FORENEVOLVE: HYPOTHESIS-DRIVEN PROGRAM EVOLUTION FOR AI-GENERATED IMAGE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Developing an AI-generated image detector involves coupled choices: which forensic cues to extract, how to preserve them under image processing, and how to adapt pretrained features without overfitting to known generators. Because these choices interact, a revision's value becomes clear only after testing across image sources and processing conditions, so automation requires feedback that shows where a revision helps or fails and a record that carries this evidence forward. We introduce **ForenEvolve**, an agentic framework for hypothesis-driven evolution of detector programs. Its search space spans three nested capability tiers: frozen forensic feature views and readouts, training-image transformations, and low-rank backbone adaptation with programmable losses. Before each experiment, a language-model agent records a hypothesis and predicted outcome; the revision is then measured across image sources and processed copies of the same images, with controlled revisions compared against their parents. Persistent experimental lineages link each outcome to its prediction, so later revisions build on accumulated evidence. The discovered detector reaches 94.34% mean balanced accuracy across 12 public benchmarks, exceeding the strongest published detector by 9.12 percentage points; it ranks first on nine, including all four excluded from search, where it reduces mean balanced error by 55% relative to the best published result. With the same backbone and training data, it exceeds standard adaptation by 4.14 points overall and outperforms three transplanted state-of-the-art recipes on every benchmark.

1 INTRODUCTION

AI-generated image (AIGI) detectors distinguish synthetic images from real photographs and must generalize across generators and image-processing conditions. They support content moderation and forensic analysis, where images may come from unfamiliar generation models and undergo compression or resizing before inspection. Existing methods achieve strong benchmark performance using pretrained visual features and low-level image statistics (Ojha et al., 2023; Tan et al., 2024b;a), yet evaluations on images collected in the wild reveal substantial generalization gaps (Li et al., 2025b; Yan et al., 2025a). The development challenge is to identify and learn forensic cues that remain informative as generation techniques and image processing change.

Developing an AIGI detector requires several coupled forensic judgments. Researchers choose between frequency statistics, neighboring-pixel relationships, reconstruction residuals, and pretrained visual features (Tan et al., 2024a;b; Wang et al., 2023; Ojha et al., 2023). They align real and synthetic training images to reduce content and compression shortcuts (Sundara Rajan et al., 2025; Chen et al., 2025), and constrain backbone adaptation to preserve transferable pretrained representations (Yan et al., 2026). The value of each choice depends on the others: training transformations alter the traces available to the backbone, while adaptation determines which information the readout can exploit. Compression and resizing further change which of these traces remain useful at evaluation. These interactions make the value of a revision difficult to predict before it is tested across image sources and processing conditions.

Automating this cycle of revision and testing requires feedback that reveals where a revision helps or fails. Program-search agents evolve code against executable evaluators and select candidates by their scores (Novikov et al., 2025; Hu et al., 2025). In detector development, an aggregate score can hide

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.

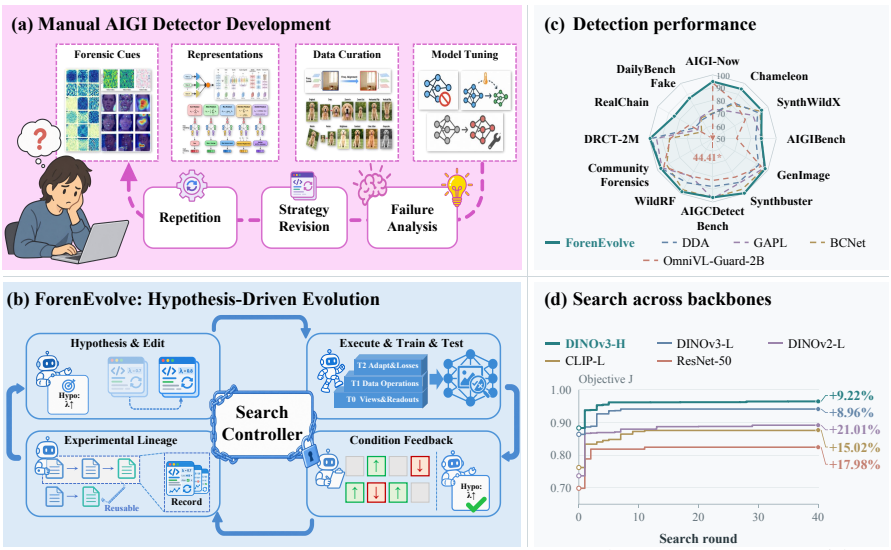


Figure 1: **ForenEvolve automates detector development.** (a–b) Manual versus evidence-guided development. (c) Pooled bACC on the 12 benchmarks of Table 1 (*: below the 50–100 range). (d) Best-so-far objective for five backbones; labels give relative gains over each initial objective.

opposing changes across image sources and processing conditions, and a score recorded without a stated expectation does not indicate which hypothesis the outcome supports. AIGI detection makes richer feedback inexpensive: benchmark images have known provenance, and processing can be applied to identical images. Holding the rest of the program fixed, an edit can therefore be measured across image sources and across processed versions of the same images.

We introduce ForenEvolve, an agentic framework for hypothesis-driven evolution of AIGI detector programs (Figure 1). The search spans three nested capability tiers: combinations and readouts of frozen forensic features (T0), training-image transformations (T1), and low-rank backbone adaptation with programmable losses (T2). Before evaluation, a language-model agent records the hypothesis and predicted outcome of its revision in a research card. Once measured, the outcome is archived with the program and its prediction, and experimental lineages carry the research question across revisions, allowing the agent to pursue a design beyond a single trial. The central contribution is to change the unit of program search from a scored candidate to a stored test of a stated prediction, with outcomes broken down by image source and processing condition. The selected detector runs with fixed parameters and no language model at inference.

The detector discovered at round 35 reaches 94.34% mean balanced accuracy across 12 public benchmarks, exceeding the strongest published detector by 9.12 percentage points and ranking first on nine. On the four benchmarks excluded from search, it ranks first on each with a mean of 91.95%, reducing mean balanced error by 55% relative to the best published result. With the same DINOv3-H backbone and training data, it exceeds standard adaptation by 4.14 points overall and 4.50 points on the excluded benchmarks, and it outperforms three state-of-the-art recipes transplanted onto this setting on all 12 benchmarks. Search also improves the objective with five backbones, three language-model proposers, and three training sets. Compared with six framework ablations and two public program-evolution frameworks under the same 240-proposal budget, the full system continues to improve after generic archive search plateaus, and its largest advantages occur on the excluded benchmarks. Under a fixed budget, the organization of experiments thus shapes how far gains extend beyond the feedback used during search.

2 Related Work

AI-generated image detection. Detector development has advanced along three design axes that ForenEvolve's capability tiers expose. Forensic representations range from frozen CLIP features (Ojha et al., 2023) to neighboring-pixel relationships (Tan et al., 2024b), diffusion reconstruction errors (Wang et al., 2023), and discrete distribution discrepancies of autoregressive-generated images (Zhang et al., 2025). Training data are augmented or transformed (Wang et al.,

2020; Li et al., 2025a) or aligned between real and synthetic images to remove content-, pixel-, and frequency-level shortcuts (Sundara Rajan et al., 2025; Chen et al., 2025). Adaptation and training objectives include orthogonal-subspace adaptation (Yan et al., 2025b), distillation with gradient surgery (Yan et al., 2026), generator-aware prototypes (Qin et al., 2026), and reduced semantic dependence (Yang et al., 2026); few-shot methods adapt to new generators from labeled examples (Wang et al., 2026a; Zhang et al., 2026c). Multimodal detectors add artifact explanation, forgery grounding, or joint generation–detection training (Wen et al., 2025; Shen et al., 2026; Zhang et al., 2026d). Each method commits to one designed combination of these choices. ForenEvolve searches over them jointly, revising reusable programs that specify how detectors are trained and applied; ForenAgent instead generates tools to inspect individual images (Zhang et al., 2026a).

Program search with language models. Executable programs give language-model agents a search space with automatic evaluation. AlphaEvolve and its open-source counterpart OpenEvolve evolve algorithm implementations against executable evaluators (Novikov et al., 2025; Sharma, 2025), ShinkaEvolve improves the sample efficiency of such evolution (Lange et al., 2026), and ADAS searches agent designs from an expanding archive of programs and evaluations (Hu et al., 2025). The Darwin Gödel Machine applies archive-based search to its own coding-agent implementation (Zhang et al., 2026b), and the Huxley–Gödel Machine guides this expansion by the performance of a lineage's descendants (Wang et al., 2026b). ForenEvolve directs program search at detector pipelines, where each evaluation trains a detector. With a few hundred affordable trials, each experiment must yield evidence beyond its score; ForenEvolve therefore records a prediction for every revision and reports its outcome by image source and processing condition. Section 4.3 compares ForenEvolve with OpenEvolve and ShinkaEvolve in the same setting and with a generic archive search control that removes this protocol.

Domain-specific experimental agents. Experimental agents use task-specific feedback to guide successive designs. PathWise retains derivation rationales, parent relationships, and performance in an entailment graph, with critics guiding subsequent proposals (Gungordu et al., 2026). R&D-Agent integrates hypothesis formation, experimental memory, code execution, and evaluation in an iterative ML workflow (Yang et al., 2025). ML-Agent trains the proposing policy through reinforcement learning from experimental interactions (Liu et al., 2026), and AutoMS steers microstructure search with physics-simulation feedback (Zhao et al., 2026). ForenEvolve specializes this loop for AIGI detection. Each revision is registered with a predicted outcome before measurement, controlled revisions change one declared component of a measured parent, and feedback reports performance across image sources and processing conditions. Persistent experimental lineages connect these comparisons to a continuing research question, so later proposals build on related outcomes.

3 METHOD

Given training data, a pretrained backbone, and a development evaluation protocol, ForenEvolve returns a detector program and an archive of experiments. A language-model agent revises detector programs and registers each revision as a test with a predicted outcome (Figure 2). Section 3.1 defines the programs and their capability tiers, Section 3.2 describes how revisions are proposed and controlled, and Section 3.3 defines the condition-level feedback and the experimental lineages that carry it forward.

3.1 PROGRAMMABLE FORENSIC DESIGN

A detector program makes forensic design choices executable: feature views determine which image statistics are available, training transformations alter the traces presented to the backbone, and adaptation losses shape the learned representation. The program also specifies training-example selection or weighting and the final readout. We denote the candidate of experiment t by p_t; training it yields a scoring function $f_t : \mathcal{X} \to \mathbb{R}$, whose larger values indicate generated content.

Three nested capability spaces, $\mathcal{P}_0 \subseteq \mathcal{P}_1 \subseteq \mathcal{P}_2$, expose these choices, and tier Tk searches $\mathcal{P}_k$. T0 edits frozen-feature views, their combination, the readout and its training loss, and score transformations; T1 adds training-image operations and synthesis plans; T2 adds backbone adaptation (LoRA (Hu et al., 2022) for Transformers) with a programmable loss, after which features are re-extracted and the readout is refitted. Nesting orders these choices by cost: a T0 revision refits a

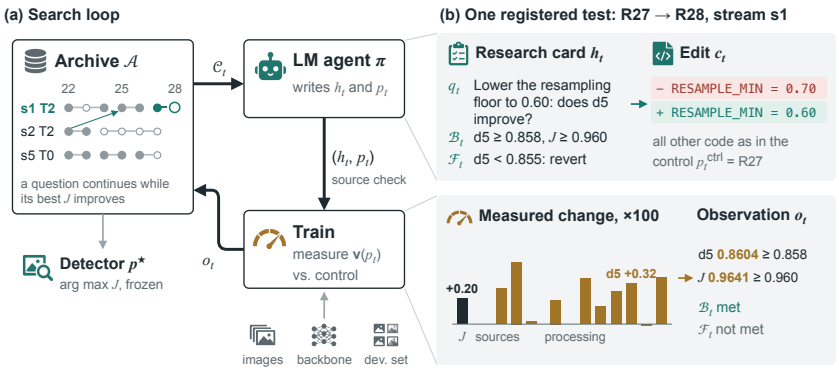


Figure 2: **Each experiment in ForenEvolve is a registered test.** (a) The search loop: the agent π reads context $\mathcal{C}_t$ from the archive $\mathcal{A}$ and writes a research card h_t and a program p_t, whose measured observation o_t returns to $\mathcal{A}$. (b) The registered test R27→R28: bounds recorded before training are checked against the measured changes ($\times 100$) of a one-attribute edit.

readout on frozen features, a T1 revision rebuilds training features, and a T2 revision retrains the backbone before refitting. Each proposal stream s searches a fixed tier $\mathcal{P}_{k_s}$, so inexpensive experiments continue alongside costly ones, and higher tiers can reuse components found by lower-tier streams (Appendix A.1).

3.2 Hypothesis-Driven Program Revision

Each revision is registered as a test before it is measured. The agent writes a research card $h_t = (q_t, c_t, \mathcal{B}_t, \mathcal{F}_t)$ stating the question and hypothesis q_t, the edited component c_t, the predicted outcome $\mathcal{B}_t$, and a falsification criterion $\mathcal{F}_t$; both $\mathcal{B}_t$ and $\mathcal{F}_t$ are bounds on the scores defined in Section 3.3 (example in Appendix A.2). Because the card is prospective, a measured outcome can contradict it instead of being explained after the fact.

Search runs concurrent proposal streams that share an archive $\mathcal{A}$ of completed experiments. The controller sets the experiment mode $m_t \in \{\text{ctrl}, \text{open}\}$ and, for a continuing question, the measured parent; the agent chooses the design to test within that mode:

$$(p_t, h_t) \sim \pi\big(\cdot \mid \mathcal{C}_t\big), \qquad p_t \in \mathcal{P}_{k_s}. \tag{1}$$

Here π is the language-model agent with fixed weights, and the context $\mathcal{C}_t$ retrieved from $\mathcal{A}$ contains the stream's question and preceding experiment, recent results, and the strongest observed programs across streams and within the tier. The preceding experiment supplies a prediction to revisit, while other programs supply components for new designs; the agent can also inspect their code and consult literature or source implementations before writing h_t.

A controlled revision ($m_t = \text{ctrl}$) attributes a measured change to one component. Its control p_t^{ctrl} is the measured parent in the same tier, and the candidate differs only in c_t, one method such as `train_image_op` or one class attribute:

$$p_t \ominus c_t = p_t^{\text{ctrl}} \ominus c_t, \qquad p_t[c_t] \neq p_t^{\text{ctrl}}[c_t], \tag{2}$$

where $p \ominus c$ removes c from the program's abstract syntax tree and $p[c]$ is its implementation of c. A source check enforces Equation 2 before execution, and the candidate is evaluated on the same images and processing conditions as its control. Because forensic design choices interact, this comparison measures the edit within the parent's configuration: changing the resampling floor while keeping the adaptation loss and readout fixed tests that transformation for this detector. Open exploration ($m_t = \text{open}$) changes several components and composes components from measured programs; the highest-objective detector combines components from two streams (Section 4.5). Measured exploratory programs can in turn serve as controls for later tests (Appendix A.3).

3.3 Forensic Feedback and Experimental Lineages

Evaluation measures both performance across image sources and sensitivity to image processing. We use G development benchmarks for the former and D processing conditions applied to a shared

Algorithm 1 ForenEvolve. Streams run concurrently; experiments are indexed in completion order.

Require: initial program p_0, training data, pretrained backbone, agent π, budget of T experiments
$\mathcal{A} \leftarrow \{(p_0, \mathbf{v}(p_0), J(p_0))\}$
for $t = 1, \dots, T$ **do**
  $(s, m_t, p_t^{\text{ctrl}}) \leftarrow \textsc{Controller}(\mathcal{A})$; $\mathcal{C}_t \leftarrow \textsc{Retrieve}(\mathcal{A}, s)$
  $(p_t, h_t) \sim \pi(\cdot \mid \mathcal{C}_t)$ with $p_t \in \mathcal{P}_{k_s}$ ▷ Eq. 1
  if $m_t = \text{ctrl}$ **then** check Eq. 2
  Train p_t; measure $\mathbf{v}(p_t)$ and $J(p_t)$ ▷ Eqs. 3–4
  $o_t \leftarrow (p_t^{\text{ctrl}}, c_t, \mathcal{B}_t, \mathcal{F}_t, \Delta J_t, \Delta \mathbf{v}_t)$ if $m_t = \text{ctrl}$, else $\varnothing$ ▷ Eq. 5
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{(p_t, h_t, \mathbf{v}(p_t), J(p_t), o_t)\}$ ▷ diagnostics on failure
end for
return $p^\star = \arg\max_{p \in \mathcal{A}} J(p)$ and $\mathcal{A}$

image pool for the latter; condition $d = 0$ is the unprocessed reference, and the others include recompression, resizing, and their compositions. A trained program p receives the condition vector

$$\mathbf{v}(p) = \left(\mathbf{v}^{\text{source}}(p), \mathbf{v}^{\text{processing}}(p)\right) \in [0,1]^{G+D}, \tag{3}$$

whose coordinates average AUROC and bACC over the eligible images of one benchmark or processing condition, pooling generators within it. AUROC measures ranking quality, and bACC measures the decision made at the threshold selected on training scores; a deployed detector needs both. The search objective is

$$J(p) = \frac{1}{G+1}\left[\sum_{g=1}^{G} v_g^{\text{source}}(p) + \frac{1}{D}\sum_{d=0}^{D-1} v_d^{\text{processing}}(p)\right]. \tag{4}$$

Averaging the processing scores first keeps the shared pool's total weight independent of the number of conditions.

The condition vector exposes changes that J can hide. For a controlled revision, we record

$$\Delta J_t = J(p_t) - J(p_t^{\text{ctrl}}), \qquad \Delta \mathbf{v}_t = \mathbf{v}(p_t) - \mathbf{v}(p_t^{\text{ctrl}}). \tag{5}$$

ΔJ_t summarizes the overall change; the source coordinates of $\Delta \mathbf{v}_t$ show where performance rises or falls across benchmarks, and the processing coordinates show how the same edit behaves under different treatments of identical images. A small ΔJ_t can therefore still reveal a tradeoff between sources and processing conditions (Section 4.5). Feedback uses stable anonymous identifiers: the agent sees condition-level changes and broad operation categories, without benchmark identities or exact processing mappings, so each revision is grounded in measured behavior on the development suite (Section 4.1).

Each completed experiment appends $(p_t, h_t, \mathbf{v}(p_t), J(p_t), o_t)$ to $\mathcal{A}$, with failure diagnostics in place of scores when execution fails. The observation o_t of a controlled revision links the control, declared edit, and prediction to ΔJ_t and $\Delta \mathbf{v}_t$, so a later proposal can retrieve what changed, what was predicted, and what was measured.

Because design choices interact, a question often needs several revisions before its value is clear: in Figure 4, successive edits to the resampling floor precede the gain that the final detector retains. An experimental lineage therefore keeps a question q open while its best objective $J_q^\star$ improves at a two-experiment checkpoint; otherwise, the controller opens a new question from the accumulated archive. Within a lineage, the agent compares o_t with $\mathcal{B}_t$ to choose the next test: a gain confined to unprocessed images, for example, motivates a follow-up on processing sensitivity. An implementation failure leads to a repair attempt, and open exploration draws on all of $\mathcal{A}$, so completed experiments also inform questions outside their original lineage. After search, $p^\star$ and its backbone state are frozen; the readout and threshold are refitted on training data and fixed for final evaluation (Appendix B.2).

4 Experiments

4.1 Experimental Setup

Search configuration. The main search uses DINOv3-H+/16 (Siméoni et al., 2025), hereafter DINOv3-H, with Qwen3.8-27B and 20,000 class-balanced OpenFake (Livernoche et al., 2025) train-

Table 1: **Main detection results by feedback level.** Pooled bACC (%, ↑). Means equally weight the four benchmarks in each level or all 12 overall, and require all constituent scores. **Bold** and underline mark the best and second-best scores among published detectors and ForenEvolve, including ties.

