EVOKNOW: CONTINUAL KNOWLEDGE EVOLUTION FOR AI-GENERATED IMAGE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

AI-generated image detectors are commonly trained on fixed generator domains and become difficult to maintain as new generative models emerge. Continual adaptation is challenging because replaying historical generated images is costly, whereas updating shared parameters with limited current-domain data can overwrite prior forensic knowledge. We propose **EvoKnow**, a replay-free framework that formulates continual AI-generated image detection as forensic knowledge evolution. EvoKnow preserves a shared forensic basis learned from base domains, incrementally adds isolated residual experts for complementary generator-relevant evidence, and retrieves expertise through an Analytical Incremental Router (AIR) updated in closed form from current-stage generated images and accumulated sufficient statistics. Experiments demonstrate effective cross-generator generalization, few-shot expansion, and long-horizon continual adaptation. With ten generated images per arriving generator, EvoKnow achieves 96.70% average accuracy on non-base GenImage generators and 94.48% accuracy on Chameleon without target-benchmark adaptation. Under a strict replay-free continual learning protocol, EvoKnow achieves state-of-the-art continual learning performance, attaining 96.32% mean stage-wise accuracy and 4.32% average forgetting.

1 INTRODUCTION

Recent advances in AI image generation have substantially improved the realism and semantic coherence of generated images. While these models enable a wide range of creative applications, their misuse also facilitates misinformation, identity impersonation, and forged visual evidence. Reliable AI-generated image detection has therefore become a fundamental problem in digital media forensics Wang et al. (2024); Zhang et al. (2025b); Guo et al. (2023).

Existing detectors are typically trained on data from a fixed set of generative models and aim to generalize beyond the observed training domains Zhu et al. (2023); Wang et al. (2026b); Xu et al. (2025). However, new generative models continuously emerge and may exhibit forensic traces absent from the original training data. When a detector fails on an emerging generator, retraining with historical and new generated images is costly and requires retaining an ever-growing image collection. Alternatively, adapting a shared detector using limited new-domain data can overwrite previously acquired forensic knowledge and cause catastrophic forgetting.

We therefore consider *generator-incremental AI-generated image detection*, where generators arrive sequentially, only a few images are available from each newly observed generator, and historical generated images cannot be replayed. The objective is to expand detection coverage to emerging generators while preserving performance on previously observed domains. The central challenge is to adapt from limited current-domain evidence without either overwriting prior forensic knowledge or retaining historical generated images.

Our key observation is that forensic evidence is neither entirely shared nor entirely domain-specific. Different generators exhibit transferable synthetic characteristics, whereas an emerging generator may require complementary evidence that is insufficiently captured by the existing representation. Continually updating a single shared model promotes transfer but is prone to forgetting, whereas learning an independent model for every generator avoids interference but redundantly models com-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

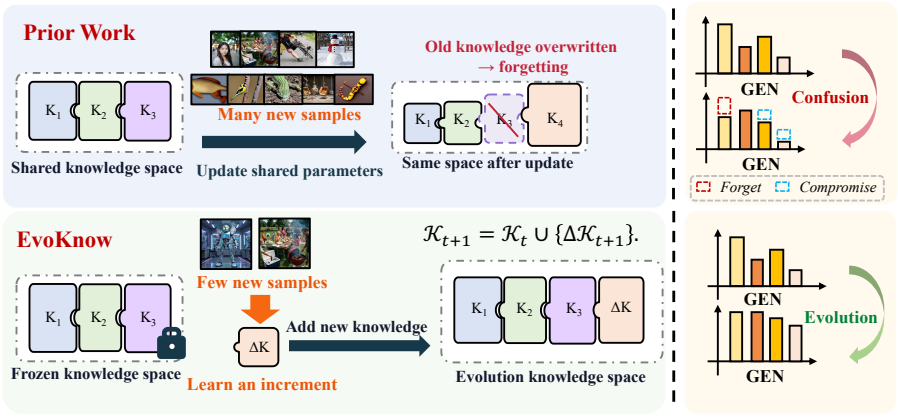


Figure 1: Continual AIGC detection with EvoKnow. Instead of repeatedly reshaping a unified forensic space, EvoKnow freezes transferable shared evidence and incrementally adds isolated residual expertise for newly arriving generators. Historical synthetic images are not replayed.

mon evidence. As illustrated in Figure 1, a practical detector should instead preserve transferable forensic knowledge and incrementally acquire complementary evidence for emerging generators.

Based on this principle, we propose **EvoKnow** (**Evo**lution of Forensic **Know**ledge), a replay-free knowledge-allocation framework for continual AI-generated image detection. EvoKnow preserves a shared forensic basis learned from base domains and frozen during incremental adaptation, while operationalizing incremental residual knowledge through isolated, unit-specific residual experts for arriving generator domains. By freezing the shared forensic learner and previously learned experts, EvoKnow avoids parameter interference during new-domain adaptation. An *Analytical Incremental Router* (AIR) performs expert retrieval through recursive closed-form updates using current-stage generated images and accumulated sufficient statistics. Our replay-free protocol excludes retaining or replaying historical AI-generated images and their instance-level features. Task-agnostic real images are sampled from a predefined shared training pool only as negative-class supervision for binary real/fake detection.

Our contributions are summarized as follows:

- **A novel perspective.** We formulate continual AI-generated image detection as replay-free forensic knowledge evolution. Rather than repeatedly updating a shared detector, this perspective preserves transferable base-domain evidence and incrementally acquires complementary forensic evidence from limited observations of emerging generators.
- **A new method.** We propose **EvoKnow**, which combines a frozen shared forensic basis, isolated incremental residual experts, and AIR for analytical expert retrieval. EvoKnow expands detection coverage from current-stage generated images and accumulated router statistics without retaining historical generated images.
- **Strong performance.** With ten generated images per arriving generator, EvoKnow achieves 96.70% average accuracy on non-base GenImage generators and 94.48% accuracy on Chameleon without target-benchmark adaptation. Under a strict replay-free continual protocol, it attains 96.32% mean stage-wise accuracy with 4.32% average forgetting, demonstrating leading continual learning performance without historical generated-image replay.

2 Related Work

2.1 Cross-Generator AI-Generated Image Detection

Early AI-generated image detectors were often designed for specific generative architectures or artifact patterns. More recent work instead learns transferable forensic representations that generalize across generators through frequency artifacts, reconstruction discrepancies, local textures,

foundation-model features, or distribution alignment Yu et al. (2025); Chen et al. (2024a); Yan et al. (2024); Tan et al. (2025); Yao et al. (2026); Liu et al. (2024b); Yang et al. (2025). These methods improve detection on unseen generators by exploiting evidence shared across generative models.

However, cross-generator detection is typically studied in a static setting, where a detector trained on fixed source domains is expected to generalize without further adaptation. In deployment, newly emerging generators may exhibit forensic characteristics insufficiently covered by the source-domain representation. Retraining on accumulated data requires repeated optimization and storage of an ever-growing collection of historical generated images. EvoKnow complements static cross-generator detection by preserving a transferable shared forensic basis and incrementally expanding detection coverage with complementary generator-relevant evidence from newly arriving domains.

2.2 Continual AI-Generated Image Detection

Continual AI-generated image detection adapts a detector to sequentially arriving generators while retaining performance on previously observed domains. Recent studies have explored continual or parameter-efficient formulations, including Wang et al. Wang et al. (2026b), Zhang et al. Zhang et al. (2025a), and OmniDFA Wu et al. (2025). More generally, continual learning mitigates forgetting through replay, regularization, constrained optimization, or parameter-efficient adaptation. Replay requires retaining historical samples, whereas repeatedly updating shared parameters can still interfere with prior knowledge.

Analytical continual learning updates regularized least-squares solutions from accumulated sufficient statistics Zhuang et al. (2022; 2024). Analytic subspace routing further combines isolated low-rank experts with recursive least-squares routing Tong et al. (2025). EvoKnow adopts this analytical update principle for AI-generated image detection, but organizes knowledge differently: it freezes a shared forensic basis and incrementally adds isolated experts for complementary generator-relevant forensic evidence. Its Analytical Incremental Router performs expert retrieval from current-stage generated images and accumulated sufficient statistics, without retaining historical generated images.

3 Method

As illustrated in Figure 2, EvoKnow instantiates continual forensic knowledge evolution through three components: a frozen shared forensic basis that preserves transferable evidence, isolated incremental residual experts that emphasize complementary generator-relevant evidence from limited current-stage generated images, and AIR for analytical expert retrieval without retaining historical generated images. The following sections formalize the replay-free setting and describe these components.

3.1 Problem Setting

We study *replay-free continual AI-generated image detection* over a sequence of emerging generator domains $\{\mathcal{D}_t\}_{t=1}^{T}$, where each domain corresponds to a specific generative model. At stage t, only a few AI-generated images $\mathcal{D}_t^{+}$ are available. For binary real/fake training, we randomly sample a task-agnostic real-image reference set $\mathcal{D}_t^{-} \subset \mathcal{D}^{\mathrm{real}}$ from a predefined shared real-image training pool. This pool is independent of generator domains and is used only as negative-class supervision; real images are excluded from AIR fitting. Historical generated images and their instance-level features are unavailable and cannot be retained or replayed.

