

ASAP: Visual Analytics for Identifying and Analyzing Image Patterns in AI-generated Images

Journal Title
XX(X):1–16
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/

SAGE

Jinbin Huang¹, Yuki Ueno¹, Chen Chen², Aditi Mishra³, Bum Chul Kwon⁴, Zhicheng Liu², and Chris Bryan¹

Abstract

Generative image models can produce highly realistic images, raising concerns about potential misuse in creating deceptive content. Current deepfake approaches face several challenges, including limited generalizability, lack of interpretability, and poor actionability. To help address these, we present ASAP, an interactive visualization system designed to empower users in the analysis and summarization of deceptive patterns in AI-generated images. ASAP introduces a novel CLIP-adapted image encoder that generates interpretable representations, enabling the extraction of influential pixel regions via calculated masks. This approach facilitates the identification of key deceptive features through influence measurement techniques. These backend techniques are integrated into a visual analytics dashboard that allows users to quantify and analyze authenticity-indicative patterns in image collections containing both authentic and AI-generated images. This approach also supports the comparative analysis of various generative models, including GANs and diffusion models. We demonstrate ASAP's efficacy through a user study and two application scenarios using established fake image detection benchmarks, showcasing its ability to effectively extract and quantify deceptive patterns.

Keywords

Generative AI, deepfakes, fake image detection, visualization

Introduction

Given the rapid advancement of generative AI (GenAI), image models such as DALL-E 3¹ and Midjourney² can now create high-fidelity, realistic images that easily deceive humans³. This paradigm shift is coinciding with rising demands for effective detection and analysis of realistic-but-fake images⁴, as their misuse can have widespread impacts including misinformation spread⁵, copyright violation⁶, and credit-blame asymmetry⁷.

While the research community has developed examples of automated detection methods for fake images (that can outperform human experts under certain conditions⁸), these tools often face significant challenges, including *generalizability*⁹, *interpretability*^{10–12}, and *actionability*¹⁰. Generalizability challenges arise, in part, from distinct visual signatures across different generative models (e.g., GANs vs. diffusion models¹³), where detectors effective for one model can fail for others⁹. Interpretability and actionability challenges can arise because black-box detector models output only predictions and do not provide transparency or explanations of decision-making processes¹⁰.

Making these challenges even more salient to real-world use, the types of users and stakeholders who are performing deepfake analysis is also broadening. Such users now include not only “AI experts” (such as GenAI developers and researchers), but also “model users”¹⁴ such as misinformation researchers, digital forensic analysts, social media moderators, computational journalists, and legal experts, who must analyze and explain deepfake images

as part of their workflows¹⁵. These users often have high computational proficiency on tasks such as pixel-based review and analysis (e.g., via photo editing tools), but lack formal training in the technical and mathematical aspects of AI/ML engineering and prediction. However, they also need to understand deceptive patterns in AI-generated images, including why an image might appear real or fake to a detector.

Along these lines, there is increasing recognition we need novel human-in-the-loop (HITL) workflows to help in analyzing and understanding AI-generated images and GenAI models⁸. Such advocacy has highlighted a need to create interpretable tools and techniques to help overcome the above-described challenges¹¹.

This paper investigates how to address these needs via a design study process¹⁶. Specifically, we develop ASAP (Analyzing and Summarizing AI-generated image Patterns), a HITL pipeline and visual analytics tool to help identify, analyze, and summarize patterns in images ensembles containing a combination of both authentic and fake images. ASAP is designed for analysis settings where

¹Arizona State University, AZ, US

²University of Maryland, MD, US

³Fujitsu Research of America, PA, US

⁴IBM Research, MA, US

Corresponding author:

Yuki Ueno, Arizona State University Sonoran Visualization Laboratory, 699 S Mill Ave, Tempe, AZ, US.

Email: yueno@asu.edu

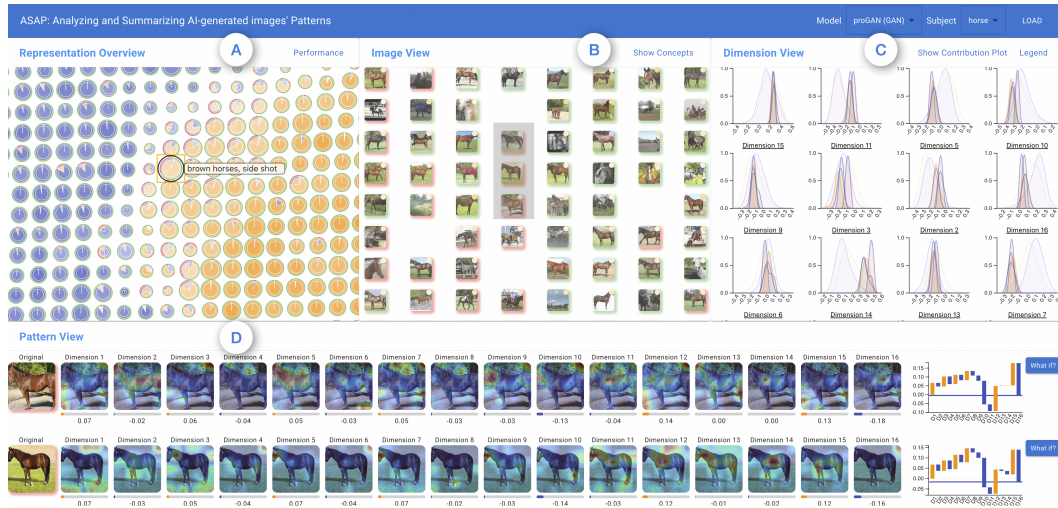


Figure 1. ASAP supports interactively analyzing deceptive patterns in AI-generated images that fool trained classifiers. In this example, a user is analyzing “brown horses, side shot” images generated by proGAN. **(A)** The Representation Overview projects image representations into 2D space, grouping them into cells with novel glyphs that summarize classification performance. **(B)** The Image View highlights fake images misclassified as authentic within selected cells. **(C)** The Dimension View visualizes embedding distributions across CLIP dimensions, correlating visual patterns with their quantitative characteristics. **(D)** The Pattern View reveals the most misleading pixel groups through saliency maps, uncovering distinct patterns in AI-generated images.

image authenticity labels are available, such as curated research datasets, benchmarking environments, or forensic investigations. It includes an end-to-end pipeline for efficient detection and analysis of image patterns in datasets that contain a combination of real and AI-generated images, including an interpretability-focused detector that produces compact embeddings with latent authenticity indicators to facilitate the discovery and explanation of how such patterns can differ among authentic and synthetic images (i.e., helping users reason about why an image appears real or fake). ASAP’s frontend is a visual analytics interface composed of several coordinated views supporting such analysis and exploration, from broad overviews to fine-grained details.

We robustly evaluate ASAP via a set of usage scenarios and a mixed methods user study. Our results demonstrate how ASAP can effectively support analyzing patterns in AI-generated and real images in an interpretable manner. Based on the process of designing, implementing, and validating ASAP, we also reflect on a number of takeaways and lessons learned, such as how HITL and visualization-driven approaches can be tailored to effectively support generalizability, interpretability, and actionability in the context of deepfake analysis, and how tools like ASAP address fast-evolving research questions in GenAI trust and authenticity.

Related Work

ASAP builds upon prior research in the detection of AI-generated images, explanations for transformer models, and applications of visual analytics for AI and GenAI.

Detecting Fake Images

Fake image detection predates generative image models, as traditional fake image creation and manipulation methods (e.g., via software like Photoshop) were known to introduce

detectable changes in images, including visual cues like compression artifacts¹⁷, resampling¹⁸, and unnatural reflections¹⁹. However, many early detectors and classifiers faced generalizability challenges, such as not handling out-of-distribution data or newer models^{9,20}, prompting the development of more versatile classifiers^{21–24}.

A significant challenge for learning-based methods is finding a robust feature space to distinguish between authentic and AI-generated images across various models²⁴. Recent research suggests using feature spaces from large pre-trained models like CLIP²³, which have shown to be effective and generalizable²⁵. In ASAP, we build upon these findings by modifying CLIP’s visual encoder to develop a linear classifier that efficiently encodes images into interpretable representations, facilitating the discovery of distinctive patterns in AI-generated images.

Specifically, CLIP (Contrastive Language-Image Pre-training)²⁶ has emerged as a powerful pre-training paradigm that combines textual supervision with visual cues, and has demonstrated remarkable adaptability and success across a range of semantic tasks including image segmentation^{27–29}, object detection³⁰, classification³¹, and medical imaging³². However, CLIP’s entanglement of text and vision poses challenges for predominantly visual tasks. Recent studies have highlighted that CLIP’s visual encoder may inadvertently embed textual information into its outputs³³. Such unintended textual embeddings can compromise the efficacy of gradient-based methods aimed at identifying crucial non-semantic pixels, consequently affecting the accurate identification of inauthentic image regions.

To address this in ASAP, we integrate a “forget-to-spell” model: an orthogonal linear projection designed to minimize textual content in CLIP’s latent space. This approach removes unwarranted textual embeddings, yielding a predominantly visual-centric representation that better serves our task of fake image detection and pattern discovery.

Measuring Token Influence In Transformer Models

As transformer models become more popular, their explanations gain importance^{34,35}. A key interpretability method for transformers is to attribute influence scores to tokens^{34,35}, with terms such as *relevance*, *importance*, and *influence* often used interchangeably³⁶.

For vision transformers, computing token relevance identifies influential pixels in model decisions³⁴; this is also crucial for detecting AI-generated images. Several methods exist for explaining predictions in transformer-based architectures^{37–39}; we employ Chefer et al.’s³⁵ approach due to its generalizability and strong performance. Specifically, we adapt their relevance map technique for our CLIP-based image encoder, with a significant modification: instead of using final logits, we utilize embedding dimensions for relevance computation (see ‘Backend’ section). This approach offers a more fine-grained analysis than the single map generated by binary classification logits.