	L0 Anonymous per-benchmark feedback					L1 Aggregate-metric feedback					L2 Search-excluded					All
Method	*AIGI-Now*	*Chameleon*	*SynthWildX*	*AIGIBench*	**Mean L0**	*GenImage*	*Synthbuster*	*AIGCDetect Bench*	*WildRF*	**Mean L1**	*Community Forensics*	*DRCT-2M*	*RealChain*	*DailyBench-Fake*	**Mean L2**	**Overall Mean**
Dedicated image detectors																
NPR (Tan et al., 2024b)	55.50	59.90	49.80	65.70	57.73	51.50	50.00	53.10	63.50	54.53	89.92	37.30	46.70	61.50	58.86	57.04
UnivFD (Ojha et al., 2023)	52.10	50.70	52.30	62.60	54.43	64.10	67.80	72.50	55.30	64.93	64.59	61.80	43.90	49.90	55.05	58.13
FatFormer (Liu et al., 2024)	51.40	51.20	52.10	69.00	55.93	62.80	56.10	85.00	58.90	65.70	56.74	52.20	48.90	50.70	52.14	57.92
AIDE (Yan et al., 2025a)	66.56	63.10	48.80	73.61	63.02	61.20	53.90	63.60	58.40	59.28	83.13	64.60	56.00	55.60	64.83	62.38
DRCT (Chen et al., 2024)	64.40	56.60	55.10	74.80	62.73	84.70	84.80	81.40	50.60	75.38	79.98	90.50	58.60	55.80	71.22	69.77
DDA (Chen et al., 2025)	69.50	82.40	90.90	83.80	81.65	91.70	90.10	87.80	90.30	89.98	96.91	98.10	66.50	66.80	82.08	84.57
GAPL (Qin et al., 2026)	73.00	74.70	86.40	88.50	80.65	96.60	90.90	**97.40**	84.30	92.30	96.09	94.20	63.60	69.10	80.75	84.57
DGS-Net (Yan et al., 2026)	62.50	52.70	50.00	82.30	61.88	98.10	45.10	96.60	60.60	75.10	89.03	19.50	50.00	56.90	53.86	63.61
BCNet (Yang et al., 2026)	69.90	80.50	92.90	**88.60**	82.98	96.90	96.20	96.60	96.70	96.60	91.77	83.90	60.10	68.60	76.09	85.22
Vision-language detectors																
OmniVL-Guard-2B (Shen et al., 2026)	93.71	83.60	79.58	68.84	81.43	93.96	85.14	83.05	84.29	86.61	94.37	83.33	71.15	44.41	73.32	80.45
UniGenDet (Zhang et al., 2026d)	94.18	60.63	66.54	64.20	71.39	96.57	96.54	84.12	61.30	84.63	92.64	84.54	67.84	48.75	73.44	76.49
FakeVLM (Wen et al., 2025)	91.54	70.30	70.27	77.98	77.52	**99.38**	60.86	92.76	68.86	80.47	93.06	87.61	71.90	58.94	77.88	78.62
Same backbone and training data (DINOv3-H, OpenFake 20K)																
Frozen-feature reference	77.51	82.82	87.78	80.57	82.17	88.33	97.78	84.18	92.85	90.79	82.76	89.28	76.44	70.67	79.79	84.25
Standard BCE adaptation	87.28	84.74	90.19	86.43	87.16	95.92	98.82	94.16	95.05	95.99	95.21	98.78	78.83	76.95	87.44	90.20
SAFE† (Li et al., 2025a)	89.61	84.97	87.74	85.37	86.92	94.84	98.48	92.58	94.09	95.00	91.58	97.77	84.19	78.79	88.08	90.00
Effort† (Yan et al., 2025b)	89.60	92.31	91.35	87.91	90.29	95.35	98.73	92.66	96.16	95.73	94.19	98.93	83.66	79.73	89.13	91.72
DDA-style data† (Chen et al., 2025)	79.59	82.64	88.59	85.27	84.02	94.52	96.80	90.38	91.57	93.32	93.90	98.62	75.56	79.60	86.92	88.09
ForenEvolve	**94.73**	**94.87**	**94.37**	88.21	**93.05**	97.57	**99.72**	96.42	**98.40**	**98.03**	**97.09**	**99.26**	**84.91**	**86.52**	**91.95**	**94.34**

The last group uses our runs on DINOv3-H and the 20K OpenFake split, with a linear readout, a training-selected threshold, and the same evaluation; adapted rows train for two epochs. †Core recipe transplanted onto this setting (Appendix D.2). Only ForenEvolve from this group enters the bold and underline ranking.

ing images. It starts from a linear readout of frozen raw-image features. Six proposal streams, one T0, two T1, and three T2, run for 40 rounds, giving 240 proposal opportunities. Each candidate has a 3,600-second execution limit, and the main search charges 105 worker-hours and 6.6M output tokens (Appendix B.4). Training data and development subsets remain fixed throughout search.

Evaluation protocol. The main comparison covers 12 benchmarks, grouped by their role in search (Table 1). **L0** provides four individual scores under anonymous identifiers; **L1** enters only aggregate feedback; **L2** contributes neither feedback nor detector selection. The agent also receives eight anonymous processing scores from differently treated copies of one image pool. It receives scores only, never benchmark images or features. Because the L0 and L1 search subsets are part of their full-coverage evaluations, we treat L0 and L1 as feedback benchmarks and L2 as the held-out test. The detector is selected by its development objective J and evaluated with a fixed threshold chosen on training data. Appendix B specifies the development subsets, full-coverage evaluation, and additional collections in aggregate feedback.

Baselines and metrics. We compare dedicated image detectors and vision-language detectors. On DINOv3-H and the same 20K OpenFake split, we also train a frozen-feature readout, standard BCE adaptation, and three state-of-the-art recipes transplanted onto this setting: Effort's orthogonal-subspace adaptation (Yan et al., 2025b), SAFE's training transformations (Li et al., 2025a), and DDA-style aligned training data (Chen et al., 2025) (Appendix D.2). We report pooled balanced accuracy (bACC), the mean of real- and fake-image accuracy within each benchmark, with equal benchmark weights in aggregate results. The objective J combines source and processing performance (Equation 4). Baseline evaluation protocols and resource accounting appear in Appendix B.

4.2 Main Results

Detection performance. The round-35 detector ranks first on nine of the 12 benchmarks, with an overall mean of 94.34% against 85.22% for the strongest published detector, BCNet (Table 1). On the other three, all feedback benchmarks, it trails the best score by 0.39–1.81 points. Its margin holds where search receives no feedback: it ranks first on all four L2 benchmarks, with a mean of 91.95%

Table 2: **Framework ablations and public frameworks.** All rows search the three capability tiers with the same 240-proposal budget; Full is the main run. $\Delta J = J - J_0$ with $J_0 = 0.8830$; parentheses give $100\Delta J/J_0$. Δ columns give the bACC (%) decrease from Full in points. †: removes research cards, single-component edits, and question continuation (Appendix C.1).

Strategy	Search objective		L0		L1		L2		All	
	Selected $J \uparrow$	Gain $\Delta J \uparrow$	bACC↑	$\Delta \downarrow$	bACC↑	$\Delta \downarrow$	bACC↑	$\Delta \downarrow$	bACC↑	$\Delta \downarrow$
Full (main)	**0.9644**	**0.0814 (+9.22%)**	**93.05**	–	**98.03**	–	**91.95**	–	**94.34**	–
Aggregate feedback	0.9592	0.0762 (+8.63%)	91.89	1.16	97.15	0.88	88.82	3.12	92.62	1.72
No hypothesis protocol†	0.9493	0.0663 (+7.51%)	90.30	2.74	96.86	1.17	88.09	3.86	91.75	2.59
Unconstrained revision	0.9569	0.0739 (+8.37%)	92.34	0.70	97.64	0.39	87.70	4.25	92.56	1.78
Archive restarts	0.9541	0.0711 (+8.05%)	91.08	1.96	97.44	0.59	87.72	4.23	92.08	2.26
Independent proposals	0.9507	0.0677 (+7.67%)	90.84	2.20	97.31	0.71	87.95	3.99	92.04	2.30
Generic archive search	0.9577	0.0747 (+8.46%)	91.90	1.15	97.74	0.29	90.55	1.40	93.39	0.94
OpenEvolve	0.9511	0.0681 (+7.71%)	90.83	2.21	96.99	1.04	87.99	3.96	91.94	2.40
ShinkaEvolve	0.9521	0.0691 (+7.83%)	91.09	1.96	97.18	0.85	88.58	3.37	92.28	2.06

against 82.08% for DDA, a 9.87-point margin close to its 10.07-point margin on L0, where per-benchmark feedback is available. Under the threshold-free AUROC, its L2 mean is 96.91% against 87.19% for the best published detector, so the held-out advantage holds for score ranking as well as for the fixed-threshold decision. Appendix D.6 analyzes robustness to image processing: under JPEG compression, resizing, and Gaussian blur stronger than the feedback conditions, the detector keeps the highest mean bACC among the evaluated detectors at every tested setting. Appendix D.1 reports AUROC; Appendix G gives generator and source-subset bACC.

Gains from search. The discovered program leads every same-backbone run on all 12 benchmarks in both bACC and AUROC (Tables 1 and 18). These runs separate the contribution of search from that of the backbone, which already rivals published detectors as a frozen readout (84.25%). The program exceeds standard BCE adaptation by 4.14 points overall and 4.50 points on L2, and the strongest transplanted recipe, Effort, by 2.62 points overall (95% bootstrap CI 2.17–3.03) and 2.82 points on L2. Two repeated searches with new seeds reach 93.87% and 94.07% overall, at least 2.15 points above Effort (Appendix D.3). Search thus adds more than any transplanted practice: the best recipe improves on standard adaptation by 1.52 points, and the other two fall below it. During search, best observed J rises from 0.8830 to 0.9644, with gains in both source and processing means; Section 4.5 follows the experiments that produced the final recipe.

4.3 Ablation Studies

Framework controls. Full exceeds every control and both public frameworks in J and in mean bACC at every feedback level (Table 2). All controls share the main run's backbone, proposer, training set, three-tier space, objective, and evaluation protocol, with six streams and 40 rounds starting from the frozen-feature design. Aggregate feedback hides condition scores; unconstrained revision allows multi-component edits; archive restarts assign fresh questions without parents but retain history. Independent proposals remove history, while generic archive search retains code and scalar scores without the hypothesis protocol. The no-protocol arm removes research cards, single-component edits, and question continuation together. Appendix C.1 specifies the protocols and evaluated candidates. OpenEvolve (Sharma, 2025) and ShinkaEvolve (Lange et al., 2026) run in the same setting with only the search framework replaced.

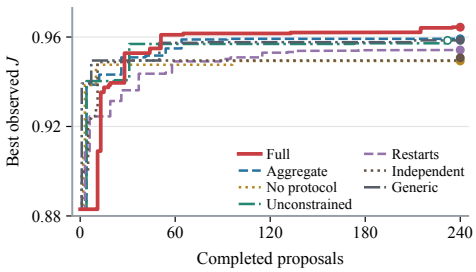


Figure 3: **Framework search progress.** Best J by completed proposals, including failures.

Sustained improvement. Full keeps improving after generic archive search plateaus (Figure 3). Generic archive search has the highest objective after 50 completed proposals ($J_{50} = 0.9574$ vs. 0.9549 for Full) but then plateaus from completion 75 through 209; Full crosses $J = 0.96$ after

Table 3: **Search with different editable capabilities.** Each configuration uses 30 proposal opportunities and the same frozen-feature reference ($J = 0.8830$). ΔJ is the best observed gain; L0, L1, L2, and All report mean bACC (%). Bold scores mark column maxima.

Available edits	Streams	Search objective		Mean bACC (%)			
		Best $J \uparrow$	$\Delta J \uparrow$	L0↑	L1↑	L2↑	All↑
Frozen views/readouts	T0, T0	0.9029	0.0199	86.34	92.75	82.60	87.23
+ Training-image edits	T0, T1	0.9032	0.0202	85.49	92.40	82.10	86.66
+ Backbone adaptation	T2, T2	**0.9526**	**0.0696**	**91.24**	**96.86**	**88.97**	**92.36**

Table 4: **Search across models and training sources.** Gains use run-specific frozen-feature references for (a,c) and a shared $J_0 = 0.8830$ for (b). $\Delta J = J_{\text{best}} - J_0$, with $100\Delta J/J_0$ in parentheses. bACC is in percent; shaded settings use the main run. Best search outcomes and bACC within each block are bold.

Setting	Search objective			Mean bACC (%)			
	J_0	$J_{\text{best}} \uparrow$	Gain $\Delta J \uparrow$	L0↑	L1↑	L2↑	All↑
(a) Visual backbone							
DINOv3-H	0.8830	**0.9644**	0.0814 (9.22%)	**93.05**	**98.03**	**91.95**	**94.34**
DINOv3-L	0.8636	0.9410	0.0774 (8.96%)	89.87	94.48	86.49	90.28
DINOv2-L	0.7368	0.8916	**0.1548 (21.01%)**	82.20	83.68	82.37	82.75
CLIP-L	0.7622	0.8767	0.1145 (15.02%)	82.57	83.73	77.04	81.11
ResNet-50	0.6987	0.8243	0.1256 (17.98%)	76.02	72.95	71.68	73.55
(b) Language-model proposer							
Qwen3.8-27B	0.8830	**0.9644**	**0.0814 (9.22%)**	**93.05**	**98.03**	91.95	**94.34**
DeepSeek-Flash	0.8830	0.9581	0.0751 (8.51%)	92.02	97.61	**92.03**	93.89
GPT-5.6-Terra	0.8830	0.9548	0.0718 (8.13%)	91.47	97.53	89.51	92.84
(c) Training set (20K images)							
OpenFake	0.8830	**0.9644**	0.0814 (9.22%)	**93.05**	98.03	**91.95**	**94.34**
GenImage	0.7933	0.8919	0.0986 (12.43%)	89.75	**98.36**	89.68	92.60
Community Forensics	0.7509	0.9026	**0.1517 (20.20%)**	86.31	95.92	87.66	89.97

51 completions and produces 31 scored programs above this level, which no control reaches. Its advantage thus comes from developing strong designs further once they appear, the behavior that experimental lineages are built to support. Appendix C.3 connects these trajectories to the agents' program revisions.

Gains beyond the feedback suite. The largest detection gaps occur on search-excluded benchmarks. Controls and public frameworks within 0.29–1.17 points of Full on L1 trail it by 1.40–4.25 points on L2, and for each of them the L2 gap exceeds its L0 and L1 gaps. Together with the search curves, this result links the organization of experiments to improvements that extend beyond the feedback suite.

Editable capabilities. Opening backbone adaptation yields the largest gain at 30 proposals (Table 3). The three searches share the frozen-feature reference and evaluation protocol. With frozen features, training-image edits change only the examples used to fit the readout, and they add little objective gain (0.0202 vs. 0.0199). Once the backbone can adapt, the same edits shape what it learns: two T2 streams reach 3.4 times the gain of the T0/T1 configuration and higher bACC at every feedback level, and the final recipe's resampling is one such edit (Section 4.5).

4.4 Search across Models and Training Sources

Visual backbones. Search raises J for all five backbones (Table 4a) and their mean bACC at every feedback level (Table 24 in Appendix D.4). Each uses 240 proposals from a frozen-feature reference, with LoRA for Transformers and full fine-tuning for ResNet-50 (He et al., 2016). DINOv2-L (Oquab et al., 2024) gains the most in J, while DINOv3-H retains the strongest endpoint: search complements a strong initial representation. Gains also extend to CLIP-L (Radford et al., 2021) and convolutional features. Appendix D.5 follows the designs found for each backbone.

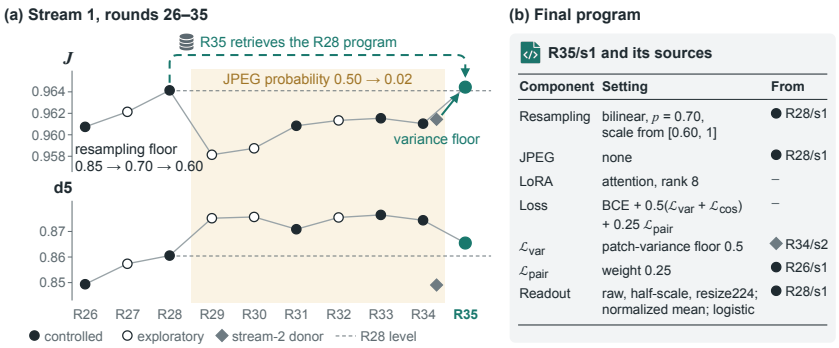


Figure 4: **Experimental evidence in program revision.** (a) J and d5 for stream 1 in rounds 26–35; filled markers are controlled revisions and hollow markers exploratory ones. The JPEG trials (shaded) raise d5 but lower J; at R35 the agent retrieves the R28 program and adds the variance floor from R34/s2. (b) The final program and the experiments its components come from.

Language-model proposers. With DINOv3-H and OpenFake fixed, all three proposers improve J by 0.072–0.081 from the shared reference, and DeepSeek-Flash exceeds the main run on L2 (92.03% vs. 91.95%; Table 4b). Their recipes differ: DeepSeek combines three views with rank-16 adaptation, while Terra uses two views and rank eight, so the loop reaches strong detectors through different paths. Their trajectories and resource use are reported in Appendices D.5 and B.4.

Training sets. We use 20,000 class-balanced images from OpenFake, GenImage (Zhu et al., 2023), or Community Forensics (Park & Owens, 2025). Each search improves J over its reference (Table 4c). OpenFake yields the strongest detector both overall and on L2. Across Table 4, settings that start below $J_0 = 0.80$ gain at least 0.098 in J, while those starting above 0.86 gain at most 0.082: search adds most where the initial representation or data leave the most room.

4.5 Analysis of the Search Process

How experiments inform revisions. The lineage shows how measured tradeoffs inform which designs to retain and combine (Figure 4). In a controlled revision, lowering the resampling floor from 0.70 to 0.60 improves both J and the weakest processing score, d5. Later JPEG-augmentation trials improve d5 but reduce J. At round 35, the agent retrieves the no-JPEG program and a patch-variance-floor loss from another stream, combining them into the highest-J detector. The search thus carries useful designs forward even when intervening trials lower the objective. Appendix E.2 documents the retrieval events and code changes; Appendix E.3 specifies the final detector.

Tradeoffs hidden by J. Condition-level feedback exposes tradeoffs that the objective hides. In the GenImage run, a patch-geometry revision raises J by only 0.0018, while its processing mean rises by 0.0116 and in-domain development AUROC falls by 0.0041. Its research card predicted a processing gain below 0.003, so the record captures both the robustness–ranking tradeoff and an underestimated processing benefit (Appendix E.2).

5 Conclusion

We introduced ForenEvolve, a framework for hypothesis-driven evolution of AI-generated image detector programs. Each revision is registered as a test with a predicted outcome, controlled comparisons across image sources and processed copies of the same images show where it helps or fails, and experimental lineages carry the resulting evidence into later revisions. The discovered detector reaches 94.34% mean balanced accuracy across 12 public benchmarks, ranks first on all four benchmarks excluded from search, and outperforms standard adaptation (by 4.14 percentage points) and three transplanted state-of-the-art recipes on the same backbone and data. With the number of experiments held fixed, their organization shapes how far detector gains extend beyond the feedback used during search. Temporal holdouts from newly released generators and repeated control searches are natural next tests of this principle (Appendix F).

AI USE STATEMENT

We used generative AI tools for literature search and paper summarization, research ideation and hypothesis refinement, experimental design and analysis, and assistance with implementing code and running experiments. The language-model agent within ForenEvolve is a component of the proposed method, described in Section 3. For manuscript preparation, the authors provided drafts that AI tools helped refine for clarity and presentation; AI assistance also supported figure and table preparation. To reduce the risk of hallucinated claims or citations, the authors checked key factual statements against original sources and verified key experimental findings against execution records and evaluation outputs. AI-generated suggestions were treated as proposals requiring verification, rather than as independent evidence. The authors reviewed the final manuscript in full and take responsibility for its accuracy, originality, and integrity, including all content produced with AI assistance.

ETHICS STATEMENT

ForenEvolve aims to support image provenance analysis through improved detection of AI-generated images. Our experiments use existing image datasets and pretrained models. Detection errors can mislabel authentic images or miss generated content, potentially affecting creators and other individuals. Detector outputs should therefore be interpreted alongside provenance information and human assessment, particularly when they inform consequential decisions. Synthetic origin alone does not establish that an image is misleading or harmful. Practical deployment should include evaluation under the intended image sources and processing conditions. Use and redistribution of the underlying datasets and models should respect their licenses, access conditions, and applicable privacy requirements.

REPRODUCIBILITY STATEMENT

We will open-source the code and release the experimental archive—including candidate programs, research cards, agent traces, and evaluation records—and the final detector weights.

Section 3 and Appendix A describe the executable program interfaces, proposal and revision procedure, controlled comparisons, and execution checks. Section 4.1 and Appendix B describe the training data, evaluation protocol, feedback visibility, detector selection, metrics, and resource budgets. Appendix E.1 reports an audit based on recorded agent interactions, program snapshots, and measurement events, while Appendix E.3 details the discovered detector.

