# META-LEARNING FOR AI-GENERATED IMAGE DETECTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Detectors of AI-generated images are usually trained once and then deployed unchanged. Their accuracy drops on generators that appear after training. We recast detection as a meta-learning problem. The detector conditions every prediction on a small labelled support set from the current deployment context instead of fixing a decision boundary at training time. We give a probabilistic account of how training episodes are built. The standard *explicit* construction keeps one class for real images and one class per generator. It needs every generator to be known and labelled in advance, as if given by an oracle. It also never pairs a query with support images of another generator, so it does not reward generalisation to a new generator. We propose an *implicit* construction over the binary real and AI-generated label space. Episode diversity then comes from a prior over the class-conditional distributions rather than from the choice of classes. This prior is stated apart from the data, and we give two instances of it. An empirical prior draws the class-conditional distributions from the domains of the training data. A synthetic prior draws them from Gaussian mixtures in embedding space and needs no images at all. We cast twelve meta-learning methods into this framework. With a nearest-centroid rule, they give thirteen context-aware detectors, which we compare with eleven static detectors. Our implicit construction degrades least when the support set comes from generators other than the query's. In this case, it matches or improves on the oracle explicit construction. Conditioning on the support set matters most on in-the-wild images, where static detectors lose their accuracy. A detector trained on the synthetic prior alone detects these images better than a zero-shot static detector.

## 1 INTRODUCTION

Diffusion models (Rombach et al., 2022; Ramesh et al., 2022; Saharia et al., 2022), rectified-flow transformers (Esser et al., 2024; Black Forest Labs et al., 2025), and autoregressive models (Tian et al., 2024; Han et al., 2025) now produce photorealistic images on demand. Fabricated evidence, non-consensual imagery, and political disinformation have thus become cheap to produce. Telling a photograph from an AI-generated image is therefore a prerequisite for trust in visual media.

**Static detectors.** Detectors built for this task are typically trained once on a pooled mixture of real and AI-generated images. They are then applied without further adaptation (Wang et al., 2020; Corvi et al., 2023; Zhong et al., 2024; Yan et al., 2025a), so what they learn is a *static decision boundary*. That boundary is anchored to the signatures of the training generators. Generative adversarial networks (GANs) imprint periodic spectral peaks through their upsampling filters, diffusion models leave structured noise residuals, and autoregressive generators produce discontinuities at the boundaries of their image tokens (Frank et al., 2020; Corvi et al., 2023; Esser et al., 2021). A new generator or a new image-processing pipeline need not show these signatures at all. More training data does not remove this problem. On million-scale benchmarks, detectors that are accurate on their training generators still lose much accuracy on held-out ones (Zhu et al., 2023; Pellegrini et al., 2025). Robustness to new generators is therefore the key requirement of a practical detector.

**Detection as meta-learning.** The difficulty is structural rather than statistical. Supervised training assumes that all images come from one fixed distribution. Every new generator violates this assumption. Retraining for each new generator is not possible, because in practice we do not know when



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.

one has appeared. In deployment, however, a detector rarely works without context. Its user usually has a handful of labelled images from the current deployment context, which we call the *support set*. The user wants a verdict on a further unlabelled image, which we call the *query*. This suggests reframing AI-generated image detection as a meta-learning problem (Finn et al., 2017; Snell et al., 2017; Vinyals et al., 2016). Instead of fixing a decision boundary at training time, the detector learns a procedure that reads the support set and adapts the boundary without parameter updates.

**Episode construction.** Adopting this view does not by itself solve the problem. Everything now rests on how training episodes are built, that is, on the prior from which support sets and queries are drawn. The standard construction is *explicit*. Real images form one class and every generator forms a class of its own. Episodes differ by the classes they sample (Wu et al., 2025b). Two difficulties follow. First, a generator absent from training has no class of its own in this label space. Nothing in the training objective rewards a detector for coping with generators outside the enumerated set. Yet this is the case that motivates the problem. Second, the construction needs the generator of every image, including the support images at test time. Only an oracle that knows every generator in advance can provide these labels. A user who collects a support set can usually guess whether an image is real or AI-generated, but rarely knows which generator produced it. What is needed instead is an episode prior over the two classes that actually matter. Such a prior has to supply variability without enumerating generators. Once it is stated separately from the data it draws on, a further question becomes available. Does the prior have to be made of images at all?

**Contributions.** The contributions of this paper are the following.

- We frame AI-generated image detection as a meta-learning problem. We give a probabilistic account of the *explicit* episode construction and identify the task prior and the class-conditional distributions it induces. That account states why this prior is ill-suited to detection. It enumerates a fixed set of generators, so a new generator has no class, and it needs oracle generator labels.
- This account lets us reformulate the prior into our *implicit* episode construction that acts only on the two classes of interest. Diversity no longer comes from selecting classes but from a prior placed directly on the class-conditional distributions. A new generator then needs no class of its own, and every episode poses a real-versus-AI-generated problem.
- Separating the prior from the data lets us ask whether the prior needs images at all. We define a fully synthetic prior over Gaussian mixtures in embedding space, under which the detector can be trained without observing a single image.
- Building on this framework, we cast twelve meta-learning methods of four paradigms into the context of AI-generated image detection. We compare thirteen context-aware detectors, which condition on the support set, with seven context-free and four domain-aware static detectors. The benchmarks are AI-GenBench (Pellegrini et al., 2025) and Chameleon (Yan et al., 2025a), and the test generators are held out by synthesis mechanism.
- Our implicit construction degrades least when the support set comes from generators other than the query's, and it then matches or improves on the oracle explicit construction. The support set pays off most on in-the-wild images, where static detectors fail. Some Transformer-based detectors, however, ignore the support set.

## 2 Problem Formulation

**Data.** Let $\mathcal{X}$ denote the space of images and let $\mathcal{Y} = \{0, 1\}$ be the binary label space, where $y = 0$ indicates a real, camera-captured image and $y = 1$ indicates an AI-generated image. We assume access to a labelled dataset

$$\mathcal{D} = \big\{(\mathbf{x}_n, y_n, d_n)\big\}_{n=1}^{N}, \quad \mathbf{x}_n \in \mathcal{X},\ y_n \in \mathcal{Y},\ d_n \in \{1, \dots, M_{y_n}\},$$

Beyond the binary label, each image carries a domain index $d_n$ that records its provenance. For a real image ($y_n = 0$), the domain identifies the photographic context, such as the camera sensor, the resolution, the scene content, and the post-capture processing. It ranges over $M_0$ values. For an AI-generated image ($y_n = 1$), the domain identifies the generative model that produced the image, which we call its *generator*. It ranges over $M_1$ values. The generators span GANs (Goodfellow et al., 2014), diffusion models (Ho et al., 2020; Song & Ermon, 2019), and autoregressive image generators (van den Oord et al., 2016; Esser et al., 2021). We index the dataset by class as $\mathcal{D}_y \coloneqq \{\bar{\mathbf{x}} : (\bar{\mathbf{x}}, y, \bar{d}) \in \mathcal{D}\}$ with $N_y \coloneqq |\mathcal{D}_y|$, and by class–domain as $\mathcal{D}_{y,d} \coloneqq \{\bar{\mathbf{x}} : (\bar{\mathbf{x}}, y, d) \in \mathcal{D}\}$ with $N_{y,d} \coloneqq |\mathcal{D}_{y,d}|$. For a

