PRIME: PROTOTYPE-ROUTED IMMUTABLE MONOTONE EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

In continual face anti-spoofing (FAS), a router sends each frame to an expert, but bonafide data from later sessions can stop it from sending a known attack to its expert. We propose Prototype-Routed Immutable Monotone Experts (PRIME) to keep these attack routes. When an attack class arrives, PRIME freezes its routing test and the bonafide evidence that the test uses, so later sessions cannot remove a route. We prove this property, conditional zero forgetting on attack routing (ZF-A). The routing test works in a fixed embedding space. PRIME learns this space with Joint Token Attention, a novel readout that weights every token of every layer of a frozen backbone. We pre-train this readout on FFHQ-SynthFAS, our new paired synthetic FAS corpus that we will release. We evaluate PRIME on two protocols. On OCIM leave-one-out with one source per session, PRIME has a mean HTER of 1.45% and outperforms the best continual (10.66%) and joint-training (1.82%) methods. It has the highest mAA (99.97%) and $mAGA$ (98.63%), above joint training on the same backbone. On a chronological chain of three public datasets, PRIME has a mean HTER of 2.21% on five unseen datasets and outperforms training-free continual classifiers on the same features. They reverse 1.5% to 15.7% of attack decisions per session, but PRIME does not lose any route.

1 INTRODUCTION

One of the key challenges with face anti-spoofing (FAS) systems is that new attack types keep evolving (Fang et al., 2024), and a deployed system is expected to continually update itself with new data distributions without forgetting the old ones. This is particularly critical in FAS because losing a previously effective defence can reopen a known vulnerability even when average performance improves. We therefore treat retention of past attack defences as a deployment requirement, and not just a tunable tradeoff against average performance. In this paper, we pose FAS as an *attack-class-incremental* problem in the broader class of continual incremental learning (CIL) architectures (van de Ven et al., 2022; Masana et al., 2023). Each attack class that comes during runtime could potentially present a new distribution, and the system has to add new experts without closing the routes that previous attacks had already taken.

Continual-learning methods freeze the statistics representing past classes or experts, yet predict by comparing an input with all currently available alternatives. For example, FeCAM stores class-wise feature means and covariances and selects the nearest class (Goswami et al., 2023), whereas CN-DPM maintains one generative density per expert and selects the highest-likelihood expert (Lee et al., 2020). Thus, even when existing statistics remain fixed, adding a new class/expert or changing class priors can alter decision regions and redirect inputs to a different expert.

In this paper, we propose PRIME - Prototype-Routed Immutable Monotone Experts. Unlike prior methods (Feng et al., 2026), PRIME freezes not only the information used for routing, but also the comparison that is used to make the routing decision. When a new attack class is introduced, PRIME learns a prototype set using only that class and stores a fixed copy of the bonafide evidence available at that time. Both remain unchanged thereafter. Routing is performed in a fixed embedding space produced by a frozen backbone and projector. An input is routed to an attack expert only if the corresponding prototype set assigns it a higher density than both a uniform baseline and its stored bonafide reference. Since neither the attack model nor its bonafide reference changes, a routing decision that was valid when the class was learned cannot later be reversed by newly added data

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.

or classes. We call this property **conditional zero forgetting for attack routing** (ZF-A, Thm. 1). Furthermore, we have experts in PRIME, wherein each expert selected by a firing prototype set independently predicts bonafide or attack. Although false attack routing cannot decrease on a fixed bonafide set, final classification error may rise or fall as experts are added. Therefore, *Monotone* in PRIME refers to permanent routing decisions, and not guaranteed accuracy retention.

We make the following *contributions* with PRIME in this paper:

Conditional zero forgetting of attack routing. In a frozen routing space, once a frame reaches an expert, later sessions cannot remove its route or change its earliest responsible expert, regardless of bonafide-data growth (Thm. 1). To our knowledge, this is the first per-input routing-permanence guarantee for continual expert routing (Table 1).

Joint Token Attention (JTA). PRIME pre-trains the projector using JTA, our novel token readout. JTA scores every patch and register token of every DINOv3 layer (Siméoni et al., 2025), and attends to spoof cues at different depths. PRIME then freezes the projector, and no session updates it.

FFHQ-SynthFAS. We generate this corpus ourselves to pre-train the projector: $\sim$ 50K identity-matched attack and benign-control pairs over 12 spoof categories, using an image-editing model on FFHQ real faces (Karras et al., 2019). To our knowledge, this is the first such corpus for FAS, and we will release this dataset along with generation prompts.

2 Related Work

Continual and multi-domain FAS: Continual FAS methods separate retained knowledge from parameters that adapt to each new domain. MDIL (Wang et al., 2024a) freezes earlier domain experts and the lower shared encoder, updates the upper shared layers, and selects an expert by cosine similarity to learned domain centres. DCL-FAS (Cai et al., 2023) updates only adapters and the classification head over a frozen ViT, without replaying past data. Both report forgetting in experiments, but neither gives a per-input retention guarantee.

Expert-based continual learning and routing: Expert-based continual learners use different routing strategies. CN-DPM (Lee et al., 2020) weights experts by likelihoods and evolving data priors, while SEED (Rypeść et al., 2024) selects one expert per task but averages expert likelihoods at inference. FlyPrompt (Yan et al., 2026) recomputes a router from accumulated statistics, and (Li et al., 2025a) show that fixed converged gates can recover task-wise expert selection in linear regression. These methods bound routing error or stabilize task-level selection. None of them keeps the route of a single input fixed as new experts arrive.

Directional densities: vMF mixtures model directional features, both for clustering (Banerjee et al., 2005) and for discriminative face-feature learning (Hasnat et al., 2017). Learned directions can be redundant: in the DINO family, several output prototypes converge to the same direction (Govindarajan et al., 2024). PRIME's prototype sets are vMF mixtures on the unit sphere, and each set's component count is chosen on held-out likelihood (Appendix E.1).

Density scores in continual classification and FAS: Density-based scores and multiple prototypes are used in continual classification and FAS. DCNet (Wang et al., 2025) models normalized features with a vMF distribution, while OTA (Li et al., 2025b) uses multiple class sub-centroids. FeCAM (Goswami et al., 2023) applies Mahalanobis scoring over frozen features, but new classes can still alter the winning class. ReCal-MP (Feng et al., 2026) preserves old-label membership in conformal prediction sets, not expert routes. In contrast, PRIME freezes both sides of its gate: an attack score is compared with a uniform null and the bonafide reference stored when the prototype set is introduced. Thus, existing routes and their earliest firing prototype sets remain unchanged (Thm. 1). Table 1 shows that some methods do keep their routing or acceptance function fixed, but their guarantees cover class scores, coverage or label inclusion, and not the route to an expert.

Table 1: Representative learning settings, routing and decision mechanisms, and the scope of each method's evidence.

Method / citation	CL setting	Transfer protocol	Adds neural experts	Fixed evidence	Routing / decision rule	Evidence / guarantee
MDIL [AAAI 2024a]	DIL	DG	✓	✓[†]	Cosine argmax	Empirical retention
OTA[fs] [CVPR 2025b]	—	SF-DA	×	—	Mean cosine	Empirical adaptation
MVP-FAS [ICCV 2025]	—	DG	×	—	Slot + head	Empirical DG
CN-DPM [ICLR 2020]	CIL/DIL	—	✓	×	Density mix.	Empirical retention
FeCAM[c] [NeurIPS 2023]	CIL	—	×	✓	Mahalanobis	Class-score identity
FeCAM[d] [NeurIPS 2023]	DIL	DG	×	×	Mahalanobis	Empirical DG
FlyPrompt [ICLR 2026]	GCL	—	✓	×	Ridge argmax	Routing-risk bounds
Li et al. [ICLR 2025a]	Recurring	—	×	—	Noisy top-1	Task-set routing
ReCal-MP[i] [SSRN 2026]	CIL	—	×	✓	Conformal set	Old-label inclusion
PRIME (Ours)	CIL/DIL[s]	DG	✓	✓	vMF gate	**Conditional ZF-A**[*]

Fixed evidence means the inherited routing or acceptance function is itself unchanged, although the winner it selects may change; for PRIME this is its frozen gates. Settings: CIL, DIL and GCL are class-incremental, domain-incremental and general continual learning; DG excludes target-domain fitting, SF-DA uses target data without source data, and Joint S+T evaluates source and target after target-only updating.

[*]**Conditional ZF-A.** For a fixed frame under frozen geometry, immutable inherited records with strictly ordered append-only insertions preserve an established expert-routed route, its earliest firing set and its session expert (Theorem 1); the expert vote can still change as more experts are called. Per-method variant markers ([†,s,c,d,i,fs]) are defined in Appendix A.

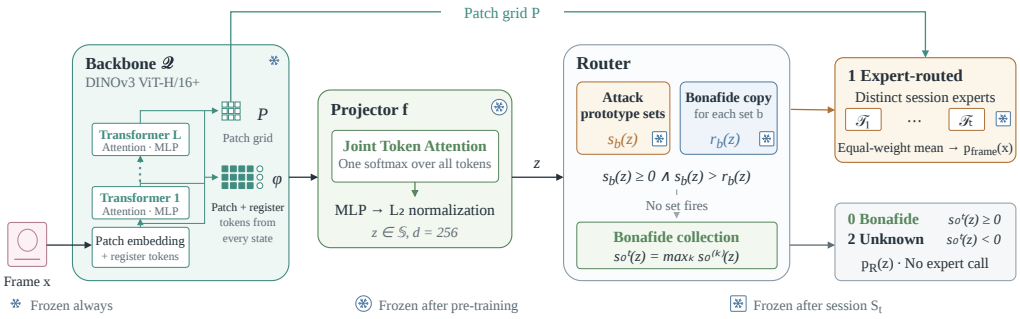


Figure 1: **PRIME.** One frozen backbone pass gives the patch grid P to the experts and the token set ϕ to the frozen projector f, which returns the unit routing embedding $\mathbf{z}$. Each attack prototype set b scores $\mathbf{z}$ against the uniform null and against its own frozen bonafide copy r_b (Eq. 1). The sets that fire select their distinct session experts, whose predictions PRIME averages. When no set fires, the router answers the frame itself from the current bonafide collection and tags it bonafide or unknown (Eqs. 2–3). Prototype sets, the bonafide collection and the experts are append-only. Plain, circled and boxed snowflakes mark components frozen always, after pre-training, and after their session S_t.

3 Method

We propose PRIME, an attack-class-incremental learning framework for face anti-spoofing that only appends state from an arriving dataset without altering earlier state. We start from a frozen DINOv3 ViT-H/16+ backbone $\mathcal{Q}$ (Siméoni et al., 2025), whose weights never change. Given an input frame x, $\mathcal{Q}$ yields two readouts: the patch and register tokens of every backbone state, ϕ, and the spatial patch grid P. The projector f reads ϕ and returns a single unit embedding $\mathbf{z} = f(\phi) \in \mathbb{S}^{d-1}$ with $d = 256$. We pre-train f once, with our novel *Joint Token Attention* mechanism, before any incremental learning begins, and we discuss its architecture and its pre-training objective in §3.1. We then freeze f, so a frame keeps the same $\mathbf{z}$ in every session that follows.

The router only reads $\mathbf{z}$, and it checks the affinity of $\mathbf{z}$ to every attack class it has seen so far. We represent each attack class by an *attack prototype set*. PRIME creates these prototype sets on the arrival of a new dataset $\mathcal{D}_t$, which marks an incremental learning session. A session models every attack class in $\mathcal{D}_t$ by one attack prototype set b. We define the affinity of set b to an embedding $\mathbf{z}$ as a score $s_b(\mathbf{z})$, which we measure against the uniform density on $\mathbb{S}^{d-1}$. Every dataset also carries the bonafide class, and we let this evidence accumulate: sessions add bonafide sets to a collection $s_0^t(\mathbf{z})$ that only grows. This growth creates a problem. A session freezes its attack prototype sets, but the collection that a set compares against keeps moving, so the set could lose a frame that it once claimed. We therefore give each attack set a frozen copy r_b of the collection as its own reference. A

session can then only append, and it never changes what an earlier session wrote. §3.2 shows how we fit these sets, compute s_b and admit a new one.

Using the frozen attack prototype sets and the frozen bonafide copies, the router now scores $\mathbf{z}$ against every attack set it holds. A set b fires when its score passes two comparisons at once:

$$g_b(\mathbf{z}) = 1 \quad \Longleftrightarrow \quad \big(s_b(\mathbf{z}) \geq 0\big) \ \wedge \ \big(s_b(\mathbf{z}) > r_b(\mathbf{z})\big). \tag{1}$$

The first comparison asks the set to explain the frame better than the uniform null. We define s_b as a log density ratio against that uniform density, so the null is zero by construction and needs no calibration. The second comparison asks the set to beat its own bonafide copy r_b.