Agent-generated prototype code and import checks run in a Bubblewrap CPU sandbox with read-only access to approved sources and feature caches, temporary scratch space, disabled network and GPU access, and execution timeouts. Prototype execution requires the sandbox backend. Candidate training and scoring run separately in worker processes with enforced wall-clock limits (Appendix A).

REFERENCES

Baoying Chen, Jishen Zeng, Jianquan Yang, and Rui Yang. DRCT: Diffusion reconstruction contrastive training towards universal detection of diffusion generated images. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7621–7639, 2024. URL https://proceedings.mlr.press/v235/chen24ay.html.

Ruoxin Chen, Junwei Xi, Zhiyuan Yan, Ke-Yue Zhang, Shuang Wu, Jingyi Xie, Xu Chen, Lei Xu, Isabel Guan, Taiping Yao, and Shouhong Ding. Dual data alignment makes AI-generated image detector easier generalizable. In *NeurIPS*, volume 38, pp. 106475–106500, 2025. doi: 10.52202/085713-3552. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/991c9f1799cfeebb4217baaacc462d86-Abstract-Conference.html.

Oguzhan Gungordu, Siheng Xiong, and Faramarz Fekri. PathWise: Planning through world model for automated heuristic design via self-evolving LLMs. In *ICML*, 2026. URL https://openreview.net/forum?id=aiOv6NhF3N.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *ICLR*, volume 2025, pp. 21344–21377, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/36b7acf6f6010652b3f2a433774a66fe-Abstract-Conference.html.

Robert Lange, Yuki Imajuku, and Edoardo Cetin. ShinkaEvolve: Towards open-ended and sample-efficient program evolution. In *ICLR*, volume 2026, pp. 74026–74078, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7886b9bafe76c52fd568db10ff9772df-Abstract-Conference.html.

Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Fuli Feng. Improving synthetic image detection towards generalization: An image transformation perspective. In *KDD*, 2025a. URL https://arxiv.org/abs/2408.06741.

Ziqiang Li, Jiazhen Yan, Ziwen He, Kai Zeng, Weiwei Jiang, Lizhi Xiong, and Zhangjie Fu. Is artificial intelligence generated image detection a solved problem? In *NeurIPS*, volume 38, 2025b. doi: 10.52202/085713-5739. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/fb693c67f61e5321746ffce8b6fdd2d0-Abstract-Datasets_and_Benchmarks_Track.html.

Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *CVPR*, pp. 10770–10780, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Liu_Forgery-aware_Adaptive_Transformer_for_Generalizable_Synthetic_Image_Detection_CVPR_2024_paper.html.

Zexi Liu, Jingyi Chai, Xinyu Zhu, Shuo Tang, Rui Ye, Weiyu Ma, Bo Zhang, Lei Bai, and Siheng Chen. ML-Agent: Reinforcing LLM agents for autonomous machine learning engineering. In *ICML*, 2026. URL https://openreview.net/forum?id=kcPPWaoegr.

Victor Livernoche, Akshatha Arodi, Andreea Musulan, Zachary Yang, Adam Salvail, Gaétan Marceau Caron, Jean-François Godbout, and Reihaneh Rabbany. OpenFake: An open dataset and platform toward real-world deepfake detection. *arXiv preprint arXiv:2509.09495*, 2025. URL https://arxiv.org/abs/2509.09495.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025. URL https://arxiv.org/abs/2506.13131.

Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, pp. 24480–24489, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Ojha_Towards_Universal_Fake_Image_Detectors_That_Generalize_Across_Generative_Models_CVPR_2023_paper.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut,

Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Jeongsoo Park and Andrew Owens. Community Forensics: Using thousands of generators to train fake image detectors. In *CVPR*, pp. 8245–8257, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Park_Community_Forensics_Using_Thousands_of_Generators_to_Train_Fake_Image_CVPR_2025_paper.html.

Ziheng Qin, Yuheng Ji, Renshuai Tao, Yuxuan Tian, Yuyang Liu, Yipu Wang, and Xiaolong Zheng. Scaling up AI-generated image detection with generator-aware prototypes. In *CVPR*, pp. 43008–43017, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Qin_Scaling_Up_AI-Generated_Image_Detection_with_Generator-Aware_Prototypes_CVPR_2026_paper.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Asankhaya Sharma. OpenEvolve: An open-source evolutionary coding agent. https://github.com/codelion/openevolve, 2025.

Jinjie Shen, Jing Wu, Yaxiong Wang, Lechao Cheng, Shengeng Tang, Tianrui Hui, Nan Pu, and Zhun Zhong. OmniVL-Guard: Towards unified vision-language forgery detection and grounding via balanced RL. In *ICML*, 2026. URL https://openreview.net/forum?id=U80CWouffM.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

Anirudh Sundara Rajan, Utkarsh Ojha, Jedidiah Schloesser, and Yong Jae Lee. Aligned datasets improve detection of latent diffusion-generated images. In *ICLR*, volume 2025, pp. 63304–63322, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9ead108421b202494d01b5060d12aa34-Abstract-Conference.html.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning. *AAAI*, 38(5):5052–5060, 2024a. doi: 10.1609/aaai.v38i5.28310. URL https://doi.org/10.1609/aaai.v38i5.28310.

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 *CVPR*, pp. 28130–28139, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Tan_Rethinking_the_Up-Sampling_Operations_in_CNN-based_Generative_Network_for_Generalizable_CVPR_2024_paper.html.

Jiaan Wang, Sirui Liu, Yu Li, Kaiyuan Yang, Juan Cao, and Sheng Tang. Fleet: Few shots lead effective AI-generated image detection. In *ICML*, 2026a. URL https://openreview.net/forum?id=3QstqZrwjO.

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. CNN-generated images are surprisingly easy to spot... for now. In *CVPR*, pp. 8695–8704, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Wang_CNN-Generated_Images_Are_Surprisingly_Easy_to_Spot..._for_Now_CVPR_2020_paper.html.

Wenyi Wang, Piotr Piękos, Li Nanbo, Firas Laakom, Yimeng Chen, Mateusz Ostaszewski, Mingchen Zhuge, and Jürgen Schmidhuber. Huxley-Gödel Machine: Human-level coding agent development by an approximation of the optimal self-improving machine. In *ICLR*, volume 2026, pp. 80356–80386, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/821d20219c2f14850af1b5220f0ed13f-Abstract-Conference.html.

Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. DIRE for diffusion-generated image detection. In *ICCV*, pp. 22445–22455, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Wang_DIRE_for_Diffusion-Generated_Image_Detection_ICCV_2023_paper.html.

Siwei Wen, Junyan Ye, Peilin Feng, Hengrui Kang, Zichen Wen, Yize Chen, Jiang Wu, Wenjun Wu, Conghui He, and Weijia Li. Spot the fake: Large multimodal model-based synthetic image detection with artifact explanation. In *NeurIPS*, volume 38, pp. 58972–59005, 2025. doi: 10.52202/085713-1974. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/551d275ae5cb1bb1ebe6e37fbe201db8-Abstract-Conference.html.

Jiazhen Yan, Ziqiang Li, Fan Wang, Boyu Wang, Ziwen He, and Zhangjie Fu. DGS-Net: Distillation-guided gradient surgery for CLIP fine-tuning in AI-generated image detection. In *ICML*, 2026. URL https://openreview.net/forum?id=G5XGej7wNt.

Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for AI-generated image detection. In *ICLR*, volume 2025, pp. 70702–70720, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b0303773962ea1b5394c3a83cc7dd066-Abstract-Conference.html.

Zhiyuan Yan, Jiangming Wang, Peng Jin, Ke-Yue Zhang, Chengchun Liu, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Orthogonal subspace decomposition for generalizable ai-generated image detection. In *ICML*, 2025b. URL https://arxiv.org/abs/2411.15633.

Jian Yang, Shibo Yao, Renshuai Tao, Chuangchuang Tan, and Yao Zhao. Revisiting deepfake detection: BCNet for robust generalization beyond semantic dependence. In *ECCV*, pp. 211–230. Springer, 2026. doi: 10.1007/978-3-032-37556-8_12. URL https://doi.org/10.1007/978-3-032-37556-8_12.

Xu Yang, Xiao Yang, Shikai Fang, Yifei Zhang, Jian Wang, Bowen Xian, Qizheng Li, Jingyuan Li, Minrui Xu, Yuante Li, Haoran Pan, Yuge Zhang, Weiqing Liu, Yelong Shen, Weizhu Chen, and Jiang Bian. R&D-Agent: An LLM-agent framework towards autonomous data science. *arXiv preprint arXiv:2505.14738*, 2025. URL https://arxiv.org/abs/2505.14738.

Fanrui Zhang, Qiang Zhang, Sizhuo Zhou, Jianwen Sun, Chuanhao Li, Jiaxin Ai, Yukang Feng, Yujie Zhang, Wenjie Li, Zizhen Li, Yifan Chang, Jiawei Liu, and Kaipeng Zhang. Code-in-the-loop forensics: Agentic tool use for image forgery detection. In *ECCV*, pp. 203–221. Springer, 2026a. doi: 10.1007/978-3-032-37592-6_12. URL https://doi.org/10.1007/978-3-032-37592-6_12.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Gödel Machine: Open-ended evolution of self-improving agents. In *ICLR*, volume 2026, pp. 104223–104294, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/aa5f5e6eb6f613ec412f1d948dfa21a5-Abstract-Conference.html.

Shan Zhang, Yongxin He, Mingming Zhang, Huiwen Tian, and Lei Ma. FiSeR: Fine-grained source representations for cross-domain AI image detection. In *ICML*, 2026c. URL https://openreview.net/forum?id=qjwFbN77kx.

Yanran Zhang, Bingyao Yu, Yu Zheng, Wenzhao Zheng, Yueqi Duan, Lei Chen, Jie Zhou, and Jiwen Lu. D3QE: Learning discrete distribution discrepancy-aware quantization error for autoregressive-generated image detection. In *ICCV*, pp. 16292–16301, 2025. URL

https://openaccess.thecvf.com/content/ICCV2025/html/Zhang_D3QE_Learning_Discrete_Distribution_Discrepancy-aware_Quantization_Error_for_Autoregressive-Generated_Image_ICCV_2025_paper.html.

Yanran Zhang, Wenzhao Zheng, Yifei Li, Bingyao Yu, Yu Zheng, Lei Chen, Jiwen Lu, and Jie Zhou. UniGenDet: A unified generative-discriminative framework for co-evolutionary image generation and generated image detection. In *CVPR*, pp. 16226–16236, 2026d. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhang_UniGenDet_A_Unified_Generative-Discriminative_Framework_for_Co-Evolutionary_Image_Generation_and_CVPR_2026_paper.html.

Zhenyuan Zhao, Yu Xing, Tianyang Xue, Lingxin Cao, Xin Yan, and Lin Lu. AutoMS: Multi-agent evolutionary search for cross-physics inverse microstructure design. In *ICML*, 2026. URL https://openreview.net/forum?id=zd6V4IqBkd.

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. In *NeurIPS*, volume 36, pp. 77771–77782, 2023. doi: 10.52202/075280-3398. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f4d4a021f9051a6c18183b059117e8b5-Abstract-Datasets_and_Benchmarks.html.

APPENDIX CONTENTS

A Method and Agent Implementation **16**

A.1 Programmable Interfaces and Execution 16

A.2 Proposal Context and Research Cards 16

A.3 Controlled Revisions and Experimental Continuation 17

B Experimental Protocol **18**

B.1 Data and Feedback Visibility 18

B.2 Detector Selection and Evaluation 18

B.3 Baselines and Search Configurations 19

B.4 Search Budgets and Resource Use 20

C Framework Ablations **22**

C.1 Intervention Protocols and Evaluated Detectors 22

C.2 Search Progress and Benchmark Results 22

C.3 How Experimental Evidence Guides Revisions 23

D Additional Experimental Results **25**

D.1 Additional Detection Results 25

D.2 Same-Backbone Recipe Baselines 26

D.3 Repeated Full Searches 27

D.4 Per-Benchmark Capability and Configuration Results 28

D.5 Search Trajectories across Models 28

D.6 Robustness to Image Processing 30

E Experimental Lineages and Discovered Detectors **33**

E.1 Trace Analysis Protocol 33

E.2 Hypothesis Tests and Cumulative Design 33

E.3 Discovered Detector Recipes 34

F Limitations **35**

G Per-Generator and Source-Subset Results **38**

Table 5: **Detector-program interfaces.** Higher tiers inherit the editable components of lower tiers. All tiers return a fitted per-image scoring function.

Tier	Editable components	Training sequence
T0	Views, feature combination, readout, readout-training loss, score transformation	Fit the readout on frozen features.
T1	T0 plus training-image operations and a synthesis plan	Rebuild training features, then fit the readout.
T2	T1 plus backbone adaptation configuration and a programmable adaptation loss	Adapt the backbone, re-extract features, then fit the readout.

A Method and Agent Implementation

This section specifies how the agent turns a forensic hypothesis into an executable comparison and carries its outcome into subsequent revisions.

A.1 Programmable Interfaces and Execution

Every detector program declares its tier and feature views and implements `fit(train)` and `score(data)`. Views expose raw-image, multiscale, frequency, and residual representations; the readout determines how their features contribute to an image's score. Table 5 specifies the editable components and training sequence for each tier. Candidate training changes within this interface, while the evaluator retains the same images and processing conditions for program comparisons.

Training-image operations act before cropping and feature encoding through `train_image_op(image, rng)`. A synthesis plan specifies operations and sample counts for additional training examples. For T2, the programmable `adapt_loss(batch, head)` receives pooled features, optional patch features, images, labels, and group and pair identifiers, together with a temporary linear head. Each candidate starts from the chosen backbone's pretrained weights. Transformers optimize LoRA parameters and the temporary head; ResNet-50 optimizes the full backbone and head. After adaptation, the runner discards the temporary head, extracts features with the adapted backbone, and fits the program's final readout. This separates the objective used to learn forensic features from the rule used to combine them at inference.

Prototype execution runs in a CPU sandbox with read-only access to program sources, registered implementations, and the training and OpenFake development feature caches. It has temporary scratch space but no network or GPU access. The runner separately performs train-only preflight checks, GPU adaptation, and scoring; benchmark images and features are supplied at the scoring stage rather than exposed through the prototype tool.

Execution checks test score reproducibility and independence from evaluation labels and batch composition. Refitting the readout twice on identical training features must produce a maximum absolute score difference below 10^{-9}; permuting evaluation labels permits a shift of at most 10^{-9}, and changing batch composition permits at most 10^{-5}. Evaluation matches row identities across candidates and uses a training-selected threshold fixed across benchmarks (Appendix B.2).

A.2 Proposal Context and Research Cards

The proposal context pairs prior designs with the measurements needed to decide what to test next (Table 6). Local records retain predictions and falsification criteria, while strong programs from other streams provide components to inspect and reuse. The agent can open their source and retrieve experimental observations or literature before registering its next research card.

The main run starts from a frozen library of 830 literature and implementation notes and nine verified code repositories registered at pinned revisions. These resources describe published methods, training recipes, evaluation protocols, and their reported results. The mapping from anonymous runtime scores to benchmark names is withheld from the agent. New search observations enter a separate

Table 6: **Context for a proposal.** Completed experiments supply both design references and evidence for the next test.

Context	Information supplied	Use in the proposal
Assignment	Tier, experiment mode, current question, parent or control, and execution allowance.	Defines the revision to deliver.
Local history	Four most recent local experiments and the current question's best measured program, with predictions and outcomes.	Identifies the effect to test next.
Shared designs	Three highest-J available programs, the best program within each tier, and the promoted frontier.	Supplies components and references for reuse.
Feedback	Anonymous source and processing scores, aggregate metrics, diagnostics, matched differences, and execution outcomes.	Locates gains, trade-offs, and implementation problems.
Retrievable records	Candidate source, experimental observations, literature, and reference implementations.	Connects an edit to its implementation and supporting evidence.

Table 7: **A prospective resampling test.** Task excerpt and condensed research card for R28/s1, recorded before evaluation. R27/s1 is the control.

Field	Recorded task or proposed test
Task excerpt	"In controlled mode, read the entire required control source and preserve everything except one named method or one class attribute"
Question	Does lowering the training-resampling floor from 0.70 to 0.60 further improve `d5` while preserving source ranking?
Hypothesis	The preceding wider resampling band improved `d5`. Stronger training rescaling should extend this gain and reduce processing spread.
Edit	Change only `RESAMPLE_MIN`; retain the kernel, operation probability, adaptation loss, and readout.
Prediction	`d5` ≥ 0.858; processing spread ≤ 0.104; mean development AUROC ≥ 0.9925; $J \geq 0.960$; processing retention ≥ 0.965.
Falsification and follow-up	If `d5` < 0.855 and spread ≥ 0.105, revert to the earlier floor and test JPEG. If mean development AUROC < 0.990 or $J < 0.955$, narrow the resampling range or reduce its application probability.

experimental archive, allowing retrieval to combine prior technical knowledge with measurements from the current run.

A research card states the question, hypothesis, predicted outcome, falsification criterion, proposed edit, and follow-up tests. It also names the parent, executable control, and supporting sources. Table 7 condenses a recorded task and card from round 28 of stream 1. The proposed resampling change targets the weak processing score `d5`, with bounds on source ranking and overall performance. Here processing spread is the maximum minus minimum score over processed conditions; retention is their mean score divided by the unprocessed reference. The card makes both the intended benefit and the acceptable trade-offs explicit before measurement.

A.3 Controlled Revisions and Experimental Continuation

A controlled revision tests one component within an already measured detector. Its control is the assigned parent in the same tier, with available source and a completed evaluation. The declared target is `fit`, `score`, `train_image_op`, `adapt_loss`, or one class attribute. Before execution, the checker normalizes class names and removes docstrings, then compares the two abstract syntax trees with that target removed. The remaining structure must match, and the target itself must dif-

fer. Open exploration permits changes to several components; measured exploratory programs can become controls for subsequent tests.

Concurrent streams share completed experiments, and each continuing stream waits for its parent to finish. An unscored outcome triggers an implementation repair. A scored exploratory parent can receive a controlled follow-up; after a matched comparison, the agent receives the measured differences and the card's proposed next tests. At the main run's two-experiment checkpoint, a question continues while its best objective improves; otherwise, the controller starts a new question using the accumulated archive. Reproducing an earlier detector's per-image score vector prompts a substantive revision. Thus, the controller sets the experiment mode, while the agent chooses the design to test within it.

For a matched pair, feedback links the declared edit to changes in J, source scores, and processing scores. An observation writer makes this comparison searchable alongside the earlier hypothesis and prediction. The round-28 observation, for example, records a 0.0020 increase in J and a 0.0032 increase in `d5` after lowering the resampling floor. Later proposals can recover the tested change together with its conditional effects; Appendix E.2 follows its use in the lineage leading to the final detector.

B Experimental Protocol

B.1 Data and Feedback Visibility

The main search uses 20,000 OpenFake training images, balanced between real and generated classes, and a fixed in-domain development split of 2,773 real and 1,913 generated images. The two splits have no shared image hashes. Training data and in-domain development features support prototype execution; benchmark evaluation remains under the runner's control.

Feedback levels describe how benchmark measurements enter the search. L0 supplies four anonymous scores: `g1` is AIGI-Now, `g2` AIGIBench, `g3` Chameleon, and `g4` SynthWildX. L1, EvalGEN, and ForenSynths contribute only through aggregate feedback. The broader development mean averages AUROC across these ten collections; its joint-score companion uses those with at least 1,000 eligible images, excluding ForenSynths. The processing pool contributes separately to J. Runtime promotion requires an improvement in J, an in-domain joint score within 0.01 of the incumbent, and a broader-development AUROC within 0.002 of the best previously promoted value. The four L2 benchmarks contribute neither feedback nor detector selection.

Search subsets are sampled from the benchmark manifests, capped at 4,000 images per collection, and filtered for the declared multiscale views. Their images are included in the corresponding full-coverage evaluations. Search uses images with short sides of at least 448 pixels; final evaluation upsamples smaller images to 448 and retains larger images at native size before extracting the program's 224-pixel feature crops. AIGI-Now combines its pixel-artifact and semantic subsets, and DailyBench-Fake denotes FakeBench.

The visibility split concerns search access; generator families and real-image sources can recur across collections. For example, OpenFake training includes FLUX and Stable Diffusion families. The available hash audit finds no exact image overlap between the 51,636-image Community Forensics evaluation and the 24,686-image OpenFake train/development pool, while both contain LAION real images.

Processing feedback measures eight treatments of the same 1,640 images (200 real and 1,440 fake images from eight generators). Table 8 specifies their operations in order. These conditions expose both individual degradations and combinations; the separate six-benchmark robustness study varies degradation severity across new evaluation supports (Appendix D.6).