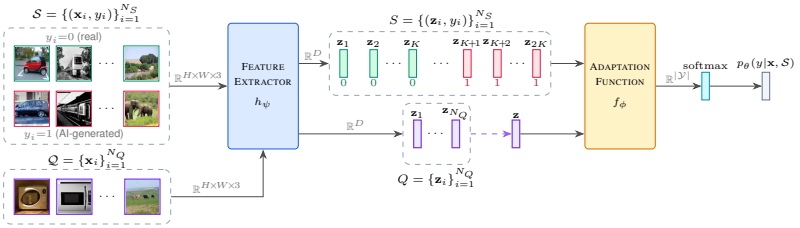



Figure 1: *Meta-learning framework for AI-generated image detection. Left:* A support set $\mathcal{S}$ holds $K$ real images ($y$=0, green border) and $K$ AI-generated images ($y$=1, red border), shown here for $N_C$=2 ways. The query set $\mathcal{Q}$ has a purple border. A shared feature extractor $h_\psi \colon \mathcal{X} \to \mathbb{R}^D$ maps every image to an embedding. This gives the labelled support embeddings $S = (S_1, S_2)$ and the labelled query embeddings $Q = (Q_1, Q_2)$, where $S_k$ and $Q_k$ collect the embedding–label pairs of the $k$-th class slot. *Right:* At each step, a *single* query embedding $\mathbf{z}$ from $Q$ is paired with the full support set $S$ and fed to the adaptation function $f_\phi$. A softmax of its output gives $q_\phi(y \mid \mathbf{z}, S)$ in (2). The query set $\mathcal{Q}$ enters only through the training objective (4).

label $y$ of the label space in use, $\mathcal{D}_y$ collects the images with that label. The domain index is never used as a label. It serves only to build diverse training episodes (Section 3).

**Static detection.** AI-generated image detection aims to learn a parametric predictive distribution $p_\theta(y \mid \mathbf{x})$ that reliably distinguishes real images from AI-generated ones. In practice, $p_\theta$ is realised by applying a softmax to the output of a neural network $f_\theta \colon \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$, i.e., $p_\theta(y \mid \mathbf{x}) = \mathrm{softmax}(f_\theta(\mathbf{x}))_y$ (Wang et al., 2020; Radford et al., 2021). During training, we optimise the parameters $\theta$ by minimising the expected negative log-likelihood over the training set:

$$\min_\theta \; \mathbb{E}_{(\mathbf{x},y)\sim p(\mathcal{D})} \left[ -\log p_\theta(y \mid \mathbf{x}) \right], \tag{1}$$

where $p(\mathcal{D})$ is the empirical data distribution. At inference time, the predicted label is $\hat{y} = \arg\max_{y\in\mathcal{Y}} p_\theta(y \mid \mathbf{x})$. Most recent detectors follow this template (Wang et al., 2020; Ojha et al., 2023; Corvi et al., 2023), and we call such a detector a *static detector*.

## 3 AI-Generated Image Detection as a Meta-Learning Problem

**Model.** Instead of a static detector $p_\theta(y \mid \mathbf{x})$, we seek a model that conditions each prediction on a support set $\mathcal{S}$. The support set holds $N_S = N_C K$ labelled images, $K$ for each of its $N_C$ classes. We call $N_C$ the number of *ways*[1]. The model is evaluated on a disjoint *query set* $\mathcal{Q}$ of $N_Q$ images. We decompose the model into two components (Figure 1). A feature extractor $h_\psi \colon \mathcal{X} \to \mathbb{R}^D$ maps each image to a $D$-dimensional embedding. Applying $h_\psi$ to the support and query images yields labelled support embeddings $S$ and labelled query embeddings $Q$, with $\mathbf{z} = h_\psi(\mathbf{x})$ for each image $\mathbf{x}$. We view both as tuples of per-class tuples, $S = (S_1, \ldots, S_{N_C})$ and $Q = (Q_1, \ldots, Q_{N_C})$. Here $S_k$ and $Q_k$ hold the support and query pairs $(\mathbf{z}, y)$ whose label matches the $k$-th class slot of the episode. We assume that $h_\psi$ maps two different images of a class to two different embeddings. Every embedding can then be identified with the image it came from. An adaptation function $f_\phi$ maps a query embedding and the labelled support embeddings to a predictive distribution over labels,

$$q_\phi(y \mid \mathbf{z}, S) = \mathrm{softmax}\big(f_\phi(\mathbf{z}, S)\big)_y. \tag{2}$$

The image-level predictor is then the composition

$$p_\theta(y \mid \mathbf{x}, \mathcal{S}) = q_\phi\big(y \mid h_\psi(\mathbf{x}), \{(h_\psi(\mathbf{x}_i), y_i)\}_{(\mathbf{x}_i,y_i)\in\mathcal{S}}\big), \tag{3}$$

where $\theta = (\psi, \phi)$. Different instantiations of $f_\phi$ recover the metric-based, optimisation-based, amortisation-based, and in-context learning paradigms of Section 4. The framework allows $h_\psi$ to be trained, but our experiments keep it frozen, so that all adaptation takes place in $f_\phi$.

**Episodic objective.** The main difference between (3) and (1) is that $p_\theta$ conditions on both the input and the support set. Since $q_\phi$ operates entirely in embedding space, we formulate training at the

[1]Under the implicit construction of Section 3.2, $N_C = 2$ and $N_S = 2K$.

embedding level. An *episode* is the pair $(S, Q)$ defined above, and $\pi(S, Q)$ denotes a distribution over episodes. The adaptation function is trained by minimising

$$\min_{\theta} \mathbb{E}_{(S,Q)\sim\pi}\left[-\frac{1}{N_Q}\sum_{(\mathbf{z},y)\in Q}\log q_\phi(y \mid \mathbf{z}, S)\right], \tag{4}$$

where $\sum_{(\mathbf{z},y)\in Q}$ ranges over all $N_Q = \sum_k |Q_k|$ entries of the per-class tuple. The episode distribution $\pi$ is the central modelling choice. It defines the tasks that $f_\phi$ must learn.

**Episode prior.** We consider two general ways to construct the episode distribution $\pi$ in (4). In both cases we introduce a task-level latent $\tau$ with prior $p(\tau)$ and factorise the joint distribution as

$$p(\tau, S, Q) = p(\tau)p(S \mid \tau)p(Q \mid S, \tau), \qquad \pi(S, Q) = \int p(\tau, S, Q)\, d\tau. \tag{5}$$

We call the conditional distribution $p(S, Q \mid \tau)$ of an episode a *task*. Two latent values that induce the same conditional distribution, up to a permutation of the class slots, define the same task. The two instantiations of Sections 3.1 and 3.2 differ in what $\tau$ represents and in $N_C$. They also differ in how $p(S \mid \tau)$ and $p(Q \mid S, \tau)$ split into a label part and a class-conditional embedding part.