We formulate continual adaptation as forensic knowledge evolution: transferable evidence learned from base domains is preserved, while complementary generator-relevant evidence is incrementally acquired. At stage t, the knowledge base is

$$\mathcal{K}_t = \underbrace{\{\mathcal{K}^{\mathrm{sh}}, U_0\}}_{\text{frozen base knowledge}} \cup \underbrace{\{\Delta\mathcal{K}_i\}_{i=1}^{t}}_{\text{incremental residual knowledge}}, \tag{1}$$

where $\mathcal{K}^{\mathrm{sh}}$ denotes the shared forensic basis learned by the frozen shared forensic learner, and U_0 denotes the base expert trained on top of this basis using offline base-domain data. Each $\Delta\mathcal{K}_i$ denotes

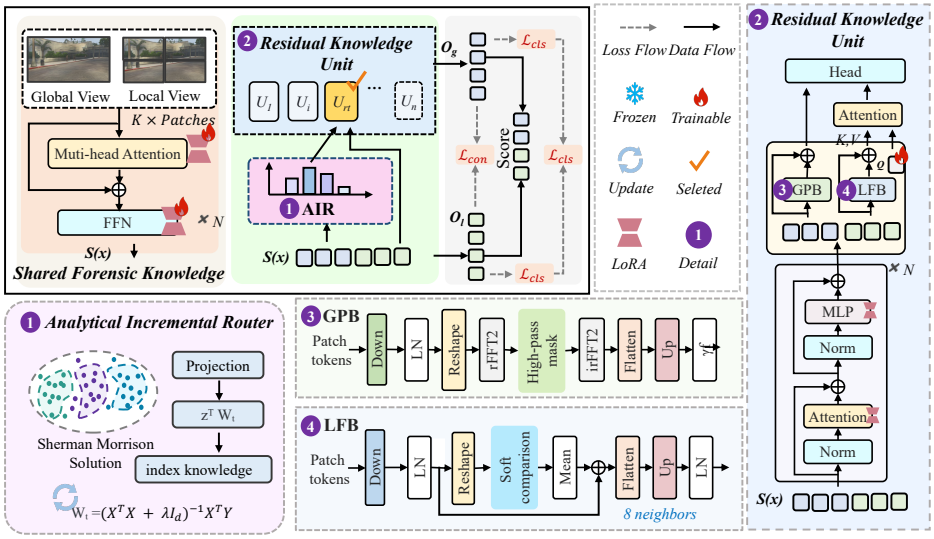


Figure 2: Overview of EvoKnow. A global view and K local views first pass through a frozen shared forensic learner. AIR performs expert retrieval over the base expert and isolated incremental residual experts. Within each expert, LoRA adapts semantic evidence shared by both branches, GPB emphasizes structural evidence in the global branch, and LFB emphasizes frequency evidence in the local branch. Cross-attention aggregates local evidence, which is fused with the global score for real-versus-synthetic prediction. When a new generator arrives, only its new incremental residual expert is trained and AIR is updated in closed form; the shared forensic learner and previously learned experts remain frozen, and historical synthetic images are not replayed.

generator-relevant residual knowledge acquired for the i-th arriving domain and is operationalized through an incremental residual expert U_i. AIR is not itself a forensic knowledge unit; rather, it performs expert retrieval among the base expert and incremental residual experts.

Given an image x, the frozen shared forensic learner outputs the routing-cut representation $S(x)$, from which AIR derives a routing feature $\mathbf{z}(x) = g(S(x))$. After learning stage t, AIR selects from the base expert and the t incremental residual experts:

$$r_t(x) = \mathcal{R}_t(\mathbf{z}(x)) \in \{0, \dots, t\}, \qquad p_t(y = 1 \mid x) = U_{r_t(x)}(S(x)). \tag{2}$$

At stage t, EvoKnow learns $\Delta\mathcal{K}_t$ from $\mathcal{D}_t^+$ and $\mathcal{D}_t^-$ while freezing the shared forensic learner, the base expert U_0, and all previously learned incremental residual experts. Starting from $\mathcal{K}_0 = \{\mathcal{K}^{\mathrm{sh}}, U_0\}$, the knowledge base is updated as

$$\mathcal{K}_t = \mathcal{K}_{t-1} \cup \{\Delta\mathcal{K}_t\}, \qquad t \geq 1. \tag{3}$$

AIR is then updated to $\mathcal{R}_t$ using current-stage generated images and accumulated sufficient statistics. Only learned model parameters and aggregated router statistics are preserved across stages.

3.2 Shared Forensic Knowledge

The shared forensic learner is built on a pretrained image encoder, e.g., DINOv3, with shared LoRA modules adapted in its first ℓ Transformer blocks. Given one global view x_g and K local views $\{x_k^{\mathrm{loc}}\}_{k=1}^K$ of image x, the patch-token features at the routing cut form the representation $S(x)$. This representation provides the shared forensic basis $\mathcal{K}^{\mathrm{sh}}$, serves as the input to the base or incremental residual experts, and is used by AIR to derive routing features.

The base expert U_0 is trained on top of $\mathcal{K}^{\mathrm{sh}}$ using offline base-domain data. Together, $\{\mathcal{K}^{\mathrm{sh}}, U_0\}$ form the frozen base knowledge after Phase I. Subsequent stages therefore preserve the shared forensic basis and adapt only through newly added incremental residual experts.

3.3 Incremental Residual Knowledge Unit

For an arriving generator domain, we operationalize $\Delta\mathcal{K}_t$ through an incremental residual expert U_t. Rather than explicitly modeling an additive correction to the output of the base expert U_0, U_t emphasizes complementary generator-relevant forensic evidence that is insufficiently captured by the frozen shared forensic basis. Each expert takes the routing-cut representation $S(x)$ as input and follows the multi-view detection paradigm of U_0 through isolated, unit-specific adaptation modules.

Specifically, unit-specific LoRA modules are inserted into the attention and MLP projections of Transformer blocks following the routing cut and are shared by the global and local branches. GPB augments the global branch with structural evidence from differentiable neighboring-patch comparisons, whereas LFB augments local branches with fine-grained spectral evidence using a learnable high-pass mask. These modules produce unit-specific residual features that are added to the corresponding branch representations. Detailed formulations of GPB and LFB are given in Appendix A.3 and Appendix A.4.

For expert U_i, unit-specific heads map global and aggregated-local features to logits o_g^i and o_l^i, respectively. Its prediction is

$$U_i(S(x)) = \left[\mathrm{softmax}\big(\mathrm{Fuse}(o_g^i, o_l^i)\big)\right]_{\mathrm{gen}}, \qquad i \in \{0, \dots, t\}, \tag{4}$$

where $[\cdot]_{\mathrm{gen}}$ denotes the AI-generated probability. After training, U_t is frozen and added to the expert set, while the shared forensic learner, the base expert U_0, and all previously learned incremental residual experts remain frozen.

3.4 Analytical Incremental Router

We introduce an *Analytical Incremental Router* (AIR) for expert retrieval. Rather than identifying the exact source generator, AIR selects the base or incremental residual expert whose forensic evidence is most compatible with an input. AIR derives a routing feature $\mathbf{z}(x) = g(S(x))$ from the routing-cut representation $S(x)$, where $g(\cdot)$ mean-pools patch tokens and applies a fixed Gaussian projection, ReLU activation, and ℓ_2 normalization. Since the shared forensic learner and $g(\cdot)$ remain frozen after base training, routing features are comparable across incremental stages.

AIR is initialized using base-domain generated images assigned to U_0. When U_t is added, AIR incorporates its expert-retrieval targets by regularized least squares with recursive updates. It retains aggregated sufficient statistics and learned router parameters, but not historical generated images or their instance-level features. At inference, AIR selects

$$r_t(x) = \arg\max_{0 \le i \le t} \mathbf{z}(x)^\top \mathbf{w}_{t,i}, \tag{5}$$

where $\mathbf{w}_{t,i}$ is the routing weight for U_i. The selected expert then performs binary real/fake detection. Detailed AIR initialization and recursive updates are provided in Appendix A.5.

3.5 Unified Multi-view Objective and Three-phase Optimization

Unified Multi-view Objective. Both the base expert and each incremental residual expert follow the same multi-view detection formulation. Let p_g, p_l, and p_f denote predictions from the global, aggregated-local, and fused branches, respectively. We use

$$\mathcal{L}_{\mathrm{cls}} = \sum_{b \in \{g,l,f\}} \omega_b \, \mathrm{CE}(y, p_b), \tag{6}$$

where CE denotes label-smoothed cross-entropy and ω_b balances branch supervision. The uniformity loss $\mathcal{L}_{\mathrm{uni}}$ encourages generated images to activate spatially distributed regions. The weak patch-level loss $\mathcal{L}_{\mathrm{patch}}$ encourages pooled patch responses to remain discriminative, and the consistency loss $\mathcal{L}_{\mathrm{cons}}$ aligns predictions across global and local views. Their detailed definitions are provided in Appendix A.1. The unified multi-view objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{cls}} + \lambda_{\mathrm{uni}}\mathcal{L}_{\mathrm{uni}} + \lambda_{\mathrm{patch}}\mathcal{L}_{\mathrm{patch}} + \lambda_{\mathrm{cons}}\mathcal{L}_{\mathrm{cons}}. \tag{7}$$

Three-phase Optimization. EvoKnow evolves forensic knowledge in three phases. Phase I learns the shared forensic basis and base expert U_0 from offline base domains. Phase II adds an incremental

residual expert for each arriving generator domain while freezing the shared forensic learner, the base expert, and all previously learned incremental residual experts. Phase III expands AIR from current-stage generated images and accumulated sufficient statistics. Phases I and II optimize the unified multi-view objective in Eq. 7 over different parameter sets, whereas Phase III updates AIR in closed form without gradient-based optimization.

In Phase I, we jointly optimize the shared forensic learner and base expert, denoted by Θ^{base}, on offline base-domain data:

$$\Theta^{\text{base}} = \arg\min_{\Theta^{\text{base}}} \mathcal{L}\big(\mathcal{D}^{+}_{\text{base}}, \mathcal{D}^{-}_{\text{base}}; \Theta^{\text{base}}\big). \tag{8}$$

The learned shared forensic learner and base expert U_0 are then frozen.

At stage t of Phase II, we instantiate U_t and optimize only its parameters Θ_t:

$$\Theta_t = \arg\min_{\Theta_t} \mathcal{L}\big(\mathcal{D}^{+}_{t}, \mathcal{D}^{-}_{t}; \Theta_t\big). \tag{9}$$

Only the unit-specific LoRA modules, GPB, LFB, and prediction heads of U_t are updated; the shared forensic learner, U_0, and $\{U_i\}_{i=1}^{t-1}$ remain frozen. The trained U_t is subsequently frozen and added to the expert set.

In Phase III, AIR is updated from routing features of current-stage generated images and accumulated sufficient statistics by recursive regularized least squares, rather than by optimizing Eq. 7.

4 Experiments

4.1 Experimental Protocol

Benchmarks and Protocol. We evaluate EvoKnow on GenImage Zhu et al. (2023), UniversalFakeDetect Ojha et al. (2023), and Chameleon Yan et al. (2025), covering controlled cross-generator expansion, diverse GAN and diffusion generators, and in-the-wild images, respectively. Unless otherwise specified, each arriving generator provides ten AI-generated images for incremental adaptation; the zero-shot Base model and adaptation-budget analysis are exceptions. At each stage, a task-agnostic real-image reference set is sampled from a predefined shared training pool for binary supervision. The pool is disjoint from evaluation splits, independent of generator domains, and excluded from AIR fitting. Historical generated images and their instance-level features are neither retained nor replayed. AIR is initialized from base-domain generated images and updated using current-stage generated images and accumulated sufficient statistics. Benchmark-specific sequences, base domains, and knowledge-unit organizations are given in the corresponding settings.