B.2 Detector Selection and Evaluation

The main detector is `cand_i3035_s1_0`, the round-35 program with the highest valid development objective, $J = 0.9644$. Final selection uses measured J; the runtime promotion checks above govern the frontier shown during search. Full-coverage evaluation loads the selected program and its archived adapter, re-extracts the original training features, and refits the program's read-

Table 8: **Processing conditions used in search feedback.** Operations are applied left to right to the same image pool. Each resize factor is relative to that operation's input; "restore" returns to its original dimensions.

ID	Treatment
`d0`	Reference RGB image
`d1`	JPEG quality 95
`d2`	JPEG quality 75
`d3`	Resize $0.70 \rightarrow$ restore $\rightarrow$ JPEG 95
`d4`	Resize $0.85 \rightarrow$ JPEG 88 $\rightarrow$ WebP 92
`d5`	Resize $0.70 \rightarrow$ JPEG 75 $\rightarrow$ resize $1.15 \rightarrow$ JPEG 85
`d6`	Gaussian blur ($\sigma = 0.6$) $\rightarrow$ Gaussian noise ($\sigma = 2/255$) $\rightarrow$ JPEG 92
`d7`	Center crop $0.90 \rightarrow$ restore $\rightarrow$ JPEG 90

Table 9: **Search and full-coverage evaluation states.** All three readouts are refitted on the original training split for full evaluation. Thresholds are selected from training scores and fixed across benchmarks.

Detector	Search J	Adaptation state	Search τ	Full τ
Frozen reference	0.8830	None	-0.1429	-0.1438722
Standard BCE	0.9219	Archived LoRA	-0.5071	-0.4856575
ForenEvolve (R35)	0.9644	Archived LoRA	-0.5957	-0.6158207

out. Training-image operations apply only to the training split. The adapter undergoes no further optimization.

The evaluation threshold maximizes training bACC over the unique score quantiles at probabilities $0.01, 0.02, \ldots, 0.99$, as in search. The resulting readout and threshold are then fixed across benchmarks. Table 9 records the search and evaluation thresholds for the discovered detector and its two same-backbone references. The full-coverage readouts are training-side refits, with the archived adaptation states retained where applicable.

Balanced accuracy pools images within each benchmark and gives equal weight to its real and generated classes:

$$\mathrm{bACC}(E) = \frac{\mathrm{TPR}_E(\tau) + \mathrm{TNR}_E(\tau)}{2}. \tag{6}$$

AUROC uses the same pooled images. Each level mean equally weights its four benchmarks, and the overall mean weights all twelve. Means require every constituent score. Generator-level breakdowns describe variation within a benchmark and are reported separately from these pooled metrics. The reported L2 balanced-error reduction is $1 - (1 - \overline{\mathrm{bACC}}_{\mathrm{ours}})/(1 - \overline{\mathrm{bACC}}_{\mathrm{ref}})$, using the strongest published L2 mean as the reference; it rounds to 55%.

B.3 Baselines and Search Configurations

The frozen-feature and standard BCE references isolate two stages of detector development on the same DINOv3-H backbone and OpenFake training split (Table 10). The frozen reference declares raw and half-scale views for common evaluation support; its linear readout uses the L2-normalized raw-view features. The BCE reference retains this readout and applies the default attention LoRA configuration: rank 8, alpha 16, zero dropout, two epochs, learning rate 10^{-4}, and batch size 32. Both reference readouts use logistic regression with $C = 1$ and up to 30 optimizer iterations. The BCE reference uses no program search. The discovered program adds resampling, a programmable adaptation loss, and three-view fusion; its full recipe appears in Appendix E.3.

External detector scores come from the comparison records summarized in Table 11. The dedicated-detector pooled score records do not identify every checkpoint revision, preprocessing pipeline, or

Table 10: **Same-backbone reference configurations.** All use DINOv3-H and the same 20K OpenFake training split. BCE and R35 share the LoRA hyperparameters given in the text.

Component	Frozen reference	Standard BCE	ForenEvolve (R35)
Declared views	Raw, half-scale	Raw, half-scale	Raw, half-scale, resize224
Training transform	None	None	Downscale and restore
Backbone update	None	Attention LoRA	Attention LoRA
Adaptation loss	–	BCE	BCE + patch regularizers
Readout input	Normalized raw	Normalized raw	Mean of normalized views, renormalized
Final readout	Linear	Linear	Linear
Program search	None	None	Three-tier search

Table 11: **Sources of external baseline comparisons.** Pooled main-table scores and source-subset exports are retained as separate records.

Methods	Available evaluation record
NPR, UnivFD, FatFormer, AIDE, DRCT, DDA, GAPL, DGS-Net, BCNet	Compiled pooled bACC comparison scores. The supplied subset export identifies AIDE-SD14 and DRCT-DRCT2M-UnivFD-SDv1; its pooled headers differ from some main-table entries.
FakeVLM, OmniVL-Guard-2B	Supplied evaluation exports with pooled and subset scores, successful sample counts, and excluded unusable predictions. The released OmniVL model is `SJJ0854/OmniVL-Guard-2B`.
UniGenDet	Supplied evaluation and subset records using $P(\text{fake}) > 0.5$, with no failed predictions reported. Run-specific real pools are retained in the generator-level tables.

threshold; these entries are compiled comparisons. The separately supplied generator-level exports retain their own source and coverage information (Appendix G).

Table 12 specifies what changes in the capability, backbone, proposer, and training-source studies. Each run starts from its frozen-feature reference and maintains its own archive. Transformer backbones use LoRA; ResNet-50 uses full fine-tuning. The capability study allocates 30 proposals to each of three two-stream configurations, while the full backbone and training-source searches allocate 240 proposals.

Each training-source run uses 20,000 class-balanced training images and 4,686 in-domain development images. The Community Forensics configuration uses 302 training model groups and 32 disjoint fake-development groups. GenImage training upsamples short sides below 448 pixels; Community Forensics uses native images with short sides of at least 448 pixels. GenImage is removed from aggregate feedback when used for training. The result tables retain the main run's L0/L1/L2 column groups, so each alternative training source is interpreted as a training-source evaluation in its own run.

B.4 Search Budgets and Resource Use

The main run and framework controls allocate 240 proposals across six streams, with a 3,600-second candidate limit and a 450,000-second total execution allowance. In Full, one T2 stream opens new questions and five streams maintain lineages. A proposal session allows up to 60,000 output tokens, giving a 14.4M-token run allowance. New-question and continuing sessions allow 14 and 18 tool turns and 2,400 and 3,000 seconds, respectively. Framework controls change the experiment organization as specified in Appendix C.1. The DeepSeek and Terra proposer runs use a 216,000-second execution allowance. The main proposer uses a 131,072-token context window. Its generation phases use temperatures 0.3 (analysis), 0 (browsing), 0.3 (interpretation), 0.8 (synthesis), 0.6 (delivery rescue), and 0 (recovery), with respective per-response output caps of 12,288, 1,024,

Table 12: **Search configurations.** Each comparison changes the listed factor from DINOv3-H, Qwen3.8-27B, and OpenFake 20K. "Main layout" is one T0, two T1, and three T2 streams over 40 rounds. Counts are proposal allowances.

Study	Variants	Streams	Proposals
Capability	T0 only; T0+T1; T2 only	T0/T0; T0/T1; T2/T2	30 each
Backbone	DINOv3-H/16, DINOv3-L/16, DINOv2-L/14, CLIP-L/14, ResNet-50	Main layout	240 each
Proposer	Qwen3.8-27B; DeepSeek-Flash; GPT-5.6-Terra	Main layout	Up to 240
Training source	OpenFake; GenImage; Community Forensics wide-dev	Main layout	240 each

Table 13: **Recorded search resource use.** Tokens are in millions. Panel (a) reports formal proposer runs; panel (b) reports the framework controls. Charged execution hours exclude proposer inference.

(a) Proposer runs

Proposer	Sessions	Input M	Output M	Charged h
Qwen3.8-27B	240	94.816	6.638	105.22
DeepSeek-Flash	188	56.815	5.630	59.33
GPT-5.6-Terra	210	24.750	0.698	59.09

(b) Framework controls

Control	Sessions	Input M	Output M
Aggregate feedback	240	70.459	3.671
No hypothesis protocol	240	68.385	3.040
Unconstrained revision	235	69.811	3.804
Archive restarts	240	86.972	4.762
Independent proposals	240	43.535	3.097
Generic archive search	240	58.049	2.350

Full is the Qwen run in panel (a). DeepSeek's separate recovery attempts add 0.450M input and 0.052M output tokens (57.265M/5.682M including recovery); its charged execution ledger includes recovery work. Main execution charges include the initial reference. Unconstrained revision has 235 recorded agent sessions and 232 scored-or-failed outcomes in the available snapshots.

10,240, 16,384, 24,576, and 2,048 tokens within the session allowance. Search execution uses A100 workers for the main run and eight/four H20 GPUs for DeepSeek/Terra, whose configured API model identifiers are `deepseek-flash` and `gpt-5.6-terra`.

Table 13 reports observed resource use. Tokens are summed from per-response usage records, including recorded responses in interrupted sessions; unused output allowances assigned to failed API requests are excluded. Input counts include repeated context. Execution time is the scheduler's charged duration for training, evaluation, preflight, and failed attempts, including fallback charges when measured durations are absent. It excludes proposer inference and is distinct from GPU utilization time. Search cost is incurred when developing a detector; subsequent inference uses the fixed program described in Appendix E.3.

Search curves order proposals by the time their scores or failures enter the archive. Initialization is plotted at zero; every completed proposal advances the count, and failures leave the best valid J unchanged. A valid-scored proposal has an available J, while an agent session can finish before scoring completes. Local round indices identify positions within individual asynchronous streams. Framework curves use completion order; the configuration curves retain local indices (Appendix D.5).

Table 14: **Executed framework interventions.** All arms retain the same evaluator, technical checks, initial reference, and static resources. Single-target edits apply to controlled follow-ups; open exploration can change several components.

Arm	Score feedback	Research card	Follow-up edits	Question continuation	Prior programs
Full	Conditions	Required	Single target	Assigned	Available
Aggregate feedback	Aggregate	Required	Single target	Assigned	Available
No hypothesis protocol	Conditions	Removed	Free	Not required	Available
Unconstrained revision	Conditions	Required	Free	Assigned	Available
Archive restarts	Conditions	Required	Voluntary controls	New question	Available
Independent proposals	Initial only	Required	Independent	New question	Unavailable
Generic archive search	Aggregate	Removed	Free	New question	Available

C Framework Ablations

C.1 Intervention Protocols and Evaluated Detectors

The framework study changes how the agent organizes experiments within the same three-tier search space. Full and all six controls use DINOv3-H, Qwen3.8-27B, OpenFake 20K, seed 0, and separate archives initialized from the frozen-feature reference. Each run allocates 240 proposals across six streams (one T0, two T1, and three T2); execution and token budgets are specified in Appendix B.4. Table 14 summarizes the information and constraints available to each proposer. Generic archive search is the code-and-score archive control implemented within this framework.

The executed no-protocol prompts remove research cards and also permit multi-component edits without sustained-question requirements, while the scheduler still supplies parent assignments. This arm tests that combined protocol change. Aggregate feedback preserves the experiment protocol but removes condition scores from prompts and retrievable records. Across the run, it has 155 missing-delivery outcomes and 158 repair-mode sessions. Its final valid score arrives at completion 93, leaving the later curve flat.

Candidate identity and evaluation. Table 15 links the detectors in Table 2 to the search trajectories. Each control's detector was selected from its development records and evaluated on all twelve benchmarks; its J and bACC are the reported results for that arm. The curve snapshot contains 240 outcomes per arm except Unconstrained, which has 232. These counts refer to completed scores or failures; its separate 235-session trace inventory also includes sessions whose scoring falls outside that snapshot.

Public frameworks. OpenEvolve (Sharma, 2025) and ShinkaEvolve (Lange et al., 2026) replace ForenEvolve's search loop while keeping the backbone, proposer, training data, three-tier program interface, evaluator, objective, and 240-proposal budget. Each reported detector is the framework's highest-J valid program, evaluated on the twelve benchmarks under the same protocol (Tables 15 and 17).

C.2 Search Progress and Benchmark Results

The trajectories separate finding a useful design from refining it further (Figure 5). Generic reaches $J = 0.9495$ at completion seven and 0.9577 at completion 75, then retains that incumbent through completion 209. Full crosses 0.96 at completion 51. Across the run, 31 scored proposals exceed this level, spanning 23 distinct program fingerprints. Table 16(a) summarizes this progression at common completion counts.

Full obtains its higher objectives from fewer valid evaluations than several controls: it scores 156 proposals, compared with 179 for Generic, 198 for Independent, and 202 for Restarts. Independent retains its early incumbent from completion 32 through 239 despite these additional evaluations.

Table 15: **Evaluated programs and search endpoints.** Candidate identifiers omit the common `cand_` prefix.

Strategy	Evaluated program		Best observed program		Outcomes
	Candidate	J	Candidate	J	
Full (main)	i3035_s1_0	0.9644	i3035_s1_0	0.9644	240
Aggregate feedback	i3007_s2_0	0.9592	i3007_s2_0	0.9592	240
No hypothesis protocol	i3012_s1_0	0.9493	i3012_s1_0	0.9493	240
Unconstrained revision	i3013_s1_0	0.9569	i3036_s2_0	0.9585	232
Archive restarts	i3029_s1_0	0.9541	i3039_s0_0	0.9542	240
Independent proposals	i3040_s0_0	0.9507	i3040_s0_0	0.9507	240
Generic archive search	i3013_s2_0	0.9577	i3035_s0_0	0.9585	240
OpenEvolve	oe_0010_35b9b3aa	0.9511	oe_0010_35b9b3aa	0.9511	240
ShinkaEvolve	sk_0010_5fd65eb9	0.9521	sk_0010_5fd65eb9	0.9521	240

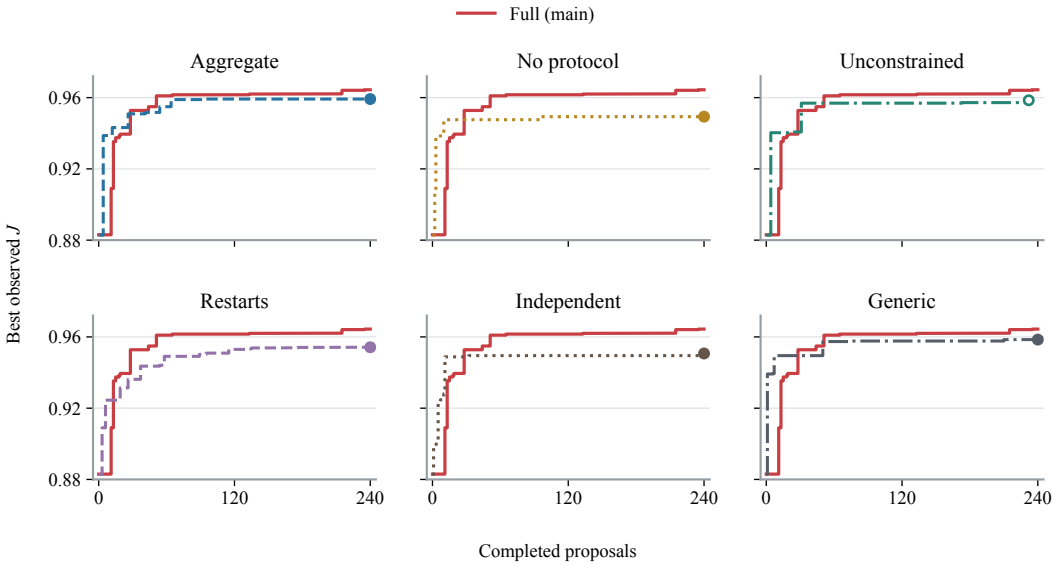


Figure 5: **Framework search trajectories.** Best observed J by completed proposals, including failures. Each panel compares Full with one control; the open endpoint marks the available Unconstrained snapshot.

Together, the counts and trajectories locate the difference in the quality of subsequent designs once a promising program has been found.

The per-benchmark results show where the evaluated designs separate (Table 17). Full's mean advantage is larger on L2 than on L1 for every control. Within L2, all arms are near ceiling on DRCT-2M, where several controls slightly exceed Full; the larger differences occur on DailyBench-Fake, where Full leads every control. This benchmark contributes the largest L2 gap for five controls; for Generic, the largest gap occurs on RealChain.

C.3 How Experimental Evidence Guides Revisions

Historical code is used extensively by both Full and Generic (Table 16(b)). Generic reads previously scored programs in 229 sessions and Full in 220; a read is counted only when a successful tool call occurs after the source program's measurement. Generic's plateau therefore coexists with substantial access to prior work. At local round 13 in stream 2 (R13/s2), it combines three scale views, per-view heads, and a margin loss to reach $J = 0.9577$. Its later R35/s0 revises the adaptation loss and downweights view scores far from their median, reaching 0.9585. The code changes document active reuse, while the curve records how far those revisions advance the search.

Table 16: **Search progress and historical program access.** J_n is the best objective after n completed proposals, including failures. Scored counts valid objectives; initialization is excluded from counts. Panel (b) counts sessions that successfully read a program after it was scored; cross-stream reads are a subset of historical reads. Both panels use the curve cohort.

(a) Progress in measurement-completion order

Strategy	Outcomes	Scored	J_{50}	J_{100}	J_{200}	Best J	# $J > 0.96$
Full	240	156	0.9549	0.9616	0.9620	0.9644	31
Aggregate	240	74	0.9517	0.9592	0.9592	0.9592	0
No protocol	240	162	0.9476	0.9493	0.9493	0.9493	0
Unconstrained	232	168	0.9569	0.9569	0.9572	0.9585	0
Restarts	240	202	0.9436	0.9509	0.9541	0.9542	0
Independent	240	198	0.9495	0.9495	0.9495	0.9507	0
Generic	240	179	0.9574	0.9577	0.9577	0.9585	0

(b) Access to previously measured programs

Strategy	Sessions	Historical reads	Cross-stream reads
Full	240	220	122
Aggregate	240	208	77
No protocol	240	229	190
Unconstrained	232	222	143
Restarts	240	227	185
Independent	240	0	0
Generic	240	229	224

Table 17: **Framework ablations: individual benchmark results.** Pooled bACC (%) for the evaluated programs in Table 2.

	L0					L1					L2					All
Strategy	*AIGI-Now*	*Chameleon*	*SynthWildX*	*AIGIBench*	***Mean***	*GenImage*	*Synthbuster*	*AIGCDetect Bench*	*WildRF*	***Mean***	*Community Forensics*	*DRCT-2M*	*RealChain*	*DailyBench-Fake*	***Mean***	**Mean**
Full (main)	94.73	94.87	94.37	88.21	**93.05**	97.57	99.72	96.42	98.40	**98.03**	97.09	99.26	84.91	86.52	**91.95**	**94.34**
Aggregate feedback	93.39	93.74	92.50	87.92	**91.89**	96.99	99.62	95.36	96.64	**97.15**	95.90	99.38	85.04	74.97	**88.82**	**92.62**
No hypothesis protocol	94.16	89.54	91.54	85.96	**90.30**	96.75	99.20	95.32	96.16	**96.86**	96.21	99.30	82.04	74.80	**88.09**	**91.75**
Unconstrained revision	91.21	93.68	93.56	90.92	**92.34**	98.03	99.60	96.51	96.40	**97.64**	96.00	99.67	83.49	71.63	**87.70**	**92.56**
Archive restarts	90.87	93.72	92.91	86.82	**91.08**	97.31	99.47	95.88	97.08	**97.44**	95.71	99.59	84.14	71.43	**87.72**	**92.08**
Independent proposals	88.11	94.00	93.82	87.45	**90.84**	97.42	99.34	95.29	97.20	**97.31**	95.78	99.59	83.70	72.73	**87.95**	**92.04**
Generic archive search	93.94	92.75	94.24	86.66	**91.90**	97.44	99.66	96.28	97.56	**97.74**	96.56	98.94	82.24	84.46	**90.55**	**93.39**
OpenEvolve	90.04	93.51	92.69	87.08	**90.83**	96.87	99.33	94.92	96.84	**96.99**	95.03	99.47	85.46	71.97	**87.99**	**91.94**
ShinkaEvolve	90.85	93.46	92.96	87.07	**91.09**	97.02	99.42	95.48	96.80	**97.18**	95.62	99.49	85.78	73.41	**88.58**	**92.28**

A transferred view exposes a measurable trade-off. Full's R5/s1 reads its measured parent R4/s1 and the already scored R21/s0 from another stream. It proposes adding R21/s0's `resize224` view to the parent's inverse-variance fusion, predicting better source scores while preserving `d5` and processing retention. Source comparison confirms that only the `views` attribute changes. The resulting J increases by 0.0070: all four source scores improve, raising their mean from 0.9576 to 0.9682, while the processing mean falls from 0.9421 to 0.9346. The targeted `d5` score barely changes (0.8690 to 0.8692), and processing retention improves; both meet the card's targets despite the lower absolute processing mean. The comparison identifies the source gain and its processing cost. In R6/s1, the agent cites the persistent `d5` floor across readout variants and tests greater adaptation capacity with rank-16 attention-and-MLP updates. That attempt times out; R7/s1 restores the measured adaptation plan and recovers $J = 0.9615$. Appendix E.2 follows the later resampling and loss comparisons that lead to the final detector.