### 3.1 Explicit episode construction

**Label space.** The explicit episode construction uses one label for real images and a distinct label for each generator. This gives the multi-class label space $\mathcal{Y}_{\text{ext}} = \{0, 1, \ldots, M_1\}$, where $y = 0$ denotes real images and $y = c$ the images of generator $c$. This is the standard $N_C$-way $K$-shot classification (Vinyals et al., 2016; Snell et al., 2017). Each episode samples $N_C$ of the $M_1{+}1$ classes and splits the images of these classes into a support set and a query set. This formulation has recently been adopted for AI-generated image detection (Wu et al., 2025b; Yao et al., 2026). The support set and the query set of an episode share the *same* $N_C$ classes, and only their images are disjoint. Episodes therefore differ only in the chosen classes. There is one task for each set of $N_C$ classes, so the number of tasks is $\binom{M_1+1}{N_C}$. More importantly, the construction never presents a query of a generator that the support set lacks. It therefore does not reward the detection of new generators. The construction also labels every support image by its generator. At test time, the label space must therefore hold a class for every generator that can occur, and the generator of every support image must be known. We call a label space with this knowledge an *oracle label space*. In deployment, a new generator is unknown before it appears, and in-the-wild images carry no generator label.

**Episode distribution.** Under this construction the task-level latent is $\tau = V = (V_1, \ldots, V_{N_C})$, an ordered tuple of distinct class indices, drawn from

$$p(V) = \frac{(M_1 + 1 - N_C)!}{(M_1 + 1)!}\,\mathbb{1}\big[V \in \mathcal{Y}_{\text{ext}}^{N_C},\ V_i \neq V_j\ \forall i \neq j\big]. \tag{6}$$

By construction, every $(\mathbf{z}, y) \in S_k \cup Q_k$ has $y = V_k$. The label is thus fixed by the class slot. For independent draws within a class, the support and query likelihoods read

$$p(S \mid V) = \prod_{k=1}^{N_C}\prod_{(\mathbf{z},y)\in S_k} p(\mathbf{z}, y \mid V_k), \qquad p(Q \mid S, V) = \prod_{k=1}^{N_C}\prod_{(\mathbf{z},y)\in Q_k} p(\mathbf{z}, y \mid V_k, S_k),$$

and each per-pair factor splits through the label indicator,

$$p(\mathbf{z}, y \mid V_k) = \mathbb{1}[y = V_k]\, p_\psi(\mathbf{z} \mid y),$$
$$p(\mathbf{z}, y \mid V_k, S_k) = \mathbb{1}[y = V_k]\, p_\psi(\mathbf{z} \mid y, S_k),$$

exposing the class-conditional embedding kernel and its $S_k$-restricted counterpart,

$$p_\psi(\mathbf{z} \mid y) = \frac{1}{N_y}\sum_{\bar{\mathbf{x}}\in\mathcal{D}_y}\delta\big(\mathbf{z} - h_\psi(\bar{\mathbf{x}})\big),$$

$$p_\psi(\mathbf{z} \mid y, S_k) = \frac{1}{N_y - |\bar{S}_k|}\sum_{\bar{\mathbf{x}}\in\mathcal{D}_y}\mathbb{1}\big[h_\psi(\bar{\mathbf{x}}) \notin \bar{S}_k\big]\,\delta\big(\mathbf{z} - h_\psi(\bar{\mathbf{x}})\big), \tag{7}$$

where $\bar{S}_k := \{\mathbf{z} : (\mathbf{z}, y) \in S_k\}$ collects the support embeddings of class $V_k$, $\delta(\cdot)$ is the Dirac delta on $\mathbb{R}^D$, and $\mathbb{1}[\cdot]$ is the Iverson indicator. Since two images of a class have two different embeddings, $|\bar{S}_k|$ counts the distinct support images of class $V_k$. The normaliser thus counts the remaining class-$V_k$ embeddings. The query kernel draws only images outside the support set, as Algorithm 1 of (Snell et al., 2017) does. The kernel (7) is also the probabilistic basis of prototypical networks (Snell et al., 2017), whose class prototype is a maximum-likelihood estimate under this kernel. Algorithm 1 in Appendix B.1 states the resulting standard $N_C$-way $K$-shot training loop.

### 3.2 Implicit episode construction

**Binary label space.** The explicit construction gives no incentive to generalise to unseen generators, and it needs an oracle label space. We therefore propose to build episodes over the binary label space $\mathcal{Y} = \{0, 1\}$, with $y = 0$ for real and $y = 1$ for AI-generated images. A new generator then needs no label of its own, and the support set needs only the binary labels, which a user can provide. The $N_C$-way construction above, however, degenerates, because only $\binom{2}{2} = 1$ task exists. Diversity must therefore come from variation in the class-conditional embedding distributions rather than from the choice of classes. Under both constructions, the number of tasks grows polynomially with $M_1$ (Appendix A.2). The two constructions differ in which tasks they contain, not in how many. The implicit construction also splits the real class into its domains, which the explicit construction pools.

**Class-level latents.** Under the implicit construction, $N_C = 2$, and $V$ follows the point mass $p(V) = \mathbb{1}[V = (0, 1)]$. We introduce per-class latents $\alpha_S = (\alpha_S^0, \alpha_S^1)$ and $\alpha_Q = (\alpha_Q^0, \alpha_Q^1)$. The latent $\alpha_S^y$ determines the class-conditional embedding distribution of class $y$ in the support set, and $\alpha_Q^y$ determines it in the query set. We write $\alpha^y$ for a statement that holds for both. The task-level latent is $\tau = (V, \alpha_S, \alpha_Q)$, with the prior

$$p(\tau) = p(V) \prod_{k=1}^{N_C} p(\alpha_S^{V_k} \mid V_k)\, p(\alpha_Q^{V_k} \mid V_k, \alpha_S^{V_k}). \tag{8}$$

The prior factorises over the classes. For each class, the first factor draws the support latent. The second factor, which we call the *coupling*, draws the query latent given the support latent. Given $\tau$, the likelihoods keep the form of Section 3.1, but the kernel is indexed by the latent of its side,

$$p(S \mid \tau) = \prod_{k=1}^{N_C} \prod_{(\mathbf{z}, y) \in S_k} \mathbb{1}[y = V_k]\, p(\mathbf{z} \mid y, \alpha_S^y), \tag{9}$$

$$p(Q \mid S, \tau) = \prod_{k=1}^{N_C} \prod_{(\mathbf{z}, y) \in Q_k} \mathbb{1}[y = V_k]\, p(\mathbf{z} \mid y, \alpha_Q^y, S_k). \tag{10}$$

Together with (5), these equations define the implicit episode distribution. The support set and the query set are drawn with different latents, so the support set informs the query only through the coupling. For $y = V_k$, the support images inform the query latent through its posterior $p(\alpha_Q^y \mid y, S_k) = \int p(\alpha_Q^y \mid y, \alpha_S^y)\, p(\alpha_S^y \mid y, S_k)\, d\alpha_S^y$. The adaptation function must learn this inference to predict the query labels. Without the coupling, the support set carries no information about the query. Under the explicit construction, the class slot plays this role, because the support labels reveal $V_k$, and $V_k$ fixes the query kernel. Each prior below states three parts, namely the support-latent prior $p(\alpha_S^y \mid y)$, the coupling $p(\alpha_Q^y \mid y, \alpha_S^y)$, and the kernel $p(\mathbf{z} \mid y, \alpha^y, S_k)$. The support kernel is the query kernel with $S_k = \emptyset$.