The router collects the sets that fired and tags the frame with $C(\mathbf{z})$, in order of precedence:

$$F(\mathbf{z}) = \{b \in \mathcal{B}_t : g_b(\mathbf{z}) = 1\}, \qquad C(\mathbf{z}) = \begin{cases} 1 \text{ (expert-routed)}, & F(\mathbf{z}) \neq \varnothing, \\ 0 \text{ (bonafide)}, & F(\mathbf{z}) = \varnothing, \ s_0^t(\mathbf{z}) \geq 0, \\ 2 \text{ (unknown)}, & F(\mathbf{z}) = \varnothing, \ s_0^t(\mathbf{z}) < 0, \end{cases} \tag{2}$$

where $\mathcal{B}_t$ holds the attack prototype sets frozen so far, and Figure 1 shows the corresponding path of a frame. When at least one set fires, PRIME sends the frame to an expert for the final decision. Every session t trains one expert $\mathcal{T}_t$ on its own dataset $\mathcal{D}_t$, and every set the session freezes points to that expert. An expert-routed frame therefore goes to the distinct session experts of its firing sets, $\Omega(\mathbf{z}) = \{\text{expert}(b) : b \in F(\mathbf{z})\}$, and PRIME averages their predictions with equal weight.

When no attack prototype set fires, that is when $F(\mathbf{z}) = \varnothing$, the router checks $\mathbf{z}$ against the current bonafide collection s_0^t. It then tags the frame bonafide or unknown, as (2) shows for $C(\mathbf{z}) \in \{0, 2\}$. The collection grows with every session, so the router can tag a frame as unknown today and as bonafide later. The firing experts therefore score an expert-routed frame, and the router scores a bonafide or unknown frame:

$$p_{\text{frame}}(x) = \begin{cases} \dfrac{1}{|\Omega(\mathbf{z})|} \displaystyle\sum_{\omega \in \Omega(\mathbf{z})} p_{\mathcal{T}_\omega}(x), & C(\mathbf{z}) = 1, \\ p_R(\mathbf{z}) = \sigma\big(\max_{b \in \mathcal{B}_t} \{\ell_b(\mathbf{z}) - \ell_0(\mathbf{z})\}\big), & C(\mathbf{z}) \in \{0, 2\}, \end{cases} \tag{3}$$

where σ is the logistic function and ℓ_0 the calibrated bonafide score, and PRIME outputs the pair $(C(\mathbf{z}), p_{\text{frame}}(x))$. Of the three tags in Eq. 2, only the expert-routed tag carries a guarantee. The session freezes every quantity that the router reads (§3.2), so a frame that reaches an expert can never be routed differently later. We call this property *conditional zero forgetting on attack routing* (ZF-A).

Theorem 1 (Conditional zero forgetting on attack routing, ZF-A)**.** *Fix a frame x and keep $\mathcal{Q}$, f and $\bar{C}_d(\cdot)$ unchanged, so $\mathbf{z}$ is fixed. Suppose the attack prototype sets are append-only, every inherited record keeps its reference, index and session expert, indices are unique, and every new index exceeds every inherited one. Let the* earliest firing set $a_t(\mathbf{z})$ *be the firing set of smallest insertion index. If $C_t(\mathbf{z}) = 1$, then for every $n > t$,*

$$C_n(\mathbf{z}) = 1, \qquad a_n(\mathbf{z}) = a_t(\mathbf{z}), \qquad \text{expert}(a_n(\mathbf{z})) = \text{expert}(a_t(\mathbf{z})).$$

The claim assumes nothing about the bonafide collection. We prove this theorem in Appendix C.

Once an attack prototype set fires, ZF-A makes the route permanent without fixing the label. A wrong route therefore stays in the system beside a correct one. PRIME can afford this because the route is not the decision: the experts that fire classify the frame, and they can label it bonafide. As more experts join, their equal-weight mean in Eq. 3 can make the error rate fall across sessions.

Inference over one frame costs one backbone pass, one projector evaluation and one inner product per stored direction in 256 dimensions. In the worst case every set fires and the expert calls grow linearly with the number of stored experts. Appendix B lists the notation.

3.1 Pre-training: building the routing space

Joint Token Attention. Spoof cues are local, and surface at the different depth of $\mathcal{Q}$, so averaging the tokens of its last state dilutes the evidence we need. We therefore let f read ϕ, every patch and register token from every layer, and learn for itself which of them carries the cue. We call this novel

readout *Joint Token Attention* (JTA). One learned vector w scores every token (El-Nouby et al., 2024), one learned scalar b_l biases each state, and one softmax turns those scores into weights:

$$s_{l,t} = w^\top \bar{x}_{l,t} + b_l, \qquad p_{l,t} = \mathrm{softmax}_{(l,t)}\big(s_{l,t}\big), \qquad \mathbf{e} = \textstyle\sum_{l,t} p_{l,t}\, \bar{x}_{l,t}, \qquad \mathbf{z} = \frac{g(\mathbf{e})}{\|g(\mathbf{e})\|_2}, \quad (4)$$

where $\bar{x}_{l,t}$ is the LayerNormed token at state l and position t, and g is an MLP that maps $\mathbf{e}$ to $\mathbb{R}^d$ with $d = 256$. We elaborate more on JTA in Appendix D.1. On held-out FFHQ-SynthFAS images (Fig. 3), 80% of its attention goes to the last eleven backbone layers, and its spatial weights land where the attack is: the mouth for a partial mouth, the cut edges for a paper cut-out, the face outline for a mask, and the image border for prints and replays. The readout learns this from class labels alone.

Pre-training objective. Equation 4 puts $\mathbf{z}$ on the unit sphere $\mathbb{S}^{d-1}$, where the von Mises–Fisher (vMF) density naturally describes a class. We therefore model every pre-training class, bonafide or attack, by one vMF density. Class c gets S subcentres with learned directions $\mu_{cj} \in \mathbb{S}^{d-1}$, concentrations $0 < \kappa_{cj} \le \kappa_{\max}$, and stick-breaking weights π_{cj} (Sethuraman, 1994). We then score a frame as $h_c(\mathbf{z})$ and form the objective $\mathcal{L}_{\mathrm{vMF}}$:

$$h_c(\mathbf{z}) = \mathrm{LSE}_{j \le S}\big[\log \pi_{cj} + \log \bar{C}_d(\kappa_{cj}) + \kappa_{cj} \mu_{cj}^\top \mathbf{z}\big], \quad \mathcal{L}_{\mathrm{vMF}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{h_{y_i}(\mathbf{z}_i)}}{1 + \sum_c e^{h_c(\mathbf{z}_i)}}, \quad (5)$$

where N is the batch size, y_i is the class label, LSE is the log-sum-exp over subcentres, and $\bar{C}_d(\kappa) = C_d(\kappa)/C_d(0)$ divides the vMF normaliser by the uniform normaliser $C_d(0)$. The denominator of $\mathcal{L}_{\mathrm{vMF}}$ adds 1 to the sum over classes which represents uniform density. It means no class matched, and its score is fixed at 0. Class c therefore claims a frame only when $h_c(\mathbf{z}) > 0$.

$\mathcal{L}_{\mathrm{vMF}}$ alone can drive every frame of a class onto a single direction, leaving a later attack prototype set nothing to model. We prevent that with two more terms in the pre-training objective. $\mathcal{L}_{\mathrm{pres}}$ trains a decoder to rebuild the features of $\mathcal{Q}$ from $\mathbf{z}$ and the label (Appendix D.2), and $\mathcal{L}_{\mathrm{inter}} = \mathbb{E}_{y_i \neq y_j}[\max(0, \mathbf{z}_i^\top \mathbf{z}_j)^2]$ penalises any positive cosine between two classes. We weight the two terms and add them to $\mathcal{L}_{\mathrm{vMF}}$, so f trains on the final objective $\mathcal{L}_{\mathrm{vMF}} + \lambda_{\mathrm{pres}}\mathcal{L}_{\mathrm{pres}} + \lambda_{\mathrm{inter}}\mathcal{L}_{\mathrm{inter}}$, over a paired synthetic corpus. After pre-training, we keep the snapshot that maximises the smallest class-to-nearest-rival margin on held-out identities (Appendix D.3), discard the training subcentres and freeze f. On the corpus's held-out split, this lifts nearest-neighbour accuracy from 79.4% on the frozen backbone's features to 99.3% (Fig. 4). More details on training in Appendix D.

FFHQ-SynthFAS. We build the pre-training corpus ourselves from FFHQ (Karras et al., 2019). We apply an image-editing generator to each real face to produce two images: a material-changing attack and a benign control. Both come from the same generator, so f gains no generator shortcut. No session ever sees this corpus. Appendix I gives the dataset details, category counts, how we build one pair and the prompts we use. *We will release FFHQ-SynthFAS and its associated prompts.*

3.2 Session Training: Continually Learned Routes

Attack prototype sets. In session t, we create one attack prototype set b per attack class of $\mathcal{D}_t$. A set because a single direction is often not enough to represent an attack class. For example, the same printed attack can produce different embeddings under two lighting rigs. Each set therefore contains K_b prototypes. A prototype j has a direction $\boldsymbol{\mu}_{bj}$, a concentration κ_{bj}, and a weight π_{bj}. For each attack class, we fit the set with capped vMF EM using its embeddings from $\mathcal{D}_t$. We choose K_b by evaluating the likelihood on embeddings held out by identity and capture session (Appendix E.1). The resulting set b assigns a score $s_b(\mathbf{z})$ to an embedding $\mathbf{z}$. This score uses Eq. 5's h_c, with the K_b prototypes in the set replacing the S subcentres of a class.

Bonafide collection and its copy. We also fit one bonafide set per contributing session, by the same procedure, and write its score $s_0^{(k)}$. The bonafide collection is then $s_0^t(\mathbf{z}) = \max_{k \in \mathcal{I}_t} s_0^{(k)}(\mathbf{z})$, where $\mathcal{I}_t$ indexes every bonafide set frozen through session t. We take a maximum, and not a weighted mixture, because a mixture renormalises: appending a set could then lower the collection score. The maximum instead satisfies $s_0^{t+1} \ge s_0^t$ pointwise, so bonafide evidence only accumulates. An attack set b never reads this live collection. It reads its own frozen copy $r_b(\mathbf{z}) = \max_{k \in \mathcal{K}_b} s_0^{(k)}(\mathbf{z})$, where $\mathcal{K}_b \subseteq \mathcal{I}_t$ indexes the bonafide sets that its session had already frozen.

Bonafide collection and its copy. We also fit one bonafide set per contributing session, by the same procedure, and write its score $s_0^{(k)}$. The bonafide collection is then $s_0^t(\mathbf{z}) = \max_{k \in \mathcal{I}_t} s_0^{(k)}(\mathbf{z})$, where $\mathcal{I}_t$ indexes every bonafide set frozen through session t. We take a maximum, and not a weighted mixture, because a mixture renormalises: appending a set could then lower the collection score. The maximum instead satisfies $s_0^{t+1} \geq s_0^t$ pointwise, so bonafide evidence only accumulates. An attack set b never reads this live collection. It reads its own frozen copy $r_b(\mathbf{z}) = \max_{k \in \mathcal{K}_b} s_0^{(k)}(\mathbf{z})$, where $\mathcal{K}_b \subseteq \mathcal{I}_t$ indexes the bonafide sets that its session had already frozen.

Session expert. We close the session with its expert $\mathcal{T}_t$, trained on $\mathcal{D}_t$ by binary cross-entropy over the patch grids. One pass through the frozen $\mathcal{Q}$ and f supplies every patch grid and embedding the session needs; Appendix E bounds the session cost. Experts share no parameters, so training expert t leaves expert $t-1$ untouched. Gradients reach the session expert and this session's own prototypes, and nothing else, so the backbone, the projector and every earlier session's state stay bit-identical.

4 EXPERIMENTS

We evaluate PRIME under two protocols. Protocol 1 is the widely used OCIM leave-one-out benchmark for FAS (Shao et al., 2019): PRIME learns three of the four datasets, one per session, and is tested on the fourth. In Protocol 2, PRIME learns three public datasets in their release order, one per session, and after every session it is tested on two out-of-distribution datasets.

Three analyses follow: 1) a comparison with prototype-based continual classifiers on the same features, with route retention for PRIME and decision reversals for the baselines (§4.3); 2) the case tags PRIME assigns to a fixed set of frames across sessions, which shows how routes accumulate and validates ZF-A on real data (§4.4); and 3) what the pre-trained projector adds over frozen DINOv3 features, and how much of the result comes from FFHQ-SynthFAS and from the backbone (§4.5). We report HTER and AUC (%), and for the continual setting the mAA, $mABT$ and $mAGA$ of (Cai et al., 2023); Appendices F and G give the training and evaluation details.

4.1 PROTOCOL 1: OCIM LEAVE-ONE-OUT

OCIM comprises OULU-NPU (O) (Boulkenafet et al., 2017), CASIA-FASD (C) (Zhang et al., 2012), Idiap Replay-Attack (I) (Chingovska et al., 2012) and MSU-MFSD (M) (Wen et al., 2015). Each run leaves out one dataset as its target and learns the other three as sessions S_1–S_3. Table 2 compares PRIME with joint-training methods and with continual methods that learn one source per session.

Among the methods that learn one source per session, MDIL has a mean HTER of 10.66% across the four targets, while PRIME reaches 1.45%. Against the published joint-training methods, which see all three sources at once with their own backbones, PRIME still has the lowest mean HTER (1.45% against 1.82% for MVP-FAS and 1.83% for GD-FAS) and the highest mean AUC (99.86%), and it leads on two of the four chains in HTER and on three in AUC. Learning one source per session therefore need not trail joint training. Part of this strength comes from the projector, pre-trained with JTA (Eq. 4) on FFHQ-SynthFAS (§3.1); §4.5 measures its share.