Table 18: **AUROC results by feedback level.** Pooled AUROC (%, ↑), with the same methods, benchmark order, and feedback levels as Table 1. Means equally weight the four benchmarks in each level or all 12 overall, and require all constituent scores. **Bold** and underline mark the best and second-best scores among published detectors and ForenEvolve in each column, including ties; * denotes unavailable AUROC results.

	L0 Anonymous per-benchmark feedback					L1 Aggregate-metric feedback					L2 Search-excluded					All
Method	AIGI-Now	Chameleon	SynthWildX	AIGIBench	**Mean L0**	GenImage	Synthbuster	AIGCDetect Bench	WildRF	**Mean L1**	Community Forensics	DRCT-2M	RealChain	DailyBench-Fake	**Mean L2**	**Overall Mean**
Dedicated image detectors																
NPR (Tan et al., 2024b)	45.32	51.98	49.85	70.87	54.51	44.61	51.44	91.68	67.38	63.78	96.82	14.95	51.82	58.22	55.45	57.91
UnivFD (Ojha et al., 2023)	64.30	55.37	47.99	66.89	58.64	83.51	79.68	91.55	59.32	78.52	79.54	87.09	39.74	50.46	64.21	67.12
FatFormer (Liu et al., 2024)	48.55	58.61	59.00	72.62	59.70	84.49	52.06	91.29	74.38	75.56	90.91	29.24	38.38	48.00	51.63	62.29
AIDE (Yan et al., 2025a)	93.10	73.60	65.34	82.85	78.72	96.89	58.00	94.62	72.94	80.61	96.46	95.67	71.47	72.29	83.97	81.10
DRCT (Chen et al., 2024)	78.93	71.82	63.18	95.19	77.28	95.19	89.76	88.99	61.05	83.75	87.78	94.13	61.52	60.14	75.89	78.97
DDA (Chen et al., 2025)	85.69	91.94	96.32	91.54	91.37	98.73	96.90	96.12	94.56	96.58	98.08	99.79	74.13	76.63	87.16	91.70
GAPL (Qin et al., 2026)	92.69	82.56	93.90	94.77	90.98	99.55	97.01	**99.75**	93.84	97.54	99.28	99.79	71.34	78.33	87.19	91.90
DGS-Net (Yan et al., 2026)	75.28	43.84	56.83	88.17	66.03	**99.88**	59.50	99.72	78.28	84.35	97.02	64.29	44.33	54.54	65.05	71.81
BCNet (Yang et al., 2026)	88.43	91.52	97.33	95.79	93.27	99.17	98.97	98.80	99.65	99.15	98.33	99.80	68.78	81.46	87.09	93.17
Vision-language detectors																
OmniVL-Guard-2B (Shen et al., 2026)	95.21	85.51	82.71	74.46	84.47	96.50	89.41	86.32	86.49	89.68	96.56	86.88	71.19	44.96	74.90	83.02
UniGenDet (Zhang et al., 2026d)	98.08	91.68	89.49	85.16	91.10	98.47	99.31	96.23	88.84	95.71	98.96	91.02	84.36	54.07	82.10	89.64
FakeVLM (Wen et al., 2025)	91.73	94.31	88.49	88.47	90.75	99.64	91.48	95.99	85.18	93.07	95.60	87.46	77.81	70.90	82.94	88.92
Same backbone and training data (DINOv3-H, OpenFake 20K)																
Frozen-feature reference	88.46	92.47	95.54	90.48	91.74	94.33	99.35	92.50	98.12	96.08	90.37	96.81	84.19	81.49	88.22	92.01
Standard BCE adaptation	95.18	95.08	97.14	94.06	95.37	97.99	99.89	97.96	98.75	98.65	98.97	99.51	86.81	87.94	93.31	95.77
SAFE† (Li et al., 2025a)	94.01	96.17	96.75	93.32	95.06	96.70	99.50	96.72	98.96	97.97	95.25	99.33	88.93	92.48	94.00	95.68
Effort† (Yan et al., 2025b)	94.50	97.42	97.34	95.01	96.07	97.53	99.74	96.77	99.06	98.28	97.67	99.64	89.32	91.20	94.46	96.27
DDA-style data† (Chen et al., 2025)	89.39	91.74	96.55	93.50	92.80	99.09	99.49	97.17	97.65	98.35	98.49	99.81	87.68	88.12	93.53	94.89
ForenEvolve	**98.13**	**99.00**	**99.10**	**95.81**	**98.01**	99.22	**100.00**	99.05	**99.90**	**99.54**	**99.41**	**99.94**	**93.77**	**94.53**	**96.91**	**98.16**

The last group uses our runs on DINOv3-H and the 20K OpenFake split, with a linear readout, a training-selected threshold, and the same evaluation; adapted rows train for two epochs. †Core recipe transplanted onto this setting (Appendix D.2). Only ForenEvolve from this group enters the bold and underline ranking.

Independent proposals cannot build on their early incumbent. Independent's R4/s2 combines training-image processing, class-conditional feature regularization, and a multiview readout, reaching $J = 0.9495$ at completion 32. Its next improvement arrives at completion 240, when R40/s0 reaches 0.9507. That final proposal still frames its hypothesis against the frozen reference: neither the earlier strong program nor its measurements are available in the context. The run repeatedly constructs designs from the initial resources, leaving the promising intermediate detector without a feedback-guided refinement path.

An archive also supports voluntary comparisons. Restarts' R39/s0 chooses the earlier R29/s1 as a control and changes only the resampling-kernel set, raising J from 0.9541 to 0.9542. Unconstrained's R36/s2 returns to R20/s2's better training recipe and adds a difference between half-scale and NPR views to its readout, reaching 0.9585. Relative to its assigned parent, both training and readout change; relative to the retrieved R20/s2, only the readout changes. These cases show why the reference program matters when interpreting an edit. Full makes such comparisons part of scheduled experimental continuation, linking the chosen reference, predicted effect, and measured trade-off for the next revision.

D Additional Experimental Results

D.1 Additional Detection Results

The gains in bACC are accompanied by better discrimination across decision thresholds. The round-35 detector raises mean AUROC from 95.77% with standard BCE adaptation to 98.16%, improving on that reference on all twelve benchmarks (Table 18). The comparison uses the same benchmark groups as the main bACC table.

Generator and source differences. Tables 32–43 show where benchmark averages conceal differences between generators or source subsets. On DRCT-2M, ForenEvolve exceeds 98% bACC on all thirteen evaluated generators. RealChain has a wider spread: Flux1 reaches 97.81%, Seedream4 reaches 71.76%, and the `i2i` subset reaches 64.71%. DailyBench-Fake spans 83.47–89.72% across its seven generators, including FLUX2, GPT-Image2, and Qwen-Image-2512. These breakdowns locate the remaining detection gaps within the search-excluded benchmarks. The complete breakdown and metric conventions appear in Appendix G.

D.2 Same-Backbone Recipe Baselines

The last group of Table 1 separates the contribution of search from that of the backbone and training data. Every row in this group uses DINOv3-H+/16 (about 841M parameters, pooled output as the feature) and the 20K OpenFake split used by the main search: 10,000 real images from 14 sources and 10,000 generated images from 55 generators. Adapted rows train for two epochs with batch size 32, AdamW (weight decay 0.01), bf16 autocast, and 224-pixel center crops at native resolution, all with seed 0. After adaptation, the baselines re-extract features and fit an ℓ_2-normalized raw-view readout by logistic regression ($C = 1$); ForenEvolve uses its discovered three-view readout. Every row freezes the threshold that maximizes training bACC. Evaluation covers all 848,940 eligible images of the 12 benchmarks with the preprocessing of the main comparison.

Each baseline transplants the core practice of a published recipe onto this shared setting (Table 19). Holding the backbone, data, budget, readout, and evaluation fixed isolates the practice itself; the table lists what each row keeps and what it replaces. AIDE's two-branch architecture does not fit a shared-backbone adapter, so SAFE represents transformation-based training.

Table 19: **Transplanted recipes.** All rows share the setting described in the text. Time is the cumulative wall-clock time of two training epochs.

Recipe	Transplanted practice	Trainable params	Time (s)	Replaced
Standard BCE	LoRA ($r = 8$, $\alpha = 16$) on query, key, value, and output projections; learning rate 10^{-4}; BCE loss.	2.62M	404	–
SAFE (Li et al., 2025a)	Official training transformations applied to the full image before cropping: horizontal flip ($p = 0.5$); rotation in $[-180°, 180°]$ with black fill; brightness, contrast, and saturation jitter in $[0.5, 1.5]$; 16×16 patch masking ($p = 0.5$, ratio in $[0, 0.75]$). LoRA as in standard BCE.	2.62M	1,827	ResNet-50 with wavelet high-frequency input → DINOv3-H RGB; 20 → 2 epochs; random → center crop.
Effort (Yan et al., 2025b)	SVD of the 128 attention projections; all but the smallest singular component stay frozen; biases frozen; learning rate 2×10^{-4} as released; BCE loss.	0.33M	840	CLIP ViT-L/14 → DINOv3-H; 10 → 2 epochs; orthogonality and singular-value terms follow the released training code, which disables them.
DDA-style data (Chen et al., 2025)	Each real training image receives an SD2.1-VAE reconstruction twin in the same batch; twins receive JPEG requantization ($p = 0.5$) and pixel and frequency blending ($p = 0.2$ each, ratio in $[0, 0.8]$). LoRA as in standard BCE.	2.62M	1,353	COCO-only training → twins added to OpenFake 20K (real:generated 1:2); DINOv2-L, 336-pixel crops, LoRA $\alpha = 1$, one epoch, and paired rescaling not adopted.

The discovered detector leads the standard BCE row and every transplanted recipe on all 12 benchmarks in both bACC and AUROC (Tables 1 and 18). Its smallest margins over Effort, the strongest recipe, are 0.30 bACC points on AIGIBench and 0.26 AUROC points on Synthbuster. Relative to standard BCE, Effort is higher on 8 of the 12 benchmarks in bACC, SAFE on 4, and DDA-style data on 1. A paired bootstrap confirms the overall differences (Table 20): within each benchmark, generator groups and real-image sources are resampled with replacement within class, over 1,000 draws on the 848,940 matched rows.

Table 20: **Paired bootstrap against transplanted recipes.** Differences are ForenEvolve minus the recipe in 12-benchmark mean score (points), with 95% intervals; every one of the 1,000 draws is positive for each recipe and metric.

Recipe	ΔbACC [95% CI]	ΔAUROC [95% CI]
Effort	+2.62 [2.17, 3.03]	+1.89 [1.50, 2.31]
SAFE	+4.34 [3.92, 4.76]	+2.48 [1.98, 3.00]
DDA-style data	+6.25 [5.22, 7.22]	+3.26 [2.60, 4.00]

Implementation notes. The Effort row computes the trainable singular component as a separate branch and is exported as an equivalent rank-2 LoRA (maximum error 1.19×10^{-7}) to share the evaluation pipeline; an earlier variant that merged it into bf16 weights gives 12-benchmark means within 0.03 points. The DDA-style row measures aligned twins added to our training set, which shifts the class ratio to 1:2 and the training-selected threshold to $+1.17$, compared with -0.70 to -0.32 for the other recipes. Recipe features are extracted with TF32 matrix multiplication and the other rows with FP32; on the standard BCE adapter, TF32 changes bACC by $+0.08$ and AUROC by $+0.001$ points.

D.3 Repeated Full Searches

The main configuration was run three times with seeds 0, 1, and 2. The runs share the backbone, proposer, training data, six proposal streams, 40 rounds, objective, and evaluation protocol of Section 4.1; seed 0 is the main run reported in the main text. Each run selects its detector by the development objective J and applies the threshold selected on training scores, so L2 enters neither search nor selection in any run. Tables 21 and 22 report the selected programs and their full-coverage bACC.

All three searches raise J from 0.8830 to 0.9626–0.9658 and select detectors with overall bACC of 93.87–94.34% (94.10 ± 0.23) and L2 bACC of 91.23–91.95% (91.66 ± 0.38). Every run exceeds the strongest published detector by 8.65–9.12 points overall and by 9.15–9.87 points on L2, and ranks first on 8–10 of the 12 benchmarks, including at least three of the four L2 benchmarks. On the same backbone and training data, every run exceeds standard BCE adaptation on all 12 benchmarks, by 3.67–4.14 points overall, and exceeds Effort by 2.15–2.62 points overall; the only benchmark on which a run trails Effort is RealChain for seed 1 (83.32% vs. 83.66%). The variation across searches is thus an order of magnitude smaller than the margin over the strongest same-backbone recipe.

The runs reach their selected programs through different lineages: seed 0 at round 35 of stream 1, seed 1 at round 21 of stream 1, and seed 2 at round 28 of stream 2. Seed 2 attains the highest J (0.9658) and seed 0 the highest overall bACC; the three detectors differ by at most 0.47 points overall, so selection by J yields comparable detectors across runs.

Table 21: **Repeated full searches.** Program Rr/sk is the candidate of round r in proposal stream k. Scored counts proposals with a valid objective. bACC (%) uses full-coverage evaluation; same-backbone rows are from Table 1.

Run	Program	Scored	J	L0	L1	L2	All
Seed 0 (main)	R35/s1	156/240	0.9644	93.05	98.03	91.95	94.34
Seed 1	R21/s1	138/240	0.9626	92.63	97.76	91.23	93.87
Seed 2	R28/s2	157/240	0.9658	92.61	97.79	91.82	94.07
Mean ± s.d.	–	–	0.9643 ± 0.0016	92.76 ± 0.25	97.86 ± 0.15	91.66 ± 0.38	94.10 ± 0.23
Standard BCE adaptation	–	–	–	87.16	95.99	87.44	90.20
Effort[†]	–	–	–	90.29	95.73	89.13	91.72

Table 22: **Benchmark bACC (%) of repeated full searches.** The last two columns repeat the same-backbone rows of Table 1.

Level	Benchmark	Seed 0	Seed 1	Seed 2	Mean ± s.d.	BCE	Effort†
L0	AIGI-Now	94.73	94.64	94.88	94.75 ± 0.12	87.28	89.60
L0	Chameleon	94.87	93.01	93.09	93.66 ± 1.05	84.74	92.31
L0	SynthWildX	94.37	94.55	93.54	94.15 ± 0.54	90.19	91.35
L0	AIGIBench	88.21	88.31	88.94	88.49 ± 0.40	86.43	87.91
L1	GenImage	97.57	97.29	96.91	97.26 ± 0.33	95.92	95.35
L1	Synthbuster	99.72	99.68	99.24	99.55 ± 0.27	98.82	98.73
L1	AIGCDetectBench	96.42	96.68	96.46	96.52 ± 0.14	94.16	92.66
L1	WildRF	98.40	97.40	98.56	98.12 ± 0.63	95.05	96.16
L2	Community Forensics	97.09	96.64	97.42	97.05 ± 0.39	95.21	94.19
L2	DRCT-2M	99.26	99.13	99.32	99.24 ± 0.10	98.78	98.93
L2	RealChain	84.91	83.32	85.20	84.48 ± 1.01	78.83	83.66
L2	DailyBench-Fake	86.52	85.82	85.33	85.89 ± 0.60	76.95	79.73

Table 23: **Capability search: individual benchmark results.** Pooled bACC (%, ↑) after 30 proposals; row labels specify the two streams.

	L0					L1					L2					All
Setting	*AIGI-Now*	*Chameleon*	*SynthWildX*	*AIGIBench*	**Mean**	*GenImage*	*Synthbuster*	*AIGCDetect Bench*	*WildRF*	**Mean**	*Community Forensics*	*DRCT-2M*	*RealChain*	*DailyBench-Fake*	**Mean**	**Mean**
Frozen reference	77.51	82.82	87.78	80.57	**82.17**	88.33	97.78	84.18	92.85	**90.79**	82.76	89.28	76.44	70.67	**79.79**	**84.25**
T0, T0	82.12	88.46	92.08	82.69	**86.34**	90.34	98.51	87.23	94.92	**92.75**	86.00	90.99	78.82	74.60	**82.60**	**87.23**
T0, T1	81.81	86.28	92.09	81.77	**85.49**	89.20	98.79	85.76	95.84	**92.40**	84.52	88.65	79.52	75.71	**82.10**	**86.66**
T2, T2	91.45	94.00	91.37	88.15	**91.24**	96.73	99.46	95.00	96.24	**96.86**	95.03	99.34	80.63	80.86	**88.97**	**92.36**

D.4 Per-Benchmark Capability and Configuration Results

Tables 23–25 expand the main-text means into pooled bACC for each benchmark, using the configurations in Appendix B.3. Community Forensics contains 51,636 evaluated images in the local runs and 51,836 in the proposer and training-source runs; each score uses its run's exported cohort.

Where adaptation helps. At 30 proposals, T2 search exceeds T0+T1 on eleven of the twelve benchmarks; SynthWildX is the exception. The advantage therefore extends across sources. With the full search budget, all three DINO backbones improve over their frozen references on every benchmark. CLIP-L and ResNet-50 improve on eleven and ten, respectively, while both lose accuracy on DailyBench-Fake and ResNet-50 also declines on Synthbuster. Higher level means thus coexist with source-specific weaknesses in the discovered programs.

Where configurations differ. Changing the proposer or training source also changes the distribution of gains. DeepSeek slightly exceeds the main run on L2, helped by RealChain, while the main run leads overall. GenImage training yields the highest L1 mean, but its lower AIGI-Now and RealChain scores leave OpenFake training stronger overall. The tables retain the main benchmark groups; GenImage and Community Forensics serve as training-source evaluations in their respective training runs.

D.5 Search Trajectories across Models

The configuration searches reach their strongest programs through different design paths (Figure 6 and Table 26). DINOv3-L reaches its best objective at local index 7, whereas CLIP-L improves through index 40. The main Qwen run combines training resampling with patch geometry; Appendix E.2 follows that lineage. The comparisons below show how other proposers and backbones use measured alternatives, with the resulting recipes collected in Appendix E.3.

Table 24: **Backbone initialization and discovered detectors.** Pooled bACC (%, ↑) before and after each 240-proposal search.

	L0					L1					L2					All
Setting	*AIGI-Now*	*Chameleon*	*SynthWildX*	*AIGIBench*	**Mean**	*GenImage*	*Synthbuster*	*AIGCDetect Bench*	*WildRF*	**Mean**	*Community Forensics*	*DRCT-2M*	*RealChain*	*DailyBench-Fake*	**Mean**	**Mean**
DINOv3-H, initial	77.51	82.82	87.78	80.57	**82.17**	88.33	97.78	84.18	92.85	**90.79**	82.76	89.28	76.44	70.67	**79.79**	**84.25**
DINOv3-H, discovered	94.73	94.87	94.37	88.21	**93.05**	97.57	99.72	96.42	98.40	**98.03**	97.09	99.26	84.91	86.52	**91.95**	**94.34**
DINOv3-L, initial	75.73	79.37	87.63	80.23	**80.74**	77.71	93.85	74.62	90.69	**84.22**	75.46	82.95	74.78	66.10	**74.82**	**79.93**
DINOv3-L, discovered	93.35	89.42	90.18	86.53	**89.87**	92.02	98.67	91.23	96.00	**94.48**	94.35	96.82	80.24	74.54	**86.49**	**90.28**
DINOv2-L, initial	61.59	61.02	76.93	72.16	**67.93**	55.74	81.13	56.22	81.42	**68.63**	62.36	56.53	69.55	55.44	**60.97**	**65.84**
DINOv2-L, discovered	87.59	76.14	84.87	80.21	**82.20**	78.77	92.09	74.24	89.61	**83.68**	86.02	91.65	77.70	74.12	**82.37**	**82.75**
CLIP-L, initial	66.39	56.62	74.66	69.95	**66.91**	64.19	80.24	62.62	81.06	**72.03**	71.17	74.26	66.54	59.03	**67.75**	**68.89**
CLIP-L, discovered	85.84	75.07	82.51	86.84	**82.57**	80.25	89.68	77.00	87.98	**83.73**	84.93	88.43	79.83	54.95	**77.04**	**81.11**
ResNet-50, initial	53.86	62.39	73.44	62.23	**62.98**	52.96	75.14	53.28	73.99	**63.84**	58.52	49.50	67.16	58.31	**58.37**	**61.73**
ResNet-50, discovered	75.51	84.61	76.71	67.24	**76.02**	76.07	70.45	66.30	78.99	**72.95**	73.41	85.35	72.46	55.49	**71.68**	**73.55**

Table 25: **Proposer and training-source results.** Pooled bACC (%, ↑). The reference uses Qwen3.8-27B and OpenFake.