**Empirical prior.** Here $\alpha^y$ is a domain index of class $y$. The kernel restricts $\mathcal{D}_y$ to the domain $d = \alpha^y$,

$$p_\psi(\mathbf{z} \mid y, d, S_k) = \frac{1}{N_{y,d} - n_{y,d}(S_k)} \sum_{\bar{\mathbf{x}} \in \mathcal{D}_{y,d}} \mathbb{1}\big[h_\psi(\bar{\mathbf{x}}) \notin \bar{S}_k\big]\, \delta\big(\mathbf{z} - h_\psi(\bar{\mathbf{x}})\big), \tag{11}$$

where $n_{y,d}(S_k) := |\{\bar{\mathbf{x}} \in \mathcal{D}_{y,d} : h_\psi(\bar{\mathbf{x}}) \in \bar{S}_k\}|$ counts the embeddings of class $y$ and domain $d$ that are already in the support set. The subscript $\psi$ marks that this kernel depends on the feature extractor. The indicator under the sum keeps every support image out of the query set. On the support side, where $\bar{S}_k = \emptyset$, it equals one. The domain latents are drawn in two stages. We first draw a non-empty subset $R \subseteq \{1, \ldots, M_0\}$ of real domains from a prior $p(R)$ and, independently, an ordered pair

of disjoint non-empty subsets $G_S, G_Q \subset \{1, \dots, M_1\}$ of generators from a prior $p(G_S, G_Q)$. The domain latents are then uniform on these subsets, with $\alpha_S^0$ and $\alpha_Q^0$ independent given $R$,

$$\alpha_S^0,\, \alpha_Q^0 \sim \mathrm{Uniform}(R), \qquad \alpha_S^1 \sim \mathrm{Uniform}(G_S), \qquad \alpha_Q^1 \sim \mathrm{Uniform}(G_Q). \tag{12}$$

Marginalising the subsets gives the support-latent prior and the coupling in (8). For the real class, the support set and the query set share $R$, so their domains coincide at least as often as under independent draws from all real domains. For the AI-generated class, the query generator always differs from the support generator. This matches the test time, where the query may come from a generator that the support set does not contain. Setting $G_S = G_Q = \{1, \dots, M_1\}$ and removing the split into support and query recovers pooled training (Wang et al., 2020). Pooled training generalises poorly to generators absent at training time (Zhu et al., 2023; Pellegrini et al., 2025).

**Synthetic prior.** Transformers trained only on data from synthetic priors can learn to predict from the labelled examples in their input, which is called *in-context learning*. They then approximate Bayesian posterior prediction without ever observing real-world data (Müller et al., 2022; Garg et al., 2022). This motivates a synthetic prior that defines $\alpha^y$ and its kernel directly in $\mathbb{R}^D$. The feature extractor $h_\psi$ is bypassed during training, so no image enters the optimisation of $p_\theta(y \mid \mathbf{x}, \mathcal{S})$. We set $\alpha^y := (m, \boldsymbol{\eta})$ with $\boldsymbol{\eta} = \{(w_j, \mu_j, \Sigma_j)\}_{j=1}^m$ to parameterise a Gaussian mixture in embedding space. The support and the query set of a class share one mixture, so the coupling is a point mass on $\alpha_S^y$,

$$p(\alpha_S^y \mid y) = p(m \mid y)\, p(w_{1:m} \mid m, y) \prod_{j=1}^{m} p(\mu_j, \Sigma_j \mid m, y), \qquad \alpha_Q^y = \alpha_S^y,$$

$$p(\mathbf{z} \mid y, \alpha^y, S_k) = \sum_{j=1}^{m} w_j\, \mathcal{N}(\mathbf{z}; \mu_j, \Sigma_j), \tag{13}$$

with $m \sim \mathrm{Poisson}^+(\rho_y)$, $(\mu_j, \Sigma_j) \sim \mathrm{NIW}(\mu_{0,y}, \lambda_y, \Lambda_y, \nu_y)$, and $w_{1:m} \sim \mathrm{Dir}(\beta_y \mathbf{1}_m)$. Here $\mathrm{Poisson}^+$ is the Poisson distribution restricted to positive integers, NIW is the normal-inverse-Wishart distribution, Dir is the Dirichlet distribution, and $\mathbf{1}_m$ is the vector of $m$ ones. Appendix C gives the hyperparameters $(\rho_y, \mu_{0,y}, \lambda_y, \Lambda_y, \nu_y, \beta_y)$. The mixture parameters vary across episodes. The adaptation function must therefore infer them from $S$ to predict $Q$ well. Unlike the empirical kernel (11), this kernel needs no indicator that keeps the support embeddings out of the query set. Its density is continuous, so a query embedding almost surely differs from every support embedding, and the query kernel equals the support kernel. Since $\pi$ does not invoke $h_\psi$, this prior only pre-trains the adaptation function $f_\phi$, and only $\phi$ is updated under it. Algorithm 2 in Appendix B.2 states the implicit training procedure for both priors. Prediction at test time is the same under both priors.

**Bayesian view.** Finally, the episodic objective (4) admits a Bayesian reading. Under it, the optimal adaptation function is the posterior predictive distribution $p(y \mid \mathbf{z}, S)$ (Appendix A.1).

## 4 Related Work

**Meta-learning.** Meta-learning methods learn to solve new tasks from few examples by training over a distribution of tasks (Vettoruzzo et al., 2024). Existing methods fall into three paradigms (Vettoruzzo et al., 2024; Wu et al., 2025a). Metric-based methods compute $p_\theta(y \mid \mathbf{x}, \mathcal{S})$ in (3) by comparing the query embedding to class-level representations derived from the support set (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018). Nearest-centroid classifiers (Snell et al., 2017) remain a competitive baseline despite their simplicity. Optimisation-based methods learn an initialisation of $\theta$ that a few gradient steps on the loss of $\mathcal{S}$ fine-tune (Finn et al., 2017; Nichol et al., 2018). Amortisation-based methods perform the adaptation in a single forward pass. This pass reads $\mathcal{S}$ and produces $p_\theta(y \mid \mathbf{x}, \mathcal{S})$ without optimisation at test time (Mishra et al., 2018; Requeima et al., 2019). A fourth paradigm is in-context learning (ICL), popularised by large language models, which can also be viewed as meta-learning (Müller et al., 2022; Wu et al., 2025a; Dong et al., 2024). Transformers trained on diverse task distributions can learn function classes from linear models to decision trees in context (Garg et al., 2022). This work applies methods of this active field (Khoee et al., 2024) and compares state-of-the-art methods of all four paradigms.

