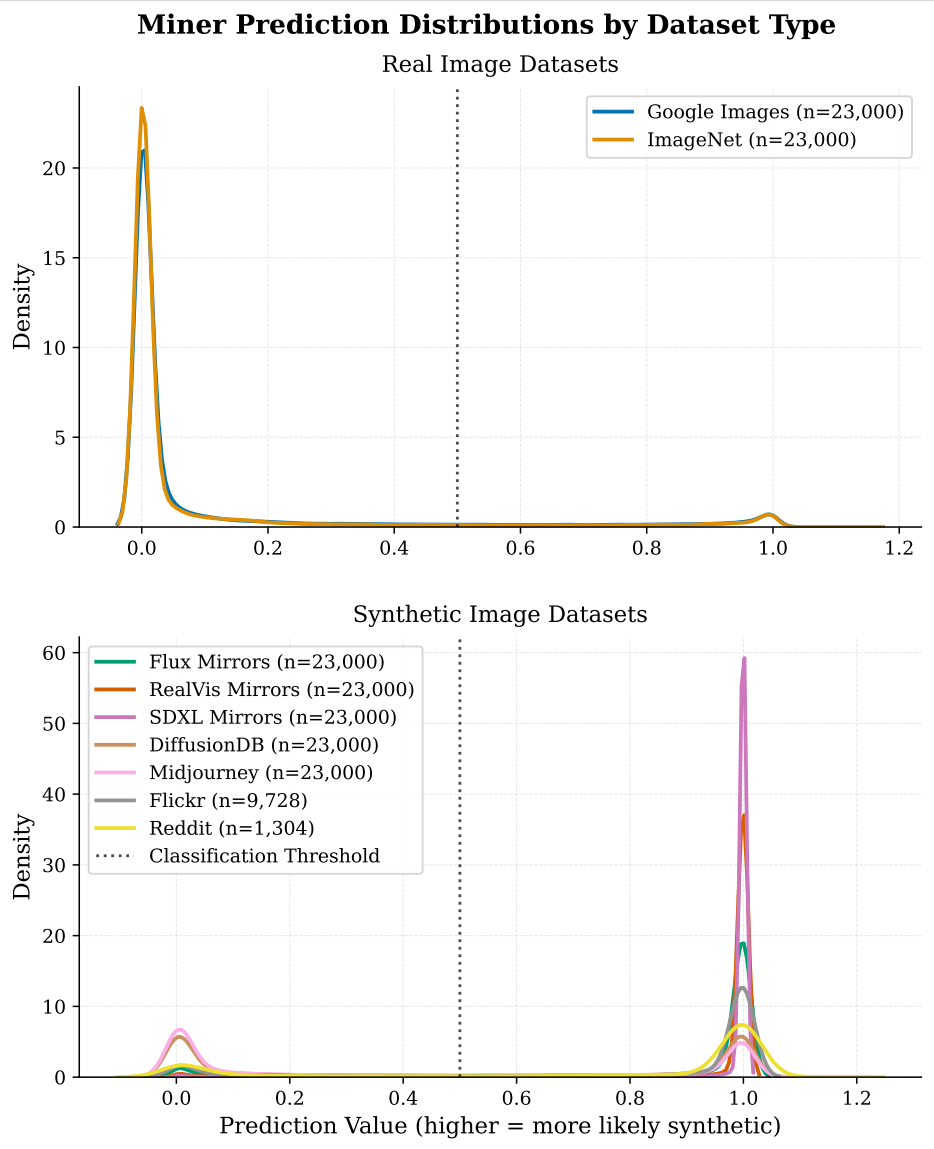

Figure 4. Density plots illustrating the distribution of miner predictions for real and synthetic image datasets, highlighting the differences in detection capabilities for in-distribution and out-of-distribution synthetic images.

6. Discussion

Our competitive, decentralized approach to deepfake detection creates an essential proving ground for innovation in this critical domain. By incentivizing contributors to refine detection algorithms across a broad range of diverse data sources, the subnet enforces the generalization necessary for a detection system to be useful in real-world deployment scenarios. The dynamic nature of our incentive mechanism allows the subnet to evolve alongside generative AI, and the extensibility of our generalized data generation infrastructure ensures that this happens quickly and seamlessly, reducing the risk of obsolescence in the face of novel generation techniques. Furthermore, the transparent evaluation process facilitated by the blockchain ensures the integrity and fairness of the model assessments.