Visual Analytics for AI and Generative AI

Visual analytics tools have been developed for a variety of HITL workflows for AI and GenAI, including data augmentation⁴⁰, labeling⁴¹, exploration and analysis⁴², bias inspection and mitigation⁴³, model explanation⁴⁴, model refinement⁴⁵, robustness testing⁴⁶, and trustworthiness promotion⁴⁷. Alongside these systems, a variety of explainable AI (XAI) methods have been developed to explain model decision-making processes, including through local surrogate models⁴⁸, feature attribution^{49,50}, class-discriminative localization⁵¹, or concept-level analysis⁵². These techniques are commonly applicable to binary classification settings (including real/fake image detection), by showing which input features, pixels, or concepts support a predicted class.

Specifically for GenAI, rapid advancements in this space have led to increased emphases on developing visual analytics tools specifically for generative models. Recent work has, for example, focused on visualizing transformer attention mechanisms⁵³, illustrating diffusion processes⁵⁴, and enhancing prompt engineering⁵⁵. Broadly, such tools can aim to unveil the inner workings of generative models (particularly those based on transformer or diffusion methodologies) or support interactions with these models.

One similar tool to ASAP is by Ye et al.⁵⁶, who develop a system for reviewing and comparing human and AI-generated paintings. While they likewise employ a CLIP-based approach, they also design and integrate a number of manual painting-focused aesthetic features (such as composition and use of color), and do not focus on differentiating nuanced pixel-specific segments and patterns that distinguish real-vs.-fake images. Put another way, ASAP targets a different analytic gap: using interactive visualization to analyze authentic and synthetic image collections beyond simple fake/real labeling, by tightly linking collection-level summaries of images to the specific image evidences (i.e., pixel patterns) responsible for detector decisions, by integrating an explainability-focused detector with coordinated views that connect dimension-specific relevance maps, detector outcomes, contribution scores, and

image collections. This design helps analysts move from simply observing that a model or collection differs, to understanding why an image succeeds or fails detection, by identifying recurring deceptive patterns and quantifying their contribution across diverse generative models.

Design Challenges and Goals

Our overarching goal is to investigate how to help users analyze collections of (potentially) fake images, such as understanding how they might contain various deceptive or authenticity-indicative patterns imbued by a creating generative image model, and if these pixel patterns fool a deepfake detector. Based on a meta-analysis of papers (i.e., from the ‘Related Work’ section) and similar HITL AI explainability work (e.g., Xu et al.⁵⁷), we identified several analytic tasks such users could perform, including exploring an ensemble of synthetic and fake images to understand their pixel-based patterns and relationships, locating suspicious or anomalous data items, identifying and analyzing groups of items (or individual instances) in the ensemble where a detector (mis)classifies, and understanding where and why such misclassifications happens. In such contexts, users would work with data that includes 2D, pixel-based images, ground-truth labeling as to their authentic or deepfake/synthetic status, and would also need transparent outputs from a deepfake detector (i.e., not just predicting if the image was authentic or not, but doing so in a way that supported explanation or interpretation).

Unfortunately, there are two high-level challenges that make this a non-trivial process: (i) we need a method to broadly extract and identify patterns that distinguish fake from real images, both qualitatively and quantitatively; (ii) we need a user experience that facilitates intuitive, interpretable, and in-depth pattern analysis (e.g., at varying levels of detail). We first formalize these into a set of four specific design challenges: **C1–C4**.

Design Challenges

C1 Automatic Discovery of Patterns Present in Fake Images: In scenarios where classifiers are trained to differentiate between real and AI-generated images, they inherently identify unique patterns of both types²⁵. The challenge is to devise a way to utilize these classifiers to efficiently extract these distinct patterns amidst irrelevant visual information that might confound the analysis.

C2 Quantifying Influence of Fake Patterns: Generative models produce diverse artificial patterns, even within the same model¹³. Analyzing these patterns requires distinctively identifying and quantitatively assessing their impact. However, existing techniques evaluate individual pixels rather than collective pixel groups forming patterns⁵⁸, and lack uniform metrics for cross-image significance comparison.

C3 Efficient Analysis and Summarization of Fake Patterns: After identifying fake patterns constituted by influential pixel groups, the subsequent challenge is to devise an interface enabling users to effectively examine and summarize these patterns. The primary obstacle is the potential for an overwhelming number of patterns⁵⁹, which could hinder clarity in analysis. At current, there are no

established guidelines about how to design such an interface or tailor the user experience for such a workflow.

C4 Adapting to a Diverse Range of Generative Models: As GenAI technology rapidly evolves, it is critical that the analysis system adapts across various models. The system’s design must also be flexible enough to accommodate new models as they emerge^{23–25}, ensuring its usefulness over time. This requires a modular approach, the specifics of which are yet to be determined.

Design Goals

Next, to address **C1–C4**, we distilled a set of six design goals **G1–G5**. The design and implementation of ASAP, described in ‘ASAP Backend’ section and ‘ASAP Interface’ section, is intended to support these goals.

G1 Develop an interpretable classifier to detect real from fake images: Discovering fake patterns (**C1**) involves: (i) distilling critical information separating real from fake images into condensed representations, and (ii) leveraging these representations to detect key pixel groups. For this, we use supervised learning, by training a classifier not just to distinguish real from fake images, but also encode authenticity information^{23,25}. To handle diverse AI-generated images (**C4**), we build the classifier on a large-scale pre-trained model with extensive training on varied datasets. We freeze the pre-trained image encoder to leverage its generic visual feature extraction capabilities and train downstream classifiers on its image embeddings. To make the classifier interpretable, we introduce a distiller layer that maps these embeddings into a compact set of orthogonal authenticity-related dimensions, each of which can be linked to classifier weights and pixel-level relevance maps in subsequent design goals.

G2 Using gradients to identify important pixels: We leverage gradients propagated back from distilled representations to reveal output sensitivity to input changes³⁵. Similar to gradient-based localization methods such as Grad-CAM⁵¹, this helps identify image regions that influence model behavior. However, rather than weighting convolutional feature maps using gradients from a final class logit, we adapt transformer relevance propagation for CLIP by propagating gradients from each dimension of the distilled authenticity representation to attention maps within CLIP’s visual transformer blocks. This yields dimension-specific token relevance maps that are mapped back to pixel sensitivities (**C1**). Given the multi-dimensional distilled representation, we craft detailed masks for each dimension, as each uniquely encodes realness or artificiality information, creating comprehensive masks that trace specific pixel groups affecting authenticity.

G3 Establish a uniform metric for evaluating pattern influence: To quantify pattern influence (**C2**), we address a limitation of gradient-based relevance maps, which primarily indicate relative importance within a single image³⁵. Such maps reveal where influential pixels are located, but do not directly show how strongly a pixel group supports a “real” or “fake” prediction, nor whether that influence is comparable across images. We therefore aim to derive a signed contribution metric for each identified pixel group by linking it to the corresponding dimension in the distilled representation. The image-specific dimension

value captures how strongly the evidence appears in the current image, while the classifier weight captures how strongly that dimension affects the real/fake decision globally. Combining and normalizing these factors yields a comparable score whose sign indicates prediction direction and whose magnitude indicates influence strength. This metric is intended to support visual comparison of pattern influence across images, groups, and prediction outcomes.

G4 Support comparative analysis of prediction outcomes and recurring patterns: Understanding fake patterns (**C3**) requires comparing images and image groups rather than inspecting isolated predictions. Inspired by contrastive explanation, which helps explain why one outcome occurs rather than another⁶⁰, users should be able to compare correctly classified and misclassified images that share apparent visual traits but differ in subtle authenticity-related patterns. Such comparisons should help users connect qualitative visual traits with quantitative representation differences, identify recurring patterns that distinguish successful detections from failures, and summarize how these patterns vary across real and fake image collections.

G5 Enable detailed examination of individual images: Finally, in-depth analysis of fake images (**C3**) likely requires examining individual examples of images of interest. This can be accomplished by providing an overview-plus-detail workflow, where users can navigate from a high-level summary view down to selecting and analyzing individual images at a fine-grained level.

ASAP Backend

We now describe ASAP’s backend, which discovers patterns indicative of image authenticity or artificiality. This process involves training binary classifiers to distinguish between fake and real images, using these classifiers as encoders to transform images into interpretable vector representations, and then leveraging these representations to uncover relevant patterns. Specific implementation details are discussed in the following subsections.

Datasets

To train classifiers for ASAP’s pattern discovery, we utilize two state-of-the-art datasets comprising AI-generated images from leading generative model paradigms: GANs and diffusion models.

(i) The proGAN dataset⁶¹ includes authentic images from the PASCAL object detection challenge and their fake counterparts generated by the proGAN model. This balanced dataset contains 30,000 real and 30,000 fake images per class across 20 classes. (ii) The DetectingSyntheticImage dataset¹³ focuses on images generated by latent diffusion models (LDMs) across various categories. We specifically examine 20,000 fake human face images generated by LDM, paired with real human face images from the Flickr-Faces-HQ (FFHQ) dataset, which contains 70,000 high-quality human face images.

For each generative model type (GAN and diffusion), we train a classifier to distinguish between “real” and “fake” images. The feature spaces of these trained classifiers are then utilized to identify deceptive patterns specific to each type of generated image.

To validate the classifier’s performance and generalizability, we evaluate against state-of-the-art baselines²³ using benchmark datasets, achieving comparable detection performance — due to page constraints, please refer to the Appendix for benchmark performance details. Critically, while our classifier demonstrates competitive accuracy, we emphasize that our focus is also to support interpretability (which is not done by SOTA detectors). Put another way, our approach enables us to not only to identify AI-generated images, but also to provide interpretable insights into the specific patterns contributing to their detection, overcoming the black box problem intrinsic to many existing detectors.