**Few-shot detection.** Two works are closest to ours. The first (Aneja & Nießner, 2020) models per-class activation statistics as class-conditional Gaussian distributions for zero- and few-shot transfer to

Table 1: *Our implicit construction against the oracle explicit construction (Experiment 1).* ACC on AI-GenBench, averaged over its thirty-six generators, when the AI-generated support images come from the query's own generator (M), from another generator of its family (S), or from other families (D). *Drop* is M minus D. Chameleon has no generator labels, so only our construction runs on it. The static detectors that read no support set have equal M, S, and D. Top-3 entries per column are coloured. N/A marks TabPFN, which handles at most ten classes. Table 4 gives the ACC per family.

| | AI-GenBench implicit (ours) | | | | AI-GenBench explicit (oracle) | | | | Chameleon implicit (ours) |
|---|---|---|---|---|---|---|---|---|---|
| Model | M | S | D | Drop | M | S | D | Drop | ACC |
| *Context-free* | | | | | | | | | |
| PatchCraft (2023) | 0.58 | 0.58 | 0.58 | 0.00 | 0.58 | 0.58 | 0.58 | 0.00 | 0.58 |
| UnivFD-FT (2023) | 0.87 | 0.84 | 0.83 | 0.04 | 0.88 | 0.83 | 0.81 | 0.07 | 0.75 |
| UnivFD-ZS (2023) | 0.84 | 0.84 | 0.84 | 0.00 | 0.83 | 0.83 | 0.83 | 0.00 | 0.63 |
| BSA (2024) | 0.64 | 0.64 | 0.64 | 0.00 | 0.58 | 0.58 | 0.58 | 0.00 | 0.51 |
| AIDE (2025) | 0.74 | 0.74 | 0.74 | 0.00 | 0.73 | 0.73 | 0.73 | 0.00 | 0.62 |
| MLEP (2025) | 0.55 | 0.55 | 0.55 | 0.00 | 0.57 | 0.57 | 0.57 | 0.00 | 0.54 |
| OSD (2025) | 0.80 | 0.80 | 0.80 | 0.00 | 0.75 | 0.75 | 0.75 | 0.00 | 0.66 |
| *Domain-aware* | | | | | | | | | |
| CORAL (2016) | 0.87 | 0.87 | **0.87** | 0.00 | 0.85 | 0.85 | **0.85** | 0.00 | 0.61 |
| IRM (2019) | 0.87 | 0.87 | **0.87** | 0.00 | 0.72 | 0.72 | 0.72 | 0.00 | 0.61 |
| GroupDRO (2020) | 0.87 | 0.87 | **0.87** | 0.00 | 0.80 | 0.80 | 0.80 | 0.00 | 0.62 |
| TENT (2021) | 0.87 | 0.87 | **0.87** | 0.00 | 0.85 | 0.85 | **0.85** | 0.00 | 0.62 |
| *Context-aware* | | | | | | | | | |
| MatchNet (2016) | 0.85 | 0.85 | 0.85 | 0.00 | 0.85 | 0.84 | 0.83 | 0.02 | 0.63 |
| MAML (2017) | 0.90 | **0.88** | 0.86 | 0.04 | **0.92** | **0.86** | 0.81 | 0.11 | 0.73 |
| ProtoNet (2017) | 0.90 | 0.87 | 0.86 | 0.04 | **0.92** | 0.85 | 0.81 | 0.11 | 0.67 |
| Reptile (2018) | 0.90 | **0.88** | **0.87** | 0.03 | **0.92** | **0.86** | 0.80 | 0.12 | 0.73 |
| CNAPs (2019) | 0.87 | 0.86 | 0.86 | 0.01 | 0.88 | 0.84 | 0.81 | 0.07 | 0.60 |
| FEAT (2020) | **0.91** | **0.88** | 0.86 | 0.05 | **0.92** | **0.86** | 0.79 | 0.13 | 0.80 |
| ICL-Causal (2022) | 0.86 | 0.86 | 0.86 | 0.00 | 0.85 | 0.84 | 0.83 | 0.02 | 0.61 |
| ICL-GP (2022) | 0.86 | 0.86 | 0.86 | 0.00 | 0.83 | 0.82 | 0.81 | 0.02 | 0.65 |
| TabPFN (2022) | 0.86 | 0.72 | 0.67 | 0.19 | N/A | N/A | N/A | N/A | **0.86** |
| GAP (2023) | 0.90 | **0.88** | 0.86 | 0.04 | **0.92** | **0.86** | 0.82 | 0.10 | 0.73 |
| CAML (2024) | 0.85 | 0.85 | 0.85 | 0.00 | 0.80 | 0.80 | 0.80 | 0.00 | 0.63 |
| AFRT (2025) | 0.89 | 0.86 | 0.85 | 0.04 | 0.91 | 0.85 | 0.79 | 0.12 | 0.76 |
| NearestCentroid | 0.90 | 0.77 | 0.73 | 0.17 | 0.90 | 0.77 | 0.69 | 0.21 | 0.85 |

unseen forgery types. The second and closer one (Wu et al., 2025b) uses a metric-based few-shot formulation based on prototypical networks (Snell et al., 2017) with the explicit episode construction. Unlike (Wu et al., 2025b), we formalise AI-generated image detection as a meta-learning problem. We further propose the implicit episode construction, which targets images of generators unknown during training and needs no oracle label space. We do not review static detectors for lack of space, but the experiments include recent ones (Ojha et al., 2023; Zhong et al., 2024; Yan et al., 2025a;b).