Appendix H adds two views. Trained on one source and scored on the other three, PRIME averages 1.71% HTER over the twelve pairs against FLIP-MCL's reported 4.84%, under different training exposure (Table 10). In the continual metrics (Table 12) PRIME has the highest mAA (99.97% vs. 99.84% for joint training) and $mAGA$ (98.63% vs. 97.71%); its $mABT$ is -0.01, against RanPAC's $+0.02$ at lower mAA and $mAGA$ (99.83% and 74.52%).

4.2 PROTOCOL 2: CHRONOLOGICAL PUBLIC-DATASET CHAIN

Protocol 2 learns a chain of public datasets in release order: S_1 trains 3D-MAD (Erdogmus & Marcel, 2013), S_2 adds HQ-WMCA (Heusch et al., 2020) and S_3 adds CelebA-Spoof (Zhang et al., 2020). No session trains on SiW-Mv2 (Guo et al., 2022) or ROSE-Youtu (Li et al., 2018).

Table 5(a) scores both after every session. HTER falls at every session, from 5.12% to 4.61% on SiW-Mv2 and from 0.58% to 0.48% on ROSE-Youtu, while AUC stays within half a point (98.86% to 98.45% and 99.98% to 100.00%). Each added session lowers the error on unseen data.

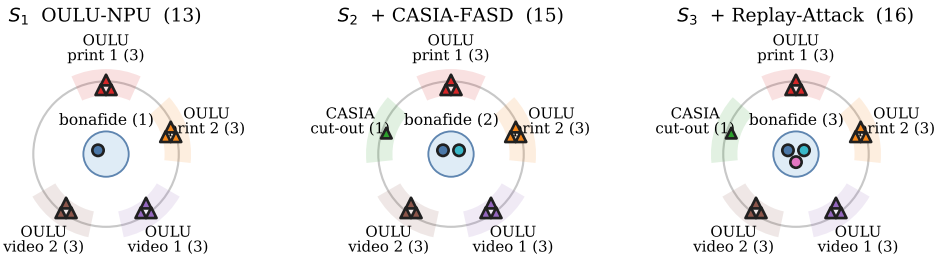


Figure 2: **Append-only growth of the prototype store** on the Protocol 1 chain OCI→M (layout schematic; parenthesised counts are prototypes). Sectors are attack prototype sets, triangles their prototypes, the central disc the bonafide collection. S_1 (OULU-NPU) admits four attack sets and one bonafide set; S_2 (+CASIA-FASD) adds one single-prototype attack set and one bonafide set; S_3 (+Replay-Attack) adds only a bonafide set, since existing sets cover its held-out frames (§3.2).

Table 2: Protocol 1: HTER and AUC (%) on OCIM leave-one-out, trained on three datasets and tested on the fourth. Rows are grouped by how the sources are used. Bold and underline mark the best and second best among rows without target data. Details in Appendix G.

Method	OCI → M		OMI → C		OCM → I		ICM → O	
	HTER↓	AUC↑	HTER↓	AUC↑	HTER↓	AUC↑	HTER↓	AUC↑
Joint multi-source training (domain generalization)								
SSDG-R [CVPR 2020]	7.38	97.17	10.44	95.94	11.71	96.59	15.61	91.54
PatchNet [CVPR 2022a]	7.10	98.46	11.33	94.58	13.40	95.67	11.82	95.07
SSAN-R [CVPR 2022b]	6.67	98.75	10.00	96.67	8.88	96.79	13.72	93.63
FLIP-MCL^{r} [ICCV 2023]	7.50	98.19	1.30	99.19	5.33	97.92	5.19	98.48
IADG [CVPR 2023]	5.41	98.19	8.70	96.44	10.62	94.50	8.86	97.14
SA-FAS [CVPR 2023]	5.95	96.55	8.78	95.37	6.58	97.54	10.00	96.23
UDG-FAS [ICCV 2023]	5.95	98.47	9.82	96.76	5.86	98.62	10.97	95.36
BUDoPT [ECCV 2024b]	<u>0.95</u>	99.70	2.85	98.03	4.40	98.54	2.26	98.78
CA-MoEiT [IJCV 2024]	2.88	98.76	7.89	97.70	6.18	98.94	9.72	96.22
CFPL [CVPR 2024a]	3.09	99.45	2.56	99.10	5.43	98.41	3.33	99.05
DiffFAS-V [ECCV 2024]	2.86	98.41	10.11	96.32	6.36	97.89	8.11	97.27
FGPL [ACM MM 2024]	2.86	98.12	3.89	98.19	3.50	99.54	<u>1.77</u>	99.23
GAC-FAS [CVPR 2024]	5.00	97.56	8.20	95.16	4.29	98.87	8.60	97.16
S-CPTL [ACM MM 2024]	1.43	99.17	0.89	99.00	6.86	98.63	4.12	99.02
TF-FAS [ECCV 2024b]	3.44	99.42	0.81	99.92	<u>2.24</u>	<u>99.67</u>	2.26	99.48
TTDG [CVPR 2024]	4.16	98.48	7.59	98.18	9.62	98.18	10.00	96.15
GD-FAS [ICCV 2025]	**0.42**	**99.88**	0.93	<u>99.99</u>	3.33	98.61	2.64	99.34
MVP-FAS [ICCV 2025]	1.71	<u>99.83</u>	<u>0.75</u>	99.89	**1.99**	99.59	2.82	<u>99.49</u>
OTA [CVPR 2025b]	2.62	99.34	2.22	99.49	5.32	98.44	3.56	99.34
Joint multi-source training with target-domain data								
SDA† [AAAI 2021b]	15.40	91.80	24.50	84.40	15.60	90.10	23.10	84.30
GDA† [ECCV 2022]	9.20	98.00	12.20	93.00	10.00	96.00	14.40	92.60
SDA-FAS† [ECCV 2022]	5.00	97.96	2.40	99.72	2.62	99.48	5.07	99.01
ViTAF‡ [ECCV 2022]	3.42	99.30	1.40	99.85	3.74	99.34	7.17	98.26
Sequential training, one source per session (continual)								
LwF^{s} [TPAMI 2018]	17.14	98.18	33.94	69.56	20.29	91.24	19.03	88.93
L2P^{s} [CVPR 2022c]	13.57	93.23	22.28	83.45	11.82	95.25	31.74	76.08
MDIL^{s} [AAAI 2024a]	5.71	98.19	13.22	91.94	11.25	95.44	12.47	94.22
PRIME	1.67	99.78	**0.56**	**100.00**	2.86	**99.69**	**0.71**	**99.96**

†: unlabeled target adaptation; ‡: five-shot target access and added CelebA-Spoof; s: sequential, one source per session; r: reproduction by Jung et al. (2025). LwF and L2P as reported by Wang et al. (2024a).

Table 3 scores the final system on five unseen sets against published methods trained on CelebA-Spoof alone. PRIME after S_3 reaches 2.21% mean HTER and 99.36% mean AUC, against 7.60% and 96.76% for I-FAS (Zhang et al., 2025), and better on every target. Two controls locate the gain. A CelebA-Spoof-only run, PRIME's first session scored through expert T_1 without routing, averages 3.53%, and sequential fine-tuning on all three sources ends at 3.57%, so additional training data alone does not lower the error. PRIME cuts the control's error by more than a third: the extra sessions help only when their experts are kept and routed to.

Table 3: HTER and AUC (%) on five unseen datasets. The three-source PRIME row follows the last Protocol 2 session; the CelebA-Spoof-only T_1 control is a first-session run scored without routing. Published rows: Zhang et al. (2025). Bold and underline rank the last group; details in Appendix G.

Method	SiW-Mv2		ROSE		OULU		CASIA		MSU		Avg.	
	HTER↓	AUC↑	HTER↓	AUC↑	HTER↓	AUC↑	HTER↓	AUC↑	HTER↓	AUC↑	HTER↓	AUC↑
Trained on CelebA-Spoof only, as reported by (Zhang et al., 2025)												
ViTAF [ECCV 2022]	26.72	80.70	69.34	74.22	26.73	81.28	3.11	99.48	12.86	93.14	27.75	85.76
ViT-B	22.60	84.59	82.69	63.22	28.53	78.59	0.70	99.86	16.67	89.89	30.24	83.23
ViT-L	17.26	90.37	80.47	71.69	29.42	78.07	0.93	99.95	20.87	85.65	29.79	85.15
FLIP [ICCV 2023]	25.95	80.78	80.73	73.60	20.57	87.30	4.88	98.48	19.37	89.95	30.30	86.02
I-FAS [AAAI 2025]	10.89	95.02	5.52	98.48	14.86	91.68	1.11	99.88	5.63	98.73	7.60	96.76
Trained on CelebA-Spoof only, our frozen backbone												
Expert T_1, no router	9.41	94.89	0.58	99.99	1.94	99.68	0.00	100.00	5.71	97.08	3.53	98.33
3D-MAD → HQ-WMCA → CelebA-Spoof, one expert per session, our frozen backbone												
Sequential fine-tuning (floor)	8.69	96.11	0.58	99.99	**2.88**	**99.77**	**0.00**	**100.00**	5.71	97.82	3.57	98.74
PRIME	**4.61**	**98.45**	**0.48**	**100.00**	3.09	98.56	**0.00**	**100.00**	**2.86**	**99.80**	**2.21**	**99.36**

Table 4: Routing rules (%) on the projector output $\mathbf{z}$ and the Protocol 2 sessions: unseen HTER after S_3, $mAGA$, attack decisions reversed at the next session, and whether the rule can abstain. Bold and underline mark the best and second best per column. Details in Appendix G.

Method	HTER↓ after S_3						$mAGA$ ↑	Reversed frames (%)↓	Abstains
	SiW-Mv2	ROSE	OULU	CASIA	MSU	Avg.			
Shared frozen backbone; method-specific readout									
PRIME	**4.61**	**0.48**	**3.09**	**0.00**	2.86	**2.21**	**99.27**	**0.00**	✓
NCM [NeurIPS workshop 2022]	4.86	0.58	4.17	**0.00**	2.86	2.49	99.24	1.5	×
FeCAM [NeurIPS 2023]	5.99	1.73	10.35	**0.00**	3.10	4.23	98.92	5.20	×
RanPAC [NeurIPS 2023]	5.75	2.22	4.72	**0.00**	**2.67**	3.07	95.56	15.72	×

4.3 What Routing Adds

Table 4 isolates the routing rule. It keeps the features fixed at the projector output $\mathbf{z}$ and the sessions fixed at Protocol 2, and replaces the router with three prototype-based continual classifiers updated once per session: NCM (Janson et al., 2022), FeCAM (Goswami et al., 2023) and RanPAC (McDonnell et al., 2023). For PRIME, a reversed decision is a route leaving Case 1.

PRIME has the lowest mean HTER in this comparison, 2.21% against 2.49% for NCM, 4.23% for FeCAM and 3.07% for RanPAC. PRIME and NCM have similar $mAGA$ (99.27% and 99.24%). Every baseline reverses some attack decisions at the next session: NCM 1.5% of its attack frames, FeCAM 5.2% and RanPAC 15.72%. No expert-routed frame of PRIME leaves Case 1, the route permanence that Theorem 1 guarantees and that real data here bears out, and PRIME is the only rule that can abstain. Routing therefore adds two things on identical features: the lowest error of the four rules, and a route stability the other three lack by design.

4.4 Routing Dynamics and ZF-A

Table 5(b) tracks Protocol 2 case tags on fixed frames that no session trains on. ZF-A forbids the moves $1 \to 0$ and $1 \to 2$, and the growing bonafide collection forbids $0 \to 2$. None occurs on the evaluated frames. The Protocol 1 audit (Table 11, Appendix H) also has zero forbidden moves, and in both protocols no frame returns to Case 2 once it leaves. Theorem 1 thus holds on every audited frame of both protocols, on data no session trained on.

In Protocol 2, the attack-frame Case 1 share rises from 3.0% after S_1 to 71.4% after S_2 and 88.6% after S_3, while Case 2 falls from 94.5% to 9.4%. Most newly expert-routed attack frames come from Case 2: 67.9% of the block's frames at S_2 and 17.2% at S_3. The bonafide Case 1 share rises from zero to 3.3% at S_2 and stays there. By S_3 almost 90% of attack frames are handled by an expert while bonafide frames stay almost entirely with the router: expert routes are earned by attacks, as designed.

Table 5: Protocol 2 across three sessions. (a) HTER and AUC (%) of PRIME's final system score on unseen SiW-Mv2 and ROSE-Youtu videos. (b) Case shares and moves on fixed frames (% of each bona fide or attack block). Dashes mark the first session; shaded forbidden moves remain zero. Appendix G defines the cases and transitions.