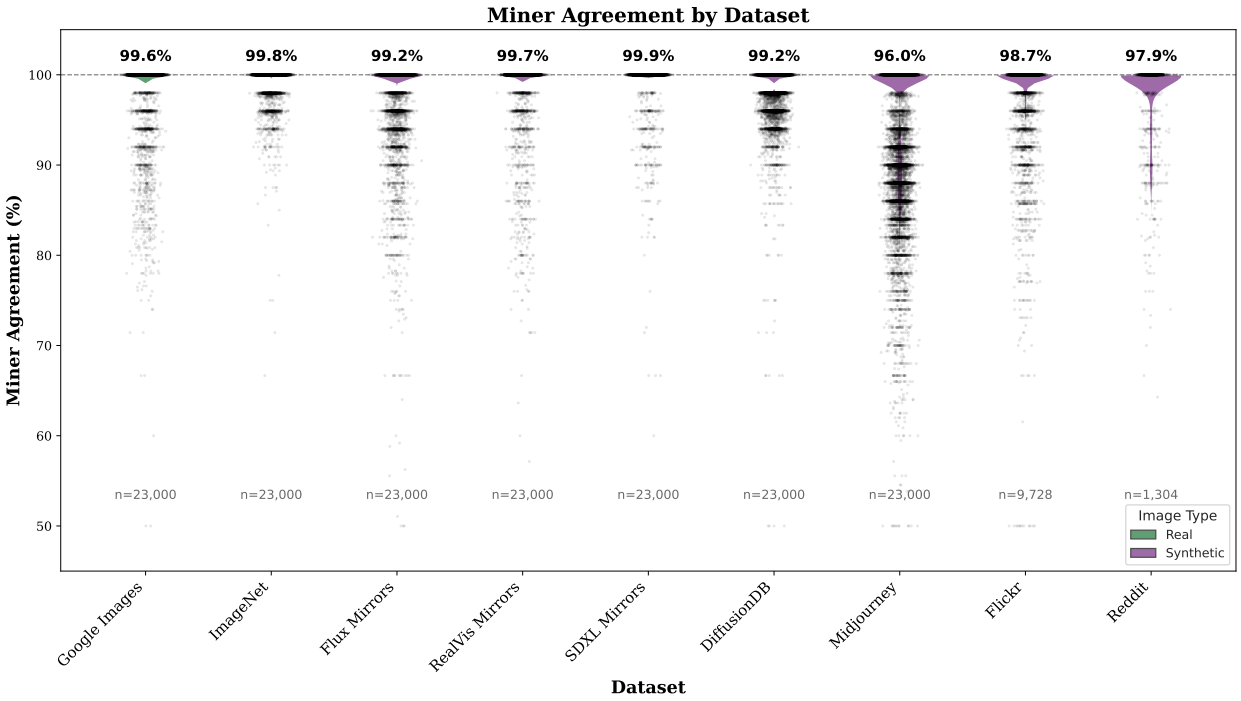

Figure 5. Miner agreement percentages across various datasets, illustrating variations between in-distribution and out-of-distribution synthetic images.

6.1. Future Work

While our initial results demonstrate the efficacy of this approach, further development is necessary to enhance the validator neuron, incentive mechanism, and overall generalization performance of the subnet. The results presented in Table 4 suggest several key areas for improvement:

6.1.1. Challenge Coverage and Subnet Generalization

Real Data Our results indicate that the subnet performs consistently well on real datasets such as our curated Google Images (90.97% accuracy with Majority Vote) and ImageNet (91.95% accuracy with Capped Mean). These metrics show that, despite the our natural constraint in real data availability, miners that have optimized their models to perform well on subnet challenges are able to generalize to a wide variety of authentic images while minimizing false positives. This is especially true in the case of the Google Images data, which contains a wide variety of image types from real world scenery to screenshots of text and scientific figures.

To ensure that we continue to increase the volume and diversity of incentivized real data, we have partnered with Eidon[2] to develop a sustainable pipeline for acquiring authentic real-world content, which will help refine the subnet's ability to distinguish real from synthetic media more effectively.