	L0					L1					L2					All
Setting	*AIGI-Now*	*Chameleon*	*SynthWildX*	*AIGIBench*	**Mean**	*GenImage*	*Synthbuster*	*AIGCDetect Bench*	*WildRF*	**Mean**	*Community Forensics*	*DRCT-2M*	*RealChain*	*DailyBench-Fake*	**Mean**	**Mean**
Main reference	94.73	94.87	94.37	88.21	**93.05**	97.57	99.72	96.42	98.40	**98.03**	97.09	99.26	84.91	86.52	**91.95**	**94.34**
DeepSeek-Flash	93.56	93.68	93.66	87.17	**92.02**	97.50	99.54	96.68	96.72	**97.61**	96.30	99.47	87.24	85.10	**92.03**	**93.89**
GPT-5.6-Terra	91.21	93.02	93.75	87.91	**91.47**	97.66	99.54	95.87	97.04	**97.53**	96.38	99.22	81.81	80.63	**89.51**	**92.84**
GenImage 20K	80.29	93.29	95.25	90.18	**89.75**	99.29	98.96	98.62	96.56	**98.36**	96.12	99.92	78.14	84.54	**89.68**	**92.60**
Community Forensics 20K	86.60	84.76	88.93	84.97	**86.31**	97.61	99.46	92.62	94.00	**95.92**	95.10	99.05	84.04	72.45	**87.66**	**89.97**

Transferring a measured adaptation configuration. DeepSeek first develops a three-view detector with JPEG, resampling, and blur during training. Widening the resampling range leaves J unchanged at 0.9555, prompting a test of an adaptation configuration already measured in another stream. This transfer changes LoRA rank and scaling, target modules, dropout, and training duration while preserving the views, image operation, loss, and readout. It raises J to 0.9581 and the processing mean from 0.9259 to 0.9352. The experiment develops the existing augmentation recipe by changing how the backbone learns from it.

Revisiting a loss construction. Terra's research card cites an earlier program using inclusive class compactness. Restoring that construction in place of leave-one-out compactness changes only the adaptation-loss method: J rises from 0.9538 to 0.9548 and source score `g1` from 0.9375 to 0.9428, while the processing mean stays nearly unchanged (0.9319 to 0.9317). The result ties the cited program and locates the recovered benefit in source performance.

Combining designs on DINO backbones. DINOv3-L retains a two-view degradation-trained recipe and adds patch supervision with a revised consistency loss. This exploratory combination raises J from 0.9362 to 0.9410 and the joint-processing score `d5` from 0.8257 to 0.8407. DINOv2-L instead combines a patch-norm regularizer from a linear-readout program with a three-view MLP, using a shorter readout fit. Its J reaches 0.8916, above both source programs (0.8878 and 0.8859). In both cases, measured programs supply complementary designs for a subsequent combination.

Separating source gains from processing gains. CLIP-L and ResNet-50 simplify expensive training operations to obtain scored revisions. CLIP-L adds a cheaper resampling branch to its

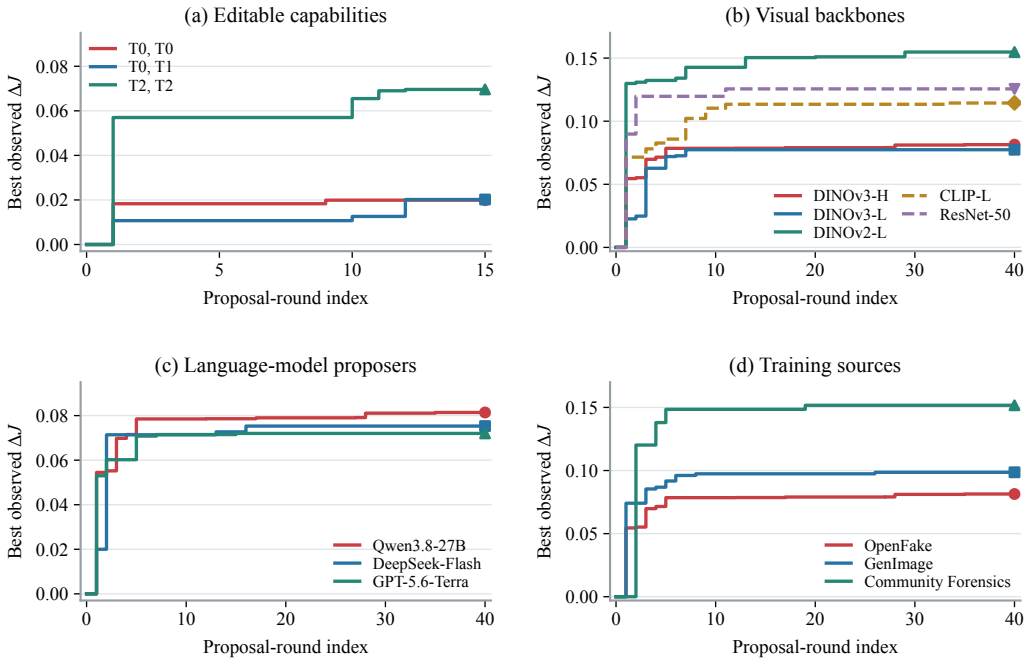


Figure 6: **Search across capabilities, models, and data sources.** Best observed ΔJ over each initialization, aggregated through each local proposal index across streams, with failed proposals retaining their positions. Streams run asynchronously, so index order differs from the completion order used in Figure 3.

Table 26: **Trajectory profiles across proposers and backbones.** Scored proposals count new accepted or rejected evaluations out of all recorded proposal outcomes, excluding initialization. The selected program is identified by local round/stream.

Configuration	Scored proposals	Best J ↑	Selected program
DINOv3-H / Qwen	156/240	0.9644	R35/s1
DINOv3-L / Qwen	171/240	0.9410	R7/s1
DINOv2-L / Qwen	143/240	0.8916	R29/s1
CLIP-L / Qwen	110/240	0.8767	R40/s0
ResNet-50 / Qwen	94/240	0.8243	R11/s2
DINOv3-H / DeepSeek	86/188	0.9581	R16/s1
DINOv3-H / Terra	166/210	0.9548	R15/s2

JPEG recipe, raising J from 0.8613 to 0.8767. Source score `g3` rises from 0.7624 to 0.8174, but `d5` falls from 0.7065 to 0.6819, contrary to the proposal's prediction. Compared with an earlier program without a custom training-image operation, ResNet-50's JPEG-only recipe has higher J (0.8243 versus 0.8184) and a lower processing mean (0.8686 versus 0.8726). These breakdowns reveal the source–processing tradeoffs behind an improved aggregate objective.

D.6 Robustness to Image Processing

We evaluate JPEG recompression, resizing, and Gaussian blur, following the three perturbation families studied by DDA (Chen et al., 2025). The six benchmarks cover each feedback level: AIGI-Now and Chameleon (L0), GenImage and Synthbuster (L1), and DRCT-2M and DailyBench-Fake (L2). We compare ForenEvolve with NPR, UnivFD, DRCT, GAPL, and BCNet.

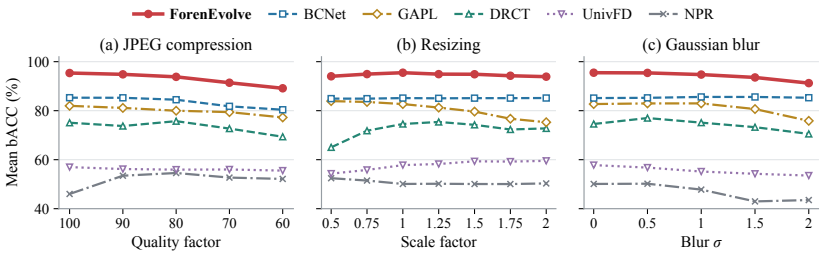


Figure 7: **Robustness to image processing.** Mean bACC over six benchmarks, equally weighted, under JPEG compression, resizing, and Gaussian blur.

Table 27: **ForenEvolve's degradation sensitivity.** Clean bACC is in percent; drops are in percentage points. Resize reports each benchmark's largest drop over nonidentity scales. The last row averages the six benchmark rows.

Level	Benchmark	Clean bACC	JPEG 60 Drop	Resize Max. drop	Blur 2 Drop
L0	AIGI-Now	94.73	6.26	3.47	3.24
L0	Chameleon	94.88	5.64	1.94	6.98
L1	GenImage	98.23	13.77	2.28	2.91
L1	Synthbuster	99.72	4.51	0.42	0.11
L2	DRCT-2M	99.26	1.51	1.39	2.11
L2	DailyBench-Fake	86.17	6.46	7.59	10.13
Six-benchmark mean		95.50	6.36	2.85	4.25

Mean results. Search feedback contains mild versions of these operations on a fixed pool of 1,640 images (JPEG quality 75–95, resizing by 0.70–1.15, and blur $\sigma = 0.6$ within a blur–noise–JPEG composite); this study sweeps stronger settings on new images. ForenEvolve achieves the highest mean bACC at every tested setting (Figure 7). At JPEG quality 60, resize factor 2, and blur $\sigma = 2$, all outside the feedback range, it reaches 89.14%, 93.88%, and 91.25%, exceeding BCNet by 8.76, 8.71, and 5.96 percentage points. The detector's robustness thus extends beyond the severities used as feedback.

Conditions and aggregation. JPEG quality factors are $q \in \{100, 90, 80, 70, 60\}$. Resizing uses factors $r \in \{0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0\}$, and Gaussian blur uses $\sigma \in \{0, 0.5, 1.0, 1.5, 2.0\}$. The clean reference is unmodified; JPEG quality 100 includes recompression, while scale 1 and blur 0 are identity conditions. Figure 7 weights the six benchmarks equally at each condition. Figures 8–10 show the individual benchmarks using consistent axes and method colors.

All three processing families also occur in search feedback (Table 8). The selected detector's training operation uses bilinear downscaling and restoration at scales $[0.60, 1]$ with probability 0.70. The final tests sweep each family separately to measure sensitivity to its severity; the resampling trials that led to the training operation are reconstructed in Appendix E.2.

Table 27 complements the absolute-score curves with clean bACC minus treated bACC, in percentage points. It reports JPEG quality 60, blur $\sigma = 2$, and each benchmark's worst nonidentity resize scale. Its resize mean averages these per-benchmark worst cases, whereas Figure 7 averages benchmarks at a common scale.

Variation across benchmarks. The most sensitive benchmark depends on the operation. GenImage has the largest JPEG-60 drop (13.77 points), whereas DailyBench-Fake loses most under blur at $\sigma = 2$ (10.13 points) and has the largest worst-case resize drop. Synthbuster remains above 99% bACC at every tested resize and blur setting. Thus, the aggregate curves combine substantial robustness on some sources with concentrated weaknesses on others; the benchmark breakdowns identify which source–operation pairs remain difficult.

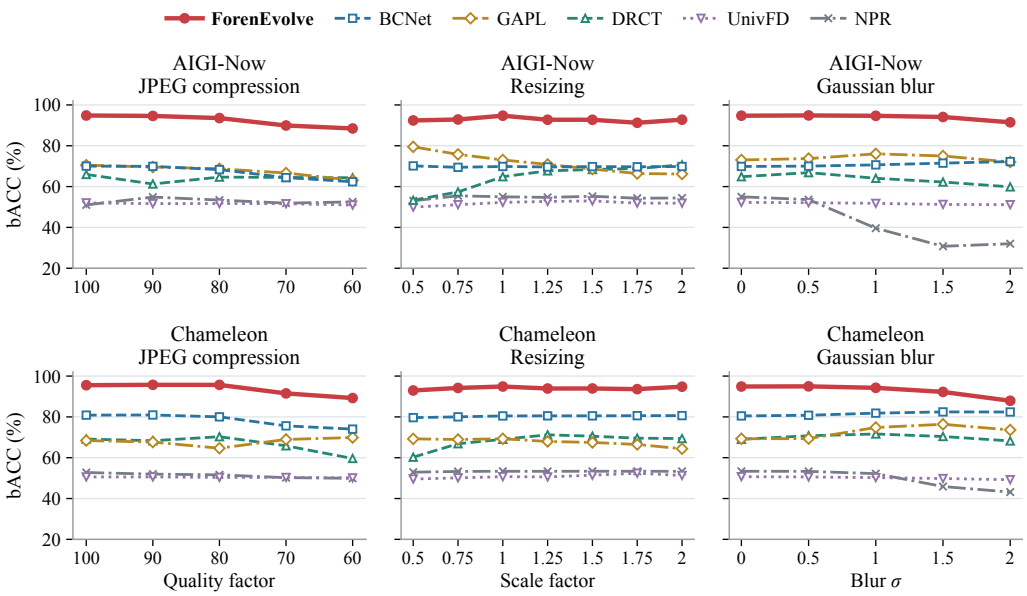


Figure 8: **Robustness on L0 benchmarks.** bACC on AIGI-Now and Chameleon under JPEG compression, resizing, and Gaussian blur.

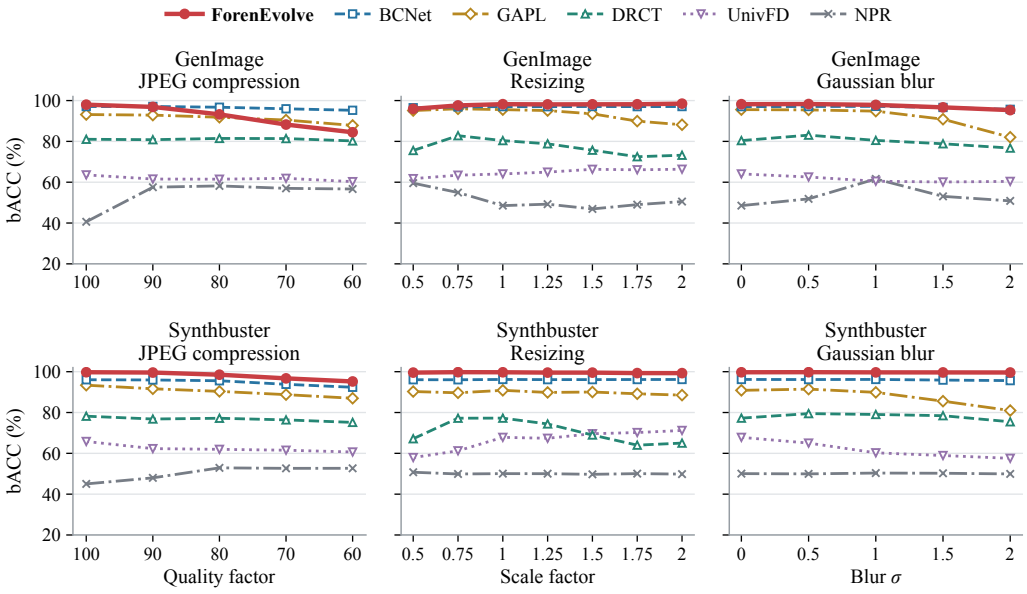


Figure 9: **Robustness on L1 benchmarks.** bACC on GenImage and Synthbuster under JPEG compression, resizing, and Gaussian blur.

Table 28: **Main-run trace coverage.** Initialization is excluded. Read counts refer to sessions with a successful retrieval of a previously measured program. Changes in J use the recorded score precision.

Statistic	Count
Proposal sessions	240
Candidates with a valid objective	156/240
Sessions reading historical programs	220/240
Sessions reading programs from another stream	122/240
Source-matched pairs with objectives	45/47
Improved / unchanged / decreased J among scored pairs	20 / 6 / 19

Table 29: **A prospective patch-geometry test in the GenImage search.** Research-card bounds and measured changes for R26/s2 relative to R24/s2. All changes are in score units.

Quantity	Predicted change	Measured change
J	< 0.002	$+0.0018$
Processing mean	< 0.003	$+0.0116$
`d5`	< 0.005	$+0.0124$
In-domain development AUROC	≥ -0.002	-0.0041

E Experimental Lineages and Discovered Detectors

E.1 Trace Analysis Protocol

We reconstruct experiments from timestamped agent interactions, research cards, program snapshots, and measurement events. A historical read counts when a tool successfully returns a program or observation after the source experiment was measured. We deduplicate repeated reads of the same record within a session. Table 28 summarizes the main run; its program-read counts use the same 240-session denominator as Table 16(b).

Code adoption is checked against the retrieved programs. We compare executable program members after removing comments, docstrings, and source locations; source-matched pairs preserve all members outside the declared component. For exploratory combinations, we identify retained components and donor expressions separately. The resulting links connect what the agent read, what it changed, and the measured outcome. The two cases below use these links to examine a revised hypothesis and a cumulative design.

E.2 Hypothesis Tests and Cumulative Design

A measured tradeoff changes the next test. In the GenImage search, R26/s2 revisits patch geometry at LoRA rank 16, using R24/s2 as its measured reference. The agent expects the patch term to be neutral or harmful for processing robustness. Its revision enables patch features and sets the patch-loss weight to 0.30, while refactoring loss and readout code; the views, training-image operation, and LoRA plan are retained. Table 29 compares the prospective bounds with the measured changes. The gain in J falls within the predicted bound, but the processing gains exceed the agent's expectations and in-domain AUROC falls more than anticipated. The condition breakdown therefore changes the interpretation of a modest objective gain.

The next research card responds to this tradeoff by proposing a lower patch-loss weight. The scored R28/s2 revision changes only `PATCH_W`, from 0.30 to 0.15, relative to R26/s2. In-domain AUROC recovers from 0.9944 to 0.9975, while the processing mean falls from 0.7092 to 0.7004. The recorded question has shifted from whether patch geometry helps to how much of its processing benefit can be retained while recovering ranking quality.

Table 30: **Measurements and sources in the final lineage.** R denotes local round and s denotes stream. C marks a source-matched comparison; E marks exploration. Sources identify the reference program, or the two design donors for R35/s1. Dev. AUROC is the mean across development benchmarks.

Program	Design	Source(s)	Type	J	d5	Dev. AUROC
R26/s1	Pair-loss weight 0.25	R25/s1	C	0.9607	0.8492	0.9936
R27/s1	Resampling floor 0.70	R26/s1	E	0.9621	0.8572	0.9938
R28/s1	Resampling floor 0.60	R27/s1	C	0.9641	0.8604	0.9944
R29/s1	JPEG probability 0.50	R28/s1	E	0.9581	0.8750	0.9911
R30/s1	JPEG probability 0.30	R29/s1	E	0.9587	0.8755	0.9917
R31/s1	JPEG probability 0.15	R30/s1	C	0.9608	0.8707	0.9924
R32/s1	JPEG probability 0.10	R31/s1	E	0.9613	0.8753	0.9929
R33/s1	JPEG probability 0.05	R32/s1	C	0.9615	0.8763	0.9933
R34/s1	JPEG probability 0.02	R33/s1	C	0.9610	0.8742	0.9932
R34/s2	Variance-floor donor	R33/s2	E	0.9614	0.8489	0.9936
R35/s1	Resampling recipe + variance floor	R28/s1, R34/s2	E	0.9644	0.8653	0.9934

A controlled resampling test supplies a reusable design. In the main run, R27/s1 widens the training-resampling interval from $[0.85, 1]$ to $[0.70, 1]$. R28/s1 then reads that program and tests a further extension to $[0.60, 1]$, changing only `RESAMPLE_MIN`. Its card predicts $\texttt{d5} \geq 0.858$ while preserving source ranking (Table 7). The measured $\Delta J = 0.0020$ and $\Delta\texttt{d5} = 0.0032$ support this extension, with mean development AUROC rising from 0.9938 to 0.9944. The archive stores the comparison alongside its prediction.

Processing gains can have a ranking cost. R29–R34/s1 test JPEG probabilities from 0.50 to 0.02 (Table 30). Every tested program raises `d5` above R28/s1, but all have lower J and mean development AUROC. The sequence contains both controlled probability changes and exploratory revisions. Together, the measurements motivate recovering the earlier resampling recipe while seeking a different way to improve processing performance.