Evaluation Metrics. We report average accuracy (Ave.) for cross-generator and in-the-wild evaluation. For continual detection, let $\mathbf{B} \in \mathbb{R}^{T\times T}$ denote the task-accuracy matrix, where $B_{i,j}$ is the accuracy on task j after learning through stage i, evaluated using the complete pipeline including AIR. We report the final average accuracy and average forgetting as

$$\text{AA} = \frac{1}{T}\sum_{j=1}^{T} B_{T,j}, \qquad \text{AF} = \frac{1}{T-1}\sum_{j=1}^{T-1}\left(B_{j,j} - B_{T,j}\right). \tag{10}$$

Higher AA and lower AF are better. For the 14-stage benchmark of Wang et al. Wang et al. (2026a), we additionally report $\overline{\text{AA}} = \frac{1}{T}\sum_{t=1}^{T}\text{AA}_t$, the mean observed-domain accuracy over the continual trajectory.

Baselines and Implementation. For cross-generator evaluation, we compare EvoKnow with detectors based on frequency artifacts, reconstruction cues, local patterns, and foundation-model features, including CNNSpot Wang et al. (2020), UnivFD Ojha et al. (2023), NPR Tan et al. (2024b), FatFormer Liu et al. (2024a), AIDE Yan et al. (2025), SAFE Yao et al. (2026), MiraGe Shi et al. (2025), OmniDFA Wu et al. (2025), F3Net Li et al. (2021), FreqNet Tan et al. (2024a), C2P-CLIP Tan et al. (2025), DIRE Wang et al. (2023), ESSP Chen et al. (2024b), and LIDA Wang et al. (2026c). These reported results follow the source-domain protocols of their respective methods and therefore provide contextual comparisons rather than strictly controlled same-source evaluations. For continual detection, we compare against sequential fine-tuning, joint training, replay-based, regularization-based, and constraint-based methods under the reported 14-stage benchmark protocol Wang et al. (2026a). Implementation details and AIR initialization are provided in Appendix A.2 and Appendix A.5.

Table 1: Cross-generator and cross-benchmark detection accuracy (%). Gray rows denote base domains and orange rows denote summary metrics; base domains are excluded from averages. *Ave.* averages non-base generators, while GAN Ave. and Diffusion Ave. summarize the corresponding UniversalFakeDetect groups. Existing-detector results follow their reported ProGAN-based protocols, whereas EvoKnow is initialized from GenImage SD v1.4; this is therefore a reported-result comparison rather than a controlled same-source evaluation. **Base** denotes the zero-shot shared forensic learner and base expert, **Exp.** denotes generator-based GenImage expansion with ten images per arriving generator, and **Transfer** evaluates **Exp.** directly on external benchmarks without target adaptation. EvoKnow columns are shaded in blue. Best and second-best summary values are shown in **bold** and underlined; † denotes OmniDFA models trained with UniversalFakeDetect data.

Dataset	Generator / Metric	Existing Detectors Base: ProGAN									EvoKnow Base: SD v1.4		
		CNNSpot	UnivFD	NPR	FatFormer	C2P-CLIP	AIDE*	SAFE*	MiraGe*	OmniDFA	**Base**	**Exp.**	**Transfer**
GenImage	SD v1.4 (base)	96.3	55.6	55.1	53.6	77.5	77.2	99.9	98.8	97.47†	99.70	98.18	–
	BigGAN	46.8	84.4	57.7	82.2	85.9	50.6	77.4	96.5	97.33†	93.80	93.77	–
	Midjourney	52.8	55.1	53.4	52.1	56.6	58.2	95.7	83.2	97.58†	93.10	96.06	–
	SD v1.5	95.9	55.7	55.0	53.8	76.9	77.4	99.8	98.5	97.75†	99.70	98.53	–
	ADM	50.1	62.5	43.8	61.4	71.6	50.4	59.5	82.7	85.50†	78.10	96.21	–
	GLIDE	39.8	61.3	41.2	65.5	73.5	54.6	91.7	91.3	96.71†	95.50	96.27	–
	Wukong	78.6	61.1	57.4	60.9	79.4	70.5	95.7	97.6	97.78†	99.70	98.64	–
	VQDM	53.4	76.9	48.4	72.5	73.7	50.8	98.0	92.4	96.78†	95.70	97.41	–
	Ave. (excl. base)	59.63	65.29	50.99	64.06	73.94	58.93	88.07	91.74	95.63†	93.66	**96.70**	–
Universal FakeDetect	*GAN family*												
	ProGAN (base)	99.99	100.00	99.84	99.89	99.98	99.99	99.86	100.00	–	93.75	–	96.50
	CycleGAN	85.20	98.50	95.00	99.32	97.31	98.49	98.87	94.30	–	90.46	–	94.80
	BigGAN	70.20	94.50	87.55	99.50	99.12	83.95	89.73	96.50	–	98.03	–	99.00
	StyleGAN	85.70	82.00	96.23	97.15	96.44	99.65	98.04	96.80	–	93.73	–	96.20
	GauGAN	78.95	99.50	86.57	99.41	99.17	73.24	91.52	93.60	–	97.99	–	98.80
	StarGAN	91.70	97.00	99.75	99.75	99.60	99.90	99.90	96.10	–	93.83	–	98.00
	GAN Ave. (excl. base)	82.35	94.30	93.02	**99.03**	98.33	91.05	95.61	95.46	–	94.81	–	97.36
	Diffusion family												
	Guided	65.66	78.14	97.35	94.20	96.10	97.85	97.25	98.00	–	98.50	–	99.10
	LDM_{200}	60.07	70.03	84.55	76.00	69.10	88.50	82.35	82.00	–	83.60	–	91.50
	LDM_{200c}	54.03	94.19	97.65	98.60	99.25	98.20	98.80	98.30	–	99.27	–	99.55
	$GLIDE_{100}$	54.96	73.76	98.00	94.90	97.25	97.45	98.65	94.60	–	98.50	–	99.10
	$GLIDE_{100/27}$	54.14	94.36	98.20	98.65	99.30	98.40	98.80	98.60	–	99.30	–	99.60
	$GLIDE_{50/27}$	60.78	79.07	96.25	94.35	95.25	98.15	95.75	97.50	–	97.97	–	98.80
	$GLIDE_{100/10}$	63.80	79.85	97.15	94.65	95.25	98.40	96.60	97.50	–	97.97	–	98.80
	DALL-E	55.58	86.78	87.15	98.75	98.55	97.50	97.50	98.60	–	98.53	–	99.20
	Diffusion Ave.	58.63	82.02	94.54	93.76	93.76	96.81	95.71	95.64	–	96.71	–	**98.21**
	Ave. (excl. base)	67.75	86.74	93.95	95.79	95.51	94.59	95.67	95.57	–	95.98	–	**97.88**
Chameleon	*Acc.*	60.11	55.62	58.13	51.20	51.10	62.60	–	–	83.48	93.92	–	**94.48**

4.2 Generalization of Shared Forensic Knowledge

We first evaluate whether the shared forensic basis learned in Phase I generalizes beyond its base domain before incremental expansion. Table 1 reports the zero-shot **Base** configuration, trained only on GenImage SD v1.4, together with the expanded configurations used in subsequent analyses. Despite this single-domain initialization, EvoKnow (Base) achieves 93.66% average accuracy on non-base GenImage generators and 95.98% on UniversalFakeDetect excluding the ProGAN reference domain. It further reaches 93.92% accuracy on the in-the-wild Chameleon benchmark. Under the respective reported source-domain protocols in Table 1, this Chameleon result is 10.44 percentage points higher than the reported OmniDFA result Wu et al. (2025). These results suggest that the frozen shared forensic basis captures transferable real-versus-generated evidence and provides a strong starting point for incremental knowledge expansion.

4.3 Few-shot Incremental Knowledge Expansion and Transfer

We next evaluate whether Phase II expands the frozen shared forensic basis using limited evidence from emerging generators. On GenImage, each arriving generator adds an incremental residual expert trained with only ten AI-generated images, while the shared forensic learner, base expert, and previously learned experts remain frozen. Under this generator-based 10-shot setting, average accu-

Table 2: Continual detection on the 14-stage multi-source benchmark of Wang et al. Wang et al. (2026a), covering GANs, deepfakes, and diffusion models. ProGAN is the base domain, and each subsequent generator provides ten generated images for adaptation. Baselines follow their reported continual-learning protocols; Joint explicitly uses accumulated historical data. In contrast, EvoKnow neither retains nor replays historical generated images or their instance-level features, and uses task-agnostic real-image supervision for binary detection. The first 14 columns report stage-wise observed-domain accuracy AA_t using the complete AIR pipeline; $\overline{\mathrm{AA}}$ averages AA_t over the trajectory, and AF is in percentage points. Baseline AF signs are converted for consistent comparison. Best non-Joint results are bolded, with ties highlighted once.

Method	Observed-domain average accuracy (AA_t, %)														**Continual summary**	
	ProGAN	Deepfake	BigGAN	StyleGAN2	DDPM	ADM	DALL-E	GLIDE	SD v1.4	Midjourney	VQDM	SD v2.1	SDXL v1.0	SD v3.0	$\overline{\mathrm{AA}}$ ↑	AF ↓
Seq	**99.99**	94.46	81.01	94.05	90.35	80.14	79.66	76.68	84.50	73.82	74.31	51.90	57.29	73.85	79.43	23.04
Joint	99.99	98.77	99.34	99.57	98.15	98.97	98.81	98.54	98.80	98.30	98.38	95.69	95.17	96.01	98.18	1.77
ER Chaudhry et al. (2019)	99.99	98.26	96.77	96.96	90.92	85.97	86.18	84.65	84.11	83.94	83.73	84.51	86.02	90.68	89.48	12.10
EWC Kirkpatrick et al. (2017)	99.99	92.57	91.63	93.73	89.44	82.87	79.85	70.47	82.35	80.11	77.10	67.97	69.39	78.11	82.54	21.52
OSLA Ritter et al. (2018)	99.99	98.37	90.25	94.98	87.69	76.65	85.05	75.53	83.32	77.86	85.92	69.61	73.33	79.08	84.12	23.77
A-GEM Chaudhry et al. (2018)	99.99	87.70	85.48	93.03	92.70	85.18	80.14	70.58	83.37	78.33	75.76	66.07	68.50	73.91	81.48	25.52
SI Zenke et al. (2017)	99.99	93.82	82.67	92.32	90.68	84.65	76.98	74.66	80.73	77.57	80.98	64.58	69.54	75.26	81.75	17.40
iCaRL Rebuffi et al. (2017)	99.99	94.38	83.19	94.63	91.52	79.87	87.93	78.18	84.64	76.18	79.38	64.26	65.29	70.51	82.14	24.72
Wang et al. Wang et al. (2026a)	99.99	**99.06**	**99.13**	**99.07**	98.38	**97.30**	96.13	95.95	94.90	93.53	**96.20**	87.41	89.55	92.32	95.64	8.91
EvoKnow (10-shot)	99.97	95.55	96.75	94.85	**98.81**	96.24	**98.39**	**96.30**	**98.16**	**97.39**	96.08	**93.28**	**92.40**	**94.31**	**96.32**	**4.32**

racy on non-base generators improves from 93.66% to 96.70%; notably, ADM accuracy increases from 78.10% to 96.21%. The expanded model also improves external evaluation without target-benchmark adaptation, raising UniversalFakeDetect Ave. from 95.98% to 97.88% and Chameleon accuracy from 93.92% to 94.48%. These results suggest that incremental residual experts provide complementary generator-relevant forensic evidence beyond the frozen shared basis, improving both observed-domain coverage and external transfer. Alternative knowledge-unit organizations are analyzed in Section 4.5.