Synthetic Data The results show that the subnet achieves high classification accuracy on incentivized generative models, particularly on SDXL (98.53% accuracy) and RealVis Mirrors (95.81%). However, its performance declines when confronted with out-of-distribution data from sources such as DiffusionDB (50.61%), MidJourney (42.66%). This confirms that the subnet's incentive structure successfully directs training efforts toward relevant generative models but also highlights the challenge of detecting models outside the incentivization pool.

It is also worth noting that the types of images present in the out-of-distribution datasets are generally of lower quality than our incentivized data, as we have prioritized the detection of high quality synthetic images from newer

[2]https://www.eidon.ai/

models. For a visual comparison of images from our incentivized challenges and out-of-distribution data, please see our supplemental materials section.

Additionally, performance on in-the-wild synthetic datasets such as Reddit (77.91%) and Flickr (82.06%) suggests that while the subnet can generalize to non-incentivized synthetic content, there is room for improvement. The classification accuracy on these datasets, though significantly better than on out-of-distribution models like MidJourney and DifffusionDB, is still lower than on models within the incentive pool. This highlights the need to continue benchmarking against evolving in-the-wild datasets, such as those from Reddit and Flickr, to prevent overfitting to static incentivized challenges and ensure adaptability to novel generative techniques.

Closed-Source Systems The inaccessibility of prominent closed-source models such as Google's Veo2 or Kling AI presents an ongoing challenge for comprehensive detection coverage. In some cases, as with OpenAI's Sora and MidJourney, we lack even API access for real-time or offline challenge data generation. Our ambitious goal remains to generalize detection capabilities to these proprietary systems' outputs without direct access, by leveraging any and all available sources—including public samples, leaked content, surrogate models, and synthetic approximations. This approach embraces the principle that robust detection systems should adapt to novel generation techniques through exposure to diverse data sources, even when the original generators remain inaccessible.

6.1.2. Validator Neuron Throughput

As we continue to expand our challenge distribution to incentivize further generalization, we must horizontally scale in order to support the additional generative models without drastically reducing the validator neuron's per-model throughput. Towards this end, we must either increase the minimum system requirements for validators, requiring multiple GPUs in order to facilitate parallel execution of generative models, or rely on compute external to the subnet for additional throughput. In the latter case, validators essentially become prompt generation machines that outsource media generation to any combination of third-party services, other Bittensor subnetworks, or BitMind-hosted infrastructure.

6.1.3. Incentive Flexibility

The subnet's current incentive mechanism forces all miners to function as generalists, requiring them to service all request types regardless of modality, source, or semantics. Specifically, equation 2 defines a miner's instantaneous reward as a weighted combination of performance across both image and video challenges, making competitive solutions for both modalities necessary to receive rewards. This approach creates an unnecessary barrier for miner who may have already developed highly accurate solutions for a single modality. We propose refactoring the reward structure to support specialization, allowing miners to excel in specific domains—whether by modality (image vs. video) or content semantics. This revised approach would enable intelligent data routing within the network, where organic traffic is directed to miners based on their historical performance with particular content types. Such specialization could potentially increase overall system accuracy while lowering the entry barrier for new participants, fostering a more diverse ecosystem of detection solutions.

7. Conclusion

This paper presents the BitMind Subnet: a functional decentralized network for deepfake detection with four key contributions: (1) a novel blockchain-incentivized framework driving dynamic detection capabilities through competitive collaboration; (2) innovative video and image detection approaches from a leading subnet miner; (3) comprehensive benchmarks across diverse datasets that validate system effectiveness in varied real-world scenarios; and (4) freely accessible consumer applications[3] that deliver immediate public utility. BitMind bridges the critical gap between research innovation and practical deployment, establishing a sustainable, adaptable system for the ongoing challenge of synthetic media authentication.

References

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives, 2014. 2

[2] James Bennett, Stanley Lanning, and Netflix Netflix. The netflix prize. 2009. 1

[3] https://bitmind.ai/apps

[3] Black Forest Labs. Announcing Black Forest Labs. https://blackforestlabs.ai/announcing-black-forest-labs/, 2024. Accessed on 2025-03-14. 6

[4] Matyas Bohacek and Hany Farid. Human action clips: Detecting ai-generated human motion, 2024. 2

[5] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025. 6

[6] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. *CoRR*, abs/1912.01865, 2019. 7

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 7

[8] Corcel.io. Mobius. https://huggingface.co/Corcelio/mobius, 2024. 6

[9] S. Dathathri, A. See, S. Ghaisas, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634: 818–823, 2024. 3