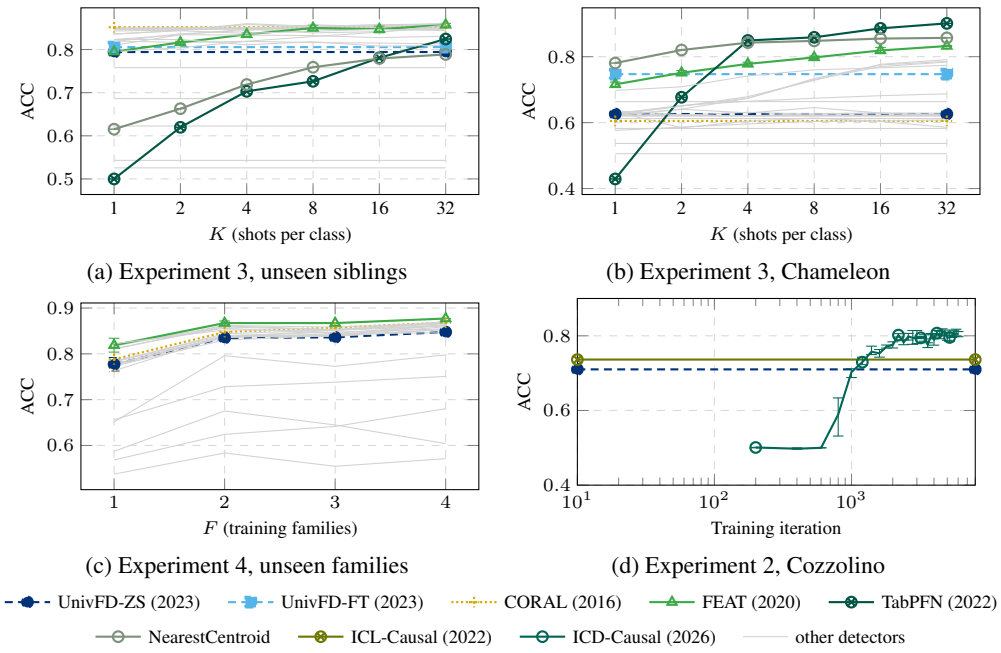
## 5 Experiments

This section tests whether our context-aware detectors stay accurate on unseen generators and images. Nine experiments (Appendices G–O) answer the five questions Q1–Q5 below.

**Datasets.** We train on AI-GenBench (Pellegrini et al., 2025) and group its thirty-six generators into eight *families* of a shared synthesis mechanism (Appendix D.1). The *training pool* holds two generators of each of four families, namely unconditional GANs (GAN-U), pixel-space diffusion (Diff-Px), token-based autoregressive models (Token-AR), and inpainting models (Inpaint). A test generator is *seen* if it is a training generator, an *unseen sibling* if it belongs to a training family, and of an *unseen family* otherwise. We call these three cases the *test groups*. We also test on Chameleon (Yan et al., 2025a), an in-the-wild benchmark of internet images without generator annotation. Experiment 2 also uses Cozzolino (Cozzolino et al., 2024) (Appendix D.3).

**Support set.** Support and query images come from disjoint halves of each test split. A *support policy* decides where the AI-generated support images come from. Under the *matched* policy, they come from the same generator as the query. Under the *sibling* policy, our default, they come from another generator of the same family. Under the *disjoint* policy, they come from another family. Chameleon is read under the matched policy. The support set holds $N_S$=16 images. Our implicit construction reads them as $N_C$=2 ways of $K$=8 images. The explicit construction reads them as one real way and three generator ways of $K$=4 images, and it compares the real logit with the largest generator logit.

**The explicit construction needs an oracle.** The explicit construction labels every support image by its generator and needs a class for every test generator. We therefore give it an oracle label space with one class per generator of AI-GenBench, including the twenty-eight it never trains on. Deployment offers neither. A new generator is unknown before it appears, and a user can usually guess whether an image is AI-generated but rarely knows its generator. Chameleon carries no generator label, so the explicit construction cannot run on it. It is thus a reference, not a deployable detector.




(a) Experiment 3, unseen siblings

(b) Experiment 3, Chameleon

(c) Experiment 4, unseen families

(d) Experiment 2, Cozzolino

Figure 2: *Accuracy as a function of (a, b) the number $K$ of support images per class, (c) the number $F$ of training families at a fixed budget, and (d) the training iterations.* Panels (a) and (c) show AI-GenBench under the sibling policy, (b) Chameleon and (d) Cozzolino under the matched policy. Solid lines are context-aware, dashed context-free, and dotted domain-aware detectors. Static detectors are horizontal lines. Grey lines are the other detectors (Figures 7 and 8). In (d), ICD-Causal is trained on our synthetic prior without any image.

**Models.** *Context-free* detectors fix one decision boundary at training time. They are UnivFD (Ojha et al., 2023), PatchCraft (Zhong et al., 2024), BSA (Zheng et al., 2024), AIDE (Yan et al., 2025a), MLEP (Yuan et al., 2025), and OSD (Yan et al., 2025b). UnivFD-ZS is the original UnivFD, and UnivFD-FT fine-tunes its linear head on the $2K$ labelled support images. *Domain-aware* detectors also fix the boundary, but use the generator as a domain label (CORAL (Sun & Saenko, 2016), IRM (Arjovsky et al., 2019), GroupDRO (Sagawa et al., 2020)) or adapt to unlabelled queries (TENT (Wang et al., 2021)). Both groups are static detectors. *Context-aware* detectors are the twelve meta-learners of Section 4, detailed in Appendix E, and NearestCentroid, which fits no parameters and assigns a query to the nearer class mean of the support embeddings. All detectors except PatchCraft, BSA, AIDE, MLEP, and OSD use frozen CLIP ViT-L/14 embeddings (Appendix F.1). We report accuracy (ACC) at the threshold $0.5$, averaged over three training seeds.

**Q1. Does our implicit construction help?** *Yes. It degrades least when the support set moves away from the query's generator, and it then matches or beats the oracle explicit construction.* In Table 1 (Experiment 1), the explicit construction is slightly ahead under the matched policy (MAML $0.92$ against $0.90$), where its support set holds the query's generator as a labelled class. From the matched to the disjoint policy, the trained meta-learners lose up to $0.13$ under the explicit construction (FEAT), but at most $0.05$ under ours, TabPFN aside. Under the sibling and the disjoint policy, every context-aware detector is therefore at least as accurate under our construction as under the explicit one. TabPFN and NearestCentroid lose most, as they depend on support images of the query's generator. At a fixed architecture, the gain of our construction grows from the seen generators ($+0.012$) to the unseen siblings ($+0.018$) and families ($+0.023$) (Experiment 6). It also lets seven of the eight trained meta-learners that follow the support labels tolerate more inverted labels (Experiment 8). The static detectors read no support set and do not depend on the policy. The domain-aware ones come close, $0.01$ below the best context-aware detectors under the sibling policy and level with them under the disjoint one. The test groups do not order the difficulty of the generators (Table 4). The Diff-Px siblings are the hardest (at most $0.74$), whereas the unseen families GAN-C and GAN-Diff are detected nearly as well as seen ones. On the Diff-Px siblings, all CLIP-based detectors stay within $0.03$ of UnivFD-ZS, while context-aware ones gain up to $0.08$ on other siblings (Experiment 9).

**Q2. When does the support set pay off?** *On in-the-wild images, where the context-aware detectors lead by a wide margin.* On Chameleon (Table 1), TabPFN reaches 0.86, NearestCentroid 0.85, and FEAT 0.80. Among the static detectors, only UnivFD-FT, fine-tuned on the same support images, comes close (0.75), and the others stay at or below 0.66. Figure 2(a, b) varies $K$ (Experiment 3). On AI-GenBench, most context-aware detectors improve up to $K{=}8$ and then match the domain-aware detectors. On Chameleon, the gains are several times larger, for example from 0.43 to 0.90 for TabPFN. With one image per class, NearestCentroid (0.78) already beats every static detector. NearestCentroid profits from the matched policy, and Q1 showed that it loses most once the support images come from other generators. Figure 2(c) varies the number $F$ of training families at a fixed budget (Experiment 4). The lead of the context-aware detectors shrinks from 0.03 at $F{=}1$ to 0.01 at $F{=}4$, so the support set makes up for a narrow training pool. On Chameleon, they lead for every $F$. These findings hold for two other frozen feature extractors (Experiment 5).