Learning Interpretable Representations

At a high level, we are motivated by the fact that CLIP-based approaches have been shown effective for general-purpose fake detection, such as recent work by Ojha et al.²³ Our approach is similar to the extent that we adapt CLIP for training a classifier. However, as discussed above, our intent is not only to achieve high classification performance but also to learn interpretable representations, which requires a customized approach. This process involves the following steps, and is summarized in Figure 2 L:

(i) **CLIP encoding:** The image is preprocessed using the CLIP:ViT-B/32 image encoder, $f_{\text{CLIP}}^{\text{image}}(\cdot)$, producing a 512-dimensional vector of generic visual features (L1 in Figure 2).

(ii) **Text information removal:** A “forget-to-spell” projection, $f_{\text{forget}}(\cdot)$, filters out unintended text-related information from CLIP’s embeddings. This is a key component in our approach (and also differentiates from prior techniques such as Ojha et al.²³) because, while CLIP’s visual embeddings provide vital cues for differentiating between authentic and synthesized images²⁵, its contrastive learning process inherently mixes visual and textual information³³. This mixing, though beneficial for tasks like zero-shot classification, can confound image authenticity verification. Our forget-to-spell projection uses an orthogonal projection technique³³ and is trained with an orthogonality loss function on a diverse dataset (including natural images, text-only images, and images with overlaid text). This approach enables the model to distinguish visual elements from textual content, resulting in a more visually focused 256-dimensional vector ($f_{\text{forget}} \circ f_{\text{CLIP}}^{\text{image}}(I)$) (L2 in Figure 2).

(iii) **Representation distillation:** A linear classifier trained on the forget-to-spell representation distinguishes between genuine and synthetic images. A distiller layer $f_{\text{distill}}(\cdot)$ reduces the 256-dimensional vector to 16 orthogonal dimensions (L3 in Figure 2), followed by a classification head $f_{\text{class}}(\cdot)$ for final real/fake prediction (L4 in Figure 2). (See ‘Representation Distillation’ section below for details.)

The combination of $f_{\text{CLIP}}^{\text{image}}(\cdot)$, $f_{\text{forget}}(\cdot)$, and $f_{\text{distill}}(\cdot)$ forms an effective image encoder, converting images into a compact space rich with information crucial for determining image authenticity, thereby facilitating subsequent pattern discovery (G1).

Representation Distillation After obtaining visual-centric 256-dimensional representations $f_{\text{forget}}(f_{\text{CLIP}}^{\text{image}}(I))$, we

further distill authenticity-indicative information through supervised learning. We train a binary classifier to distinguish real from fake images, guiding it to extract authenticity-relevant dimensions. This approach condenses 256 dimensions into 16 orthogonal dimensions (the authors empirically assessed this size as a good balance between performance and explainability by comparing against other potential values, though ASAP can support different dimensional spaces and we plan to investigate tradeoffs in larger/smaller dimensional spaces as a future work), enabling gradient-based analysis to identify influential pixel groups (i.e., patterns) for real/fake image classification.

Architecturally, our classifier is intentionally “simple,” only consisting of two linear layers: A distiller layer $f_{\text{distill}}(\cdot)$ reduces dimensionality retaining primarily image authenticity-relevant information, and a classification head $f_{\text{class}}(\cdot)$ processes the distilled representations through a sigmoid function to produce the final binary classification. We train the classifier on the datasets described in ‘Datasets’ section, initializing the distiller’s weights orthogonally⁶² and incorporating an orthogonality penalty in the loss function:

$$\text{Loss}(y, \hat{y}) = \lambda_{\text{BCE}} \text{BCE}(y, \hat{y}) + \lambda_{\text{ortho}} R(W) \quad (1)$$

Here, y is the image label, $\hat{y}$ is the classifier’s output, and $R(W) = \|Id - WW^T\|_2$ is the orthogonality penalty for the distiller’s weight W (Id is the identity matrix). We set λ_{BCE} to 3 and λ_{ortho} to 1. The model is trained using Adam optimizer (batch size 32, learning rate 1e-3) for 12 epochs on a MacBook Pro with M1 chip, in 68 seconds.

This approach yields an encoder that condenses the forget-to-spell representations into a compact, 16-dimensional vector enriched with image authenticity information. We then apply a gradient-based method to these vectors to identify influential pixels and quantify their impact.

Pixel Relevance Computation and Pattern Discovery

Using the distilled representations, we identify influential fake image patterns by computing pixel relevances and selecting those with high relevance. Unlike conventional techniques that generate a single saliency map from final logits, we compute pixel relevance from each of the 16 dimensions in our distilled representation, producing 16 distinct masks, each highlighting unique influential regions: $\frac{\partial A}{\partial x_i}$, $i \in 1, \dots, 16$, where A represents the attention map in CLIP’s transformer block (G2). This approach has three key advantages:

(i) **Mask diversity:** Orthogonality among the 16 dimensions (see ‘Representation Distillation’ section) ensures propagated gradients capture pixel importance from 16 distinct perspectives.

(ii) **Enhanced separability:** Our method isolates individual pixel regions and their contributions (unlike traditional aggregated saliency maps), supporting identification of influential areas.

(iii) **Fine-grained analysis:** The approach also enables detailed examination of individual image regions (patterns), potentially surfacing subtle artificiality indicators often overlooked in black box analyses.

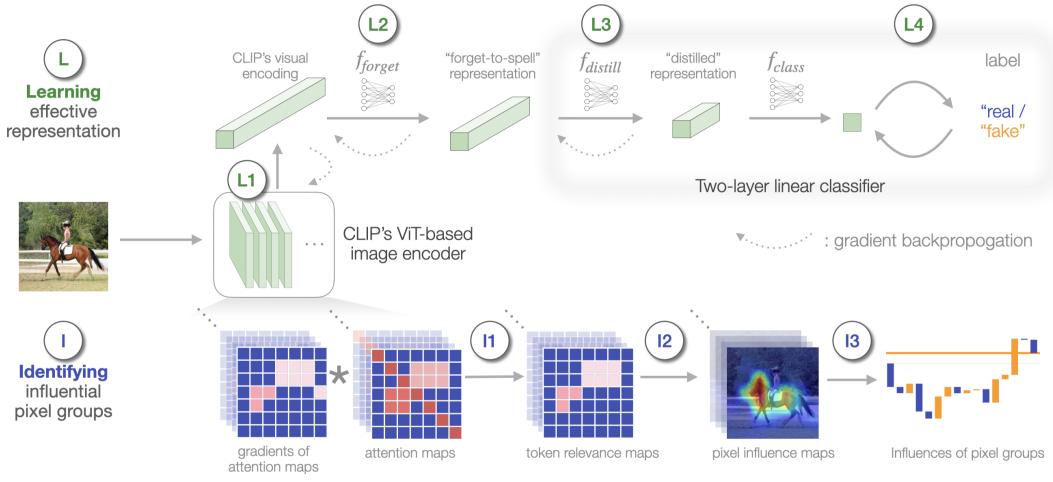


Figure 2. We employ a two-step approach for detecting AI-generated images and identifying their underlying patterns. Step L (Learning) begins by encoding images using CLIP’s image encoder (L1) and then proceeds to strip textual information via a ‘forget-to-spell’ projection (L2). Subsequently, we train a classifier that includes a distiller layer, which maps the ‘forget-to-spell’ representation to a distilled space (L3), and a classification head that predicts the authenticity of images (real or fake) based on this distilled representation (L4). In Step I (Identification), we identify influential pixel groups by computing token relevance using a gradient-based technique (I1), translating token maps to individual pixel maps (I2), and computing uniform contribution metrics for pixel groups based on the classifier’s weights (I3).

After identifying influential pixel groups, we quantify their impact using the classifier f_{class} ’s global weights. This approach ensures consistent influence measurements across images, enabling direct cross-image comparisons. These relevance maps also facilitate segmentation and clustering of pixel groups into distinct visual concepts, enhancing analysis of common fake image patterns (G4). The discovered patterns are then presented in ASAP’s interactive interface for user analysis and summarization (see ‘ASAP Interface’ section).

Identifying Influential Pixel Groups To compute a pixel relevance map for identifying pixels crucial to the final real/fake prediction, we utilize gradients of attention maps in the transformer blocks within CLIP: ViT-B/32.

In ViT, an image I is divided into $k \times k$ patches (where $k = 224/32 = 7$ for CLIP: ViT-B/32). These patches are tokenized into vectors, resulting in k^2 tokens per image, plus an additional [CLS] token, forming a $(k^2 + 1) \times d$ matrix, where $d = 512$ is the embedding space dimension. For an $h \times w$ image, we first compute a $k \times k$ token relevance map, $R_{k \times k}$, indicating each token’s influence. This map is then upsampled to the original image size and normalized between 0 and 1. Initially, $R = [1]_{k \times k}$, considering all tokens equally contributory. We then refine R by backpropagating from the last transformer block:

$$R = R + \bar{A}_i R, i \in m, \dots, 1 \quad (2)$$

Here, $\bar{A}_i$ is derived from gradients of attention maps in the i -th transformer block:

$$\bar{A}_i = \mathbb{E}_h (\nabla A_i \odot A_i)^+ \quad (3)$$

∇A_i is the gradient of attention map A_i , and $\odot$ denotes the Hadamard product. Then, to focus on the most influential pixels, we simplify the final relevance computation to use only the last transformer block:

$$R = \mathbb{E}_h (\nabla A_{last} \odot A_{last})^+ \quad (4)$$

This method involves a forward pass to obtain the 16-dimensional distilled representation and attention maps,

followed by dimension-wise gradient calculation. The resulting 16 relevance maps, each resized to the original image dimensions, serve as a set of masks highlighting different regions crucial for distinguishing real from fake images.

A Uniform Metric for Pattern Contribution Based on the 16 relevance maps derived for each image, we introduce a method to quantify and compare contributions of different pixel groups. The original relevance scores in these maps are image-specific, preventing direct comparisons of pixel group influences across images. This limitation poses challenges in comparing pattern influences (G3, G4). To address this, we develop a uniform metric that integrates both global and local dimensional information, yielding a normalized scalar between $[-1, 1]$. The scalar’s sign indicates the pattern’s contribution direction (positive for fake, negative for real), while its magnitude reflects the extent.