(a) Final-system performance

PRIME after each session	video level (%)	
Session	HTER↓	AUC↑
SiW-Mv2		
S_1	5.12	98.86
S_2	4.86	98.34
S_3	4.61	98.45
ROSE-Youtu		
S_1	0.58	99.98
S_2	0.53	99.97
S_3	0.48	100.00

(b) Routing cases and transitions

	Share of frames			Moves since previous session					
	Unknown	Bona fide	Expert	allowed			forbidden		
Session	Case 2	Case 0	Case 1	$2 \to 0$	$2 \to 1$	$0 \to 1$	$1 \to 0$	$1 \to 2$	$0 \to 2$
Bona fide frames									
S_1	12.9	87.1	0	–	–	–	–	–	–
S_2	8.2	88.5	3.3	1.4	3.3	0	0	0	0
S_3	4.2	92.5	3.3	4.0	<0.1	0	0	0	0
Attack frames									
S_1	94.5	2.5	3.0	–	–	–	–	–	–
S_2	26.6	2.0	71.4	0	67.9	0.5	0	0	0
S_3	9.4	2.0	88.6	0	17.2	0	0	0	0

Table 6: Linear probe (%) on frozen representations, trained on the named corpus and read on SiW-Mv2 and ROSE-Youtu at a fixed 0.5 threshold. Rows 1–2 change the corpus, row 3 adds PRIME's projector, rows 3–4 change the backbone. Bold and underline mark the best and second best per column. Details in Appendix G.

		SiW-Mv2		ROSE-Youtu	
Representation	Trained on	HTER (%) ↓	AUC (%) ↑	HTER (%) ↓	AUC (%) ↑
Frozen DINOv3 ViT-H/16+	FFHQ-SynthFAS	<u>7.41</u>	**99.39**	16.77	99.58
Frozen DINOv3 ViT-H/16+	CelebA-Spoof	17.89	88.64	32.47	<u>99.59</u>
Projector on DINOv3 ViT-H/16+	FFHQ-SynthFAS	**5.11**	98.47	**1.16**	**99.92**
Projector on CLIP ViT-H/14	FFHQ-SynthFAS	9.47	<u>98.61</u>	<u>9.13</u>	99.18

4.5 What the Projector Adds

Table 6 makes three comparisons on SiW-Mv2 and ROSE-Youtu, each changing one ingredient. *Projector against frozen features:* on the same corpus, the projector lowers HTER from 7.41% to 5.11% on SiW-Mv2 and from 16.77% to 1.16% on ROSE-Youtu; its AUC is 0.92 points below the frozen-feature probe on SiW-Mv2 and 0.34 above on ROSE-Youtu. *FFHQ-SynthFAS against CelebA-Spoof:* on the same frozen features, FFHQ-SynthFAS gives 7.41% against 17.89% on SiW-Mv2 and 16.77% against 32.47% on ROSE-Youtu, less than half the error of a captured-attack corpus. *Backbone against CLIP:* the same projector and corpus on CLIP ViT-H/14 reach 9.47% and 9.13%, about twice the HTER of DINOv3 on SiW-Mv2 and eight times on ROSE-Youtu. All three ingredients contribute, and the projector's gain is largest where the frozen features are weakest.

5 Conclusion, Limitations, and Broader Impact

PRIME separates continual expansion from the routing decisions that must persist. Frozen geometry and introduction-time bonafide references let the prototype store and the experts grow without rewriting established gates. Theorem 1 makes an established expert route, its earliest firing set and its session expert permanent, which protects access to retained expertise. FFHQ-SynthFAS supplies paired attack and benign-control images for projector pretraining; we will release its images, prompts, scripts and metadata.

Limitations. New bonafide prototype sets cannot revise old references or retract existing gates. A bonafide frame can therefore retain an attack-expert route while its selected binary experts still classify it correctly. Verdicts can change as more experts are called. The next step is specialist aggregation that adapts to domain shift while retaining this routing contract.

Broader impact. PRIME targets face anti-spoofing for device authentication, including replay and mask attacks. Immutable prototype set records make routing provenance auditable across updates. Deployment should include subgroup fairness audits and safeguards against misuse for surveillance.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and editing, literature comparison, and LaTeX preparation. The paired attack and benign-control images used for projector pretraining were produced with an image-editing model, as described in §3.1. The authors are responsible for the final text, claims, and artifacts.

ETHICS STATEMENT

Source data, licence and terms. We build FFHQ-SynthFAS from FFHQ (Karras et al., 2019), a public set of photographs of real people collected from Flickr. We collected no new images and we photographed or recruited nobody. We keep only the source photographs whose licence permits redistribution of a derived work, and we drop every source whose terms do not; this removes a large part of FFHQ, which is why our release is smaller than the full set. We redistribute no FFHQ original. We release the generated images alone, together with the prompt, the generator, the random seed and the index of the source photograph for each one, so that any result can be reproduced and any image can be traced back.

Privacy and likeness. Our attacks keep the identity of the source person on purpose: a model must learn the material of an attack and not the face, and that holds only when both classes carry the same faces. The dataset therefore holds images in which a real person's likeness appears on a printed sheet, on a screen or on a worn mask. The people in FFHQ did not agree to this use, and a permissive photograph licence is not consent. We work within three limits. We release no name, no Flickr handle and no other metadata about the person. We generate no image that places a person in a setting that could damage them: the prompts set a desk, a booth or an ordinary room, and they never name an act, a place or an object that carries a claim about the person. We remove an image, and every image derived from the same source, when the owner of the source photograph asks.

Release controls. We release FFHQ-SynthFAS for research on presentation attack detection, under terms that forbid three things: any attempt to identify or contact a person in the data, any use of the images to train or test a face recognition system for deployment, and any redistribution of the images outside those terms. We ask for an institutional e-mail and a signed agreement before we grant access, and we keep the right to withdraw it. We publish the prompts and the code in full, because a detector team needs them and because they carry no image of a person.

Bias and subgroup evaluation. FFHQ is not balanced by age, sex or skin tone, and FFHQ-SynthFAS inherits its distribution. We report overall APCER, BPCER and ACER only. We do not report per-group results, because FFHQ carries no reliable group labels and we will not attach inferred ones to photographs of real people. So we make no claim that a model trained on this data performs equally across groups, and a team that deploys such a model should measure that on data of its own, with labels its subjects agreed to.

Dual use. The images are presentation attacks, so the pipeline could be aimed at a face recognition system instead of at a detector. Three things limit that. The images are single face crops, not video, so they do not carry the motion a liveness check looks for. The generator is a commercial image editing API with its own safety filters, which refused a small number of our prompts. The release terms forbid the use. We accept that the prompts also tell an attacker which materials work, and we judge the gain to the defence to be larger: a detector team needs many attack types and cannot buy them, while an attacker needs one type and can already build it.

REPRODUCIBILITY STATEMENT

The Method section (§3) provides the architecture and conditional routing proof; §4 specifies evaluation protocols and identifies comparison sources. We will release the paired synthetic attack and benign-control corpus together with its generation and preparation materials.

REFERENCES

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von Mises–Fisher distributions. *Journal of Machine Learning Research*, 6: 1345–1382, 2005. 2

Zinelabidine Boulkenafet, Jukka Komulainen, Lei Li, Xiaoyi Feng, and Abdenour Hadid. OULU-NPU: A mobile face presentation attack database with real-world variations. In *IEEE FG*, pp. 612–618, 2017. 6

Rizhao Cai, Yawen Cui, Zhi Li, Zitong Yu, Haoliang Li, Yongjian Hu, and Alex Kot. Rehearsal-free domain continual face anti-spoofing: Generalize more and forget less. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8037–8048, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Cai_Rehearsal-Free_Domain_Continual_Face_Anti-Spoofing_Generalize_More_and_Forget_Less_ICCV_2023_paper.html. 2, 6, 21, 22

Ivana Chingovska, André Anjos, and Sébastien Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *IEEE BIOSIG*, pp. 1–7, 2012. 6

Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Ángel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M. Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. 5

Nesli Erdogmus and Sébastien Marcel. Spoofing in 2D face recognition with 3D masks and anti-spoofing with Kinect. In *IEEE 6th International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2013. 6

Hao Fang, Ajian Liu, Haocheng Yuan, Junze Zheng, Dingheng Zeng, Yanhong Liu, Jiankang Deng, Sergio Escalera, Xiaoming Liu, Jun Wan, and Zhen Lei. Unified physical-digital face attack detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 749–757, 2024. doi: 10.24963/ijcai.2024/83. URL https://doi.org/10.24963/ijcai.2024/83. 1

Kun Feng, Xingyu Gao, Xin Qiu, and Rui Guo. Temporal conformal contracts for uncertainty-tolerant class-incremental learning. SSRN preprint, 2026. URL https://ssrn.com/abstract=7309608. Not peer reviewed. 1, 2, 3

Xinxu Ge, Xin Liu, Zitong Yu, Jingang Shi, Chun Qi, Jie Li, and Heikki Kälviäinen. DiffFAS: Face anti-spoofing via generative diffusion models. In *Computer Vision – ECCV 2024*, pp. 144–161, 2024. doi: 10.1007/978-3-031-72949-2_9. URL https://link.springer.com/chapter/10.1007/978-3-031-72949-2_9. 7

Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Computer Vision – ECCV 2016*, volume 9908 of *Lecture Notes in Computer Science*, pp. 597–613. Springer, 2016. doi: 10.1007/978-3-319-46493-0_36. 22

Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. FeCAM: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 6582–6595, 2023. doi: 10.52202/075280-0288. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/15294ba2dcfb4521274f7aa1c26f4dd4-Abstract-Conference.html. 1, 2, 3, 8, 23

Hariprasath Govindarajan, Per Sidén, Jacob Roll, and Fredrik Lindsten. On partial prototype collapse in the DINO family of self-supervised methods. In *BMVC*, 2024. URL https://bmva-archive.org.uk/bmvc/2024/papers/Paper_949/paper.pdf. 2

Jiabao Guo, Huan Liu, Yizhi Luo, Xueli Hu, Hang Zou, Yuan Zhang, Hui Liu, and Bo Zhao. Style-conditional prompt token learning for generalizable face anti-spoofing. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 994–1003, 2024. doi: 10.1145/3664647.3680857. URL https://dl.acm.org/doi/10.1145/3664647.3680857. 7

Xiao Guo, Yaojie Liu, Anil Jain, and Xiaoming Liu. Multi-domain learning for updating face anti-spoofing models. In *ECCV*, pp. 230–249. Springer, 2022. 6

Md Abul Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, and Liming Chen. von Mises–Fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264*, 2017. 2

Guillaume Heusch, Anjith George, David Geissbühler, Zohreh Mostaani, and Sébastien Marcel. Deep models and shortwave infrared information to detect face presentation attacks. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2020. URL https://arxiv.org/abs/2007.11469. 6

Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1498–1507, 2018. doi: 10.1109/CVPR.2018.00162. 22

Xueli Hu, Huan Liu, Haocheng Yuan, Zhiyang Fu, Yizhi Luo, Ning Zhang, Hang Zou, Jianwen Gan, and Yuan Zhang. Fine-grained prompt learning for face anti-spoofing. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 7619–7628, 2024. doi: 10.1145/3664647.3680855. URL https://dl.acm.org/doi/10.1145/3664647.3680855. 7

Hsin-Ping Huang, Deqing Sun, Yaojie Liu, Wen-Sheng Chu, Taihong Xiao, Jinwei Yuan, Hartwig Adam, and Ming-Hsuan Yang. Adaptive transformers for robust few-shot cross-domain face anti-spoofing. In *ECCV*, 2022. 7, 8

Paul Janson, Wenxuan Zhang, Rahaf Aljundi, and Mohamed Elhoseiny. A simple baseline that questions the use of pretrained-models in continual learning. In *NeurIPS Workshop on Distribution Shifts*, 2022. 8, 23

Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Single-side domain generalization for face anti-spoofing. In *CVPR*, pp. 8484–8493, 2020. 7

Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Unified unsupervised and semi-supervised domain adaptation network for cross-scenario face anti-spoofing. *Pattern Recognition*, 115:107888, 2021. doi: 10.1016/j.patcog.2021.107888. 22

Seungjin Jung, Kanghee Lee, Yonghyun Jeong, Haeun Noh, Jungmin Lee, and Jongwon Choi. Group-wise scaling and orthogonal decomposition for domain-invariant feature extraction in face anti-spoofing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13372–13381, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Jung_Group-wise_Scaling_and_Orthogonal_Decomposition_for_Domain-Invariant_Feature_Extraction_in_ICCV_2025_paper.html. 7

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. URL https://arxiv.org/abs/1812.04948. 2, 5, 10, 23

Binh M. Le and Simon S. Woo. Gradient alignment for cross-domain face anti-spoofing. In *CVPR*, pp. 188–199, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Le_Gradient_Alignment_for_Cross-Domain_Face_Anti-Spoofing_CVPR_2024_paper.html. 7

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural Dirichlet process mixture model for task-free continual learning. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://iclr.cc/virtual_2020/poster_SJxSOJStPr.html. 1, 2, 3

Haoliang Li, Wen Li, Hong Cao, Shiqi Wang, Feiyue Huang, and Alex C. Kot. Unsupervised domain adaptation for face anti-spoofing. *IEEE Transactions on Information Forensics and Security*, 13(7): 1794–1809, 2018. doi: 10.1109/TIFS.2018.2801312. 6, 22

Hongbo Li, Sen Lin, Lingjie Duan, Yingbin Liang, and Ness B. Shroff. Theory on mixture-of-experts in continual learning. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/17a234c91f746d9625a75cf8a8731ee2-Paper-Conference.pdf. 2, 3, 16

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE TPAMI*, 40(12):2935–2947, 2018. 7, 21, 23

Zhuowei Li, Tianchen Zhao, Xiang Xu, Zheng Zhang, Zhihua Li, Xuanbai Chen, Qin Zhang, Alessandro Bergamo, Anil K. Jain, and Yifan Xing. Optimal transport-guided source-free adaptation for face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24351–24363, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Li_Optimal_Transport-Guided_Source-Free_Adaptation_for_Face_Anti-Spoofing_CVPR_2025_paper.html. 2, 3, 7