Retrieval turns an earlier result into a later component. At R35/s1, the proposer reads both R28/s1 and the variance-floor program R34/s2. The new program retains twenty executable members from R28/s1, including its training-image operation and readout; it revises the adaptation loss and geometry helper and adds a variance-floor attribute. The three assignment expressions that compute the patch variance, floor, and penalty match the donor. This exploratory combination reaches $J = 0.9644$, with higher `d5` and lower mean development AUROC than R28/s1. The program reads and source matches show how designs from separate experiments become components of the final detector.

E.3 Discovered Detector Recipes

Training and inference. The selected main-run program, R35/s1 (`cand_i3035_s1_0`), uses attention LoRA with rank eight, alpha 16, zero dropout, two epochs, learning rate 10^{-4}, batch size 32, and 20,000 training images. For images whose short side is at least 256 pixels, its training operation applies bilinear downscaling and restoration with probability 0.70, drawing the scale uniformly from $[0.60, 1]$. Both labels receive the same operation. At inference, the program normalizes the raw, half-scale, and resize224 feature vectors individually, averages them, and normalizes the result. A linear logistic readout uses inverse regularization strength one and a maximum of 30 fitting iterations. The archived adapter and training-side readout and threshold fitting follow Appendix B.2.

Adaptation objective. The loss combines BCE with constraints on patch variation, class-mean directions, and distances between images. Let $P \in \mathbb{R}^{B\times N\times C}$ contain patch features for B images,

N patches, and C feature channels. The implemented objective is

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + 0.5(\mathcal{L}_{\text{var}} + \mathcal{L}_{\cos}) + 0.25\mathcal{L}_{\text{pair}}, \tag{7}$$

$$\mathcal{L}_{\text{var}} = \frac{1}{BC}\sum_{b,c}[0.5 - \text{Var}_n(P_{bnc})]_+, \tag{8}$$

$$\mathcal{L}_{\cos} = \frac{1}{N}\sum_{n}[1 - \cos(\mu_{\text{fake},n}, \mu_{\text{real},n})]. \tag{9}$$

Here $[a]_+ = \max(a, 0)$, variance uses the population normalization, and $\mu_{\text{fake},n}$ and $\mu_{\text{real},n}$ are class means at patch position n, with class counts clamped to at least one. The variance term penalizes within-image patch variance below 0.5 on the unnormalized features. Minimizing the cosine term aligns the two class-mean directions at each patch position.

The pairwise term acts on normalized mean patch features $z_b = \text{norm}(N^{-1}\sum_n P_{bn})$:

$$\mathcal{L}_{\text{pair}} = \frac{1}{|S|}\sum_{(i,j)\in S}\left(\mathbf{1}_{y_i=y_j}\|z_i - z_j\|^2 + \mathbf{1}_{y_i\neq y_j}[2 - \|z_i - z_j\|^2]_+\right). \tag{10}$$

It contracts same-class image pairs and imposes a squared-distance margin of two on different-class pairs. The set S contains all unordered pairs when there are at most 64, otherwise a subset of 64 drawn with seed 12345. The auxiliary terms return zero for absent patch features or a singleton batch.

Strategies across configurations. Multiview features recur in all seven selected programs, with different fusion and adaptation designs (Table 31). DINOv2-L uses an MLP, DINOv3-L averages standardized per-view linear scores, and the main DINOv3-H program averages normalized feature vectors. The table records the resulting designs; Appendix D.5 describes how their experiments lead to them.

The auxiliary objectives constrain different representations. DINOv3-L uses a patch-to-class softplus margin and fixed-direction consistency; DINOv2-L penalizes relative patch-norm variance with weight 0.1, and CLIP-L weights pooled angular compactness by 0.1. On DINOv3-H, DeepSeek constrains the within-class radius to $[0.35, 0.85]$ and the centroid distance to a margin of 1.0, with total geometry weight 0.1. Terra weights inclusive compactness and centroid separation by 0.1 each, and patch-to-image agreement by 0.15; its detached patch weights use temperature 0.25. These recipes specify representation-dependent choices for fusion and regularization within the common program interfaces.

F LIMITATIONS

Evaluation scope. The evaluation covers public image benchmarks and prescribed processing conditions. Repeated use of fixed development feedback can favor designs adapted to that suite; L2 measures transfer beyond it. Later generator releases and evolving processing distributions would extend this assessment, with an independent evaluation stream maintained as the feedback suite changes.

Search variability and attribution. The main configuration is repeated with three seeds (Appendix D.3); each other configuration is represented by one search trajectory. Repeated runs of the framework controls would quantify their variability from proposals, training, and asynchronous execution. Framework controls compare experimental protocols, some of which change several constraints together (Appendix C.1). More targeted controls would separate the contributions of hypothesis registration, edit restrictions, and question continuation.

Search space and cost. The search explores designs expressible through the program interfaces, available pretrained representations, and candidate budgets. Training and evaluation require substantial computation (Appendix B.4). Adaptive allocation between inexpensive screening and full evaluation could support longer training and broader architectural changes within a fixed budget. Evaluating that allocation would require accounting for both proposal generation and candidate execution.

Table 31: **Programs discovered under different configurations.** Each row describes the highest-objective program reported for that run. Qwen denotes Qwen3.8-27B. Training operations apply equally to both labels; all readouts use fixed training-fitted parameters at evaluation.

Configuration	Views and final readout	Adaptation and training design
DINOv3-H / Qwen	Raw, half-scale, resize224. Normalize each view, average, normalize again; linear head.	Rank-8 attention LoRA, two epochs. Resample with probability 0.70, scale $[0.60, 1]$. BCE plus patch-variance floor, cosine-distance, and pairwise terms.
DINOv3-L / Qwen	Raw, half-scale. Separate linear heads; standardize scores using training statistics, then average.	Rank-8 attention LoRA, two epochs. A 50% clean-image branch; otherwise JPEG and resampling. BCE plus cosine class margin, feature/score consistency, and patch-to-class margin.
DINOv2-L / Qwen	Raw, half-scale, median residual. Concatenate normalized views; 256-unit MLP trained for 30 epochs.	Rank-32 attention LoRA, two epochs. Apply a mild JPEG/resampling chain to 15% of images. BCE plus pooled cosine-margin and relative patch-norm variance penalties.
CLIP-L / Qwen	Raw, high-pass, nearest-neighbor resampling residual. Concatenate views, then normalize; linear head.	Rank-8 attention LoRA, two epochs. JPEG with probability 0.60 and resampling with probability 0.35. BCE plus within-class angular compactness on pooled features.
ResNet-50 / Qwen	Raw, half-scale. Concatenate normalized views; linear head.	Full backbone fine-tuning, one epoch. JPEG with probability 0.35, quality $[20, 95]$. Standard BCE adaptation.
DINOv3-H / DeepSeek	Raw, half-scale, log-DCT. Concatenate normalized views; linear head.	Rank-16 query/value LoRA, three epochs. Random-order JPEG/resampling/blur chain with 20% clean images. BCE plus within-class radius-band and between-class centroid-margin penalties.
DINOv3-H / Terra	Raw, half-scale. Normalize and fuse; concatenate a fixed 256-dimensional random projection with a class-centroid margin; linear head.	Rank-8 query/value LoRA, two epochs. Identity training-image operation. BCE plus inclusive class compactness, centroid separation, and softly weighted patch-to-image agreement.

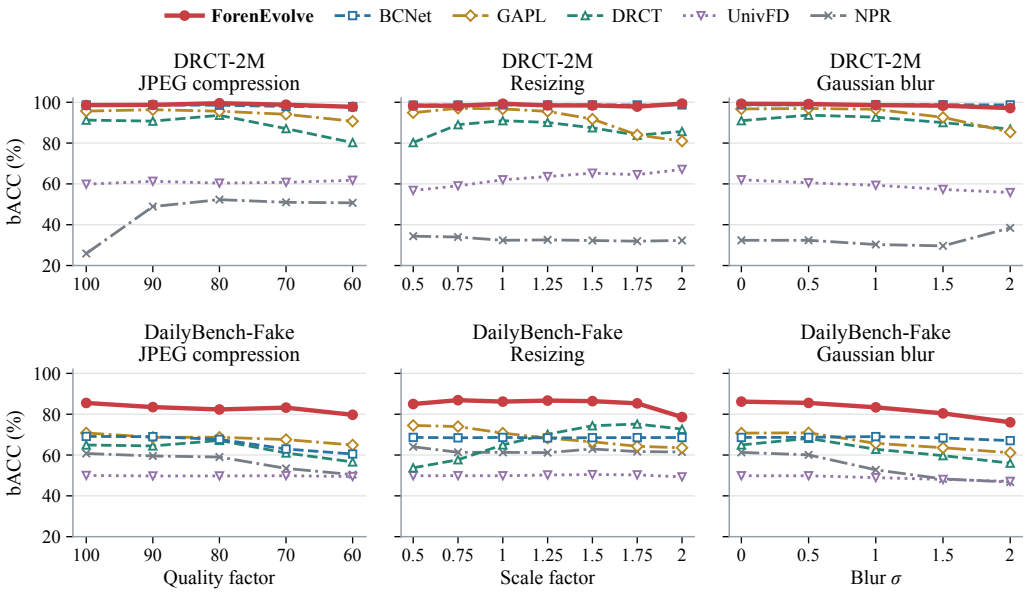


Figure 10: **Robustness on L2 benchmarks.** bACC on DRCT-2M and DailyBench-Fake under JPEG compression, resizing, and Gaussian blur.

G PER-GENERATOR AND SOURCE-SUBSET RESULTS

Reading the subset tables. Each subset's bACC averages fake recall and real recall on its paired or shared real pool. ForenEvolve uses the threshold in Appendix B.2; the supplied baseline exports use $P(\text{fake}) > 0.5$. AIDE uses SD14 and DRCT uses DRCT2M-UnivFD-SDv1. The tables preserve the exports' evaluation support and real-pool conventions, so their baseline aggregates can differ from Table 1. Summary rows give unweighted source means and population standard deviations, describing variation across sources. Main-table pooled bACC weights images within each class. AIGI-Now summaries first average pixel and semantic results within each generator; DRCT-2M summaries use the thirteen generators evaluated by ForenEvolve.

The n_{r} and n_{f} columns count ForenEvolve's evaluated images; shared real pools count once in benchmark totals. Superscript $*$ marks exclusion of unusable predictions, † denotes a table-specific pooling or coverage exception, and – denotes an unavailable result. Captions specify support differences, including Community Forensics' shared real pool and UniGenDet's differing subsets. Scores retain source precision; summaries use unrounded values where available.

Table 32: **AIGI-Now (L0).** Per-generator/source-subset bACC (%). $n_{\mathrm{r}}, n_{\mathrm{f}}$ count ForenEvolve's evaluated real and fake images. Pixel and semantic views remain separate; summary statistics first average the two views for each of nine generators. $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors		ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	Fake VLM	n_{r}	n_{f}	bACC
Nano/pixel	78.75	50.95	55.75	96.30	68.90	81.55	84.05	99.75	64.50	98.95	99.30	1,000	1,000	98.65
Nano/semantic	45.65	51.35	49.95	57.40	63.40	56.20	65.50	49.65	70.80	97.60	97.85	1,000	1,000	97.05
flux-dev/pixel	80.90	50.15	52.80	89.20	86.55	91.55	95.85	96.70	82.95	97.65	98.60*	1,000	1,000	99.00
flux-dev/semantic	45.45	49.25	50.00	61.25	62.65	51.15	67.15	49.35	76.55	96.45	92.94*	1,000	1,000	98.50
flux-kera/pixel	48.65	51.30	50.30	51.95	82.30	59.35	79.35	50.15	66.15	94.50	89.30	1,000	1,000	96.65
flux-kera/semantic	45.25	51.10	50.00	59.80	63.35	50.00	61.05	49.50	58.25	93.60	83.95	1,000	1,000	89.25
flux-kontext/pixel	80.35	57.50	64.50	88.85	86.95	82.70	92.30	98.55	68.25	75.65	82.83*	1,000	1,000	91.15
flux-kontext/semantic	46.65	55.20	49.95	63.55	60.85	52.85	64.15	49.90	63.95	85.60	71.00	1,000	1,000	82.70
flux-pro/pixel	34.20	52.30	47.95	57.75	74.00	76.60	76.00	49.75	65.10	92.40	91.55	1,000	1,000	95.40
flux-pro/semantic	46.60	51.45	50.00	59.45	63.00	55.00	61.20	49.70	79.30	95.50	94.09*	1,000	1,000	96.80
gpt4o/pixel	80.40	53.80	51.65	81.15	75.05	92.30	90.60	85.05	71.75	95.05	99.35*	1,000	1,000	96.75
gpt4o/semantic	46.30	53.55	49.90	56.30	61.60	65.35	56.40	49.15	75.20	95.70	87.73*	1,000	1,000	95.60
jimeng/pixel	30.55	50.60	47.95	60.25	81.05	86.95	75.35	49.80	61.75	95.45	97.05*	1,000	1,000	93.85
jimeng/semantic	46.05	52.40	49.95	54.65	63.20	55.70	55.65	49.70	75.45	96.25	91.70	1,000	1,000	97.45
keling/pixel	80.45	51.35	56.45	90.15	66.85	96.00	98.25	98.95	79.85	98.80	99.30*	1,000	1,000	99.15
keling/semantic	47.25	53.65	50.00	58.85	62.10	64.55	58.05	49.40	84.50	95.85	92.45	1,000	1,000	94.15
minimax/pixel	69.65	50.50	48.50	52.90	78.55	83.25	79.60	49.80	55.30	90.35	93.14*	1,000	1,000	94.40
minimax/semantic	45.25	52.25	50.00	58.40	61.15	50.50	54.20	49.25	57.90	91.35	85.54*	1,000	1,000	88.70
Mean (9 groups)	55.46	52.15	51.42	66.56	70.08	69.53	73.04	62.45	69.86	93.71	91.54	–	–	94.73
Population std.	10.00	1.80	2.53	8.95	3.32	7.10	5.27	11.59	7.62	5.08	6.17	–	–	3.47

Table 33: **Chameleon (L0).** Per-generator/source-subset bACC (%). $n_{\mathrm{r}}, n_{\mathrm{f}}$ count ForenEvolve's evaluated real and fake images. The export provides one mixed subset rather than individual generator identities. $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors			ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	UniGen Det	Fake VLM	n_{r}	n_{f}	bACC
mixed	53.36	50.71	51.11	59.56	60.90	82.41	74.42	52.70	80.47	83.60	60.62	70.30*	14,863	11,170	94.87
Mean (1 groups)	53.36	50.71	51.11	59.56	60.90	82.41	74.42	52.70	80.47	83.60	60.62	70.30	–	–	94.87
Population std.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	–	–	0.00

Table 34: **SynthWildX (L0).** Per-generator/source-subset bACC (%). $n_{\mathrm{r}}, n_{\mathrm{f}}$ count ForenEvolve's evaluated real and fake images. ForenEvolve and the three vision-language exports use 428 shared real images; the reference manifest lists 429. $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors			ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	UniGen Det	Fake VLM	n_{r}	n_{f}	bACC
dalle3	40.86	45.51	48.12	67.86	62.34	92.35	87.14	49.07	95.05	84.59	66.82	70.48	428	438	95.33
firefly	60.77	64.72	61.55	52.10	58.91	87.28	87.45	51.89	88.59	75.10	66.19	70.62	428	478	92.61
midjourney_v5	42.35	45.74	50.52	52.91	59.83	93.03	84.63	49.18	95.05	79.45	66.59	69.68	428	438	95.33
Mean (3 groups)	47.99	51.99	53.40	57.62	60.36	90.89	86.41	50.05	92.90	79.71	66.53	70.26	–	–	94.42
Population std.	9.05	9.00	5.85	7.25	1.45	2.57	1.26	1.30	3.05	3.88	0.26	0.41	–	–	1.28

Table 35: **AIGIBench (L0).** Per-generator/source-subset bACC (%). $n_{\mathrm{r}}, n_{\mathrm{f}}$ count ForenEvolve's evaluated real and fake images. Each generator uses its paired real split, including UniGenDet; this differs from its pooled-real macro convention in the main table. $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors			ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	UniGen Det	Fake VLM	n_{r}	n_{f}	bACC
BLIP	90.18	58.44	81.39	96.60	65.42	97.89	96.93	98.51	96.49	71.02	73.82	83.37*	4,500	4,500	94.74
BlendFace	23.76	50.60	44.19	50.49	78.88	78.62	59.42	68.31	63.97	64.40	36.14	61.83*	4,500	4,500	58.02
CommunityAI	55.70	51.14	52.02	61.75	67.65	84.62	76.59	54.62	82.53	81.75	56.42	65.49*	6,000	6,000	96.78
DALLE-3	66.91	53.36	50.84	71.34	74.94	94.89	91.93	42.00	94.49	67.74	67.88	82.50*	4,002	4,000	94.65
E4S	24.73	74.37	68.44	53.06	80.21	84.66	78.64	56.64	83.79	67.36	52.26	73.37*	4,500	4,500	67.53
FLUX1-dev	79.87	48.33	47.37	79.01	52.61	92.99	94.81	88.61	90.32	67.28	67.78	82.68*	4,500	4,500	94.20
FaceSwap	35.85	73.80	76.56	60.55	75.20	72.92	69.46	76.19	71.25	70.21	69.21	82.26	4,500	4,400	57.76
GLIDE	80.17	59.89	83.49	83.29	70.94	84.70	97.19	92.17	96.13	70.30	71.66	83.27	4,500	4,500	93.19
IP-Adapter	79.52	52.53	67.08	84.27	67.61	97.43	95.41	90.23	92.31	62.12	67.91	82.31*	4,500	4,500	94.41
Imagen3	80.02	50.54	46.22	75.66	54.48	85.73	91.32	89.83	67.11	66.22	67.84	82.22*	4,500	4,500	93.88
InSwap	36.03	51.53	56.64	65.51	73.89	84.04	59.52	78.41	70.64	70.64	57.20	71.70*	4,500	4,400	66.91
Infinite-ID	77.30	62.49	82.20	82.70	70.63	97.86	95.52	90.96	96.17	71.31	71.33	82.91*	4,500	4,500	94.63
InstantID	69.11	51.54	69.09	85.98	74.61	97.82	96.84	89.47	96.02	67.66	67.99	83.09*	4,500	4,500	94.48
Midjourney	61.32	52.28	46.55	62.45	64.75	96.57	93.57	73.80	87.45	52.45	51.77	80.67	3,000	3,000	95.15
PhotoMaker	58.03	51.86	45.43	74.67	75.84	84.14	82.24	88.30	83.69	61.39	67.82	72.55*	4,500	4,500	93.13
ProGAN	99.79	99.81	99.89	69.27	70.40	92.30	99.91	99.67	99.75	74.92	52.56	80.50*	4,000	4,000	90.38
R3GAN	78.03	78.43	93.08	82.54	45.37	95.99	96.83	90.50	95.98	67.96	67.63	82.02*	4,500	4,500	94.29
SD3	79.03	56.32	62.63	88.76	59.99	97.74	96.86	91.57	96.69	75.58	74.74	84.43*	4,500	4,500	95.57
SDXL	78.99	59.04	74.29	87.36	64.29	97.76	96.99	91.58	97.13	75.72	74.82	84.46*	4,500	4,500	95.57
SimSwap	35.02	57.26	60.53	63.46	74.61	84.14	76.47	83.61	73.37	70.33	51.62	58.89*	4,500	4,500	71.97
SocialRF	54.83	54.60	56.27	60.48	53.97	81.73	80.10	59.23	94.98	76.00	62.82	63.06*	3,000	3,000	93.67
StyleGAN-XL	76.01	80.42	92.41	83.66	50.33	48.12	96.71	89.93	96.32	67.84	68.20	81.58*	4,500	4,500	94.28
StyleGAN3	78.96	64.14	90.39	76.18	66.71	59.14	98.17	90.83	95.73	60.37	69.66	83.38*	4,500	4,500	96.59
StyleSwim	77.32	84.11	92.80	64.67	73.72	52.53	96.91	89.92	95.76	67.91	68.11	82.44*	4,500	4,500	94.30
WFIR	65.40	86.90	85.35	76.50	76.20	51.50	95.20	91.70	95.70	54.75	50.00	84.10	1,000	1,000	97.10
Mean (25 groups)	65.68	62.55	69.01	73.61	67.33	83.83	88.54	82.26	88.55	68.13	63.49	78.20	–	–	88.53
Population std.	20.08	13.74	17.56	11.76	9.38	15.25	11.93	14.70	10.70	6.42	9.55	7.77	–	–	12.37