For an image I , we identify 16 pixel groups $pg_i, i \in 1, \dots, 16$, each corresponding to a dimension in the distilled representation $v_{distill} = (v_1, \dots, v_{16})$. Using the classification head’s weight $w = (w_1, \dots, w_{16})$, we calculate $s_i = v_i \times w_i$ for each dimension. To enable cross-image comparisons, we normalize s_i into a contribution score c_i :

$$c_i = \frac{|v_i \times w_i|}{\sum_{i=1}^{16} |v_i \times w_i|} * \text{sign}\left(\frac{v_i \times w_i}{\sum_{i=1}^{16} |v_i \times w_i|}\right)$$

This preserves the sign of s_i while avoiding potential amplification due to cancellations in summation. The resulting score encapsulates both local and global dimensional data for each pixel group, maintaining the original scalar’s sign within the $[-1, 1]$ range. This uniform metric also facilitates efficient comparison of pixel groups’ influence on real/fake image classification both within and across images, providing a consistent measure of their impact.

Revealing Commonality of Similar Images To illustrate commonalities among fake images exhibiting similar artificial patterns (G4), we leverage concept-based methods⁵².

Initially, we utilize the relevance maps generated in ‘Identifying Influential Pixel Groups’ section as masks to identify influential image segments (pixel groups) within a collection of similar images. These images are considered similar based on the proximity of their “distilled” representations in the L2 distance metric. To refine our analysis, we exclude overlapping segments within images by discarding those with high Intersection over Union (IoU) scores relative to already identified segments, as they are nearly identical to the existing ones. Subsequently, we encode these segments using the combined encoder $(f_{\text{distill}} \circ f_{\text{forget}} \circ f_{\text{CLIP}}^{\text{image}})(\cdot)$ into distilled representations. We group these representations into three clusters (these were empirically assessed as a good balance in the number of groupings for our tested datasets, determined based on the number of images per cell) for analysis using k-means clustering.

This process enables us to identify and highlight clusters of visual concepts in a human-interpretable manner, effectively revealing the common visual patterns shared by groups of images with similar authentic or fake patterns.

System Stack

ASAP is built on a Flask server to handle computations like interpretable image embeddings, influential pixel segmentation, and influence measurement in response to user queries. The frontend interface (see next section) is built with React and d3.js.

ASAP Interface

Building upon our design goals (‘Design Goals’ section), techniques for extracting critical image authenticity information (‘Representation Distillation’ section), and methods for identifying influential pixel groups (‘Identifying Influential Pixel Groups’ section), we introduce ASAP’s frontend interface. The system facilitates interactive and interpretable analysis and summarization of fake patterns in images produced by various generative models, across multiple levels of data granularity (i.e., global, subset, and local levels). At a high level, users begin using ASAP by selecting a generative model and subject for comparison (e.g., real versus fake “horses” generated by the “proGAN” model) in a load menu. The main interface consists of four main views, shown in Figure 1 and described in detail below.

Representation Overview

The Representation Overview (Figure 1A) summarizes fake patterns, facilitates navigation and comparison, and supports user annotations. It groups similar image representations into distinct, non-overlapping cells to minimize visual clutter (G4), and uses a custom cell glyph (described below) that summarizes the detector’s performance on the image subsets contained within each glyph, with the goal of intuitively presenting the proportions of undetected fake images. The overview plot is designed around the following design considerations:

Avoiding visual clutters via local aggregation. Using the combined encoder $f_{\text{distill}} \circ f_{\text{forget}} \circ f_{\text{CLIP}}^{\text{image}}$, we transform images into lower-dimensional distilled representations, placing images with similar patterns close together. To

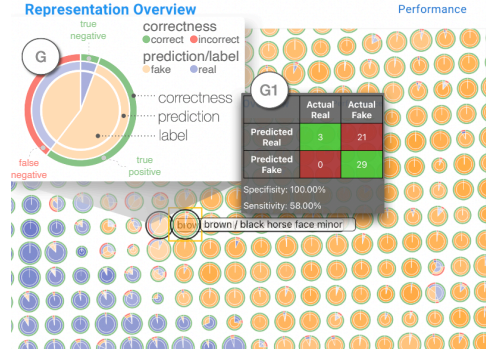


Figure 3. Our cell glyph design aggregates information for images within a cell. Users can investigate more detailed insights by hovering. Here, the tooltip shows details about a cell with severe sensitivity (see Figure 4).

avoid overlap from traditional dimensionality reduction, we use IsoMatch⁶³, which preserves pairwise distances in a structured 2D grid layout. We divide the space into a user-definable $m \times m$ grid of non-overlapping cells (m defaults to 30). Deriving cells from distilled representations in this way ensures adjacent cells reflect similar real/fake information, supporting comparative analysis while also reducing clutter/overlap. Users can explore a cell and examine adjacent cells to recognize statistical differences and real/fake boundaries (e.g., where cells go from majority blue-to-orange).

Supporting identification of suspicious cells with cell glyphs. We employ custom cell glyphs (see Figure 3) to help users quickly identify cells likely to contain deceptive or misclassified cases by visualizing the classifier’s specificity and sensitivity. For this design, we initially considered matrix-like summaries of the four confusion outcomes, but because the overview contains many small cells, we prioritized rapid visual scanning over precise reading of individual values, which can be seen via a hover tooltip. The glyph design is a pie chart with three radial layers and four sectors: true positives (fake images correctly predicted as fake), true negatives (real images correctly predicted as real), false positives (real images incorrectly predicted as fake), and false negatives (fake images incorrectly predicted as real). The outer arc uses consistent color coding: real images in blue, fake in orange, correct predictions in green, incorrect in red. Saturation indicates classifier confidence, with paler colors showing uncertainty; hovering displays a details-on-demand tooltip for exact values. The Representation Overview also supports navigation to better see glyph sectors and arcs, such as zooming and panning across glyphs along a blue-to-orange boundary in the IsoMatch grid. Glyph size also reflects the number of examples, allowing users to quickly identify challenging examples near the decision boundary. Selecting a cell highlights it with a gold square and loads its contained images into linked panels (described below) for detailed analysis; users can pan and zoom to navigate.

Annotation for incremental analysis. Users can right-click to annotate cells with pattern descriptions. Annotations persist until removed, enabling incremental understanding of fake patterns.

Image View

The Image View (Figure 1 B) activates when users select a cell, displaying its contained images and supporting tasks such as identifying shared fake patterns (G4). In this panel, displayed images are arranged in a 2D grid; we again utilize IsoMatch⁶³ for placing images in this grid (this means “similar” images will be placed close together). Each image is labeled (in its top right corner) with a circle: the color hue indicate the image’s ground truth (blue for real, orange for fake); to the right and bottom of the image, a drop shadow shows its detector prediction. Discrepancies between circle label and drop shadow (e.g., for fake images wrongly identified as real) can indicate a need for more in-depth analysis.

Above, a “Show Concept” feature displays influential visual concepts in a Concept View popup (see Figure 6 for an example), using methods from ‘A Uniform Metric for Pattern Contribution’ section, and intended to provide more insights into why misclassifications might occur. For more detailed examination, users can select images via lassoing, which selects and loads them into the below-described Pattern View for fine-grained analysis of the critical pixels and potential fake patterns they contain.

Pattern View

The Pattern View (Figure 1 D) supports detailed examination of individual images (G5), based on lassoing images in the Image View. The set of selected images is loaded into this view, which showcases the influential patterns of each selected image across the 16 distilled dimensions.

Specifically, each row shows information for a single image (when many images are loaded, users can scroll to see hidden rows), and each column information for one of the 16 dimensions. The original image is shown at the left-most position in the row (with drop shadow indicating prediction correctness), and then the image is duplicated 16 times across the row. Each of these images is overlaid with a heat map representing a pixel relevance map for its dimension (see ‘Identifying Influential Pixel Groups’ section). Contribution bars below each heatmap show the dimension’s influences on predictions (based on the methods in ‘A Uniform Metric for Pattern Contribution’ section): orange for positive influences (pushing towards ‘fake’), and blue for negative (pushing towards ‘real’). For instance, in Figure 1 D, dimension 10 shows significant negative contribution for brown horse images, which misleads the classifier.

At each row’s end, a waterfall chart summarizes the contribution of each dimension to the image’s perceived authenticity. The chart exposes dimensions that are highly influential and the corresponding pixel groups, allowing users to identify the most significant patterns (E.g., in Figure 1D, dimension 10 is key to predicting fake “brown horse, side shot” as real.)

Dimension View

To support contrastive analysis from a quantitative perspective (G4), the Dimension View (Figure 1 C) allows users to examine quantitative attributes across cells, providing insights into each dimension’s role. Users can toggle between *dimension value plots* (shown in Figure 1 C) and

a *dimension contribution plot* (shown in Figure 4 H2 and H5), revealing correlations between visual patterns and latent distributions.

Specifically, the dimension value plot uses small multiples to present dimension values across 16 dimensions. Upon initialization, it shows global distribution of real versus synthetic images. Dimensions are ordered by Kullback-Leibler divergence between real and fake distributions, with least divergence first, emphasizing dimensions with overlapping distributions requiring closer examination. For example, as shown in Figure 1 C, dimension 1 shows nearly identical distributions. When selecting a cell, plots refresh to overlay the cell’s distribution atop the global distribution, enabling users to gauge deviation from the global norm.

Toggling to the dimension contribution plot shows a set of violin plots with boxplot overlays showing key statistics on hover. This enables users to identify primary dimensions accounting for differences between visually similar images with divergent predictions (G4). Upon selecting a cell, the plot incorporates parallel coordinates displaying each image’s distilled representation. Users can use “show correct” and “show incorrect” buttons to filter contribution distributions, facilitating comparison to pinpoint critical dimensions contributing to prediction discrepancies (see Figure 4).