4.4 Replay-free Continual Knowledge Evolution

We next evaluate long-horizon continual adaptation on the 14-stage multi-source benchmark of Wang et al. Wang et al. (2026a), spanning GANs, deepfakes, and diffusion models. Unlike GenImage experiments, this benchmark uses ProGAN as the base domain to initialize the shared forensic learner, base expert U_0, and AIR expert-0 routing target. Each subsequent generator provides ten AI-generated images, while historical generated images are neither retained nor replayed.

Table 2 reports stage-wise observed-domain accuracy AA_t, its trajectory average $\overline{\mathrm{AA}}$, and average forgetting (AF). EvoKnow achieves 96.32% mean stage-wise accuracy with 4.32% average forgetting, yielding the highest reported mean stage-wise accuracy and the lowest reported average forgetting among non-Joint methods. Unlike the Joint reference, EvoKnow never accesses accumulated historical generated images. These results support the knowledge-evolution design: the frozen shared forensic learner preserves transferable evidence, incremental residual experts add complementary evidence without modifying prior experts, and AIR expands expert retrieval through accumulated sufficient statistics.

4.5 Ablation and Analysis

We analyze the multi-view forensic formulation, routing representation, and adaptation-sample budget of EvoKnow. Additional analysis of residual knowledge organization is provided in Appendix B.3.

Multi-view Forensic Modeling. Table 3(a) evaluates the multi-view formulation, forensic cues, and auxiliary supervision. Combining global and local views improves balanced accuracy from 89.18% to 93.54%, showing that image-level structural context and local appearance evidence are complementary. Cross-view consistency and weak patch supervision further improve the full model to 94.40%. These results support the use of multi-view supervision for learning spatially distributed and cross-view-consistent forensic evidence.

Routing Representation and Expert Specialization. Table 3(b) shows that routing features improve linear-probe accuracy from 68.58% to 79.42% and k-NN accuracy from 67.83% to 78.83%,

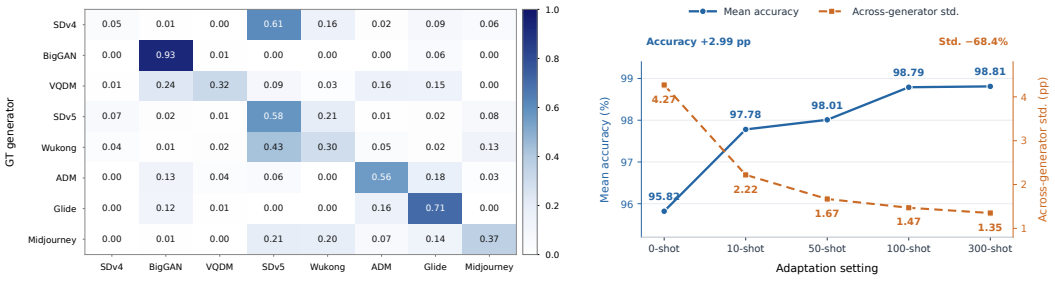
(a) Multi-view modeling ablation (BA, %)

Views	Cues	Cons.	Patch.	BA
Global	none			88.99
Global	structure			89.16
Global	frequency			89.03
Global	structure+frequency			89.18
Global+local	structure+frequency			93.54
Global+local	structure+frequency	✓		93.98
Global+local	structure+frequency		✓	93.87
Global+local	structure+frequency	✓	✓	**94.40**

(b) Frozen-representation probes (%)

Representation	Linear	k-NN
Shared	68.58	67.83
Shared+structure	69.50	67.92
Shared+structure+frequency	69.50	68.42
Router	**79.42**	**78.83**

Table 3: Ablation and routing-representation analysis of EvoKnow. (a) Balanced accuracy (BA) under different view configurations, forensic cues, cross-view consistency (Cons.), and weak patch-level supervision (Patch.). (b) Five-fold linear-probe and k-NN accuracy of frozen shared and routing representations.



(a) Generator-to-expert routing assignments

(b) Accuracy and standard deviation trends

Figure 3: Expert retrieval and adaptation-sample analysis. (a) Generator-to-expert routing assignments. (b) Effect of the GenImage adaptation budget on cross-generator evaluation over UniversalFakeDetect, reported by average accuracy and across-generator standard deviation. UniversalFakeDetect is used only for evaluation.

indicating that they expose expert-relevant forensic variation more clearly than the shared representation. As visualized in Figure 3(a), generators with related forensic characteristics exhibit similar expert-selection patterns, whereas more distinctive generators activate specialized incremental residual experts. AIR is therefore used for expert retrieval rather than exact source attribution; additional diagnostics are provided in Appendix B.4.

Adaptation-Sample Budget. Figure 3(b) varies the number of generated adaptation images in the fixed GenImage stream and evaluates transfer on UniversalFakeDetect. Larger budgets improve average accuracy and reduce across-generator variation. Notably, ten images per arriving generator already improve coverage beyond the frozen shared forensic basis, whereas additional samples mainly benefit generators that are challenging under zero-shot transfer. Complete generator-wise results are reported in Appendix B.2.

5 Conclusion

We presented **EvoKnow**, a replay-free knowledge-allocation framework for continual AI-generated image detection. EvoKnow preserves a shared forensic basis learned from base domains, incrementally adds isolated residual experts for complementary generator-relevant evidence from limited current-stage generated images, and uses AIR for analytical expert retrieval through accumulated sufficient statistics. This design expands detection coverage without retaining or replaying historical generated images or their instance-level features. Experiments demonstrate strong cross-generator and in-the-wild generalization, effective few-shot expansion, and robust continual adaptation. On the 14-stage benchmark, EvoKnow achieves the highest reported mean stage-wise accuracy and lowest reported average forgetting among non-Joint methods. These results suggest that continual detectors can preserve transferable forensic evidence, incrementally acquire complementary knowledge, and retrieve relevant expertise without image replay.

AI USE STATEMENT

We used generative AI tools to improve the clarity, conciseness, grammar, organization, and consistency of author-written text, including notation, terminology, experimental protocols, and table/figure presentation. We also used these tools to help document author-written implementation code and express GPB and LFB in mathematical form; all such descriptions were checked against the implementation and revised by the authors.

Generative AI was not used to develop the research ideas, formulate the core hypotheses, design the methodology, generate data, run experiments, produce or interpret results, or make scientific claims. The authors developed, verified, and approved all methodological decisions, experiments, analyses, figures, tables, and claims, and take full responsibility for the final manuscript.

ETHICS STATEMENT

EvoKnow aims to improve the robustness and continual adaptability of AI-generated image detection, which may help mitigate misinformation, impersonation, and forged visual evidence. Our experiments use publicly available benchmarks, including GenImage, UniversalFakeDetect, and Chameleon, and do not collect new personal data or involve human subjects.

AI-generated image detection remains imperfect: false positives may label authentic images as synthetic, while false negatives may miss generated or manipulated content. Performance may also vary across generators, sources, content, demographic groups, geographic contexts, and post-processing conditions. EvoKnow should therefore not serve as the sole basis for high-stakes decisions, including legal judgments, content moderation, journalism, or decisions affecting individuals; its outputs should support human judgment and context-specific verification.

Although our replay-free setting does not retain or replay historical generated images, it retains model parameters, incremental residual experts, and aggregated router statistics. Replay-free learning therefore does not eliminate all privacy risks. We encourage responsible evaluation of reliability, fairness, privacy, and downstream impacts before deployment.

REPRODUCIBILITY STATEMENT

Section 3 specifies the replay-free problem setting, shared forensic learner, incremental residual experts, AIR, and three-phase optimization. Section 4 describes the benchmarks, evaluation metrics, baselines, and continual-learning protocols. The appendix provides auxiliary-loss definitions (Appendix A.1), implementation details (Appendix A.2 and Table 5), GPB and LFB formulations (Appendices A.3–A.4), and AIR initialization and recursive updates (Appendix A.5). Appendix B further reports continual trajectories, retention analyses, adaptation-budget studies, router-state complexity, and attribution diagnostics. We will release source code, training and evaluation scripts, model configurations, and reproduction instructions upon acceptance.

REFERENCES

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Baoying Chen, Jishen Zeng, Jianquan Yang, and Rui Yang. Drct: Diffusion reconstruction contrastive training towards universal detection of diffusion generated images. In *Forty-first International Conference on Machine Learning*, 2024a.

Jiaxuan Chen, Jieteng Yao, and Li Niu. A single simple patch is all you need for ai-generated image detection. *arXiv preprint arXiv:2402.01123*, 2024b.

Danhuai Guo, Huixuan Chen, Ruoling Wu, and Yangang Wang. Aigc challenges and opportunities related to public safety: A case study of chatgpt. *Journal of Safety Science and Resilience*, 4(4): 329–339, 2023.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Jiaming Li, Hongtao Xie, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6454–6463. IEEE, 2021.

Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10770–10780, 2024a.

Jiawei Liu, Fanrui Zhang, Jiaying Zhu, Esther Sun, Qiang Zhang, and Zheng-Jun Zha. Forgerygpt: Multimodal large language model for explainable image forgery detection and localization. *arXiv preprint arXiv:2410.10238*, 2024b.

Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. 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. IEEE, 2023.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 31, 2018.

Kuo Shi, Jie Lu, Shanshan Ye, Guangquan Zhang, and Zhen Fang. Mirage: Multimodal discriminative representation learning for generalizable ai-generated image detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 353–361, 2025.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5052–5060, 2024a.