[10] DeepFloyd. Deepfloyd IF-I-XL v1.0. https://huggingface.co/DeepFloyd/IF-I-XL-v1.0, 2023. 6

[11] DeepFloyd. Deepfloyd IF-II-L v1.0. https://huggingface.co/DeepFloyd/IF-II-L-v1.0, 2023. 6

[12] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton-Ferrer. The deepfake detection challenge dataset. *ArXiv*, abs/2006.07397, 2020. 2

[13] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models, 2023. 3

[14] Aaron Grattafiori and et al. The llama 3 herd of models, 2024. 5

[15] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech 256, 2022. 7

[16] Jiashang Hu, Shilin Wang, and Xiaoyong Li. Improving the generalization ability of deepfake detection via disentangled representation learning. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3577–3581, 2021. 2

[17] Gary Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. *Tech. rep.*, 2008. 7

[18] Yonghyun Jeong, Doyeon Kim, Seungjai Min, Seongho Joe, Youngjune Gwon, and Jongwon Choi. Bihpf: Bilateral high-pass filters for robust deepfake detection. *CoRR*, abs/2109.00911, 2021. 2

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. 6, 7

[20] Cagliostro Research Lab. Animagine-xl-4.0. https://huggingface.co/cagliostrolab/animagine-xl-4.0, 2024. 6

[21] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, 2022. 7

[22] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 5

[23] Shanchuan Lin and Xiao Yang. Animatediff-lightning: Cross-model diffusion distillation, 2024. 6

[24] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 7

[25] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 772–781, 2021. 2

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 7

[27] Google LLC. Open images dataset. https://storage.googleapis.com/openimages/web/index.html, 2020. Accessed: 2020-09-01. 7

[28] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. *CoRR*, abs/2103.12376, 2021. 2

[29] Lykon. Dreamshaper 8 inpainting. https://huggingface.co/Lykon/dreamshaper-8-inpainting, 2024. 6

[30] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation, 2025. 7

[31] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models, 2024. 2

[32] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 6

[33] PromptHero. Openjourney-v4. https://huggingface.co/prompthero/openjourney-v4, 2023. 6

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. 2

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 6

[36] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. *CoRR*, abs/1901.08971, 2019. 2

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. 12

[38] SG161222. Realvisxl v4.0. https://huggingface.co/SG161222/RealVisXL_V4.0, 2023. 6

[39] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image generation models, 2023. 2

[40] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. Video visual relation detection. In *ACM International Conference on Multimedia*, Mountain View, CA USA, 2017. 7

[41] Stability AI. Stable diffusion XL 1.0 inpainting. https://huggingface.co/diffusers/stable-diffusion-xl-1.0-inpainting-0.1, 2023. 6

[42] Stability AI. Stable diffusion XL base 1.0. https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0, 2023. 6

[43] Chuangchuang Tan, Huan Liu, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection, 2023. 2, 10

[44] Genmo Team. Mochi 1. https://github.com/genmoai/models, 2024. 6

[45] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 5

[46] Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv:2210.14896 [cs]*, 2022. 12

[47] Zijian Zhang Rox Min Zuozhuo Dai Jin Zhou Jiangfeng Xiong Xin Li Bo Wu Jianwei Zhang Kathrina Wu Qin Lin Aladdin Wang Andong Wang Changlin Li Duojun Huang Fang Yang Hao Tan Hongmei Wang Jacob Song Jiawang Bai Jianbing Wu Jinbao Xue Joey Wang Junkun Yuan Kai Wang Mengyang Liu Pengyu Li Shuai Li Weiyan Wang Wenqing Yu Xinchi Deng Yang Li Yanxin Long Yi Chen Yutao Cui Yuanbo Peng Zhentao Yu Zhiyu He Zhiyong Xu Zixiang Zhou Zunnan Xu Yangyu Tao Qinglin Lu Songtao Liu Dax Zhou Hongfa Wang Yong Yang Di Wang Yuhong Liu Weijie Kong, Qi Tian and along with Caesar Zhong Jie Jiang. Hunyuanvideo: A systematic framework for large video generative models, 2024. 6