Usage Scenarios

To help demonstrate the capabilities of ASAP, we briefly present two usage scenarios from the perspective of a hypothetical user Ryan, who has to analyze various synthetic patterns produced by different generative models. The first scenario focuses on horse images created by the proGAN model, while the second investigates synthetic human faces generated by the LDM. In these scenarios, the analyst works with a curated dataset in which images are already grouped by generator and semantic class, and therefore begins by focusing on a specific subject (e.g., horses or human faces).

Scenario 1: Analyzing Patterns in proGAN Images. Ryan begins by selecting the proGAN model and “horse” as the subject in the interface. Upon loading, the Representation Overview (Figure 1 A) displays images organized into local cells. Ryan notices a pale central region suggestive of a decision boundary, surrounded by numerous orange cells with red arcs, indicating deceptive synthetic patterns (G4). He selects a prominent orange cell in the center, revealing multiple synthetic images of brown horses in Image View (Figure 1 B). Many of these images are misclassified as real. Ryan tags this cell as “brown horse, side shot” and explores adjacent cells with similar content (G4). Further exploration leads Ryan to a cell with a high misclassification rate featuring horse heads, which he tags as “brown/black horse head” (this cell’s glyph is shown in Figure 3).

For a more detailed analysis, Ryan investigates the “brown/black horse head” pattern by comparing two adjacent cells containing similar images but with notably different accuracy rates (Figure 4 H1, H4). Using the “Show Concepts” feature in the Concept View, he identifies shared key visual concepts within the images, such as fur and sky patterns. He then refers to the Dimension Value Plot and observes that the dimension distributions for these two cells are quite similar



Figure 4. The user compares dimension contributions (H2 and H5) of two “brown/black horse head” cells with severe (H1) and minor sensitivity (H4). Analysis revealed notable differences in dimension 2. Investigation of pixel groups linked to dimension 2 showed strong correlation with pixels near horse eyes (H3 and H6). Severe sensitivity cell images had high dimension 2 values, while minor sensitivity images showed low values, suggesting horse eye portrayal significantly influences detection.

across most dimensions, indicating that their differences are subtle. To reveal these subtle differences, Ryan examines the Dimension Contribution Plot (Figure 4 H2, H5), which reveals a significant difference in dimension 2’s contribution between the two cells. The Pattern View (Figure 4 H3, H6) further highlights that the pixel groups linked to dimension 2, which correlate with areas near the horse’s eyes, significantly impact the classification (G5). Put another way, Ryan has discovered that horse eyes and ears — even when appearing similar between real and fake images — can nonetheless play a key role in (mis) detection of fake images based on their latter pixel patterns. More broadly, these detailed analyses demonstrate how ASAP enables users to uncover subtle-yet-significant factors that contribute to the classification of synthetic images, via supporting both pattern identification and interpretation with quantitative model-derived cues.

As an addendum to this scenario, when zooming out, Ryan discovers a red-bordered blue cell contained amidst orange cells, which indicates real images were confusingly misclassified as fake. We report his analysis on this cell (which predominantly contains real horse images in 3D video games) in the Appendix.

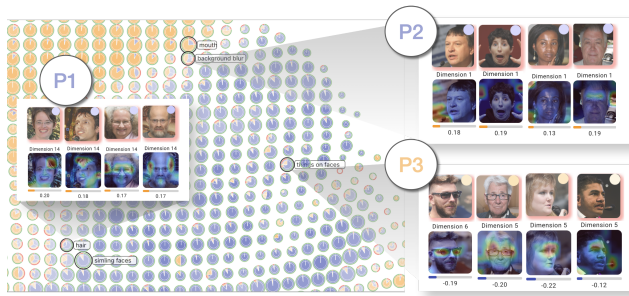


Figure 5. The user identifies common patterns in misclassified real faces, including hair (P1) and background blur (P2), as well as factors contributing to the misclassification of fake images, such as “things on faces” (P3).

Scenario 2: Analyzing Patterns in LDM Images. Ryan now switches to the LDM, with “human face” as the subject. The representation overview (shown in Figure 5) again shows a clear decision boundary between real and synthetic images. He tags two noteworthy cells: one with low sensitivity (true positive rate) containing misclassified

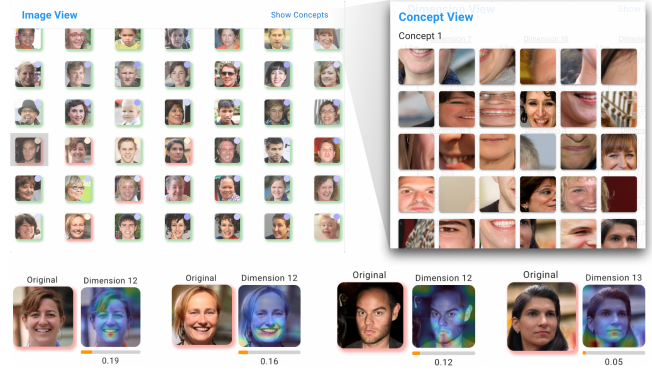


Figure 6. The user identifies a cell with misclassified images, mostly smiling faces. The Concept View reveals that the mouth significantly influences classification. Despite this, the smiles are correctly detected by the model as they have a positive contribution, indicating the classifier recognizes “fake smiles.”

synthetic images featuring “things on faces,” and the another with images of smiling individuals, which he labels “smiling faces.” Based on this, he conducts a more detailed analysis of these two patterns:

For the “things on faces” pattern, images in this cell (Figure 5 P3) show human faces with realistic items like sunglasses, eyeglasses, and microphones. These near-face objects significantly influence the classifier to predict these images as real (G4, G5). For the “smiling faces” pattern, the Concept View (Figure 6) highlights the mouth region’s influence on classification. Surprisingly, the mouth region in misclassified images contributes to a “fake” classification. Further investigation reveals that misclassifications often stem from the forehead region (G4).

In addition to analyzing fake image patterns, Ryan also investigates real images that are flagged as synthetic (G4). By examining cells with high false-negative rates (predominantly blue with red borders), he identifies recurring misclassification patterns: real images are often misjudged based on hair features associated with dimension 14 (Figure 5 P1) or background blur associated with dimension 1 (Figure 5 P2) (G4). These insights points to the classifier’s vulnerabilities, suggesting that to compromise an image’s perceived authenticity, one might only need to introduce specific patterns to trigger a false positive classification.

User Study

To empirically evaluate ASAP, we conducted a user study with 10 participants. The study aimed to answer two primary questions: (i) How well does the system support summarizing patterns of fake images (G1–G2)? (ii) How well does the system support analysis of specific fake image patterns using quantitative model-derived cues (G3–G5)? Both quantitative and qualitative data were collected to assess the system’s ability to support the design goals and its overall usability. This study was performed under the oversight of the author’s institutional review board, and we obtained informed consent from all participants prior to beginning the study.

Study Design

Participants followed a three-stage procedure:

(1) **Training:** After completing a demographics questionnaire, participants watched a 5-minute video explaining ASAP’s functionality and interface, with opportunities to ask questions. Participants then completed a training task identifying a fake pattern in bird images and reasoning about contributing dimensions to familiarize themselves with the system.

(2) **Tasks:** Participants completed two tasks designed to evaluate key aspects of the system:

T1: Identify fake patterns. This task assessed summarization capabilities (G1, G2, G4). Participants identified five fake patterns in two image classes: horses (proGAN-generated, easier) and human faces (LDM-generated, harder), without time constraints, though completion time was recorded to measure efficiency. A think-aloud protocol captured differences in mental workload between the datasets, and task correctness was assessed based on accurately identifying and describing actual patterns to the study proctor.

T2: Analyze a specific fake pattern. This task evaluated participants’ ability to use quantitative model-derived cues to interpret a specific fake pattern (G3, G4). Participants analyzed the “person-on-horse” pattern and specified relevant dimensions, with completion time recorded. Task correctness was assessed based on the accurately interpreting the pattern and describing its relevant dimensions to the study proctor.

(3) **Review:** After completing both tasks, participants completed a usability survey, rating system components using 7-point Likert scales and providing comments.

Participants and Apparatus

We recruited ten participants: eight CS graduate students and one post-doc, and one graduate student in education technology (mean age = 28.4, SD = 2.6; 7 males, 3 females). All participants had experience with generative image AI for image creation using natural language prompts (e.g., via Dall-E or MidJourney) and all were computationally proficient at image review and editing (e.g., via Photoshop), and self-rated as competent at using visual analytics for data-driven analysis. Study sessions were conducted via Zoom (avg. 35 min, SD = 12).

Usability and Effectiveness



Functionality of Specific Features

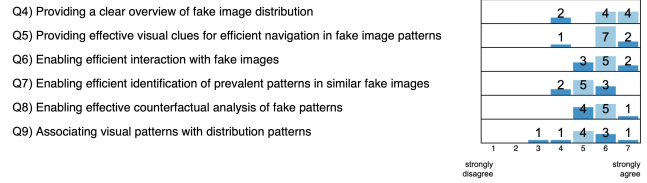


Figure 7. Participant usability ratings of ASAP from the survey during the user study review stage. Median ratings are shown in brighter blue.

Study Results

For T1, average completion time was 8 minutes (SD = 1). For T2, average completion time was 18 minutes (SD = 10), though this was expected due to its open-ended and reflective nature (e.g., exploring and reasoning about patterns). All participants successfully completed both tasks by providing the requested pattern labels and dimension-based explanations. We report quantitative ratings from the usability survey and qualitative verbal comments collected via the think-aloud protocol and summary answers. Verbal comments were thematically coded using an open coding approach⁶⁴, focusing on how the system promotes insights and supports G1–G5.

System Usability and Functionality Ratings. Figure 7 summarizes usability and functionality ratings from the review stage survey in two categories: (Q1–Q3) usability and effectiveness, and (Q4–Q9) functionality of specific features. Feedback was overall positive, with the system rated as easy to use (Q1), effective at summarizing fake patterns (Q2), and supporting analysis (Q3). The representation overview panel (Q4–Q5), image view (Q6–Q7), pattern view (Q8), and dimension view (Q9) likewise received high ratings.