Ajian Liu. CA-MoEiT: Generalizable face anti-spoofing via dual cross-attention and semi-fixed mixture-of-expert. *International Journal of Computer Vision*, 132(11):5439–5452, 2024. doi: 10.1007/s11263-024-02135-2. URL https://link.springer.com/article/10.1007/s11263-024-02135-2. 7

Ajian Liu, Shuai Xue, Jianwen Gan, Jun Wan, Yanyan Liang, Jiankang Deng, Sergio Escalera, and Zhen Lei. CFPL-FAS: Class free prompt learning for generalizable face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 222–232, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Liu_CFPL-FAS_Class_Free_Prompt_Learning_for_Generalizable_Face_Anti-spoofing_CVPR_2024_paper.html. 7

Si-Qi Liu, Qirui Wang, and Pong C. Yuen. Bottom-up domain prompt tuning for generalized face anti-spoofing. In *Computer Vision – ECCV 2024*, pp. 170–187, 2024b. doi: 10.1007/978-3-031-72897-6_10. URL https://link.springer.com/chapter/10.1007/978-3-031-72897-6_10. 7

Yuchen Liu, Yabo Chen, Wenrui Dai, Mengran Gou, Chun-Ting Huang, and Hongkai Xiong. Source-free domain adaptation with contrastive domain alignment and self-supervised exploration for face anti-spoofing. In *Computer Vision – ECCV 2022*, pp. 511–528, 2022. doi: 10.1007/978-3-031-19775-8_30. URL https://link.springer.com/chapter/10.1007/978-3-031-19775-8_30. 7

Yuchen Liu, Yabo Chen, Mengran Gou, Chun-Ting Huang, Yaoming Wang, Wenrui Dai, and Hongkai Xiong. Towards unsupervised domain generalization for face anti-spoofing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20654–20664, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Liu_Towards_Unsupervised_Domain_Generalization_for_Face_Anti-Spoofing_ICCV_2023_paper.html. 7

Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:5513–5533, 2023. doi: 10.1109/TPAMI.2022.3213473. 1

Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. RanPAC: Random projections and pre-trained models for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 8, 23

OpenAI. GPT-Image-2 model. OpenAI API documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-image-2. Model ID `gpt-image-2`; released April 2026. 24

Grzegorz Rypeść, Sebastian Cygert, Valeriya Khan, Tomasz Trzciński, Bartosz Zieliński, and Bartłomiej Twardowski. Divide and not forget: Ensemble of selectively trained experts in continual learning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5f96a21345c138da929e99871fda138e-Abstract-Conference.html. 2

Jayaram Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, pp. 639–650, 1994. 5

Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C. Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10023–10031, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Shao_Multi-Adversarial_Discriminative_Deep_Domain_Generalization_for_Face_Presentation_Attack_Detection_CVPR_2019_paper.html. 6

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. 2, 3

Koushik Srivatsan, Muzammal Naseer, and Karthik Nandakumar. Flip: Cross-domain face anti-spoofing with language guidance. In *ICCV*, pp. 19685–19696, 2023. 7, 8, 21, 22

Yiyou Sun, Yaojie Liu, Xiaoming Liu, Yixuan Li, and Wen-Sheng Chu. Rethinking domain generalization for face anti-spoofing: Separability and alignment. In *CVPR*, pp. 24563–24574, 2023. 7

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7167–7176, 2017. doi: 10.1109/CVPR.2017.316. 22

Gido M. van de Ven, Tinne Tuytelaars, and Andreas S. Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4:1185–1197, 2022. doi: 10.1038/s42256-022-00568-3. 1

Chien-Yi Wang, Yu-Ding Lu, Shang-Ta Yang, and Shang-Hong Lai. PatchNet: A simple face anti-spoofing framework via fine-grained patch recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20281–20290, 2022a. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_PatchNet_A_Simple_Face_Anti-Spoofing_Framework_via_Fine-Grained_Patch_Recognition_CVPR_2022_paper.html. 7

Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Improving cross-database face presentation attack detection via adversarial domain adaptation. In *Proceedings of the International Conference on Biometrics (ICB)*, pp. 1–8. IEEE, 2019. doi: 10.1109/ICB45273.2019.8987254. 22

Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6678–6687, 2020. doi: 10.1109/CVPR42600.2020.00671. 22

Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Unsupervised adversarial domain adaptation for cross-domain face presentation attack detection. *IEEE Transactions on Information Forensics and Security*, 16:56–69, 2021a. doi: 10.1109/TIFS.2020.3002390. 22

Jingjing Wang, Jingyi Zhang, Ying Bian, Youyi Cai, Chunmao Wang, and Shiliang Pu. Self-domain adaptation for face anti-spoofing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):2746–2754, 2021b. doi: 10.1609/aaai.v35i4.16379. URL https://ojs.aaai.org/index.php/AAAI/article/view/16379. 7

Keyao Wang, Guosheng Zhang, Haixiao Yue, Ajian Liu, Gang Zhang, Haocheng Feng, Junyu Han, Errui Ding, and Jingdong Wang. Multi-domain incremental learning for face presentation attack detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(6):5499–5507, 2024a. doi: 10.1609/aaai.v38i6.28359. URL https://ojs.aaai.org/index.php/AAAI/article/view/28359. 2, 3, 7

Tianqi Wang, Jingcai Guo, Depeng Li, and Zhi Chen. On the discrimination and consistency for exemplar-free class incremental learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 6424–6432, 2025. doi: 10.24963/ijcai.2025/715. URL https://www.ijcai.org/proceedings/2025/715. 2

Xudong Wang, Ke-Yue Zhang, Taiping Yao, Qianyu Zhou, Shouhong Ding, Pingyang Dai, and Rongrong Ji. TF-FAS: Twofold-element fine-grained semantic guidance for generalizable face anti-spoofing. In *Computer Vision – ECCV 2024*, pp. 148–168, 2024b. doi: 10.1007/978-3-031-72667-5_9. URL https://link.springer.com/chapter/10.1007/978-3-031-72667-5_9. 7

Zhuo Wang, Zezheng Wang, Zitong Yu, Weihong Deng, Jiahong Li, Tingting Gao, and Zhongyuan Wang. Domain generalization via shuffled style assembly for face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4123–4133, 2022b. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_Domain_Generalization_via_Shuffled_Style_Assembly_for_Face_Anti-Spoofing_CVPR_2022_paper.html. 7

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, pp. 139–149, 2022c. 7

Di Wen, Hu Han, and Anil K. Jain. Face spoof detection with image distortion analysis. *IEEE TIFS*, 10(4):746–761, 2015. 6

Hongwei Yan, Guanglong Sun, Kanglei Zhou, Qian Li, Liyuan Wang, and Yi Zhong. FlyPrompt: Brain-inspired random-expanded routing with temporal-ensemble experts for general continual learning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://iclr.cc/virtual/2026/poster/10011172. 2, 3

Jeongmin Yu, Susang Kim, Kisu Lee, Taekyoung Kwon, Won-Yong Shin, and Ha Young Kim. Multi-view slot attention using paraphrased texts for face anti-spoofing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21117–21128, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Yu_Multi-View_Slot_Attention_Using_Paraphrased_Texts_for_Face_Anti-Spoofing_ICCV_2025_paper.html. 3, 7

Haixiao Yue, Keyao Wang, Guosheng Zhang, Haocheng Feng, Junyu Han, Errui Ding, and Jingdong Wang. Cyclically disentangled feature translation for face anti-spoofing. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, pp. 3358–3366, 2023. doi: 10.1609/aaai.v37i3.25443. 22

Guosheng Zhang, Keyao Wang, Haixiao Yue, Ajian Liu, Gang Zhang, Kun Yao, Errui Ding, and Jingdong Wang. Interpretable face anti-spoofing: Enhancing generalization with multimodal large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(9):9896–9904, 2025. doi: 10.1609/aaai.v39i9.33073. URL https://ojs.aaai.org/index.php/AAAI/article/view/33073. 7, 8, 21

Yuanhan Zhang, ZhenFei Yin, Yidong Li, Guojun Yin, Junjie Yan, Jing Shao, and Ziwei Liu. CelebA-Spoof: Large-scale face anti-spoofing dataset with rich annotations. In *ECCV*, pp. 70–85. Springer, 2020. 6

Zhiwei Zhang, Junjie Yan, Sifei Liu, Zhen Lei, Dong Yi, and Stan Z. Li. A face antispoofing database with diverse attacks. In *IEEE ICB*, pp. 26–31, 2012. 6

Qianyu Zhou, Ke-Yue Zhang, Taiping Yao, Ran Yi, Kekai Sheng, Shouhong Ding, and Lizhuang Ma. Generative domain adaptation for face anti-spoofing. In *Computer Vision – ECCV 2022*, pp. 335–356, 2022. doi: 10.1007/978-3-031-20065-6_20. URL https://link.springer.com/chapter/10.1007/978-3-031-20065-6_20. 7, 22

Qianyu Zhou, Ke-Yue Zhang, Taiping Yao, Xuequan Lu, Ran Yi, Shouhong Ding, and Lizhuang Ma. Instance-aware domain generalization for face anti-spoofing. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20453–20463, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Zhou_Instance-Aware_Domain_Generalization_for_Face_Anti-Spoofing_CVPR_2023_paper.html. 7

Qianyu Zhou, Ke-Yue Zhang, Taiping Yao, Xuequan Lu, Shouhong Ding, and Lizhuang Ma. Test-time domain generalization for face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 175–187, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Zhou_Test-Time_Domain_Generalization_for_Face_Anti-Spoofing_CVPR_2024_paper.html. 7

A Comparison Table: Per-Method Notes

This section defines the variant markers used in Table 1.

Expert growth. The *adds neural experts* column counts neural experts only, excluding added statistics and prototypes.

Variants. FeCAM[c] retains old classwise statistics; FeCAM[d] updates covariance across domains and evaluates on held-out CoRe50 domains. [†]We infer MDIL's fixed routing scores from its frozen encoder and retained centres; the paper does not state this property directly. OTA[fs] uses labelled, training-free few-shot adaptation, whereas Table 2 uses OTA's zero-shot row. ReCal-MP[c] and ReCal-MP[i] (unrefereed preprint) denote the current-rival and introduction-score variants; both provide sessionwise class-conditional coverage under exchangeability and symmetric scoring, but only [i] preserves old-label inclusion. Li et al. (2025a) study a recurring-linear model with gate training stopped after expert convergence.

PRIME. [s]Attack and bonafide prototype sets grow through class- and domain-incremental updates, while the FAS readout remains binary throughout.

B Notation

Table 7 lists the symbols we use in the main text, in the order the method introduces them.

C Proof of Theorem 1 (ZF-A)

Let $\mathcal{B}_t$ denote the attack prototype sets after session t. Every inherited set evaluates Eq. 1 from functions that no later session changes, at the same fixed $\mathbf{z}$, so each inherited $g_b(\mathbf{z})$ keeps its value. Since $C_n(\mathbf{z}) = 1 \iff \bigvee_{b\in\mathcal{B}_t} g_b(\mathbf{z}) \vee \bigvee_{b\in\mathcal{B}_n\setminus\mathcal{B}_t} g_b(\mathbf{z})$, the first disjunction stays true, so $C_n(\mathbf{z}) = 1$. Its earliest firing set remains earliest because every appended index exceeds every inherited one, and that set's session expert is immutable. □

D Pre-training the Projector

D.1 Joint Token Attention

The backbone exposes 33 states: the patch embedding, the outputs of its first 31 blocks, and the last block after the closing LayerNorm. We drop each state's class token and keep its 196 patch tokens and 4 register tokens, which gives 6,600 tokens $x_{l,t} \in \mathbb{R}^{1280}$ per frame for the softmax of Eq. 4. The MLP g maps $1280 \to 1024 \to 768 \to 512 \to 256$ with LayerNorm and GELU between layers. Both w and b start at zero, so training begins from a plain token average. The learned weights factor as $p_{l,t} = \alpha_l\, p(t \mid l)$ with $\alpha_l = \mathrm{softmax}_l\big(b_l + \log \sum_t e^{w^\top \bar{x}_{l,t}}\big)$, so one set of scores yields both the depth profile α_l and the within-layer spatial map, and every layer's weight stays readable.

Table 7: Notation used throughout the method.