Table 36: **GenImage (L1).** Per-generator/source-subset bACC (%). $n_{\mathrm{r}}, n_{\mathrm{f}}$ count ForenEvolve's evaluated real and fake images. †: UniGenDet uses 5,750 real images per generator (7,750 for SD v1.5); the reference splits contain 6,000 (8,000). $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors			ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	UniGen Det	Fake VLM	n_{r}	n_{f}	bACC
ADM	34.12	62.52	78.44	78.48	75.84	89.55	95.03	97.37	91.22	96.23	96.41†	99.68*	6,000	6,000	86.04
BigGAN	76.78	84.38	96.79	66.85	88.21	86.52	97.82	95.74	99.81	91.00	96.68†	99.72*	6,000	6,000	98.92
Midjourney	49.96	55.10	56.09	81.39	86.48	95.60	90.03	93.77	85.11	87.09	96.36†	98.07*	6,000	6,000	98.29
VQDM	37.03	76.88	86.87	80.25	89.90	76.58	98.02	99.55	99.76	95.62	96.50†	99.44*	6,000	6,000	99.11
glide	33.13	61.28	88.03	91.76	88.17	89.58	97.88	99.30	99.62	99.04	96.61†	99.64*	6,000	6,000	98.58
stable_diffusion_v_1_4	52.44	55.58	67.83	99.74	93.06	98.72	97.98	99.74	99.86	95.33	96.42†	99.57*	6,000	6,000	99.67
stable_diffusion_v_1_5	52.26	55.67	68.06	99.74	92.58	98.55	97.82	99.57	99.77	94.79	96.83†	99.44*	8,000	8,000	99.64
wukong	52.52	61.08	73.06	98.85	93.03	98.78	97.96	99.67	99.79	92.27	96.70†	99.42*	6,000	6,000	99.59
Mean (8 groups)	48.53	64.06	76.90	87.13	88.41	91.73	96.57	98.09	96.87	93.92	96.56	99.37	–	–	97.48
Population std.	13.33	10.12	12.40	11.42	5.30	7.30	2.65	2.11	5.25	3.45	0.16	0.50	–	–	4.35

Table 37: **Synthbuster (L1).** Per-generator/source-subset bACC (%). $n_{\mathrm{r}}, n_{\mathrm{f}}$ count ForenEvolve's evaluated real and fake images. The nine generators share a pool of 1,000 real images. $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors			ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	UniGen Det	Fake VLM	n_{r}	n_{f}	bACC
dalle2	50.55	83.50	54.10	37.25	77.10	86.30	91.90	53.05	97.40	83.55	98.55	60.05	1,000	1,000	99.15
dalle3	51.55	47.35	35.85	37.00	86.50	89.95	74.75	4.10	98.55	99.55	98.55	61.50	1,000	1,000	99.95
firefly	46.10	89.95	60.20	23.75	79.85	91.90	91.75	27.35	77.80	70.70	92.40	60.45	1,000	1,000	99.15
glide	48.40	53.25	67.75	63.90	82.60	76.45	93.50	53.30	99.60	98.70	98.55	61.55*	1,000	1,000	99.55
midjourney-v5	52.55	52.45	44.45	57.70	73.65	93.45	90.90	52.20	93.80	61.30	94.70	61.35	1,000	1,000	99.95
stable-diffusion-1-3	51.40	70.40	52.95	73.15	86.50	92.85	93.80	54.00	99.70	97.95	98.55	61.30	1,000	1,000	99.95
stable-diffusion-1-4	51.50	69.90	53.70	72.90	86.55	92.65	93.80	54.00	99.70	98.20	98.55	61.20	1,000	1,000	99.95
stable-diffusion-2	45.90	75.65	51.55	53.95	83.15	93.30	92.90	53.50	99.65	74.30	91.95	58.90	1,000	1,000	99.85
stable-diffusion-xl	52.55	68.00	66.65	68.70	71.20	93.45	94.50	54.00	99.60	82.00	97.10	61.45*	1,000	1,000	99.95
Mean (9 groups)	50.06	67.83	54.13	54.26	80.79	90.03	90.87	45.06	96.20	85.14	96.54	60.86	–	–	99.72
Population std.	2.46	13.64	9.52	16.79	5.42	5.28	5.80	16.62	6.76	13.47	2.63	0.85	–	–	0.33

Table 38: **AIGCDetectBench (L1).** Per-generator/source-subset bACC (%). $n_{\mathrm{r}}, n_{\mathrm{f}}$ count ForenEvolve's evaluated real and fake images. †: UniGenDet uses the benchmark-wide real pool for CycleGAN, ProGAN, StyleGAN, and StyleGAN2 after deduplication. $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors			ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	UniGen Det	Fake VLM	n_{r}	n_{f}	bACC
ADM	69.91	66.85	78.45	78.48	75.84	88.20	95.08	97.34	91.22	96.42	96.41	99.68*	6,000	6,000	86.04
DALLE2	64.87*	50.75*	69.60*	95.00*	92.10*	92.90*	95.80*	99.55*	92.85*	88.90*	99.25	98.15*	999	1,000	97.40
Glide	78.52	62.44	88.01	91.76	88.17	86.29	97.86	99.17	99.62	99.02	96.58	99.64*	6,000	6,000	98.58
Midjourney	78.03	56.14	56.08	81.39	86.48	96.16	90.02	93.74	85.11	87.57	96.33	98.07*	6,000	6,000	98.29
VQDM	78.47	85.32	86.87	80.25	89.90	72.00	97.95	99.59	99.76	96.03	96.46	99.44*	6,000	6,000	99.11
biggan	84.35	95.08	99.50	77.20	82.20	83.65	98.67	84.35	99.75	71.15	52.73	94.32*	2,000	2,000	99.60
cyclegan	96.10	98.33	99.32	74.38	89.40	63.89	98.45	98.83	97.96	78.08	89.75†	98.86	1,321	1,321	97.27
gaugan	82.50	99.47	99.41	64.35	83.20	89.95	99.65	79.09	99.91	67.18	50.18	90.39*	5,000	5,000	98.91
progan	99.79	99.81	99.89	69.27	70.40	92.34	99.91	99.80	99.75	74.90	89.75†	80.50*	4,000	4,000	90.38
sd_xl	78.53	60.13	73.47	93.40	80.38	99.38	99.48	99.72	99.60	97.92	98.85	99.40*	2,000	2,000	99.55
stable_diffusion_v_1_4	78.88	63.68	67.79	99.74	93.06	98.70	98.03	99.72	99.86	95.62	96.42	99.57*	6,000	6,000	99.67
stable_diffusion_v_1_5	79.04	63.47	68.04	99.74	92.58	98.68	97.76	99.56	99.77	95.12	96.82	99.44*	8,000	8,000	99.64
stargan	99.35	95.72	99.75	80.27	53.03	66.21	97.05	99.72	90.57	50.98	82.29	98.80	1,999	1,999	97.45
stylegan	97.70	84.94	97.15	71.32	64.63	86.07	98.10	99.70	98.22	59.91	78.09†	74.50	5,991	5,991	92.00
stylegan2	98.38	74.94	98.80	72.50	60.95	87.32	99.57	99.79	92.42	61.40	81.21†	73.73*	7,988	7,988	93.76
whichfaceisreal	65.40	86.95	88.10	76.50	76.20	50.70	95.20	92.50	95.70	54.75	50.00	84.10	1,000	1,000	97.10
wukong	76.33	70.91	73.05	98.85	93.03	98.75	97.92	99.79	99.79	92.73	96.69	99.42*	6,000	6,000	99.59
Mean (17 groups)	82.71	77.35	84.90	82.61	80.68	85.36	97.44	96.59	96.58	80.45	85.17	93.41	–	–	96.73
Population std.	11.24	16.29	14.14	11.14	11.88	13.72	2.33	5.89	4.36	16.23	17.01	8.98	–	–	3.80

Table 39: **WildRF (L1).** Per-generator/source-subset bACC (%). n_r, n_f count ForenEvolve's evaluated real and fake images. Rows denote social-media source subsets rather than generator architectures. $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors			ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	UniGen Det	Fake VLM	n_r	n_f	bACC
facebook	75.62	48.75	54.69	59.38	49.06	92.81	85.31	53.12	95.00	73.44	54.69	64.06	160	160	96.88
reddit	65.20	59.47	69.67	71.93	58.60	86.80	81.40	75.93	97.80	85.60	68.20	71.18*	750	750	99.13
twitter	50.52	55.97	55.11	59.20	50.48	90.93	86.08	52.63	97.37	86.52	61.00	65.99	341	342	97.51
Mean (3 groups)	63.78	54.73	59.82	63.50	52.71	90.18	84.26	60.56	96.72	81.85	61.30	67.08	–	–	97.84
Population std.	10.30	4.46	6.96	5.96	4.20	2.51	2.05	10.87	1.23	5.96	5.52	3.01	–	–	0.95

Table 40: **Community Forensics (L2).** Per-generator/source-subset bACC (%). n_r, n_f count ForenEvolve's evaluated real and fake images. †: ForenEvolve uses a shared pool of 25,918 real images; other available columns use paired real subsets. UniGenDet's Hourglass subset has 1,000 fake images. FakeVLM/OmniVL provide only pooled bACC (93.06/94.37), so their subset cells remain unavailable. $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors			ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	UniGen Det	Fake VLM	n_r	n_f	bACC
DFGAN	95.50	98.80	100.00	76.30	89.60	96.15	98.70	99.90	99.95	–	94.00	–	25,918	912	99.66†
Dalle2	98.45	77.85	61.45	75.70	79.55	93.50	97.40	99.65	94.25	–	93.75	–	25,918	1,000	97.06†
Dalle3	99.40	59.70	68.65	99.05	87.60	98.95	98.55	99.90	96.40	–	93.70	–	25,918	1,000	99.66†
DeciDiffusionV2	97.70	60.00	57.35	92.80	88.10	99.10	98.00	99.65	99.80	–	93.55	–	25,918	1,000	99.66†
FLUX-dev	99.45	49.20	54.10	93.65	74.70	95.60	98.10	98.40	89.10	–	92.95	–	25,918	994	99.41†
FLUX-schnell	97.80	50.05	55.85	88.25	72.90	97.95	98.15	94.70	89.85	–	92.80	–	25,918	991	99.31†
Firefly_Image2	64.13	83.71	54.19	60.85	88.57	93.91	97.19	50.05	87.82	–	94.07	–	25,918	1,877	94.78†
Firefly_Image3	56.49	71.36	54.54	64.32	87.60	89.30	95.46	50.03	92.12	–	93.84	–	25,918	1,928	95.77†
GALIP	95.65	86.85	93.85	74.95	90.75	89.05	96.60	96.80	99.85	–	93.05	–	25,918	1,000	99.41†
Hourglass	78.75	58.17	72.83	73.90	67.07	64.80	79.70	85.00	79.72	–	72.15†	–	25,918	2,000	81.38†
IdeogramV1	99.35	54.00	55.35	87.70	65.55	98.35	98.80	98.30	81.25	–	92.50	–	25,918	1,000	98.31†
IdeogramV2	99.25	52.45	52.85	84.20	58.00	98.00	97.85	96.75	64.80	–	93.75	–	25,918	1,000	97.06†
Imagen3	99.57	53.22	58.56	81.84	75.12	94.65	96.12	99.81	78.33	–	93.61	–	25,918	1,053	95.91†
LCM_lora_sdv15	99.20	50.55	59.00	93.90	85.05	99.10	99.05	99.50	99.40	–	93.80	–	25,918	993	99.61†
LCM_lora_sdxl	76.85	77.55	87.85	72.50	87.65	96.70	93.15	80.35	99.60	–	94.25	–	25,918	1,000	99.56†
LCM_lora_ssd1b	84.10	64.15	62.90	77.40	88.25	96.30	95.70	89.25	99.90	–	93.95	–	25,918	1,000	99.66†
MidjourneyV5_2	99.22	55.42	65.00	80.58	63.85	99.12	98.67	94.68	96.39	–	94.23	–	25,918	1,985	99.53†
MidjourneyV6_1	98.55	56.60	58.98	84.97	78.09	94.52	97.22	98.45	91.27	–	93.47	–	25,918	1,991	99.06†
kandinsky_2_2	99.35	56.75	58.25	89.05	89.75	99.20	98.30	99.90	100.00	–	93.65	–	25,918	1,000	99.66†
kvikontent_midjourney_v6	98.45	72.80	77.20	99.10	89.70	99.35	99.05	99.70	99.60	–	94.00	–	25,918	994	99.66†
stable_cascade	99.30	67.35	72.05	94.80	88.10	99.40	98.40	100.00	99.90	–	94.35	–	25,918	1,000	99.66†
Mean (21 groups)	92.21	64.60	65.75	83.13	80.74	94.90	96.67	91.94	92.35	–	92.64	–	–	–	97.80
Population std.	12.36	13.50	13.35	10.49	9.96	7.37	4.05	14.54	9.28	–	4.61	–	–	–	3.97

Table 41: **DRCT-2M (L2).** Per-generator/source-subset bACC (%). $n_{\mathrm{r}}, n_{\mathrm{f}}$ count ForenEvolve's evaluated real and fake images. The first 13 generators define ForenEvolve's 57K view. †: UniGenDet uses 2,000 shared real images and different fake counts. The final three inpainting rows are available only for the vision-language detectors and are excluded from the 13-generator summaries. $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors			ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	UniGen Det	Fake VLM	n_{r}	n_{f}	bACC
controlnet-canny-sdxl-1.0	35.18	74.87	60.94	57.11	76.08	99.19	95.22	52.15	99.37	79.63	81.83†	87.70*	5,000	4,000	99.48
lcm-lora-sdv1-5	33.09	54.41	48.64	75.05	77.98	94.33	98.73	49.94	99.88	99.25	96.45†	97.70*	5,000	4,000	99.59
lcm-lora-sdxl	31.03	65.81	50.08	55.74	87.78	98.31	95.38	49.84	97.99	95.26	96.45†	94.13*	5,000	4,000	99.39
ldm-text2im-large-256	31.14	85.41	54.75	64.21	90.91	99.03	99.32	50.49	99.89	96.87	96.23†	60.64*	5,000	4,000	99.63
sd-controlnet-canny	30.66	67.34	49.77	70.67	84.85	98.56	98.29	50.13	99.75	66.83	81.59†	85.03*	5,000	4,000	99.34
sd-turbo	32.23	56.19	48.58	54.69	77.54	97.04	95.56	49.84	99.42	98.52	96.45†	98.24*	5,000	4,000	99.49
sd21-controlnet-canny	30.63	64.85	50.23	55.90	78.46	99.03	96.59	49.90	99.62	60.99	81.46†	87.66*	5,000	4,000	99.18
sdxl-turbo	32.92	52.75	48.54	55.11	78.34	92.88	93.57	49.84	98.30	98.06	96.45†	97.83*	5,000	4,000	99.12
stable-diffusion-2-1	33.52	56.25	48.54	61.05	83.72	98.39	96.88	49.85	99.56	95.16	95.12†	95.17*	5,000	4,000	99.14
stable-diffusion-v1-4	32.72	56.85	48.54	79.05	85.35	98.65	98.16	50.11	99.60	99.01	96.45†	97.10*	5,000	4,000	99.09
stable-diffusion-v1-5	33.03	56.36	48.54	78.75	85.18	98.67	98.21	50.04	99.52	98.93	96.45†	96.83*	5,000	4,000	99.05
stable-diffusion-xl-base-1.0	32.37	60.61	48.61	56.97	87.73	98.33	95.17	49.84	90.58	93.77	96.45†	93.33*	5,000	4,000	98.23
stable-diffusion-xl-refiner-1.0	32.16	54.49	48.61	72.95	87.35	95.43	96.42	49.84	99.69	99.62	96.45†	98.80*	5,000	4,000	99.65
Mean (13 groups)	32.36	62.01	50.34	64.40	83.17	97.53	96.73	50.14	98.71	90.92	92.91	91.55	–	–	99.26
Population std.	1.24	9.16	3.48	9.17	4.69	1.95	1.65	0.61	2.41	12.61	6.19	9.94	–	–	0.36
Additional inpainting generators (outside the 13-generator summaries)															
stable-diffusion-2-inpainting	–	–	–	–	–	–	–	–	–	50.48	48.53†	63.80*	–	–	–
stable-diffusion-inpainting	–	–	–	–	–	–	–	–	–	50.58	48.66†	90.56*	–	–	–
stable-diffusion-xl-1.0-inpainting-0.1	–	–	–	–	–	–	–	–	–	50.38	47.55†	57.24*	–	–	–

Table 42: **RealChain (L2).** Per-generator/source-subset bACC (%). $n_{\mathrm{r}}, n_{\mathrm{f}}$ count ForenEvolve's evaluated real and fake images. Rows share each export's real pool: 7,000 nominal images, 6,999 evaluated by ForenEvolve and OmniVL, and 6,995 by FakeVLM. $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors			ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	UniGen Det	Fake VLM	n_{r}	n_{f}	bACC
Flux1	46.49	39.61	48.64	55.15*	58.75	58.62	75.35	50.00	67.05	80.60*	71.86	78.70*	6,999	1,000	97.81
Hunyuan3	45.19	40.41	48.64	57.65*	59.15	57.97	68.90	50.00	67.00	77.15*	70.46	77.20*	6,999	1,000	94.46
NanoBanana	45.94	39.71	48.64	52.60*	57.35	55.72	53.90	50.00	51.40	79.45*	71.66	78.95*	6,999	1,000	90.21
QwenImage	46.09	44.21	48.84	56.35*	57.65	75.02	66.40	50.00	72.85	79.15*	71.26	73.50*	6,999	1,000	90.61
SDv3.5	46.59	43.61	48.69	51.40*	57.65	71.02	61.00	50.00	58.25	75.75*	69.26	67.99*	6,999	1,000	84.81
Seedream4	48.44	47.06	49.04	62.55*	60.10	73.22	59.05	50.00	51.05	49.85*	65.06	72.60*	6,999	1,000	71.76
i2i	48.34	52.36	49.99	56.00*	59.75	73.97	60.75	50.00	52.80	56.15*	55.31	54.37*	6,999	1,000	64.71
Mean (7 groups)	46.73	43.85	48.93	55.96	58.63	66.51	63.62	50.00	60.06	71.15	67.84	71.90	–	–	84.91
Population std.	1.13	4.31	0.46	3.36	1.02	7.97	6.59	0.00	8.22	11.70	5.56	8.01	–	–	11.33

Table 43: **DailyBench-Fake (L2).** Per-generator/source-subset bACC (%). $n_{\mathrm{r}}, n_{\mathrm{f}}$ count ForenEvolve's evaluated real and fake images. Rows preserve the seven generator groups in the supplied export, including the two small recent-generator subsets. $*$ marks excluded unusable predictions; – denotes unavailable results.

Generator / source subset	Dedicated image detectors									Vision-language detectors			ForenEvolve		
	NPR	Univ FD	Fat Former	AIDE SD14	DRCT	DDA	GAPL	DGS Net	BC Net	OmniVL Guard 2B	UniGen Det	Fake VLM	n_{r}	n_{f}	bACC
FLUX	49.82	48.68	49.44	57.67	57.19	56.94	71.23	49.95	77.24	52.51	49.80	62.00*	20,328	20,328	89.72
FLUX2	54.11	49.09	49.60	53.16	53.86	53.24	63.84	50.02	66.41	40.84	48.37	56.78*	12,456	12,456	84.75
GPT-Image2	97.99	49.82	55.66	55.11	43.80	70.07	78.47	98.36	62.77	50.18	47.99	59.85	274	274	87.41
NanoBanana2	62.45	48.37	50.00	51.22	62.04	65.31	59.39	50.00	54.90	57.55	50.00	56.98*	245	245	83.47
Qwen-Image-2512	61.88	51.22	49.69	59.83	62.72	70.48	70.66	50.08	65.70	43.36	48.47	59.58*	20,272	20,272	84.81
SD35	51.32	50.91	50.57	53.78	53.70	79.17	71.70	50.01	80.13	41.21	48.33	57.78*	19,398	19,398	87.66
ZImage	52.89	50.84	49.97	58.24	57.41	72.61	68.79	50.06	73.24	42.23	48.33	57.60*	19,321	19,321	84.97
Mean (7 groups)	61.49	49.85	50.70	55.57	55.82	66.83	69.15	56.93	68.63	46.84	48.76	58.65	–	–	86.11
Population std.	15.59	1.08	2.05	2.87	5.90	8.40	5.65	16.92	8.14	6.08	0.74	1.76	–	–	2.03