Chuangchuang Tan, 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 28130–28139, 2024b.

Chuangchuang Tan, Renshuai Tao, Huan Liu, Guanghua Gu, Baoyuan Wu, Yao Zhao, and Yunchao Wei. C2p-clip: Injecting category common prompt in clip to enhance generalization in deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7184–7192, 2025.

Kai Tong, Kang Pan, Xiao Zhang, Erli Meng, Run He, Yawen Cui, Nuoyan Guo, and Huiping Zhuang. Analytic subspace routing: How recursive least squares works in continual learning of large language model. *arXiv preprint arXiv:2503.13575*, 2025.

Hanyi Wang, Jun Lan, Yaoyu Kang, Huijia Zhu, Weiqiang Wang, Zhuosheng Zhang, and Shilin Wang. Generalizable and adaptive continual learning framework for ai-generated image detection. *IEEE Transactions on Multimedia*, 28:2693–2704, 2026a. doi: 10.1109/TMM.2026.3651067.

Hanyi Wang, Jun Lan, Yaoyu Kang, Huijia Zhu, Weiqiang Wang, Zhuosheng Zhang, and Shilin Wang. Generalizable and adaptive continual learning framework for ai-generated image detection. *IEEE Transactions on Multimedia*, 2026b.

Hongsong Wang, Renxi Cheng, Chaolei Han, and Jie Gui. Attribution as retrieval: Model-agnostic ai-generated image attribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14062–14072, 2026c.

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. 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, 2020. doi: 10.1109/CVPR42600.2020.00872.

Tao Wang, Yushu Zhang, Shuren Qi, Ruoyu Zhao, Zhihua Xia, and Jian Weng. Security and privacy on generative data in aigc: A survey. *ACM Computing Surveys*, 57(4):1–34, 2024.

Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22445–22455, 2023.

Shiyu Wu, Shuyan Li, Jing Li, Jing Liu, and Yequan Wang. Omnidfa: A unified framework for open set synthesis image detection and few-shot attribution. *arXiv preprint arXiv:2509.25682*, 2025.

Qijie Xu, Can Wang, Jiawei Chen, Siwei Lyu, and Defang Chen. Fully ai-generated image detection: Definition, recent advances and challenges. *arXiv preprint arXiv:2502.19716*, 2025.

Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for ai-generated image detection. In *International Conference on Learning Representations*, volume 2025, pp. 70702–70720, 2025.

Zhiyuan Yan, Jiangming Wang, Zhendong Wang, Peng Jin, Ke-Yue Zhang, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Effort: Efficient orthogonal modeling for generalizable ai-generated image detection. *arXiv preprint arXiv:2411.15633*, 2(6):7, 2024.

Zheng Yang, Ruoxin Chen, Zhiyuan Yan, Ke-Yue Zhang, Xinghe Fu, Shuang Wu, Xiujun Shu, Taiping Yao, Shouhong Ding, Zequn Qin, et al. All patches matter, more patches better: Enhance ai-generated image detection via panoptic patch learning. *arXiv preprint arXiv:2504.01396*, 2025.

Yulin Yao, Kangfeng Zheng, Bin Wu, Chunhua Wu, Jujie Wang, Jiaqi Gao, Minjiao Yang, and Dan Luo. Safe: Semantic-and frequency-enhanced curriculum for cross-domain deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11919–11927, 2026.

Peipeng Yu, Jianwei Fei, Hui Gao, Xuan Feng, Zhihua Xia, and Chip Hong Chang. Unlocking the capabilities of large vision-language models for generalizable and explainable deepfake detection. *arXiv preprint arXiv:2503.14853*, 2025.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pp. 3987–3995. Pmlr, 2017.

Hongyu Zhang, Yijia Duan, Chenxuan Liu, and Yuexuan Guo. Dynamic risk image detection and early warning in mines via aigc-driven generative adversarial augmentation and incremental learning. In *2025 7th International Conference on Frontier Technologies of Information and Computer (ICFTIC)*, pp. 919–923. IEEE, 2025a.

Wenhua Zhang, Weicheng Li, Xuanrong Rao, Lixin Zou, Xiangyang Luo, Chubin Zhuang, Yongjie Hong, Zhen Qin, Hengyu Chang, Chenliang Li, et al. Aiguard: A benchmark and lightweight detection for e-commerce aigc risks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 12437–12450, 2025b.

Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. Genimage: A million-scale benchmark for detecting ai-generated image. *Advances in neural information processing systems*, 36:77771–77782, 2023.

Huiping Zhuang, Zhen Weng, Hong Wei, Rui Xie, Kok-Aun Toh, and Zhouchen Lin. Acil: Analytic class-incremental learning with absolute memorization and privacy protection. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11602–11614, 2022.

Huiping Zhuang, Yifan Chen, Dong Fang, Ran He, Kaizhi Tong, Hong Wei, Zhifeng Zeng, and Chun Chen. Gacl: Exemplar-free generalized analytic continual learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 83024–83047, 2024.

A Additional Methodological Details

This appendix details the auxiliary losses, implementation protocol, GPB, LFB, and analytical router updates used by EvoKnow.

A.1 Detailed Definitions of Auxiliary Multi-view Losses

This section details the auxiliary losses in Eq. 7, which are shared by Phases I and II. Let $\mathcal{B}$ denote a mini-batch and $\mathcal{B}_+ \subseteq \mathcal{B}$ its generated-image subset. Expectations are averaged over the indicated samples.

Patch-response Uniformity. For a generated image x and a view $v \in \{g, l_1, \ldots, l_K\}$, let $s^v(x) \in \mathbb{R}^N$ denote its auxiliary patch-logit vector, where N is the number of patches. We obtain a spatial response distribution

$$a^v(x) = \mathrm{softmax}\left(\frac{s^v(x)}{\tau}\right), \tag{11}$$

where τ is a temperature parameter, set to 1 in all experiments. Let $u \in \mathbb{R}^N$ be the uniform distribution, with $u_j = 1/N$. The uniformity loss is

$$\mathcal{L}_{\mathrm{uni}} = \mathbb{E}_{x\sim\mathcal{B}_+}\left[D_{\mathrm{KL}}\big(a^g(x)\|u\big) + \frac{1}{K}\sum_{k=1}^{K} D_{\mathrm{KL}}\big(a^{l_k}(x)\|u\big)\right], \tag{12}$$

where

$$D_{\mathrm{KL}}(a\|u) = \sum_{j=1}^{N} a_j \log\frac{a_j}{u_j}. \tag{13}$$

This loss is applied only to generated images and is set to zero when $\mathcal{B}_+$ is empty.

Weak Patch-level Supervision. To ensure that spatially distributed patch responses remain discriminative, we weakly supervise them using image-level real/fake labels. For each image x, we average the patch logits within the global view and across all local views:

$$\bar{s}_g(x) = \frac{1}{N}\sum_{j=1}^{N} s_j^g(x), \qquad \bar{s}_l(x) = \frac{1}{KN}\sum_{k=1}^{K}\sum_{j=1}^{N} s_j^{l_k}(x). \tag{14}$$

The weak patch-level classification loss is

$$\mathcal{L}_{\mathrm{patch}} = \mathbb{E}_{(x,y)\sim\mathcal{B}}\left[\ell_{\mathrm{BCE}}\big(\bar{s}_g(x), y\big) + \ell_{\mathrm{BCE}}\big(\bar{s}_l(x), y\big)\right], \tag{15}$$

where

$$\ell_{\mathrm{BCE}}(s, y) = -y\log\sigma(s) - (1-y)\log\big(1-\sigma(s)\big). \tag{16}$$

This objective requires only image-level annotations and does not use patch-level labels.

Cross-view Consistency. The global and local views describe the authenticity of the same image and should therefore yield consistent real-versus-generated decisions. Let $o_g(x)$, $o_l(x)$, and $o_{l_k}(x)$ denote the two-class logits of the global view, aggregated-local representation, and k-th local view, respectively. We define the generated-versus-real margin as

$$m_v(x) = o_{v,1}(x) - o_{v,0}(x). \tag{17}$$

The cross-view consistency loss is

$$\mathcal{L}_{\mathrm{cons}} = \mathbb{E}_{(x,y)\sim\mathcal{B}}\left[\frac{1}{K}\sum_{k=1}^{K}\big(m_{l_k}(x) - m_g(x)\big)^2 + \big(m_l(x) - m_g(x)\big)^2\right]. \tag{18}$$

The first term aligns individual local views with the global view, whereas the second aligns the aggregated-local representation with the global prediction. Together with image-level classification supervision, these objectives encourage spatially distributed, discriminative, and cross-view-consistent forensic responses.

A.2 Implementation and No-Historical-Generated-Image-Replay Protocol

This section specifies the architecture, optimization, and data-access protocol. Unless stated otherwise, experiments use the settings in Table 4.

Model Configuration. We use DINOv3-L/16 with 224×224 inputs, one global view, and four local views. The first six encoder blocks form the frozen shared forensic learner and define the routing cut, whose patch-token output is denoted by $S(x)$. AIR derives the $10{,}000$-dimensional routing feature $\mathbf{z}(x)$ by mean-pooling $S(x)$, followed by a fixed Gaussian projection, ReLU, and ℓ_2 normalization. Transformer blocks after the routing cut belong to the base or incremental residual experts. Each incremental residual expert uses rank-8 LoRA modules, with GPB and LFB inserted into three deeper blocks.

Three-phase Optimization and Data Access. In Phase I, shared LoRA modules and the base expert U_0 are trained on the base domain, whose generated images also initialize AIR's expert-0 target. In Phase II, only the newly added incremental residual expert is optimized; the shared forensic learner and all prior experts remain frozen. Unless otherwise specified, each arriving generator provides ten generated images.

At each stage, we sample $1{,}000$ task-agnostic real images from a predefined training pool for negative-class supervision. The pool is disjoint from evaluation splits, independent of generator domains, and excluded from AIR fitting. In Phase III, AIR is updated using only current-stage generated images and the new expert identifier. Historical generated images and instance-level routing features are discarded after each stage; only aggregated router statistics and learned parameters are retained.

In Phase III, AIR is expanded using only current-stage generated images and the identifier of the newly added incremental residual expert. After each stage, historical generated images and all instance-level routing features are discarded. Routing information from earlier stages is preserved only through aggregated sufficient statistics and learned router parameters, as detailed in Appendix A.5 and Appendix B.4.