[48] Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection, 2023. 2, 10

[49] Zhiyuan Yan, Yong Zhang, Xinhang Yuan, Siwei Lyu, and Baoyuan Wu. Deepfakebench: A comprehensive benchmark of deepfake detection, 2023. 2

[50] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 6

[51] Yuma Rao. Bittensor: A peer-to-peer intelligence market. https://bittensor.com/whitepaper, 2023. Accessed: [Insert access date]. 2, 3

[52] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. Exploring temporal coherence for more general video face forgery detection. *CoRR*, abs/2108.06693, 2021. 2

Survival of the Fittest Detectors: A Decentralized Framework for Evolving Deepfake Detection

Supplementary Material

8. Generalized Deepfake Detection: A Top Miner's Approach

This section presents previous iterations of a top miner's image and video detection models to showcase novel approaches that achieved success on the subnet.

8.1. Image Detection

Image deepfake detection requires identifying artifacts across multiple domains while generalizing to unseen generation techniques. Our approach combines text analysis, frequency domain processing, and spatial feature extraction in a multi-modal architecture.

Our detection model processes input images through three parallel pathways: an OCR model that analyzes text regions, a filter bank that examines frequency patterns, and EfficientNet that extracts spatial features. This multi-pathway design allows the model to detect different types of artifacts depending on the image content.

Our detection model processes an input image through three parallel pathways (see Figure 6):

1. **OCR Pathway:** An Optical Character Recognition (OCR) model extracts and analyzes text regions, identifying inconsistencies in character spacing, alignment, and contextual integration with the background.
2. **Frequency Domain Pathway:** A filter bank separates image frequencies to isolate subtle artifacts. This includes high-frequency filters to detect noise and edge artifacts, low-frequency filters to assess color distribution anomalies, and spatial filters (such as Sobel filters) to capture gradient discontinuities.
3. **Spatial Feature Pathway:** An EfficientNet-based model processes the entire image to capture object boundaries, texture patterns, and lighting irregularities that are indicative of synthetic content.

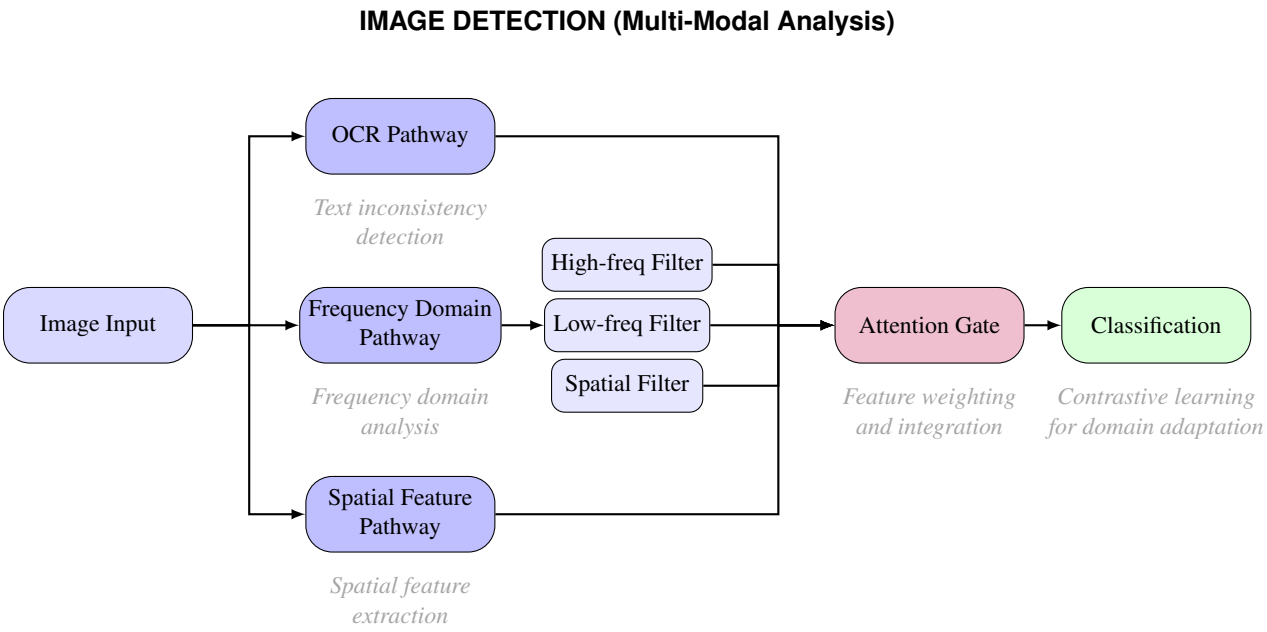

Figure 6. Image detection architecture with three analysis pathways and an attention mechanism for feature integration.