Symbol	Meaning
$\mathcal{Q}, f, \mathbf{z}$	Frozen backbone, projector, unit routing embedding (256 dimensions).
$\mathbb{S}^{d-1}, d$	Unit sphere the projector maps onto; embedding dimension, $d = 256$.
ϕ, P	Patch and register tokens of every backbone state; patch grid read by the experts.
w, b_l, g	Joint Token Attention scoring vector and per-layer bias; MLP to $d = 256$.
$\mathcal{D}_t, t$	Dataset that arrives in session t; the session index.
$\mathcal{T}_t$	Session-t binary expert.
b, K_b	Attack prototype set index; frozen component count.
$\boldsymbol{\mu}_{bj}, \kappa_{bj}, \pi_{bj}$	Component direction, concentration and mixture weight.
$S, \mu_{cj}, \kappa_{\max}$	Subcentres per pre-training class; a subcentre direction; the concentration cap.
$\bar{C}_d(\kappa)$, LSE	vMF normaliser relative to uniform; log-sum-exp.
$s_b(\mathbf{z}), s_0(\mathbf{z})$	Attack mixture log-density ratio to uniform; best current bonafide prototype set score.
$r_b(\mathbf{z}), \mathcal{K}_b$	Frozen bonafide reference function; bonafide prototype sets copied by value at introduction.
$g_b(\mathbf{z})$	Gate of a prototype set.
$\mathcal{B}_t, \mathcal{I}_t$	Attack prototype sets frozen through session t; indices of the bonafide sets frozen by then.
$\mathrm{idx}_b, \mathrm{expert}(b)$	Insertion index; session expert.
$F(\mathbf{z}), a_t(\mathbf{z}), \Omega(\mathbf{z})$	Firing prototype sets; earliest firing set; distinct session experts called.
$\tilde{a}_b, \delta_b$	Reporting row of a set: the two scalars that place its raw score on a comparable scale.
$\ell_b(\mathbf{z}), \ell_0(\mathbf{z})$	Reported score of an attack set and of the bonafide collection.
$C, p_R, p_{\mathrm{frame}}$	Case tag; router (bonafide and unknown cases) and frame attack probabilities.

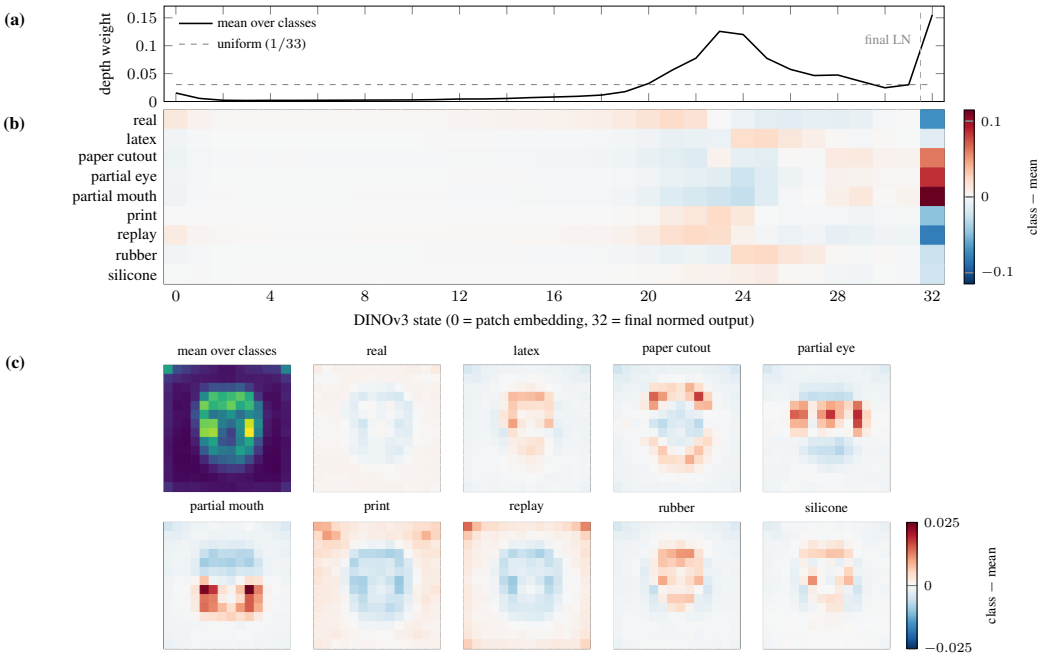


Figure 3: **What Joint Token Attention reads**, on held-out FFHQ-SynthFAS images. (a) Depth profile α_l averaged over classes, against the uniform $1/33$. (b) Each class's depth profile minus that mean. (c) The token weights summed over depth on the 14×14 patch grid: the class mean (brighter is heavier) and each class's departure from it.

Figure 3 shows the learned weights on held-out FFHQ-SynthFAS images. Depth is far from uniform: states 22–32 carry 80% of the weight and the first 19 states under 10%, with peaks at states 23–24 (0.13 and 0.12) and at the final state (0.16, five times its predecessor). The classes split by depth in a way that matches their physics (Fig. 3b): partial attacks, whose cue is a small object pasted on a real face, lean on the final state (partial mouth $+0.11$, partial eye $+0.08$, paper cut-out $+0.06$ above the mean), whereas real faces, replays and prints, whose cue is a texture or a medium spread over the whole frame, draw more on states 21–23. Spatially (Fig. 3c) the class maps follow the attack: partial-mouth attacks move weight to the mouth, partial-eye attacks to the eyes, paper cut-outs to the cut edges, and the three mask materials to the face outline and cheeks, while prints and replays move it off the face to the frame border, where the medium shows. None of this is supervised: w and b_l start at zero, so the readout begins as a plain average of all 6,600 tokens and learns these preferences from the class labels alone.

D.2 Preservation term

$\mathcal{L}_{\mathrm{vMF}}$ alone drives every frame of a class onto one direction, which leaves a later prototype set nothing to model. $\mathcal{L}_{\mathrm{pres}}$ counteracts it by requiring $\mathbf{z}$ to retain enough of the backbone's own description of the frame to rebuild it:

$$\mathcal{L}_{\mathrm{pres}} = \mathbb{E}_{(x,y)}\Big[\big\|D_\phi(\mathbf{z}, y) - \mathrm{sg}(m)\big\|_2^2\Big], \tag{6}$$

where m is the backbone's last-layer patch mean, $\mathrm{sg}(\cdot)$ stops the gradient into it, and D_ϕ is a small decoder conditioned on the class label. The decoder belongs to the loss and not to f, so it leaves the frozen projector state untouched and we discard it once pre-training ends.

D.3 Snapshot selection

We save one snapshot per epoch and keep the one that maximises the worst class's held-out margin M, the fifth-percentile prototype margin under leave-one-group-out validation on a population declared in advance. We select on the worst class rather than on an average because the closest class is what limits routing. In each fold we fit up to six unit prototypes U_c per class by spherical k-means++ on the in-fold frames, using fewer when groups are scarce:

$$\begin{aligned} m_i &= \max_{u \in U_{y_i}} z_i^\top u - \max_{c \neq y_i} \max_{u \in U_c} z_i^\top u, \\ M &= \min_{c:\, n_c \geq 2} q_{0.05}\{m_i : y_i = c\}, \end{aligned} \tag{7}$$

where $q_{0.05}$ is the empirical fifth percentile and n_c counts held-out groups per class. A class needs two groups to be scored, one to fit on and one to score, and we report a class with fewer as unscorable. We require a strict improvement on the best M so far, so the earlier epoch wins a tie.

Figure 4 shows what this training buys on the corpus's held-out split. The frozen backbone already keeps glasses and partial attacks apart, but its bonafide, replay, print and mask frames share one cloud, and a ten-nearest-neighbour classifier on its features is right 79.4% of the time. The projector separates every family into compact clusters at a tenth of the dimension, and the same classifier reaches 99.3%. The projector was trained on these classes and the backbone never saw them, so the pair measures the training and not the backbone; Table 6 is the test on datasets neither has seen.

E Session Training

E.1 Choosing the number of prototypes

We build a prototype set in two steps. We first place a starting set of prototypes, then we reduce it. The number that survives is K_b.

Placing the starting set. We seed six directions per class by spherical k-means++ on that class's embeddings, using fewer when the class holds fewer frames. In the first session we then fit the directions $\boldsymbol{\mu}_{bj}$, concentrations κ_{bj} and weights π_{bj} by capped vMF expectation maximisation in double precision. We hold the directions on $\mathbb{S}^{d-1}$, we hold the weights summing to one, and we

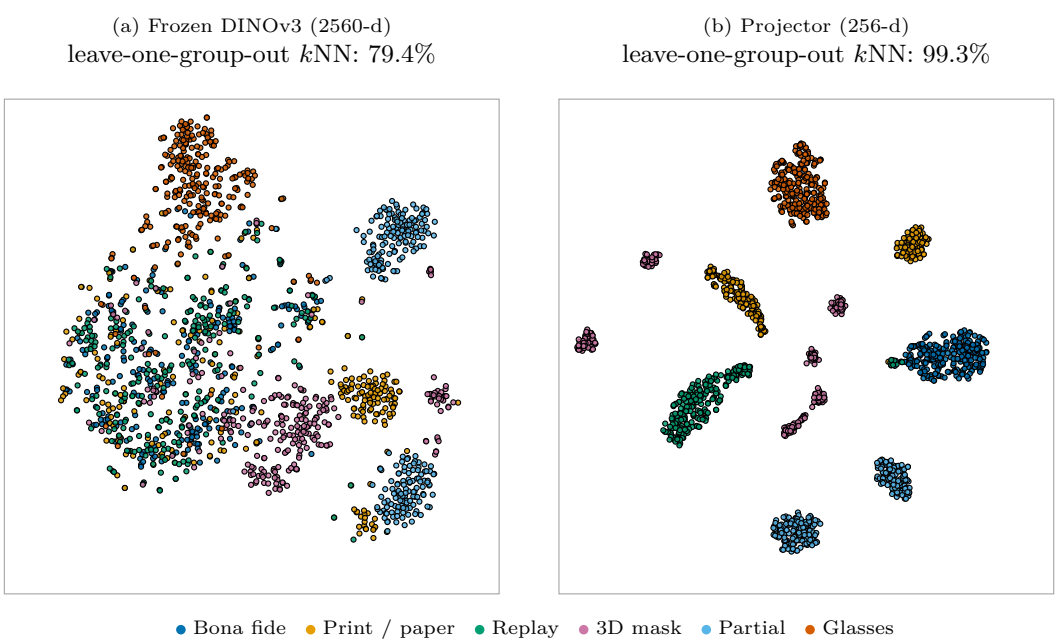


Figure 4: **What pre-training buys**, on held-out FFHQ-SynthFAS data. Each panel is a t-SNE map of the same frames from L_2-normalised features under identical settings, so only the representation differs. (a) The frozen backbone's own readout, its class token beside its patch-token mean (2,560-d). (b) The projector output $\mathbf{z}$ (256-d). The accuracy in each title is measured on the full-dimensional features, not on the map: a ten-nearest-neighbour classifier with every frame of one source face held out at a time, so a bonafide frame and the attacks made from it never vote for each other. Colour is the attack family; several clusters of one colour are the distinct classes inside it.

solve each concentration inside the range over which we evaluate the vMF normaliser, which caps it. Expectation maximisation returns a local solution and we do not assume a global one.

In later sessions we refine the seeded set instead, by gradient descent, while we train the session expert. The refinement loss pulls each prototype toward the embeddings of its own class and pushes it away from the frozen prototypes of other classes and from the frozen bonafide prototypes. Every frozen prototype enters this loss as a constant, so no earlier set moves. Here we cap each new concentration at the largest concentration among the frozen prototypes of its group, and we start it at that group's median. A single prototype that out-concentrated every frozen one would dominate the log-sum-exp of Eq. 5 near its own direction, and could then take frames that an earlier session's set answers for.

Reducing the set. We then drop prototypes in two passes. Both read embeddings held out by identity and capture session, which we declare before the session starts together with a minimum number of held-out identities.

1. *Redundancy.* When two prototypes of a set point within a cosine of 0.85 of each other, we keep the one that more held-out embeddings select and drop the other.
2. *Contribution.* We drop a prototype when removing it does not lower the held-out log-likelihood of the set.

We repeat both passes until the set stops changing, and we always keep at least one prototype. K_b is the count that remains, and it freezes with the set. We reach it by dropping prototypes from one fitted set, not by fitting a separate mixture for every candidate count and comparing them.

E.2 Frozen state

Table 8 lists what a session freezes and what a later session may still append to.

Table 8: Frozen state. Frozen rows describe individual items at introduction; their collections can grow by appending new items. The current bonafide collection itself grows by appending immutable prototype sets.

Frozen item	Later sessions
Expert $\mathcal{T}_t$	Parameters frozen at introduction
Attack density $\{\boldsymbol{\mu}_{bj}, \kappa_{bj}, \pi_{bj}\}_{j=1}^{K_b}$	Component count and all parameters frozen
Bonafide reference $\mathcal{K}_b$	Every bonafide prototype set frozen at that time, copied by value; all copied parameters frozen
Insertion index idx_b and session expert $\mathrm{expert}(b)$	Assigned at introduction; frozen
Reporting row $\widetilde{a}_b, \delta_b$	Fitted before introduction; frozen
Current bonafide collection	One prototype set appended per contributing session; inherited sets bit-identical; score is the best set's score

E.3 Session cost

One pass of $\mathcal{D}_t$ through the frozen backbone and the frozen projector yields every patch grid and every embedding the session uses. Neither receives gradients, so the session needs no second pass over the images. Each fit afterwards reads the d-dimensional embeddings alone, and in the first session one expectation-maximisation iteration over a K-component mixture costs $O(|\mathcal{D}_t|Kd)$. In later sessions the refinement adds no pass of its own, because we add its loss to the expert's loss and train the two together.

Gradients reach the session expert and this session's own prototypes, and nothing else. Every frozen prototype set enters the loss as a constant, which is why the backbone, the projector and every earlier session's state stay bit-identical. The expert's parameter count is fixed and does not grow with t.