User Feedback. Participant verbal comments provided additional insights into the types of insights users can generate, in the context of design goals. We summarize several themes below:

(G1, G2) Representation Overview enables efficient navigation and exploration. The Representation Overview and Image View were positively regarded for navigating the image space and exploring fake patterns. Participants tended to first inspect large cells with major orange portions bordered by red, indicating high numbers of misclassified fakes, then use the Image View for more fine-grained review. “I’m starting on cells with a lot of misclassified fake images” (p4). “This [overview] makes it easy to browse the data” (p2). “I like the efficiency [of the representation overview]. I can check a lot of images at one time” (p9).

(G4, G5) Pattern View allows in-depth analysis of specific fake patterns. Participants found the Pattern View effective for analyzing fake patterns by highlighting influential pixel groups, measuring their impact, and enabling focused analysis. “I like this view. You can see the detector checks the horse and doesn’t have an issue there, but it also checks other pixels in the background and finds the fakeness. So it knows the fake pattern among these images comes from the background” (p4). “It’s clear the neural net

picks up the person [pattern]. You see it looks at the person [rider on horse]. When the person is realistic, it passes [i.e., predicts true], and when the person is fake, it predicts fake" (p8).

(G3, G4) Dimension View allows quantitative association of fake patterns with dimension distributions. Participants were able to link fake patterns with dimension distributions. *"I can see that d8 is for the person [pattern]. It's positive [indicating fake] when it sees a fake person and near zero when it doesn't"* (p6, analyzing the "rider-on-horse" pattern in the horse class). *"d3 is a background dimension, it looks at the background and it has very little bias"* (p7, analyzing the "rider-on-horse" pattern in the horse class). *"You can see d16 and d10 are for the horse body"* (p10, analyzing the "brown horse" pattern). Some potential improvements were suggested here, such as enlarging images and providing natural language descriptions of patterns. We intend to explore these refinements as future work.

These comments also aligned with our observations of system use during the study. For example, participants tended to consistently identify the more salient patterns during their work (e.g., the above-discussed rider-on-horse pattern), though finer-grained patterns varied across individuals.

Discussion and Conclusion

ASAP addresses three major challenges currently faced in fake image analysis workflows: limited *generalizability*, lack of *interpretability*, and poor *actionability*. For generalizability, we develop a detector that leverages pre-trained models and supports seamless integration of future models. For interpretability, we use an interpretability-focused training paradigm that distills interpretable embeddings and introduces novel techniques for segmenting influential pixel regions and assessing influence. This uncovers latent, recurrent patterns in AI-generated images. An actionable interface organizes these patterns, enabling users to investigate and reason about the causes of detector successes and failures.

Overall, our usage scenarios and user study demonstrate ASAP's effectiveness in supporting design goals (G1–G5). Based on our process of developing and validating ASAP, we also identified several areas for future visual analysis and explainability of GenAI images and models:

Limits in detector capacity: Our detector's capacity is currently bound by the CLIP architecture. However, our pipeline's flexible design allows for integration of alternative or emerging detectors, imposing these two requirements: (i) there should be a differentiable feature space from which the lightweight forget-to-spell projection and distiller layer can be trained, and (ii) relevance maps should be computable. For a transformer-based encoder, swapping in a new backbone would mainly require retraining the two small linear layers on top. For non-transformers, requirement (ii) would need to be replaced with an architecture-appropriate attribution method (e.g., a CNN-style saliency technique), which is a more involved adaptation.

Real-time labeling and evaluation: Our current labeling method operates in a post-hoc manner, focusing on interactively and interpretably support AI-generated content analysis where ground truth image labels are known. While this makes ASAP highly effective as a tool for analyzing

image datasets that contain mixtures of AI-generated and real images, future iterations can adapt the system for scenarios where users want to analyze an image's authenticity in real-time (i.e., without knowing its true label).

Scalability considerations: While ASAP supports analyzing analysis at multiple granularities (from the overall image dataset in the Representation Overview down to individual image details in the Pattern and Dimension Views), we note the current system design imposes some limitations on scalability. For one, Isomatch has poor computational complexity ($O(n^3)$), which we address by pre-computing for the default layout for the Representation View grid to maintain real-time interaction. However, for users who wish to explore changing the grid size, especially for larger image ensemble datasets, this could lead to a computation bottleneck. In such cases, alternative layout algorithms could be explored, though it would be important that the results still supported workflows such as being able to recognize real/fake boundaries where cells go from majority blue-to-orange.

Another challenge for the Representation View is that, when a glyph is selected, all of its contained images are shown in the Image View. In our tested datasets, each glyph usually aggregated to at most 75 images, though this size would increase if the Representation View's grid size was shrunk or an image dataset containing millions of images was tested. In such cases, the Image View would likely need to be redesigned to better support larger amounts of selected images.

Limits of saliency-based explanations: While our relevance maps help identify image regions that influence predictions, they do not by themselves explain why those regions are influential or what specific visual attributes within them drive the model's decision. For example, a highlighted mouth region may indicate that the area is important for predicting an image as fake, but not whether this is due to shape, texture, boundary artifacts, or some other property. Richer explanations of the specific attributes driving model behavior remain an important direction for future work.

Interactive base model selection: While ASAP currently employs CLIP as its foundational visual feature encoder, the rapid advancement of multi-modal pre-trained models presents an opportunity to support interactive base model selection. This enhancement could significantly improve the detection of nuanced patterns in AI-generated images.

Integration with LLMs: As multi-modal LLMs become more powerful, there is potential to integrate into them deepfake analysis workflows, such as using them to verbally summarize fake patterns. This could provide users with a more efficient overview of fake patterns and their distribution. However, given the current limitations in LLMs' vision understanding capabilities⁶⁵, this integration will likely require more mature models (or advanced RAG approaches) than are currently available.

Ethical and privacy considerations: Finally, deepfake analysis tools such as ASAP can also raise ethical concerns, due to their potential dual-use nature: while they can support media forensics and authenticity analysis, discovered patterns could also be misused to help deceptive content evade detection. In addition, scenarios involving human face images introduce privacy concerns when facial content

is inspected and compared at scale, even if (as in our case) public benchmark datasets are used. More broadly, generative ecosystems continue to face concerns related to bias and copyright, motivating not only calls for responsible use but also the development of guardrails to constrain unintended outcomes.

Ultimately, ASAP presents a novel approach for detecting patterns and signatures in AI-generated images and facilitating their in-depth analysis. In particular, ASAP addresses key challenges in existing approaches by supporting generalizability, interpretability, and actionability, via a combination of backend techniques and interactive, human-in-the-loop visual analytics. While ASAP currently focuses on image content, as the landscape of GenAI continues to rapidly evolve, we posit systems like ASAP will play an increasingly crucial role in supporting digital media integrity and fostering trust in content.

References

1. DALL-E 3 — OpenAI. <https://openai.com/dall-e-3>.
2. Midjourney. <https://www.midjourney.com/home>.
3. Cao Y, Li S, Liu Y et al. A Survey of AI-Generated Content (AIGC). *ACM Computing Surveys* 2025; 57(5): 1–38. DOI: 10.1145/3704262.
4. Hamed A, Zachara-Szymanska M and Wu X. Safeguarding Authenticity for Mitigating the Harms of Generative AI: Issues, Research Agenda, and Policies for Detection, Fact-Checking, and Ethical AI. *iScience* 2024; 27(2): 1–6. DOI:10.1016/j.isci.2024.108782.
5. Xu D, Fan S and Kankanhalli M. Combating Misinformation in the Era of Generative AI Models. In *Proceedings of the 31st ACM International Conference on Multimedia*. p. 9291–9298. DOI:10.1145/3581783.3612704.
6. Sag M. Copyright Safety for Generative AI. *Houston Law Review* 2023; 61(2): 295–347. DOI:10.2139/ssrn.4438593.
7. Mann S, Earp B, Nyholm S et al. Generative AI Entails a Credit–Blame Asymmetry. *Nature Machine Intelligence* 2023; 5(5): 472–475. DOI:10.1038/s42256-023-00653-1.
8. Ha A, Passananti J, Bhaskar R et al. Organic or Diffused: Can We Distinguish Human Art from AI-generated Images? In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. p. 4822–4836. DOI: 10.1145/3658644.3670306.
9. Zhang X, Karaman S and Chang SF. Detecting and Simulating Artifacts in GAN Fake Images. In *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*. pp. 1–6. DOI:10.1109/WIFS47025.2019.9035107.
10. Lloyd T, Reagle J and Naaman M. ‘there has to be a lot that we’re missing’: Moderating ai-generated content on reddit. *Proc ACM Hum-Comput Interact* 2025; 9(7): 1–24. DOI: 10.1145/3757445.
11. Weber-Wulff D, Anohina-Naumeca A, Bjelobaba S et al. Testing of Detection Tools for AI-Generated Text. *International Journal for Educational Integrity* 2023; 19(26): 1–39. DOI: 10.1007/s40979-023-00146-z.
12. Sadasivan V, Kumar A, Balasubramanian S et al. Can AI-Generated Text be Reliably Detected? *arXiv preprint arXiv:230311156* 2023; DOI:10.48550/arXiv.2303.11156.
13. Corvi R, Cozzolino D, Zingarini G et al. On the Detection of Synthetic Images Generated by Diffusion Models. In *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10095167.
14. Hohman F, Kahng M, Pienta R et al. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *IEEE Transactions on Visualization and Computer Graphics* 2019; 25(8): 2674–2693. DOI:10.1109/TVCG.2018.2843369.
15. Sohrawardi S, Wu Y, Hickerson A et al. Dungeons & Deepfakes: Using Scenario-Based Role-Play to Study Journalists’ Behavior Towards Using AI-Based Verification Tools for Video Content. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. pp. 1–17. DOI:10.1145/3613904.3641973.
16. Sedlmair M, Meyer M and Munzner T. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Transactions on Visualization and Computer Graphics* 2012; 18(12): 2431–2440. DOI:10.1109/TVCG.2012.213.
17. Agarwal S and Farid H. Photo forensics from JPEG dimples. In *2017 IEEE Workshop on Information Forensics and Security*. pp. 1–6. DOI:10.1109/WIFS.2017.8267641.
18. Popescu A and Farid H. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing* 2005; 53(2): 758–767. DOI:10.1109/TSP.2004.839932.
19. O’Brien J and Farid H. Exposing Photo Manipulation with Inconsistent Reflections. *ACM Transactions on Graphics* 2012; 31(1): 1–11. DOI:10.1145/2077341.2077345.
20. Cozzolino D, Thies J, Rössler A et al. ForensicTransfer: Weakly-Supervised Domain Adaptation for Forgery Detection. *arXiv preprint arXiv:181202510* 2018; DOI:10.48550/arXiv.1812.02510.
21. Chai L, Bau D, Lim SN et al. What Makes Fake Images Detectable? Understanding Properties that Generalize. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI* 16. pp. 103–120. DOI:10.1007/978-3-030-58574-7_7.
22. Asnani V, Yin X, Hassner T et al. Proactive Image Manipulation Detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 15365–15374. DOI:10.1109/CVPR52688.2022.01495.
23. Ojha U, Li Y and Lee Y. Towards Universal Fake Image Detectors that Generalize across Generative Models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 24480–24489. DOI:10.1109/CVPR52729.2023.02345.
24. Zhu M, Chen H, Huang M et al. GenDet: Towards Good Generalizations for AI-Generated Image Detection. *arXiv preprint arXiv:231208880* 2023; DOI:10.48550/arXiv.2312.08880.
25. Cozzolino D, Poggi G, Corvi R et al. Raising the Bar of AI-generated Image Detection with CLIP. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. pp. 4356–4366. DOI: 10.1109/CVPRW63382.2024.00439.
26. Radford A, Kim JW, Hallacy C et al. Learning Transferable Visual Models from Natural Language Supervision. In *International Conference on Machine Learning*. pp. 8748–8763.
27. Li B, Weinberger K, Belongie S et al. Language-Driven Semantic Segmentation. *arXiv preprint arXiv:220103546*