8.1.1. Detection Pathways

The OCR pathway analyzes text regions in images, looking for inconsistencies that often appear in AI-generated content. Text in deepfakes frequently shows problems with character spacing, alignment, and integration with the

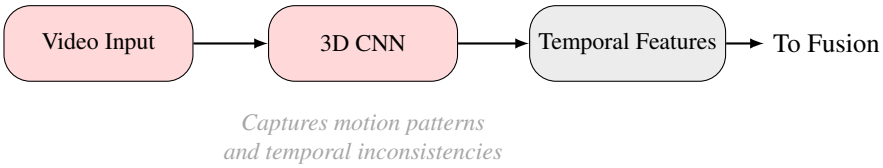

Figure 7. Video branch using 3D CNNs to analyze temporal patterns across frames.

background. The model examines these aspects along with the contextual appropriateness of the text content.

The frequency domain pathway uses a filter bank to separate image frequencies and isolate artifacts that are often invisible to the naked eye. We implement high-frequency filters to detect noise patterns and edge artifacts, low-frequency filters to identify color distribution anomalies, and spatial filters to find texture inconsistencies.

The filter bank contains several specialized kernels for deepfake detection:

$$K_{\text{high}} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \quad K_{\text{low}} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{9}$$

$$K_{\text{sobel-x}} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad K_{\text{sobel-y}} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{10}$$

Each kernel detects specific artifacts: high-pass filters find unnatural sharpness, low-pass filters identify color blending issues, Sobel filters capture gradient discontinuities, and others detect various texture and edge anomalies.

The spatial feature pathway uses EfficientNet to process the full image and identify object boundary irregularities, unusual texture patterns, and lighting inconsistencies that are common in synthetic images.

8.1.2. Feature Integration and Training

An attention mechanism combines outputs from all pathways, weighting features based on their relevance to the specific image:

$$\alpha_i = \frac{\exp(W_i \cdot f_i + b_i)}{\sum_j \exp(W_j \cdot f_j + b_j)} \tag{11}$$

This approach allows the model to focus on the most relevant signals for each image. For example, when analyzing an image with text, the OCR pathway might receive higher weight, while images with unusual textures might prioritize the frequency domain pathway.

The final feature representation combines these weighted features:

$$\mathbf{f}_{\text{final}} = \sum_i \alpha_i \mathbf{f}_i \tag{12}$$

For training, we use focal loss to address class imbalance and contrastive learning to improve generalization across different generation techniques. The training process uses balanced datasets with both real and synthetic images from multiple generation models, along with data augmentation to simulate various capture conditions.

8.2. Video Detection

Video deepfake detection requires analyzing both spatial and temporal dimensions to identify artifacts that may not be apparent in single frames. Our approach uses a dual-branch architecture that processes temporal sequences and individual frames in parallel, allowing the model to detect a wide range of inconsistencies.