A session leaves behind one expert, K_b directions and $2K_b$ scalars per admitted set, and the calibration data behind its reporting rows.

F Implementation Details

Table 9 gives the training and routing settings. The projector is pre-trained once and frozen across both protocols. From the second session onward, candidate attack prototypes are refined and checked for coverage before admission.

Table 9: Training settings of every run behind the result tables.

Stage	Setting	Value
Projector pre-training	Optimizer	Adam ($\beta = 0.9, 0.999$), projector learning rate 7×10^{-4}, weight decay 5×10^{-4}, seed 1024
	Batch, schedule	global batch 1,792 over 8 GPUs; 50-epoch run, learning rate halved every 20 epochs
	Input	224 px, data augmentation on
	Readout	Joint Token Attention over 33 states, one shared scorer; $d = 256$
	Objective	$S = 6$ subcentres; $\lambda_{\mathrm{pres}} = 0.05$, $\lambda_{\mathrm{inter}} = 0.5$
	End of training	drop any subcentre holding under 5% of its class's mass
Session expert	Optimizer	Adam, learning rate 10^{-4}, weight decay 5×10^{-4}, seed 1024
	Head	hidden width 512, dropout 0.1, input 224 px, data augmentation on
Prototype sets	Proxy refiner	SGD, learning rate 0.05, momentum 0.9, refine weight 0.5
	Refiner terms	repel weight 1.0 (margin 0.1), spread weight 0.5 (threshold 0.5), bonafide repel weight 1.0 (threshold 0.1), $\sigma_\kappa^2 = 0.05$
	Coverage	coverage threshold 0.7, attack coverage 0.5, prune threshold 0.85, minimum assignment fraction 0.05; uncovered tail $\theta = 0.1$ with at least 256 frames
Router calibration	Optimizer	1,500 steps, batch 256, learning rate 2×10^{-3} to 10^{-4}; $\sigma_a^2 = 1.0$
	Loss weights	new-session supervision 1.0; earlier attacks 0.5, on raw embeddings (S_2+)

G Evaluation Details

Scoring. PRIME scores each frame by Eq. 3. In Case 1, the firing experts provide the score; their predictions are averaged only when more than one expert fires. In Cases 0 and 2, the router provides the score. Expert and router scores are not blended for a frame. The case tags do not use a fitted threshold. We compute HTER at each target set's equal-error threshold and AUC from the video scores. The continual metrics are defined in Appendix H.2.

Protocol 1 comparison (Table 2). Each OCIM run learns its three sources in the order named by the chain, so OMI→C learns OULU-NPU, then MSU-MFSD, then Idiap Replay-Attack; the fourth is held out. The other methods use the published settings shown in the table, grouped by training exposure. Bold and underline rank rows without target data, including PRIME.

Protocol 2 and the unseen sets (Tables 5 and 3). The first group of Table 3 reproduces the reported HTER and AUC from Zhang et al. (2025); those methods train on CelebA-Spoof alone. Our rows use a frozen DINOv3 ViT-H/16+ backbone. The CelebA-Spoof-only control is a separate PRIME run: it trains CelebA-Spoof as S_1 and scores expert T_1 without routing. Full PRIME and sequential fine-tuning use all three Protocol 2 sources; PRIME is scored after S_3, while sequential fine-tuning updates one expert head without routing. *Avg.* is the mean over the five sets.

Routing rules (Table 4). NCM, FeCAM and RanPAC are updated once per Protocol 2 session. In this table, each method reads the same frozen projector output $\mathbf{z}$. HTER is measured after S_3 and $mAGA$ across the sessions on SiW-Mv2 and ROSE-Youtu. *Reversed frames* pools adjacent-session transitions on those two sets: for a baseline, an attack decision becomes bonafide; for PRIME, an inherited Case 1 route is lost. *Abstains* denotes PRIME's Case 2.

Case tags (Tables 5 and 11). Case 2 is unknown (no gate fires and $s_0^t(\mathbf{z}) < 0$), Case 0 is tagged bonafide (no gate fires and $s_0^t(\mathbf{z}) \geq 0$), and Case 1 is expert-routed (a gate fires). Shares are percentages within the bonafide or attack block, and moves count transitions from the previous session. A new bonafide set permits $2 \to 0$; a new attack gate permits $2 \to 1$ or $0 \to 1$. The shaded $1 \to 0$, $1 \to 2$ and $0 \to 2$ moves are forbidden by the routing rules. A printed 0 is exact; < 0.1 marks a smaller nonzero percentage. Protocol 1 pools the four OCIM held-out chains; Protocol 2 uses SiW-Mv2 and ROSE-Youtu, neither used for training.

Linear probe (Table 6). Rows 1 and 2 read frozen DINOv3 and differ only in the corpus the head is trained on, so their gap is the training data's. Rows 3 and 4 replace those features with a projector's 256-dimensional output on the same corpus as row 1: row 1 against row 3 is what the projector adds, and row 3 against row 4 is what the backbone's pre-training contributes. Row 3 is PRIME's frozen projector (§3.1). All probes use the same training and video-scoring procedure, with a fixed 0.5 decision threshold.

Single-source transfer (Table 10). A PRIME entry is a first session trained on the source alone and scored by the final system. The baseline values are as reported by Srivatsan et al. (2023); their rows train with CelebA-Spoof as supplementary data, and several of them adapt to unlabeled target data.

Protocol 1 continual metrics (Table 12). The metrics follow Eq. 7 of Cai et al. (2023) and Eq. 8; each printed value averages the four OCIM chains. mAA and $mABT$ use held-out slices of the learned datasets, while $mAGA$ uses SiW-Mv2 and ROSE-Youtu after each session. CASIA-FASD is omitted from mAA and $mABT$ when it has no usable held-out slice. The MSU-MFSD slice is video-disjoint from training but shares subjects. Joint training has no backward change; sequential fine-tuning and LwF (Li & Hoiem, 2018) update a single expert head. NCM, FeCAM and RanPAC update without earlier-session data and use the frozen backbone's pooled features here, unlike the projector features used in Table 4.

H Additional Results

Single-source transfer. Table 10 trains PRIME's first session on one OCIM source and scores each of the other three datasets.

Table 10: Single-source HTER (%) on OCIM: PRIME's first session trained on one dataset and tested on another. Baselines are as reported by Srivatsan et al. (2023), whose rows also train on CelebA-Spoof; a dash marks a pair they do not report. Bold and underline mark the best and second best per column.

Method	C → I	C → M	C → O	I → C	I → M	I → O	M → C	M → I	M → O	O → C	O → I	O → M	Avg.
ADDA [CVPR 2017]	41.8	36.6	–	49.8	35.1	–	39.0	35.2	–	–	–	–	39.6
DRCN [ECCV 2016]	44.4	27.6	–	48.9	42.0	–	28.9	36.8	–	–	–	–	38.1
DupGAN [CVPR 2018]	42.4	33.4	–	46.5	36.2	–	27.1	35.4	–	–	–	–	36.8
KSA [TIFS 2018]	39.3	15.1	–	12.3	33.3	–	9.1	34.9	–	–	–	–	24.0
DR-UDA [TIFS 2021a]	15.6	9.0	28.7	34.2	29.0	38.5	16.8	3.0	30.2	19.5	25.4	27.4	23.1
MDDR [CVPR 2020]	26.1	20.2	24.7	39.2	23.2	33.6	34.3	8.7	31.7	21.8	27.6	22.0	26.1
ADA [ICB 2019]	17.5	9.3	29.1	41.5	30.5	39.6	17.7	5.1	31.2	19.8	26.8	31.5	25.0
USDAN-Un [PR 2021]	16.0	9.2	–	30.2	25.8	–	13.3	3.4	–	–	–	–	16.3
GDA [ECCV 2022]	15.10	5.8	–	29.7	20.8	–	12.2	2.5	–	–	–	–	14.4
CDFTN-L [AAAI 2023]	**1.7**	8.1	29.9	11.9	9.6	29.9	8.8	**1.3**	25.6	19.1	5.8	6.3	13.2
FLIP-V [ICCV 2023]	15.08	13.73	12.34	4.30	9.68	7.87	0.56	3.96	4.79	2.09	5.01	6.00	7.12
FLIP-IT [ICCV 2023]	12.33	15.18	7.98	1.12	8.37	6.98	**0.19**	5.21	4.96	**0.16**	4.27	5.63	6.03
FLIP-MCL [ICCV 2023]	10.57	7.15	3.91	0.68	7.22	4.22	**0.19**	5.88	3.95	0.19	5.69	8.40	4.84
PRIME	2.29	**2.86**	**0.69**	**0.00**	**2.86**	**0.71**	0.68	2.86	**0.56**	0.56	**3.57**	**2.86**	**1.71**

H.1 Protocol 1 Routing Cases

Table 11 pools the four OCIM held-out sets. The attack-frame Case 1 share rises from 84.7% after S_1 to 93.5% after S_3. The 4.9% of bona fide frames routed to an expert at S_1 retain that route, while the Case 2 share falls from 6.9% at S_2 to 3.9% at S_3 through a $2 \to 0$ move. The forbidden moves remain zero across all three sessions.

Table 11: Protocol 1 case tags after each session, pooled across four OCIM held-out sets (% of each bona fide or attack block). Dashes mark the first session; shaded columns are forbidden moves. Case definitions and transition semantics are in Appendix G.

	Share of frames			Moves since the previous session					
				allowed			forbidden (must be 0)		
Session	Unknown Case 2	Bona fide Case 0	Attack Case 1	$2 \to 0$	$2 \to 1$	$0 \to 1$	$1 \to 0$	$1 \to 2$	$0 \to 2$
Four OCIM held-out sets pooled									
Bona fide frames									
S_1	7.4	87.7	4.9	–	–	–	–	–	–
S_2	6.9	88.2	4.9	0.6	0	0	0	0	0
S_3	3.9	91.2	4.9	3.0	0	0	0	0	0
Attack frames									
S_1	10.8	4.5	84.7	–	–	–	–	–	–
S_2	8.3	2.2	89.5	0	2.5	2.3	0	0	0
S_3	4.3	2.2	93.5	0	4.0	0	0	0	0

H.2 Continual-Learning Metrics for Protocol 1

We follow Eq. 7 of Cai et al. (2023) over the sessions $S_1, \ldots, S_T$, with $T = 3$ in every chain. The pre-training stage ships no expert and no prototype set, so no system exists before S_1 and every sum starts at $t = 1$. $R_{i,t}$ is the AUC after session i on the held-out slice of the dataset learned at session t

($i \geq t$), and R_t^u the AUC after session t on an unseen set $u \in \mathcal{U} = \{\text{SiW-Mv2},\ \text{ROSE-Youtu}\}$:

$$
\begin{aligned}
mAA &= \frac{1}{T}\sum_{t=1}^{T}\frac{1}{T-t+1}\sum_{i=t}^{T}R_{i,t}, \\
mABT &= \frac{1}{T-1}\sum_{t=1}^{T-1}\Big(\frac{1}{T-t}\sum_{i=t+1}^{T}R_{i,t}-R_{t,t}\Big), \\
AGA_u &= \frac{1}{T}\sum_{t=1}^{T}R_t^u, \qquad mAGA = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}AGA_u.
\end{aligned}
\tag{8}
$$

A dataset with no held-out slice leaves both mAA and $mABT$, and every printed value is the mean over the four chains. Appendix G gives the slices.

Table 12: Protocol 1 continual metrics (AUC, %), averaged over the four OCIM chains: mAA and $mABT$ on the seen datasets, $mAGA$ on SiW-Mv2 and ROSE-Youtu. Every row uses PRIME's frozen backbone; joint training has no backward change (–). Details in Appendix G.

	Seen session datasets		Unseen sets
Method	$mAA\uparrow$	$mABT\uparrow$	$mAGA\uparrow$
Reference rows on the same frozen backbone			
Joint training (joint reference)	99.84	–	97.71
Sequential fine-tuning (floor)	99.81	−0.05	96.68
Continual baselines on the same frozen backbone			
LwF [TPAMI 2018]	99.65	−0.04	96.38
NCM [NeurIPS workshop 2022]	90.46	−0.28	73.59
FeCAM [NeurIPS 2023]	99.03	−0.26	83.30
RanPAC [NeurIPS 2023]	99.83	0.02	74.52
PRIME	99.97	−0.01	98.63

I Dataset details

I.1 What the dataset contains

FFHQ-SynthFAS contains 95,360 face images: 47,680 presentation attacks and 47,680 genuine images, across 23,960 identities. We take one FFHQ photograph (Karras et al., 2019) per identity as the source. We split the dataset into 76,142 training images and 19,218 test images, and no identity appears in both splits.

We generate two pairs from each source photograph. A pair is one attack image and one genuine image of the same person. We draw the prompt axes again for the second pair, so the two attacks of one photograph are independent draws and usually name different attack categories. A few generations fail the checks in §I.2 and we drop them, which is why the totals sit a little under two pairs per photograph.

We make the genuine image of a pair with the same generator, from the same source photograph, in the same scene, under the same light and on the same camera as its attack. Only the material differs. A model therefore cannot separate the two classes by the room, the pose, the lighting, the camera or the generator. It has to read the material.

We release the generated images only, and not the FFHQ source photographs.