- 2022; DOI:10.48550/arXiv.2201.03546.
28. Rao Y, Zhao W, Chen G et al. DenseCLIP: Language-Guided Dense Prediction with Context-Aware Prompting. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 18061–18070. DOI:10.1109/CVPR52688.2022.01755.
 29. Luo H, Bao J, Wu Y et al. SegCLIP: Patch Aggregation with Learnable Centers for Open-Vocabulary Semantic Segmentation. In *International Conference on Machine Learning*. pp. 23033–23044.
 30. Bangalath H, Maaz M, Khattak MU et al. Bridging the Gap between Object and Image-Level Representations for Open-Vocabulary Detection. *Advances in Neural Information Processing Systems* 2022; 35: 33781–33794.
 31. Zhou K, Yang J, Loy CC et al. Conditional Prompt Learning for Vision-Language Models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 16795–16804. DOI:10.1109/CVPR52688.2022.01631.
 32. Zhao Z, Liu Y, Wu H et al. CLIP in medical imaging: A comprehensive survey. *arXiv preprint arXiv:231207353* 2023; DOI:10.48550/arXiv.2312.07353.
 33. Materzyńska J, Torralba A and Bau D. Disentangling Visual and Written Concepts in CLIP. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 16389–16398. DOI:10.1109/CVPR52688.2022.01592.
 34. Barkan O, Haulon E, Caciularu A et al. Grad-SAM: Explaining Transformers via Gradient Self-Attention Maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. pp. 2882–2887. DOI: 10.1145/3459637.3482126.
 35. Chefer H, Gur S and Wolf L. Generic Attention-model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 387–396. DOI:10.1109/ICCV48922.2021.00045.
 36. Yuan T, Li X, Xiong H et al. Explaining Information Flow Inside Vision Transformers Using Markov Chain. In *NEURIPS Workshop on eXplainable AI Approaches for Debugging and Diagnosis*.
 37. Abnar S and Zuidema W. Quantifying Attention Flow in Transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. pp. 4190–4197. DOI:10.18653/v1/2020.acl-main.385.
 38. Chefer H, Gur S and Wolf L. Transformer Interpretability Beyond Attention Visualization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 782–791. DOI:10.1109/CVPR46437.2021.00084.
 39. Voita E, Talbot D, Moiseev F et al. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. pp. 5797–5808. DOI:10.18653/v1/P19-1580.
 40. Zhao Z, Xu P, Scheidegger C et al. Human-in-the-loop Extraction of Interpretable Concepts in Deep Learning Models. *IEEE Transactions on Visualization and Computer Graphics* 2021; 28(1): 780–790. DOI:10.1109/TVCG.2021.3114837.
 41. Hoque M, He W, Shekar A et al. Visual Concept Programming: A Visual Analytics Approach to Injecting Human Intelligence at Scale. *IEEE Transactions on Visualization and Computer Graphics* 2023; 29(1): 74–83. DOI:10.1109/TVCG.2022.3209466.
 42. Gou L, Zou L, Li N et al. VATLD: A Visual Analytics System to Assess, Understand and Improve Traffic Light Detection. *IEEE Transactions on Visualization and Computer Graphics* 2020; 27(2): 261–271. DOI:10.1109/TVCG.2020.3030350.
 43. Xie T, Ma Y, Kang J et al. FairRankVis: A Visual Analytics Framework for Exploring Algorithmic Fairness in Graph Mining Models. *IEEE Transactions on Visualization and Computer Graphics* 2022; 28(1): 368–377. DOI:10.1109/TVCG.2021.3114850.
 44. Huang J, Mishra A, Kwon BC et al. ConceptExplainer: Interactive Explanation for Deep Neural Networks from a Concept Perspective. *IEEE Transactions on Visualization and Computer Graphics* 2023; 29(1): 831–841. DOI:10.1109/TVCG.2022.3209384.
 45. Wongsuphasawat K, Smilkov D, Wexler J et al. Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow. *IEEE Transactions on Visualization and Computer Graphics* 2018; 24(1): 1–12. DOI:10.1109/TVCG.2017.2744878.
 46. Li J and Kuang J. Robustmap: Visual exploration of dnn adversarial robustness in generative latent space. *IEEE Transactions on Visualization and Computer Graphics* 2025; 31(9): 5801–5815. DOI:10.1109/TVCG.2024.3471551.
 47. Chatzimpampas A. Visual analytics for explainable and trustworthy artificial intelligence. *IEEE Computer Graphics and Applications* 2025; 45(2): 100–111. DOI:10.1109/MCG.2025.3533806.
 48. Ribeiro MT, Singh S and Guestrin C. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16, Association for Computing Machinery, p. 1135–1144. DOI:10.1145/2939672.2939778.
 49. Lundberg SM and Lee SI. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30.
 50. Sundararajan M, Taly A and Yan Q. Axiomatic Attribution for Deep Networks. In *International Conference on Machine Learning*. PMLR, pp. 3319–3328.
 51. Selvaraju R, Cogswell M, Das A et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*. pp. 618–626. DOI:10.1109/ICCV.2017.74.
 52. Kim B, Wattenberg M, Gilmer J et al. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International Conference on Machine Learning*. pp. 2668–2677.
 53. Yeh C, Chen Y, Wu A et al. AttentionViz: A Global View of Transformer Attention. *IEEE Transactions on Visualization and Computer Graphics* 2024; 30(1): 262–272. DOI:10.1109/TVCG.2023.3327163.
 54. Lee S, Hoover B, Strobel H et al. Diffusion Explainer: Visual Explanation for Text-to-Image Stable Diffusion. In *2024 IEEE Visualization and Visual Analytics (VIS)*. pp. 96–100. DOI: 10.1109/VIS55277.2024.00027.
 55. Feng Y, Wang X, Wong K et al. PromptMagician: Interactive Prompt Engineering for Text-to-Image Creation. *IEEE Transactions on Visualization and Computer Graphics* 2024; 30(1): 295–305. DOI:10.1109/TVCG.2023.3327168.

56. Ye Y, Huang R, Zhang K et al. Unified visual comparison framework for human and ai paintings using neural embeddings and computational aesthetics. *IEEE Computer Graphics and Applications* 2025; 45(2): 19–30. DOI:10.1109/MCG.2025.3555122.
57. Xu W, Dainoff M, Ge L et al. Transitioning to Human Interaction with AI Systems: New Challenges and Opportunities for HCI Professionals to Enable Human-Centered AI. *International Journal of Human-Computer Interaction* 2023; 39(3): 494–518. DOI:10.1080/10447318.2022.2041900.
58. Park N and Kim S. How do Vision Transformers Work? In *International Conference on Learning Representations*.
59. Wang Z, Hohman F and Chau D. WizMap: Scalable Interactive Visualization for Exploring Large Machine Learning Embeddings. In *Proceedings of the 61st Annual Meeting Of The Association For Computational Linguistics*. pp. 516–523. DOI: 10.18653/v1/2023.acl-demo.50.
60. Stepin I, Alonso J, Catala A et al. A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence. *IEEE Access* 2021; 9: 11974–12001. DOI:10.1109/ACCESS.2021.3051315.
61. Wang SY, Wang O, Zhang R et al. CNN-Generated Images are Surprisingly Easy to Spot... for Now. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 8692–8701. DOI:10.1109/CVPR42600.2020.00872.
62. Saxe A, McClelland J and Ganguli S. Exact Solutions to the Nonlinear Dynamics of Learning in Deep Linear Neural Networks. *arXiv preprint arXiv:13126120* 2013; DOI:10.48550/arXiv.1312.6120.
63. Fried O, DiVerdi S, Halber M et al. IsoMatch: Creating Informative Grid Layouts. *Computer Graphics Forum* 2015; 34(2): 155–166. DOI:10.1111/cgf.12549.
64. Thornberg R and Charmaz K. Grounded Theory and Theoretical Coding. *The SAGE Handbook of Qualitative Data Analysis* 2014; 5: 153–169. DOI:10.4135/9781446282243.n11.
65. Zhang J, Hu J, Khayatkhoei M et al. Exploring Perceptual Limitation of Multimodal Large Language Models. *arXiv preprint arXiv:240207384* 2024; DOI:10.48550/arXiv.2402.07384.