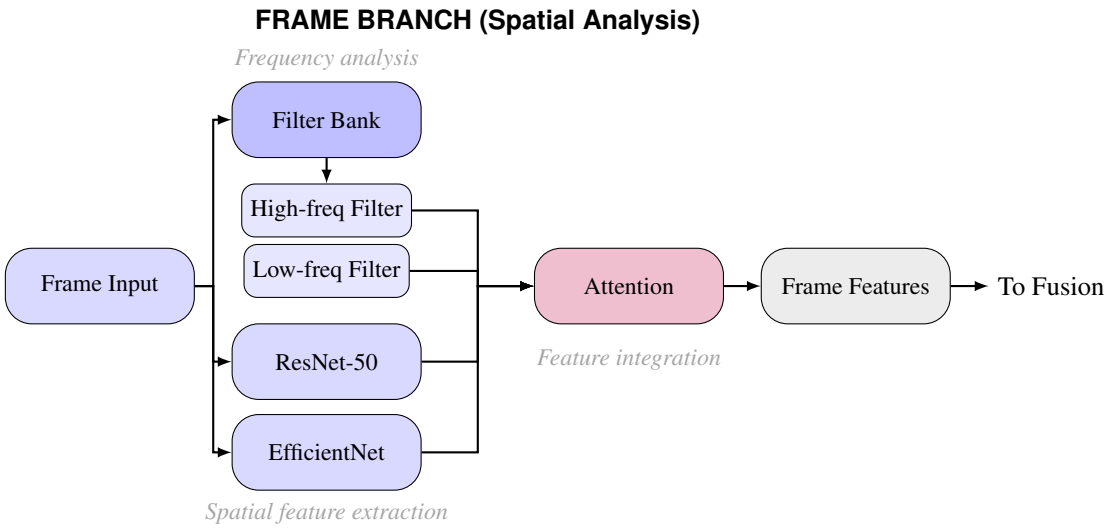

Figure 8. Frame branch analyzing individual frames through filter banks and CNN models.

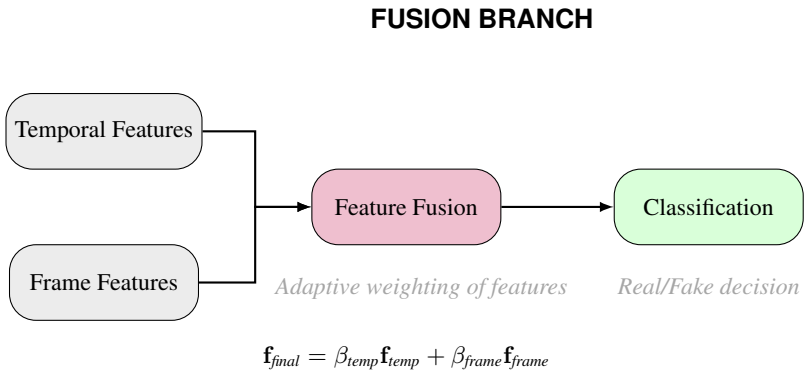

Figure 9. Fusion branch combining temporal and spatial features for final classification.

AI-generated videos contain distinct artifacts that require specialized detection approaches. Our dual-branch architecture targets these artifacts through complementary analysis methods:

The temporal branch uses 3D convolutional neural networks to process video sequences as three-dimensional data (width × height × time). This approach is particularly effective for detecting:

$$\mathbf{Z}(t,h,w,c_{out}) = \sum_{i=0}^{k_t-1}\sum_{j=0}^{k_h-1}\sum_{k=0}^{k_w-1}\sum_{c=0}^{C_{in}-1} \mathbf{V}(t+i,h+j,w+k,c)\cdot\mathbf{K}(i,j,k,c,c_{out}) \tag{13}$$

- **Motion inconsistencies** - Unnatural movements that violate physics or human biomechanics
- **Temporal flickering** - Subtle changes in lighting, texture, or facial features between consecutive frames
- **Synchronization issues** - Misalignment between audio and visual elements or between different moving parts

The frame branch analyzes individual frames using a combination of frequency domain analysis and spatial feature extraction. This branch targets:

- **Boundary artifacts** - Unnatural edges or blending issues at object boundaries, detected by high-frequency filters
- **Texture inconsistencies** - Unrealistic skin textures or surface details, captured by CNN models
- **Color distribution anomalies** - Unusual saturation or color patterns, identified by low-frequency filters
- **Anatomical anomalies** - Structural inconsistencies in faces or bodies, detected by pretrained CNN models

The fusion mechanism combines features from both branches using adaptive weights that adjust based on the

content being analyzed:

$$\mathbf{f}_{\text{final}} = \beta_{\text{temp}}\mathbf{f}_{\text{temp}} + \beta_{\text{frame}}\mathbf{f}_{\text{frame}} \tag{14}$$

For videos with significant motion, the system gives more weight to temporal features. When spatial artifacts are prominent, it emphasizes frame-level features. This adaptive approach allows the model to focus on the most reliable signals for each video type.

The weights are learned during training through an attention mechanism that considers both the confidence of each branch and the characteristics of the input video. This approach makes the system effective against a wide range of deepfake techniques, including those that focus primarily on improving either spatial quality or temporal coherence.

Our training methodology uses a diverse dataset of real and synthetic videos, with augmentation techniques that simulate various capture conditions and post-processing effects. This ensures that the model generalizes well to new deepfake generation methods rather than overfitting to specific artifacts.