I.2 How we make one pair

We start from a face crop of the FFHQ photograph. We draw one value for each prompt axis, then write two prompts from that draw: one for the attack and one for its control. Four axes are shared: we write the same drawn value into both prompts. We send each prompt with the source crop to the

`gpt-image-2` edit endpoint (OpenAI, 2026) and ask for about one megapixel, keeping the aspect ratio of the crop.

Two checks run on every returned image. The first decodes the image, compares its size against the request and rejects a constant-colour frame. The second measures the mean absolute grey-level difference against the input and rejects anything below 4.0 grey levels on the 0–255 scale, because a near-copy of the input filed as an attack teaches the model nothing. We retry a refusal up to three times, then drop the item.

I.3 Attack categories

We group the twelve categories into three families. Two-dimensional attacks hold a flat medium in front of the camera: print, replay and paper_cutout (Figure 5). Partial attacks cover one region of the face and leave the rest as live skin: partial_eye, partial_mouth and funny_eye_glasses (Figure 6). Three-dimensional attacks are worn or moulded objects with volume: cloth_mask, three_d_mask, mannequin, latex, rubber and silicone (Figure 7).

Table 13 gives the share of the attack half. We give print and replay the largest shares on purpose: these two attacks occur most often in the field, and every earlier benchmark carries many of them. The other ten categories take near-equal shares, between 6.6% and 7.8%.

I.4 The prompts

We write every prompt in five blocks. SCENE names the attack object and the situation. KEEP locks the identity. SURFACE names the material. LIGHT says how that material takes light. CAPTURE sets the camera. The control prompt uses the same five blocks and the same KEEP text; its SCENE says the person is photographed live, and its SURFACE names live skin.

We do not use one fixed sentence per category. Each category has 3 to 11 axes, and we draw a value for each one. Four axes are shared with the control:

- **capture context**: candid, webcam station, phone selfie, ID booth.
- **lighting**: nine values, from indoor office to outdoor backlit.
- **camera**: laptop webcam, phone front, phone rear, kiosk camera, old webcam.
- **capture defect**: none, soft focus, motion blur, underexposed, overexposed, noisy.

The candid value names no scene, so the prompt keeps the pose and the room of the source photograph. The other three values move the head and name a room, so we drop the clause that claims the pose and the room are unchanged. A prompt that names a new scene and also claims the old one stays makes the model choose, and it chooses the scene.

We show three pairs below, one per prompt family: a printed photograph, a printed paper strip and a worn mask. The text is what we sent to the API. The generator does not return the same image twice for the same prompt, so these prompts reproduce the described attack but not these exact images.

I.4.1 print

A person holds a paper print of the subject in front of the camera. The print fills most of the frame, so almost every pixel of the face is ink.

Attack prompt.

> **SCENE:** A picture of a picture. This exact photograph was printed as ink on letter-size thin matte copier paper by a cheap office colour copier, creased and a little wrinkled, leaning back so its top edge is a little further away, and is now being photographed a second time. The sheet is held up and covers most of the frame — the face, clothing and background on it are all ink — but two of its cut edges and the fingertips gripping them are in shot, with the real room behind out of focus behind. In the picture they are sitting at a desk and looking straight into the camera, head square to the lens, shoulders level, face centred, with a plain painted wall close behind them and no expression.

Table 13: Attack categories in FFHQ-SynthFAS. Shares are of the 47,680 attack images. Every attack image has one genuine image of the same person, so the genuine half comes to the same 47,680 images.

Category	Images	Share	Train	Test
replay	7,481	15.69%	5,965	1,516
print	7,160	15.02%	5,668	1,492
partial_eye	3,704	7.77%	2,978	726
funny_eye_glasses	3,363	7.05%	2,705	658
mannequin	3,346	7.02%	2,635	711
paper_cutout	3,329	6.98%	2,651	678
cloth_mask	3,327	6.98%	2,725	602
three_d_mask	3,230	6.77%	2,646	584
partial_mouth	3,208	6.73%	2,579	629
rubber	3,185	6.68%	2,524	661
latex	3,182	6.67%	2,500	682
silicone	3,165	6.64%	2,495	670
Total	47,680	100%	38,071	9,609

> **KEEP:** The same person: same face shape and width, same nose, same eye spacing, same hairline, same age, the same uneven teeth, the same asymmetry, the same moles and marks in the same places.
>
> **SURFACE:** On the printed sheet only — Second-generation ink, not skin: flat blocked colour with no pores and no fine texture, individual hair strands merged into solid dark shapes, heavy contrast with the shadows gone to one flat dark mass, a heavy warm orange cast over the whole sheet, faint horizontal printer banding and visible paper grain.
>
> **LIGHT:** The sheet is matte and takes no reflections from even office LED ceiling lighting; brightness is even right across the sheet, face and background alike.
>
> **CAPTURE:** Shot with a phone rear camera, a trace of motion blur from an unsteady hand.

Matched control prompt.

> **SCENE:** A close-up photo of this same real person on a different camera, photographed live. They are sitting at a desk and looking straight into the camera, head square to the lens, shoulders level, face centred and filling the frame, with a plain painted wall close behind them and no expression.
>
> **KEEP:** The same person: same face shape and width, same nose, same eye spacing, same hairline, same age, the same uneven teeth, the same asymmetry, the same moles and marks in the same places.
>
> **SURFACE:** Real skin, not ink: visible pores and fine skin texture, small blemishes and uneven tone, individual hairs catching the light, and real depth between the face and what is behind it.
>
> **LIGHT:** Even office LED ceiling lighting models the face in three dimensions — wrapping around the nose and brow, a soft shadow under the chin — with the background falling away at its own distance behind.
>
> **CAPTURE:** Shot with a phone rear camera, a trace of motion blur from an unsteady hand.

I.4.2 PARTIAL EYE

Somebody wears a printed paper strip of the subject's eyes over their own eyes. Only that band is paper; the rest of the face stays live skin.

Attack prompt.

> **SCENE:** Somebody else has a printed paper strip of this exact face over the eyes, doing a face-spoofing attack. The paper is cut to a band across both eyes and the brow, with two small holes over the pupils, and it is worn against the face, hooked behind the ears. In the

picture they are sitting at a desk and looking straight into the camera, head square to the lens, shoulders level, face centred, with a plain painted wall close behind them and no expression.

KEEP: The strip carries a life-size print of this person's eyes and brows and nothing else — same eye shape, same eye spacing, same brows — printed at exactly the size of the wearer's own eyes and lying directly over them, so the two sets of eyes land on each other. The print stops at the cut edges: above the strip is the wearer's own forehead, below it their own cheeks and mouth, all live skin, and their own nose runs on down from under the strip as one unbroken nose. The person wearing it is somebody else, with their own hair, ears, neck and shoulders.

SURFACE: Solid opaque paper you cannot see through — it completely hides the skin behind it. Flat printed ink, a warm orange cast, no pores, lighter and less red than the live skin around it, so its edges stand out against the face. Cut edge clean, with a small paper tab left at one side, with a thin hard shadow on the skin below. Their own eyes show through the small holes, wet and glossy against the flat paper. The flat sheet cannot follow the face: it creases hard and bridges straight across the hollows.

LIGHT: The paper is matte and takes no reflections from even office LED ceiling lighting; it sits evenly lit and noticeably flatter than the live skin around it.

CAPTURE: Shot with a phone rear camera, a trace of motion blur from an unsteady hand.

Matched control prompt.

SCENE: A close-up photo of this same real person, re-photographed with the faint marks a pair of glasses left on the nose. They are sitting at a desk and looking straight into the camera, head square to the lens, shoulders level, face centred and filling the frame, with a plain painted wall close behind them and no expression.

KEEP: The same person: same face shape and width, same nose, same eye spacing, same hairline, same age, the same uneven teeth, the same asymmetry, the same moles and marks in the same places.

SURFACE: Live skin: visible pores and fine texture, small blemishes and uneven tone, fine hairs at the hairline and brow, the eyes wet and set into the face.

LIGHT: Even office LED ceiling lighting models the face in three dimensions — wrapping around the nose and brow, a soft shadow under the chin — and the same light carries on across the whole face and onto the neck and ears without a break.

CAPTURE: Shot with a phone rear camera, a trace of motion blur from an unsteady hand.

I.4.3 SILICONE

Somebody wears a moulded silicone mask of the subject. The material has thickness, a seam and its own way of taking light.

Attack prompt.

SCENE: A different person — a different build, with their own neck, ears and shoulders — has pulled an SPFX silicone disguise mask of this exact face on over their own head, doing a face-spoofing attack. It is one whole piece of rubbery skin, pulled on like a swimming cap, covering the face, the ears and the neck together. The wearer's own real eyes look out through small cut eye holes, with a dark rim of shadow all around each one, and no eyelashes at all. It sits very slightly off-centre on the head, so it is closer to one ear than the other. The mask's edge runs in front of the ears, so the wearer's own ears sit outside it. The mask's crown is bare shell with only a thin painted hairline. Both hands are out of frame. In the picture they are sitting at a desk and looking straight into the camera, head square to the lens, shoulders level, face centred, with a plain painted wall close behind them and no expression.

KEEP: The mask is moulded and painted as this same person — same face shape and width, same nose, same eye spacing, the same moles and marks in the same places. It is a copy, and a poor one: the cheeks and jaw are too full and heavy for this face, and the skull is wider and rounder than the head inside it. The face is inert: no expression at all. The person wearing it is somebody else: their own neck, ears and shoulders are live skin in a clearly different tone from the mask face, and the line where the two meet is visible.

SURFACE: A worn object, not a face. It has thickness: the eyelid and lip edges are a few millimetres thick at the openings, and the lips sit doubled on top of the wearer's own. The wrinkles are moulded in, deeper and more even than real creases. The surface is not smooth:

fine grain shows across it. It is skin-coloured, not grey, but painted: the blush sits in soft patches that do not follow the bone, and the face is a different shade from the live skin past that edge.

LIGHT: Even office LED ceiling lighting lies on the mask as broad soft patches of gloss on the forehead, cheekbones, nose and chin — large smooth areas of sheen, never the small bright pinpoints live skin makes.

CAPTURE: Shot with a phone rear camera, a trace of motion blur from an unsteady hand.

Matched control prompt.

SCENE: A close-up photo of this same real person, re-photographed with a little shine on the forehead and nose. Both hands are out of frame. They are sitting at a desk and looking straight into the camera, head square to the lens, shoulders level, face centred and filling the frame, with a plain painted wall close behind them and no expression.

KEEP: The same person: same face shape and width, same nose, same eye spacing, same hairline, same age, the same uneven teeth, the same asymmetry, the same moles and marks in the same places.

SURFACE: Live skin: pores that vary in size across the face, fine hairs at the hairline and brow, small blemishes and uneven colour, extra redness at the cheeks, nose and ears, and the eyes wet and set into the face.

LIGHT: Even office LED ceiling lighting models the face in three dimensions and passes a little way into the skin, so the ears and the edge of the nose glow warm and red where the light comes through them; the same light carries on across the neck, the ears and the shoulders without a break or a seam anywhere.

CAPTURE: Shot with a phone rear camera, a trace of motion blur from an unsteady hand.



Figure 5: Two-dimensional attacks in FFHQ-SynthFAS. The medium is a flat surface held in front of the camera: a printed sheet, a screen, or a printed mask cut to the face. Each row holds one identity. Column 1 is the FFHQ source photograph, column 2 is a generated genuine control, and the remaining columns are the attack categories. We draw the scene, the lighting and the camera once per pair and write them into the attack prompt and into its control prompt alike, so these factors carry no information about the label. We generate the images shown here for this figure, with the pipeline the dataset uses, so that one face can show every category of the family; in the dataset each identity carries two categories, each with its own control. We sample the prompt axes at random and we do not select images by appearance. A replay attack therefore leaves only a weak cue in some draws, as it does in a face crop of a real replay attack. We mark Paper Cutout as 2D/3D: the piece is flat ink and cannot take the shape of the face, which puts it with the two-dimensional attacks, but somebody wears it.



Figure 6: Partial attacks in FFHQ-SynthFAS. The attack covers one region of the face and the rest of the face stays live skin, so the cue sits at the cut edge where the printed piece meets the skin. Each row holds one identity. Column 1 is the FFHQ source photograph, column 2 is a generated genuine control, and the remaining columns are the attack categories. We draw the scene, the lighting and the camera once per pair and write them into the attack prompt and into its control prompt alike, so these factors carry no information about the label. We generate the images shown here for this figure, with the pipeline the dataset uses, so that one face can show every category of the family; in the dataset each identity carries two categories, each with its own control. We sample the prompt axes at random and we do not select images by appearance. The piece is life-size and lands on the region it copies, so the attack does not change the face; it changes the material over one part of it.



Figure 7: Three-dimensional attacks in FFHQ-SynthFAS. The attack is an object with volume, worn over the head or, for the mannequin, photographed on its own, so the cue is the material and the way it takes light. Each row holds one identity. Column 1 is the FFHQ source photograph, column 2 is a generated genuine control, and the remaining columns are the attack categories. We draw the scene, the lighting and the camera once per pair and write them into the attack prompt and into its control prompt alike, so these factors carry no information about the label. We generate the images shown here for this figure, with the pipeline the dataset uses, so that one face can show every category of the family; in the dataset each identity carries two categories, each with its own control. We sample the prompt axes at random and we do not select images by appearance. A rigid shell carries a quieter cue than a painted or a glossy one, so the 3D mask reads as a smoothed face in some draws.