A.3 Global Pattern-aware Block: Differentiable Structural Comparison

The Global Pattern-aware Block (GPB) augments global-view features with explicit structural evidence by comparing each patch with its spatial neighbors. We use a smooth relaxation of hard comparisons so that the detection objective can optimize structural responses end-to-end.

Let $\mathbf{F} \in \mathbb{R}^{HW \times C}$ denote the patch-token features after a Transformer block, excluding class and storage tokens. An affine down-projection D_g, followed by LayerNorm and spatial reshaping, produces

$$\mathbf{Z}_g = \mathrm{reshape}(\mathrm{LN}(D_g(\mathbf{F}))), \tag{19}$$

where $\mathbf{Z}_g$ has $C' = \lfloor C/8 \rfloor$ channels. For a patch location i and one of its eight neighbors $j \in \mathcal{N}_8(i)$, let $d_{ij} = Z_{g,j} - Z_{g,i}$ denote their feature difference in a given channel. A hard comparison would produce $h(d) = \mathbb{1}[d \geq 0]$, whose derivative is zero away from the threshold and undefined at $d = 0$. GPB therefore replaces the hard comparison with the following differentiable relaxation:

$$\begin{gathered} h(d) = \mathbb{1}[d \geq 0] \quad \longrightarrow \quad s_k(d) = \sigma(kd), \\ k = \mathrm{clip}(e^{\eta}, 1, 50), \end{gathered} \tag{20}$$

where σ is the sigmoid function and η is a learnable scalar initialized to $\log 10$. The corresponding feature-difference derivative is

$$\frac{\partial s_k(d)}{\partial d} = k\, s_k(d)\big(1 - s_k(d)\big). \tag{21}$$

This derivative allows the detection objective to adjust both the center and neighboring features, with opposite signs through their difference. Larger k sharpens the transition and concentrates sensitivity near $d = 0$, whereas smaller k spreads sensitivity over a wider range of feature differences. The derivative at $d = 0$ is $k/4$. The sharpness is trainable within its clipping bounds; clipping can suppress its gradient outside that interval.

Table 4: Implementation details and hyperparameter settings.

Phase	Hyperparameter	Value
Shared knowledge learning	Backbone	DINOv3-L/16
	Input resolution	224×224
	Global / local views	1 / 4
	Optimizer	AdamW
	Learning rate	5×10^{-6}
	Weight decay	10^{-2}
	Batch size	8
	Optimization steps	3,000
Residual knowledge learning	Generated images per stage	10
	Task-agnostic real images sampled per stage	1,000
	Optimizer	AdamW
	Learning rate	5×10^{-6}
	Batch size	4
	Training epochs	2
	LoRA rank	8
	Trainable modules	LoRA, GPB, LFB, and heads
Analytical router	Feature source	First 6 encoder blocks
	Routing feature dimension	10,000
	Feature mapping	Gaussian projection + ReLU + ℓ_2 norm.
	Update rule	Recursive least squares
	Ridge coefficient λ_{RLS}	1,000
	Router fitting data	Current generated images only
Multi-view objective	λ_{cons}	0.2
	λ_{uni}	0.02
	λ_{patch}	0.1
	$(\omega_g, \omega_l, \omega_f)$	$(0.3, 0.3, 1.0)$

GPB applies the comparison independently to each bottleneck channel and averages the eight neighboring responses:

$$\mathbf{R}_{g,i} = \frac{1}{8} \sum_{j \in \mathcal{N}_8(i)} \sigma(k(\mathbf{Z}_{g,j} - \mathbf{Z}_{g,i})) \,. \tag{22}$$

Reflection padding defines the neighborhood at feature-map boundaries. The averaging produces a local relative-order response inspired by binary texture comparisons; it does not encode the neighbors as a directional, bit-weighted LBP integer. The response is combined with the bottleneck features and projected back to the backbone dimension:

$$\begin{aligned} \Delta\mathbf{F}^{\mathrm{GPB}} &= \gamma_g \ \mathrm{LN}(U_g(\mathrm{flat}(\mathbf{Z}_g + \mathbf{R}_g))) \,, \\ \mathbf{F}' &= \mathbf{F} + \Delta\mathbf{F}^{\mathrm{GPB}}, \end{aligned} \tag{23}$$

where U_g is an affine up-projection and flat restores the patch-token sequence. The learnable residual scale γ_g is initialized to 10^{-3} in the block wrapper. GPB directly modifies only patch tokens; its residual entries for the class and storage tokens are zero. Subsequent attention layers can still propagate the updated patch information to those tokens. Figure 4 illustrates the comparison responses and their gradients.

A.4 Local Frequency-aware Block: Learnable Spectral Selection

The Local Frequency-aware Block (LFB) augments local-view features with learnable spectral residuals. It uses a smooth high-pass transition so that the cutoff can be optimized jointly with the detection objective.

An independent affine down-projection D_f and LayerNorm produce the bottleneck map $\mathbf{Z}_f = \mathrm{reshape}(\mathrm{LN}(D_f(\mathbf{F})))$, again with $C' = \lfloor C/8 \rfloor$ channels. For a Fourier bin with frequencies

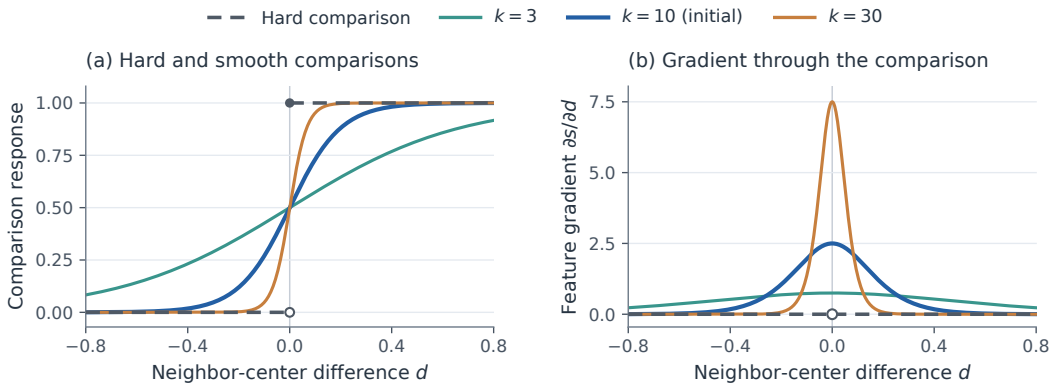


Figure 4: Analytical illustration of the structural comparison in GPB. (a) A hard neighbor–center comparison and its sigmoid relaxations. (b) Derivatives with respect to the feature difference. The initial sharpness is $k = 10$; the other values illustrate how sharpness changes the transition and gradient concentration. The hard comparison has zero derivative for $d \neq 0$ and no ordinary derivative at $d = 0$, indicated by an open marker.

(f_x, f_y), define $\rho = \sqrt{f_x^2 + f_y^2}$. The implementation uses the frequencies returned by `rfftfreq` and `fftfreq`, measured in cycles per patch spacing; ρ is not rescaled to $[0, 1]$. A hard high-pass mask is $M_{\text{hard}}(\rho; c) = \mathbb{1}[\rho \geq c]$. At a fixed Fourier bin, this mask has zero derivative with respect to c away from the threshold. Although filtering with a fixed mask remains differentiable with respect to its input, the hard cutoff does not provide an ordinary gradient for learning its location.

LFB therefore uses a smooth radial mask:

$$M_{\text{hard}}(\rho; c) = \mathbb{1}[\rho \geq c] \quad \longrightarrow \quad M(\rho; c) = \sigma(\alpha(\rho - c)), \qquad c = \tfrac{1}{2}\sigma(\theta), \tag{24}$$

where θ is learnable and $\alpha = 30$ is fixed. The parameterization constrains c to $(0, 0.5)$, with the initial value $c = 0.25$ obtained from $\theta = 0$. The same scalar cutoff and radial mask are shared across channels within each LFB. The derivatives are

$$\frac{\partial M}{\partial c} = -\alpha M(1 - M), \qquad \frac{\partial M}{\partial \theta} = -\frac{\alpha}{2}\,\sigma(\theta)\big(1 - \sigma(\theta)\big)\, M(1 - M). \tag{25}$$

These derivatives provide a gradient path from the detection loss to the cutoff parameter, with the strongest mask sensitivity near $\rho = c$. The smooth transition also retains partial responses near the cutoff. Figure 5 compares hard and smooth masks and illustrates their cutoff sensitivities.

Frequency residual. LFB applies the mask to the real-valued two-dimensional Fourier transform of each bottleneck feature channel:

$$\mathbf{Z}_{\text{hp}} = \mathcal{F}_r^{-1}(M \odot \mathcal{F}_r(\mathbf{Z}_f)), \qquad \Delta\mathbf{F}^{\text{LFB}} = \gamma_f U_f(\text{flat}(\mathbf{Z}_{\text{hp}})). \tag{26}$$

Here, $\mathcal{F}_r$ and $\mathcal{F}_r^{-1}$ correspond to `rfft2` and `irfft2` with orthonormal normalization; the inverse transform uses the original spatial dimensions. The transforms are evaluated in FP32 and the filtered features are cast back to the input dtype before the up-projection. The residual is added to the patch-token output of the wrapped Transformer block, with zero residual entries for the class and storage tokens. The wrapper initializes the learnable scale γ_f to 10^{-3}. This attenuates the initial residual without making it identically zero.

Placement and branch selection. For DINOv3-L/16, the two side modules are attached after blocks 7, 13, and 19 in one-based indexing, corresponding to indices 6, 12, and 18 in the code.

Their bottleneck width is $1024/8 = 128$. In the global/local design, GPB processes the global branch and LFB processes the local branches. The wrapper exposes independent activation flags for these modules; when both are enabled in one call, the frequency module processes the tokens after the structural residual has been added. The smooth operators above remain part of the feature computation; their hard counterparts serve as reference operations for explaining the relaxation.

A.5 Analytical Incremental Router Details

This section supplements the AIR overview in Section 3.4 by specifying its base-expert initialization and recursive expansion. AIR selects among the base expert U_0 and incremental residual experts $\{U_i\}_{i=1}^{t}$ in a fixed routing-feature space.

Given the routing-cut representation $S(x)$ produced by the frozen shared forensic learner, AIR computes $\mathbf{z}(x) = g(S(x))$, where $g(\cdot)$ mean-pools the patch tokens of $S(x)$ and applies a fixed Gaussian projection, ReLU activation, and ℓ_2 normalization. Since both the shared forensic learner and $g(\cdot)$ remain fixed after base training, routing features are comparable across incremental stages.