Validating the Generalizability of the Detector

We evaluate our detector’s generalizability following the methodology of Ojha et al. ²³. Our approach involves training the detector on the proGAN dataset, which contains real images from ImageNet and fake images generated by the proGAN model ⁶¹. We then test the detector on unseen datasets to assess its generalizability.

In Tables 1 and 2, we report the performance of various detection methods across different image generation techniques. The models included are:

- GAN-based: CycleGAN[?], BigGAN[?], StyleGAN[?], GauGAN[?], and StarGAN[?]
- Diffusion-based and hybrid techniques^{??}
- Other generative technologies^{????}

We report both accuracy and precision for each method. Accuracy represents the model’s ability to make correct predictions overall, while precision indicates its ability to correctly identify fake images.

Our method outperforms many of the baseline models, and is comparable to that of Ojha et al. ²³, another state-of-the-art model that also leverages CLIP. Differences in performance are likely attributed to our forget-to-spell projection and distillation process, which is needed for interpretability purposes (and which Ojha et al. lacks). In our case, this process necessarily discards some information from the CLIP embeddings, resulting in slightly less informative input for detection. However, as discussed in the main paper body, our primary goal is to discover fake patterns rather than maximize prediction accuracy. This trade-off is thus acceptable given the overall comparable performance.

Patterns of Confusing Real Horse Images

At the end of Usage Scenario 1, hypothetical user Ryan finds a cell containing real horse images that are confusingly misclassified as synthetic/fake. These are predominantly real horse images from 3D games, and dimension 6 (associated with the horse’s body) contributes significantly to their synthetic classification, which suggests that 3D-generated horses lack certain realistic features.

To reach such a finding, Figure 8 shows Ryan interacting with the cell glyph in the Representation Overview. Here, Ryan first observes that most images in one cell (O1) depict horses in 3D games. A nearby cell (O2) also contains mostly real images, but these are correctly classified. Both cells’ images feature horses with grassy backgrounds.

By comparing these cells, their dimension values, and dimensional contributions, Ryan identifies dimension 6 as the key differentiator (O3, O4). Figure 9 illustrates that dimension 6 focuses on pixels near the horse’s body or the rider. While this dimension successfully predicts real images in cell O2, it fails to do so for cell O1.

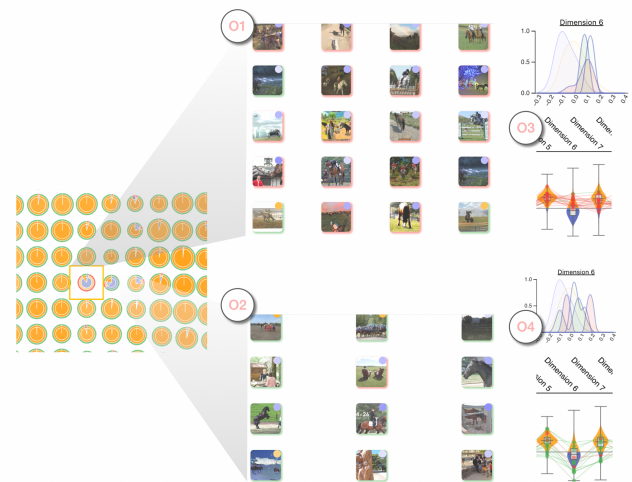


Figure 8. Real horse images from 3D games misclassified as synthetic (red-bordered cell) and nearby cells.

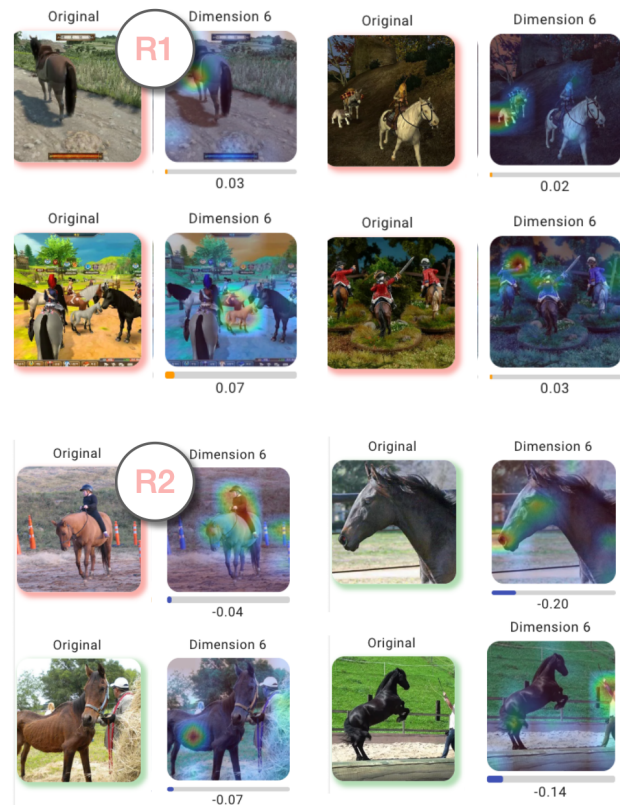


Figure 9. Dimension 6 focuses on pixels near the horse body or the rider. It correctly predicts real images (assign negative contributions) in cell O2 but fails for cell O1 (assign positive contributions).

Detection Method	Generative Adversarial Networks							Low-level Vision		Perceptual Loss		Guided	LDM			Glide			DALL-E	Avg. Acc.
	ProGAN	CycleGAN	BigGAN	StyleGAN	GauGAN	StarGAN	Deepfakes	SITD	SAN	CRN	IMLE		200 steps	200 w/ CFG	100 steps	100/27	50/27	100/10		
Wang et al. ⁶¹	99.99	85.20	70.20	85.7	78.95	91.7	53.47	66.67	48.69	86.31	86.26	60.07	54.03	54.96	54.14	60.78	63.8	65.66	55.58	69.58
Chai et al. ²¹	75.03	68.97	68.47	79.16	64.23	63.94	75.54	75.14	75.28	72.33	55.3	67.41	76.5	76.1	75.77	74.81	73.28	68.52	67.91	71.24
Nataraj et al. ²	97.70	63.15	53.75	92.50	51.1	54.7	57.1	63.06	55.85	65.65	65.80	60.50	70.7	70.55	71.00	70.25	69.60	69.90	67.55	66.86
Zhang et al. ⁹	49.90	99.90	50.50	49.90	50.30	99.70	50.10	50.00	48.00	50.60	50.10	50.90	50.40	50.40	50.30	51.70	51.40	50.40	50.00	55.45
Ojha et al. ²³	100.0	98.50	94.50	82.00	99.50	97.00	66.60	63.00	57.50	59.5	72.00	70.03	94.19	73.76	94.36	79.07	79.85	78.14	86.78	81.38
Ours	97.45	91.14	71.33	82.15	78.01	91.92	67.12	64.17	69.18	65.02	73.28	79.95	79.00	63.50	80.50	87.00	88.55	88.10	60.60	77.79

Table 1. Generalization Results (Accuracy). This table compares the classification accuracy of various methods for detecting real and fake images. Our approach achieves performance ahead of most existing methods and is comparable to the state-of-the-art method²³ with a slight difference attributable to our focus on interpretability. Our detector incorporates an orthogonality loss during training, which regularizes the model's weights. This design choice balances classification performance with the generation of orthogonal dimensions in the distiller layer, facilitating subsequent relevance-based mask generation and fake pattern discovery.

Detection Method	Generative Adversarial Networks							Low-level Vision		Perceptual Loss		Guided	LDM			Glide			DALL-E	Avg. mAP
	ProGAN	CycleGAN	BigGAN	StyleGAN	GauGAN	StarGAN	Deepfakes	SITD	SAN	CRN	IMLE		200 steps	200 w/ CFG	100 steps	100/27	50/27	100/10		
Wang et al. ⁶¹	100.0	93.47	84.5	99.54	89.49	98.15	89.02	73.75	59.47	98.24	98.4	73.72	70.62	71.0	70.54	80.65	84.91	82.07	70.59	83.58
Chai et al. ²¹	80.88	72.84	71.66	85.75	65.99	69.25	76.55	76.19	76.34	74.52	68.52	75.03	87.1	86.72	86.4	85.37	83.73	78.38	75.67	77.73
Nataraj et al. ²	99.74	80.95	50.61	98.63	53.11	67.99	59.14	68.98	60.42	73.06	87.21	70.20	91.21	89.02	92.39	89.32	88.35	82.79	80.96	78.11
Zhang et al. ⁹	55.39	100.0	75.08	55.11	66.08	100.0	45.18	47.46	57.12	53.61	50.98	57.72	77.72	77.25	76.47	68.58	64.58	61.92	67.77	66.21
Ojha et al. ²³	100.00	99.46	99.59	97.24	99.98	99.60	82.45	61.32	79.02	96.72	99.00	87.77	99.14	92.15	99.17	94.74	95.34	94.57	97.15	93.38
Ours	99.43	96.62	76.30	92.73	85.79	97.20	83.64	74.87	77.59	62.52	71.44	90.63	92.31	80.78	92.52	95.65	96.38	95.97	78.79	86.38

Table 2. Generalization Results (Average Precisions). This table extends the analysis from Table 1, focusing on the average precision of our detector when applied to diverse datasets after training on proGAN. Our method demonstrates superior precision compared to most alternatives and achieves results comparable to the state-of-the-art approach, underscoring its effectiveness in minimizing false positives across various image datasets.