**Q3. Do context-aware detectors read the support set?** *Most do, but three Transformer-based detectors ignore it.* Swapping the real and the AI-generated support labels (Experiment 7) drives nine meta-learners and NearestCentroid below chance, to 0.12–0.27, so they follow the labels. ICL-Causal, ICL-GP, and CAML keep their accuracy. They have learned a static decision boundary, although their architecture can read the support set. This is why their accuracy does not change with the support policy and why they fall behind on Chameleon (0.61–0.65). For the others, resampling the support set changes far more decisions on Chameleon than on AI-GenBench (62% against 14% for MAML).

**Q4. Can the episode prior do without images?** *In part. A detector trained on our synthetic prior alone beats a zero-shot static detector in the wild, but our empirical prior stays more accurate.* ICD-Causal is a causal Transformer that differs from ICL-Causal in its tokens and its size (Appendix E). It is trained only on our synthetic prior and never sees an image (Experiment 2). In Figure 2(d), it stays at chance for about 600 iterations, then reaches about 0.80 on Cozzolino, above ICL-Causal (0.74) and UnivFD-ZS (0.71). At $K{=}32$ on Chameleon, ICD-Causal reaches 0.75 and beats UnivFD-ZS (0.63) and ICL-Causal (0.59), but not UnivFD-FT (0.81) or NearestCentroid (0.86).

**Q5. Do the findings hold beyond accuracy?** *Yes. Context-aware detectors trained under our construction are the best calibrated and match the best ranking metrics, but two limits remain.* Under the sibling policy, MAML, Reptile, GAP, and AFRT reach the lowest adaptive calibration error (0.07), against 0.09–0.10 for the domain-aware detectors. ProtoNet, CNAPs, and FEAT share the highest area under the ROC curve (0.94) with the domain-aware detectors, and six meta-learners share the highest average precision (0.93) with them. At a false-positive rate of 1%, ICL-Causal misses the fewest AI-generated images (45%), ahead of the domain-aware detectors (46–48%). FEAT (88%) and NearestCentroid (97%) are the exceptions. First, removing the constraint that the support and the query generators differ changes the mean accuracy by at most 0.001 (Experiment 6). The gain of our construction thus comes from the binary label space and the split of the real class into its domains. Second, the predictions of UnivFD-ZS can replace the support labels on AI-GenBench, where 84% of them are correct, but not on Chameleon, where only 62% are (Experiment 8).

## 6 Conclusion and Limitations

We treated AI-generated image detection as a meta-learning problem and proposed an implicit construction of training episodes over the binary label space. Five findings stand out. (i) Our construction degrades least when the support set comes from other generators, and it then matches or beats the explicit construction with an oracle label space. (ii) In the wild, a parameter-free nearest-centroid rule with one labelled image per class beats every static detector. (iii) Some detectors that can read the support set ignore it and fall behind in the wild. (iv) A detector trained on Gaussian mixtures alone beats a zero-shot static detector in the wild. (v) The gain of our construction comes from the binary label space, not from keeping the support and the query generators apart.

**Limitations and societal impact.** Our setting needs a labelled support set at test time, which pseudo-labels replace only where the labelling detector is accurate. At a false-positive rate of 1%, even the best detectors miss almost half of the AI-generated images. Frozen CLIP embeddings limit the detection of some diffusion generators, and Chameleon is read only under the matched policy. The setting does not attribute generators and costs more than a static detector. Adaptive detection strengthens defences against deepfakes. Statistical detectors, however, are susceptible to adversarial manipulation and carry unknown demographic biases, so their output is a clue, not a definitive answer.

### AI USE STATEMENT

Generative AI tools were used for coding assistance, editing prose, preparing figures and tables, and running the experiments. They were not used to generate research ideas or to draft the mathematical expressions. All AI-assisted output was reviewed by the authors, and we take responsibility for the final content of this work.

### REPRODUCIBILITY STATEMENT

Algorithms 1 and 2 in Appendix B state both episodic training procedures. Appendix E describes every compared model, and Appendix F lists the feature extractor, the preprocessing, the optimisation schedule, and the model-specific hyperparameters. Both benchmarks are available for academic research (Appendix D).

## REFERENCES

Shivangi Aneja and Matthias Nießner. Generalized zero and few-shot transfer for facial forgery detection. *arXiv preprint arXiv:2006.11863*, 2020. URL https://arxiv.org/abs/2006.11863.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint*, 2019. URL https://arxiv.org/abs/1907.02893.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. URL https://arxiv.org/abs/2506.15742.

Stephanie C.Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.05055.

Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2023. URL https://arxiv.org/abs/2211.00680.

Davide Cozzolino, Giovanni Poggi, Riccardo Corvi, Matthias Nießner, and Luisa Verdoliva. Raising the bar of AI-generated image detection with CLIP. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 4356–4366, 2024. URL https://openaccess.thecvf.com/content/CVPR2024W/WMF/html/Cozzolino_Raising_the_Bar_of_AI-generated_Image_Detection_with_CLIP_CVPRW_2024_paper.html.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2024. URL https://arxiv.org/abs/2301.00234.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. URL https://arxiv.org/abs/2012.09841.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, pp. 12606–12633, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Christopher Fifty, Dennis Duan, Ronald G. Junkins, Ehsan Amid, Jure Leskovec, Christopher Ré, and Sebastian Thrun. Context-aware meta-learning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.10971.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017. URL https://arxiv.org/abs/1703.03400.

Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/2003.08685.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2208.01066.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. URL https://arxiv.org/abs/1406.2661.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2412.04431.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2006.11239.

Suhyun Kang, Duhun Hwang, Moonjung Eo, Taesup Kim, and Wonjong Rhee. Meta-learning with a geometry-adaptive preconditioner. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2304.01552.

Arsham Gholamzadeh Khoee, Yinan Yu, and Robert Feldt. Domain generalization through meta-learning: a survey. *Artificial Intelligence Review*, 57(10):285, 2024. URL https://doi.org/10.1007/s10462-024-10922-z.

Louis Kirsch, James Harrison, Jascha Sohl-Dickstein, and Luke Metz. General-purpose in-context learning by meta-learning transformers. In *NeurIPS 2022 Workshop on Meta-Learning*, 2022. URL https://arxiv.org/abs/2212.04458.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1707.03141.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2112.10510.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. URL https://arxiv.org/abs/1803.02999.

Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2302.10174.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint*, 2023. URL https://arxiv.org/abs/2304.07193.

Lorenzo Pellegrini, Davide Cozzolino, Serafino Pandolfini, Davide Maltoni, Matteo Ferrara, Luisa Verdoliva, Marco Prati, and Marco Ramilli. AI-GenBench: A new ongoing benchmark for AI-generated image detection. In *IEEE International Joint Conference on Neural Networks (IJCNN), Verimedia Workshop*, 2025. URL https://arxiv.org/abs/2504.20865.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. URL https://arxiv.org/abs/2103.00020.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022. URL https://arxiv.org/abs/2204.06125.