Base-router Initialization. AIR is initialized using generated images from the base domain, all assigned to the base expert U_0. Let $\mathbf{Z}_0 \in \mathbb{R}^{n_0 \times m}$ contain their routing features, where m is the routing-feature dimension, and let $\mathbf{Y}_0 \in \mathbb{R}^{n_0 \times 1}$ be the corresponding single-column target matrix. We initialize

$$\begin{aligned} \mathbf{G}_0 &= \lambda_{\mathrm{RLS}} \mathbf{I} + \mathbf{Z}_0^\top \mathbf{Z}_0, \\ \mathbf{R}_0 &= \mathbf{G}_0^{-1}, \\ \mathbf{Q}_0 &= \mathbf{Z}_0^\top \mathbf{Y}_0, \\ \mathbf{W}_0 &= \mathbf{R}_0 \mathbf{Q}_0, \end{aligned} \tag{27}$$

where λ_{RLS} is the ridge regularization coefficient. The recursive implementation retains $\mathbf{R}_0$ rather than explicitly recomputing the inverse of $\mathbf{G}_0$ at later stages.

Incremental Router Expansion. At stage t, let $\mathbf{Z}_t \in \mathbb{R}^{n_t \times m}$ contain routing features of the current-stage generated images. Their one-hot expert-selection targets $\mathbf{Y}_t \in \mathbb{R}^{n_t \times (t+1)}$ assign all current samples to the newly added expert U_t, while retaining columns for the base expert and previously added incremental residual experts. These targets supervise expert retrieval rather than exact generator-source attribution. To expand the output space, we append a zero column to the previous feature–target statistic:

$$\widetilde{\mathbf{Q}}_{t-1} = [\mathbf{Q}_{t-1}, \mathbf{0}]. \tag{28}$$

AIR then updates its sufficient statistics and router state as

$$\begin{aligned} \mathbf{G}_t &= \mathbf{G}_{t-1} + \mathbf{Z}_t^\top \mathbf{Z}_t, \\ \mathbf{R}_t &= \mathbf{R}_{t-1} - \mathbf{R}_{t-1} \mathbf{Z}_t^\top \left(\mathbf{I} + \mathbf{Z}_t \mathbf{R}_{t-1} \mathbf{Z}_t^\top \right)^{-1} \mathbf{Z}_t \mathbf{R}_{t-1}, \\ \mathbf{Q}_t &= \widetilde{\mathbf{Q}}_{t-1} + \mathbf{Z}_t^\top \mathbf{Y}_t, \\ \mathbf{W}_t &= \mathbf{R}_t \mathbf{Q}_t, \end{aligned} \tag{29}$$

where $\mathbf{R}_t = \mathbf{G}_t^{-1}$ and $\mathbf{W}_t \in \mathbb{R}^{m \times (t+1)}$. The update of $\mathbf{R}_t$ follows the Woodbury identity and avoids explicitly inverting the $m \times m$ Gram matrix at every stage. With fixed routing features, a fixed ridge coefficient, and the same one-hot routing targets, this recursive update yields the same solution as refitting the AIR ridge-regression objective on all previously observed routing samples. This equivalence applies only to the AIR routing objective; it does not involve joint optimization of expert parameters. The update avoids storing historical routing samples and does not require gradient-based router optimization.

Routing and Detection. AIR is fitted only on generated images because it retrieves the expert whose residual forensic evidence is most compatible with the input. Its objective is expert retrieval for binary detection rather than exact generator-source attribution. Real images are excluded from router fitting but supervise every expert for binary real/fake detection. At inference, AIR selects

$$r_t(x) = \arg \max_{0 \le i \le t} \mathbf{z}(x)^\top \mathbf{w}_{t,i}, \tag{30}$$

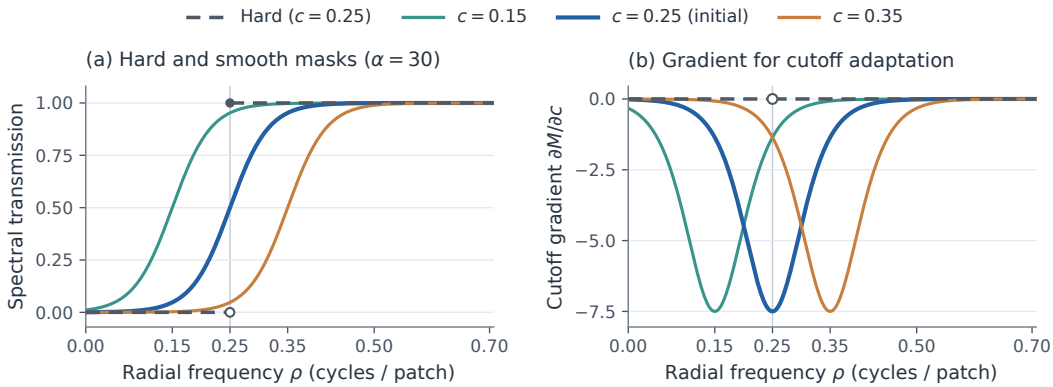


Figure 5: Analytical illustration of spectral selection in LFB. (a) A hard high-pass mask at $c = 0.25$ and smooth masks with $\alpha = 30$ at three cutoff values. (b) Their derivatives with respect to the cutoff. The initial cutoff is $c = 0.25$. The smooth mask provides cutoff gradients near the transition; the hard mask has zero cutoff derivative away from $\rho = c$. Frequency is measured on the patch-feature grid.

where $\mathbf{w}_{t,i}$ is the i-th column of $\mathbf{W}_t$. The selected expert outputs

$$p_t(y = 1 \mid x) = U_{r_t(x)}(S(x)). \tag{31}$$

B Supplementary Experiments

Table 5: Complete generator-wise detection trajectory on GenImage. Generators arrive in the column order. Seen averages accuracy over the generators observed by the current stage; All averages accuracy over all eight generators, including those not yet observed. Bold marks the highest values in the Seen and All columns. All entries are percentages; averages are computed from the displayed generator-wise results.

Stage	SD v1.4	BigGAN	VQDM	SD v1.5	Wukong	ADM	GLIDE	Midjourney	Seen	All
T1	99.69	93.78	95.67	99.69	99.74	78.05	95.52	93.10	**99.69**	94.41
T2	100.00	97.54	97.78	99.93	99.96	83.19	97.59	94.43	98.77	96.30
T3	100.00	97.54	97.77	99.93	99.96	83.19	97.59	94.42	98.44	96.30
T4	100.00	97.54	97.52	99.93	99.96	82.87	97.59	94.31	98.75	96.22
T5	100.00	97.32	97.42	99.93	99.96	81.54	97.03	94.08	98.93	95.91
T6	100.00	97.41	97.54	99.93	99.96	83.96	97.47	94.72	96.47	**96.37**
T7	100.00	97.18	97.46	99.93	99.96	83.83	97.34	94.64	96.53	96.29
T8	100.00	97.18	97.46	99.93	99.96	83.85	97.35	94.78	96.31	96.31

This section provides supplementary experiments supporting EvoKnow's continual knowledge-evolution mechanism. We first analyze complete continual trajectories and historical knowledge retention, then study the effect of the adaptation budget on cross-generator transfer. Finally, we examine the persistent state, storage scaling, and auxiliary attribution behavior of the analytical router.

B.1 Continual Learning Trajectories and Knowledge Retention

We analyze how EvoKnow evolves as incremental residual experts are added sequentially. The GenImage sequence starts from the Phase-I base domain, SD v1.4, followed by BigGAN, VQDM, SD v1.5, Wukong, ADM, GLIDE, and Midjourney. The shared forensic learner is initialized on SD v1.4 and remains frozen throughout subsequent adaptation.

After each stage, we evaluate the detector on all eight generator domains, including generators not yet observed at that stage. Let $B_{t,j}$ denote detection accuracy on generator j after stage t. We dis-

Table 6: Stage-wise continual detection performance on GenImage. Each generator denotes its arrival stage. AA averages accuracy over the generators observed so far; AF measures the average signed performance drop relative to the accuracy obtained immediately after each generator is learned. AA is reported in percentages and AF in percentage points. AF is undefined at the first stage. Final-stage results are bolded.

Method	SD v1.4		BigGAN		VQDM		SD v1.5		Wukong		ADM		GLIDE		Midjourney	
	AA ↑	AF ↓	AA ↑	AF ↓	AA ↑	AF ↓	AA ↑	AF ↓	AA ↑	AF ↓	AA ↑	AF ↓	AA ↑	AF ↓	AA ↑	AF ↓
EvoKnow	99.69	–	98.77	−0.31	98.44	0.00	98.75	0.08	98.93	0.14	96.47	0.07	96.53	0.13	**96.31**	**0.11**

Table 7: Generator-wise retention at the final stage on GenImage. The post-acquisition accuracy is measured immediately after the corresponding generator is learned, and the final accuracy is measured at T8. The signed drop is their difference in percentage points, with a negative value indicating improvement. Midjourney, introduced at T8, is excluded from AF.

Generator	Post-acquisition accuracy	Final accuracy	Drop
SD v1.4	100.00	100.00	0.00
BigGAN	97.54	97.18	0.36
VQDM	97.77	97.46	0.31
SD v1.5	99.93	99.93	0.00
Wukong	99.96	99.96	0.00
ADM	83.96	83.85	0.11
GLIDE	97.34	97.35	−0.01
AF ↓	–	–	**0.11**

Table 8: Persistent state of the analytical router and incremental residual knowledge units. m is the routing-feature dimension, d is the input dimension of the fixed projection, M is the number of residual knowledge units, and p is the number of additional parameters per unit. The frozen shared backbone is excluded.

State	Shape / count	Storage order
$\mathbf{R}$ (Gram inverse)	$m \times m$	$O(m^2)$
$\mathbf{Q}$ (feature–target statistic)	$m \times M$	$O(mM)$
$\mathbf{W} = \mathbf{RQ}$ (router weights)	$m \times M$	$O(mM)$
$\mathbf{P}$ (fixed projection)	$m \times d$	$O(md)$
Incremental residual expert parameters	M units	$O(Mp)$
Unit metadata	M entries	$O(M)$

tinguish accuracy on observed domains and all evaluation domains:

$$\mathrm{Seen}_t = \frac{1}{t}\sum_{j=1}^{t} B_{t,j}, \qquad \mathrm{All}_t = \frac{1}{8}\sum_{j=1}^{8} B_{t,j}. \tag{32}$$

Here, Seen_t measures performance on acquired domains, whereas All_t additionally measures transfer to generators not yet observed. At the final stage, the two quantities coincide.

Table 5 reports the complete generator-wise trajectory. The all-domain average improves overall from 94.41% at the first stage to 96.31% after the final stage, despite moderate fluctuations as domains with different difficulty levels are introduced. A decrease in Seen_t does not necessarily indicate forgetting because a new and potentially more difficult generator is added to the average at every stage. For example, when ADM arrives at Stage 6, Seen_t decreases while All_t increases and AF remains low.

Table 6 and 7 summarize stage-wise and generator-wise knowledge retention. The low final forgetting indicates that residual experts are added without directly overwriting shared knowledge or historical experts. Since these components remain frozen, changes in historical-domain performance can arise through the evolving router, which may retrieve a different expert after knowledge expansion.

B.2 Few-shot Adaptation and Cross-Generator Transfer

Starting from the same GenImage-trained shared basis, we vary the number of generated images available for each incoming generator over $\{10, 50, 100, 300\}$ while fixing the knowledge-unit organization and arrival sequence. The resulting detectors are evaluated on UniversalFakeDetect. Thus, the shot budget refers to the GenImage adaptation stream, whereas UniversalFakeDetect is used only for evaluation.

Table 9: UniversalFakeDetect detection accuracy (%) under zero-shot and few-shot knowledge expansion. Existing-detector results follow their reported ProGAN-based training protocols. For EvoKnow, the adaptation budget denotes the number of generated images per arriving generator in the GenImage stream; UniversalFakeDetect is used only for evaluation. The knowledge-unit organization and arrival sequence are fixed across adaptation budgets. The final column reports average accuracy and population standard deviation across the 14 displayed generators; the standard deviation measures cross-generator variation rather than variation across repeated runs.

Method	Adaptation budget	GAN						Guided	LDM		GLIDE				DALL-E	Ave. ± Std.
		ProG.	Cycle	BigG.	Style	GauG.	StarG.		200	200-cfg	100	100/27	50/27	100/10		
CNNSpot	CVPR'20	99.99	85.20	70.20	85.70	78.95	91.70	65.66	60.07	54.03	54.96	54.14	60.78	63.80	55.58	70.05±14.90
PatchFor.	ECCV'20	75.03	68.97	68.47	79.16	64.23	63.94	68.52	67.41	76.50	76.10	75.77	74.81	73.28	67.91	71.44±4.73
Co-occur.	EI'20	97.70	63.15	53.75	92.50	51.10	54.70	69.90	60.50	70.70	70.55	71.00	70.25	69.60	67.55	68.78±12.69
Freq-spec	WIFS'19	49.90	99.90	50.50	49.90	50.30	99.70	50.40	50.90	50.40	50.40	50.30	51.70	51.40	50.00	57.55±17.26
F3Net	ECCV'20	99.38	76.38	65.33	92.56	58.10	100.00	83.05	69.20	68.15	75.35	68.80	81.65	83.25	66.30	77.68±12.48
UnivFD	CVPR'23	100.00	98.50	94.50	82.00	99.50	97.00	78.14	70.03	94.19	73.76	94.36	79.07	79.85	86.78	87.69±9.98
LGrad	CVPR'23	99.84	85.39	82.88	94.83	72.45	99.62	89.55	77.50	94.20	95.85	94.80	87.40	90.70	88.35	89.53±7.70
FreqNet	AAAI'24	97.90	95.84	90.45	97.55	90.24	93.41	88.15	86.70	84.55	99.58	65.56	85.69	97.40	59.06	88.01±11.57
NPR	CVPR'24	99.84	95.00	87.55	96.23	86.57	99.75	97.35	84.55	97.65	98.00	98.20	96.25	97.15	87.15	94.37±5.19
FatFormer	CVPR'24	99.89	99.32	99.50	97.15	99.41	99.75	94.20	76.00	98.60	94.90	98.65	94.35	94.65	98.75	96.08±5.95
C2P-CLIP	AAAI'25	99.98	97.31	99.12	96.44	99.17	99.60	96.10	69.10	99.25	97.25	99.30	95.25	95.25	98.55	95.83±7.58
EvoKnow	0-shot	93.75	90.46	98.03	93.73	97.99	93.83	98.50	83.60	99.27	98.50	99.30	97.97	97.97	98.53	95.82±4.27
	10-shot	96.50	94.80	99.00	96.20	98.80	98.00	99.10	91.50	99.55	99.10	99.60	98.80	98.80	99.00	97.78±2.22
	50-shot	97.80	95.37	99.10	97.35	98.53	99.75	98.55	93.32	99.05	98.00	99.30	98.00	98.75	99.20	98.01±1.67
	100-shot	98.37	97.80	99.32	97.75	99.32	99.17	99.65	94.06	99.40	99.54	99.63	99.71	99.82	99.65	98.79±1.47
	300-shot	98.54	98.50	99.50	97.55	98.75	99.55	99.85	94.47	99.58	99.27	99.54	99.17	99.20	99.84	98.81±1.35

Table 10: Effect of incremental residual knowledge organization on GenImage (%). Generator-based organization assigns one incremental residual expert per generator, whereas category-based organization groups generators into GAN, diffusion, and unknown-source categories. Both settings use ten generated images per arriving generator. Unlike Table 1, *Ave.* here includes all eight GenImage domains, including the SD v1.4 base domain.

Organization	MidJ.	SD1.4	SD1.5	ADM	GLIDE	Wuk.	VQDM	BigGAN	Ave.
Generator-based	96.06	98.18	98.53	96.21	96.27	**98.64**	97.41	93.77	96.88
Category-based	**97.39**	**99.32**	**99.62**	**97.30**	**98.32**	98.61	**99.21**	**99.64**	**98.68**
Δ	+1.33	+1.14	+1.09	+1.09	+2.05	-0.03	+1.80	+5.87	+1.79

B.3 Incremental Residual Knowledge Organization

Table 9 reports complete generator-wise results. Average accuracy increases from 95.82% without adaptation to 97.78% with ten-shot adaptation and 98.81% with 300-shot adaptation. Meanwhile, across-generator variation decreases substantially, from 4.27 to 1.35 percentage points. The largest gains occur on initially challenging domains, indicating that additional residual knowledge primarily compensates for generator-relevant gaps in the shared forensic basis.

The default EvoKnow setting assigns one incremental residual expert to each arriving generator domain. As an additional analysis, we compare this generator-based organization with a category-based organization that groups generators into GAN, diffusion, and unknown-source categories. Both settings use ten generated images per arriving generator.

Table 10 shows that category-based organization improves seven of eight GenImage domains and raises the all-domain average by 1.79 percentage points. This result suggests that incrementally acquired residual knowledge is not entirely generator-exclusive: related generators can benefit from partially shared complementary forensic evidence.

B.4 Analytical Router State, Resource Scaling, and Attribution Diagnostics

This section analyzes the persistent state required for replay-free analytical routing, its storage scaling, and its auxiliary source-attribution behavior. EvoKnow is designed for binary real/fake detection and residual knowledge retrieval rather than exact source attribution; attribution is therefore reported only as a diagnostic.

Table 11: Auxiliary source-attribution diagnostic (rank-1 accuracy, %). The evaluation includes one real-image class and eight generator classes. The real label is determined by the binary detection head, whereas generator labels are supplied by the analytical router. Avg. is the unweighted mean over all nine classes. Big, Mid, Wuk, GLI, and VQ abbreviate BigGAN, Midjourney, Wukong, GLIDE, and VQDM, respectively.

Method	Real	Big	Mid	Wuk	SD14	SD15	ADM	GLI	VQ	Avg.
ResNet	16.1	10.6	56.3	27.6	12.1	22.6	17.1	20.1	10.1	21.4
DIRE	22.6	11.6	0.0	26.6	22.1	20.6	18.1	22.6	10.6	17.2
ESSP	17.6	13.6	49.7	27.1	19.6	17.6	16.6	27.1	13.1	22.4
LIDA	83.4	98.5	69.3	13.1	23.6	50.3	47.2	55.3	45.7	54.0
EvoKnow (router)	98.5	93.1	29.5	35.7	2.1	65.0	63.5	68.0	38.1	**54.8**

Persistent State and Storage Scaling. In this work, "replay-free" specifically means that historical AI-generated images, image features, and per-sample routing features are not retained or replayed during incremental adaptation. This does not imply zero persistent state. To continue routing updates without storing historical generated images, the router retains the regularized Gram inverse $\mathbf{R}$, the feature–target statistic $\mathbf{Q}$, and router weights $\mathbf{W} = \mathbf{RQ}$. Given

$$\mathbf{G} = \lambda_{\mathrm{RLS}}\mathbf{I} + \mathbf{X}^\top\mathbf{X}, \qquad \mathbf{R} = \mathbf{G}^{-1}, \tag{33}$$

the recursive implementation maintains $\mathbf{R}$ directly. The checkpoint additionally contains the fixed projection $\mathbf{P}$, the mapping between router outputs and residual units, and update metadata.

For routing dimension m and M incremental residual experts, $\mathbf{R}$ requires $O(m^2)$ values, whereas $\mathbf{Q}$ and $\mathbf{W}$ each require $O(mM)$. The projection requires $O(md)$ values, and expert parameters grow as $O(Mp)$. At $m = 10{,}000$, $\mathbf{R}$ occupies approximately 400 MB in FP32. Table 8 summarizes the persistent router state and its storage scaling. Thus, replay-free learning in EvoKnow avoids retaining or replaying historical generated images and their instance-level features, while retaining aggregated router statistics, learned router parameters, and an expanding set of incremental residual experts. It therefore avoids an image replay buffer but does not imply zero persistent state.

Auxiliary Source-attribution Diagnostics. The following attribution diagnostic is conducted under generator-based residual-unit organization, where each incremental residual expert has a one-to-one association with an observed generator domain. AIR selects residual knowledge units rather than explicitly predicting exact source identities. Since multiple generators can share related forensic differences, source attribution is treated only as an auxiliary diagnostic. In the generator-based setting, router outputs can be mapped to source labels as

$$\hat{c}(x) = \begin{cases} \mathrm{Real}, & \hat{y}(x) = 0, \\ g(r(x)), & \hat{y}(x) = 1, \end{cases} \tag{34}$$

where $r(x)$ is the selected expert index, $\hat{y}(x)$ is the binary prediction, and $g(\cdot)$ maps an expert to its associated generator.

Table 11 reports auxiliary nine-class attribution results. Its nonuniform generator-wise behavior is consistent with the router's intended role: it retrieves residual evidence useful for detection rather than enforcing a one-to-one mapping between generators and experts.