James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E. Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1906.07697.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. URL https://arxiv.org/abs/2112.10752.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1911.08731.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.11487.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1703.05175.

Wei Song and Yaobin Huang. Adaptive feature recalibration transformer for enhancing few-shot image classification. *The Visual Computer*, 41(10):7269–7283, 2025. URL https://doi.org/10.1007/s00371-025-03804-0.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1907.05600.

Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *Computer Vision – ECCV 2016 Workshops*, volume 9915 of *Lecture Notes in Computer Science*, pp. 443–450, 2016. URL https://link.springer.com/chapter/10.1007/978-3-319-49409-8_35.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. URL https://arxiv.org/abs/1711.06025.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2404.02905.

Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 2016. URL https://arxiv.org/abs/1601.06759.

Anna Vettoruzzo, Mohamed-Rafik Bouguelia, Joaquin Vanschoren, Thorsteinn Rögnvaldsson, and KC Santosh. Advances and challenges in meta-learning: A technical review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7):4763–4779, 2024. URL https://arxiv.org/abs/2307.04722.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016. URL https://arxiv.org/abs/1606.04080.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2212.07677.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2006.10726.

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. CNN-generated images are surprisingly easy to spot... for now. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. URL https://arxiv.org/abs/1912.11035.

Shiguang Wu, Yaqing Wang, Yatao Bian, and Quanming Yao. Learning to learn with contrastive meta-objective. In *Advances in Neural Information Processing Systems*, 2025a. URL https://arxiv.org/abs/2410.05975.

Shiyu Wu, Jing Liu, Jing Li, and Yequan Wang. Few-shot learner generalizes across AI-generated image detection. In *International Conference on Machine Learning*, 2025b. URL https://arxiv.org/abs/2501.08763.

Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for AI-generated image detection. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://arxiv.org/abs/2406.19435.

Zhiyuan Yan, Jiangming Wang, Peng Jin, Ke-Yue Zhang, Chengchun Liu, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Orthogonal subspace decomposition for generalizable AI-generated image detection. In *International Conference on Machine Learning*. PMLR, 2025b. URL https://proceedings.mlr.press/v267/yan25b.html.

Shibo Yao, Renshuai Tao, Xiaolong Zheng, Chao Liang, and Chunjie Zhang. Leveraging failed samples: A few-shot and training-free framework for generalized deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27764–27772, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/39998.

Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. URL https://arxiv.org/abs/1812.03664.

Lin Yuan, Xiaowan Li, Yan Zhang, Jiawei Zhang, Hongbo Li, and Xinbo Gao. MLEP: Multi-granularity local entropy patterns for universal AI-generated image detection. *arXiv preprint*, 2025. URL https://arxiv.org/abs/2504.13726.

Chende Zheng, Chenhao Lin, Zhengyu Zhao, Hang Wang, Xu Guo, Shuai Liu, and Chao Shen. Breaking semantic artifacts for generalized AI-generated image detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6dddcff5b115b40c998a08fbd1cea4d7-Abstract-Conference.html.

Nan Zhong, Yiran Xu, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. PatchCraft: Exploring texture patch for efficient AI-generated image detection. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2311.12397.

Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. GenImage: A million-scale benchmark for detecting AI-generated images. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.08571.

## APPENDIX CONTENTS


**A Notes on the Binary Detection Case** **16**

A.1 Bayesian interpretation . . . . . 16

A.2 Episode complexity of the two constructions . . . . . 16

**B Episodic Meta-Learning Algorithms** **17**

B.1 Explicit construction . . . . . 18

B.2 Implicit construction . . . . . 18

B.3 Prediction at test time . . . . . 19

**C Synthetic Prior Hyperparameters** **19**

**D Datasets** **19**

D.1 AI-GenBench . . . . . 19

D.2 Chameleon . . . . . 19

D.3 Cozzolino . . . . . 19

D.4 Licences . . . . . 19

**E Additional Details on Compared Models** **20**

E.1 Context-free detectors . . . . . 20

E.2 Domain-aware detectors . . . . . 21

E.3 Metric-based meta-learners . . . . . 21

E.4 Optimisation-based meta-learners . . . . . 22

E.5 Amortisation-based meta-learners . . . . . 22

E.6 In-context learners . . . . . 22

**F Hyperparameter Settings** **23**

F.1 Feature extractor . . . . . 23

F.2 Image preprocessing . . . . . 23

F.3 Optimisation . . . . . 23

F.4 Random seeds . . . . . 23

F.5 Compute . . . . . 23

**G Experiment 1: Implicit and Explicit Episode Construction** **24**

G.1 Experiment description . . . . . 24

G.2 Support policies . . . . . 24

G.3 Chameleon . . . . . 25

G.4 Other metrics . . . . . 25

**H Experiment 2: Synthetic Prior** **49**

H.1 Experiment description . . . . . 49
H.2 Results . . . . . 49

**I Experiment 3: Support-Set Size** **52**
I.1 Experiment description . . . . . 52
I.2 Results . . . . . 52

**J Experiment 4: Diversity of Training Families** **53**
J.1 Experiment description . . . . . 53
J.2 Results . . . . . 53

**K Experiment 5: Feature Space** **53**
K.1 Experiment description . . . . . 53
K.2 Results . . . . . 53

**L Experiment 6: Decomposition of the Gain** **54**
L.1 Experiment description . . . . . 54
L.2 Results . . . . . 54

**M Experiment 7: Support-Set Counterfactuals** **55**
M.1 Experiment description . . . . . 55
M.2 Results . . . . . 55

**N Experiment 8: Support-Set Corruption** **56**
N.1 Experiment description . . . . . 56
N.2 Results . . . . . 57

**O Experiment 9: Semantic and Sub-Semantic Generators** **58**
O.1 Experiment description . . . . . 58
O.2 Results . . . . . 58

# A Notes on the Binary Detection Case

## A.1 Bayesian interpretation

The episodic objective (4) admits a Bayesian reading. The ideal predictor for a query embedding $\mathbf{z}$ given support embeddings $S$ is the posterior predictive distribution (Müller et al., 2022)

$$p(y \mid \mathbf{z}, S) = \int p(y \mid \mathbf{z}, \tau) p(\tau \mid \mathbf{z}, S)\, d\tau, \qquad p(\tau \mid \mathbf{z}, S) \propto p(\mathbf{z} \mid \tau, S) p(S \mid \tau) p(\tau), \tag{14}$$

which integrates over all latent values $\tau$ weighted by their posterior given $S$ and $\mathbf{z}$. Minimising (4) is equivalent to minimising $\mathbb{E}_\tau \mathbb{E}_{\mathbf{z},S}\big[\mathrm{KL}\big(p(\cdot \mid \mathbf{z}, S) \,\|\, q_\phi(\cdot \mid \mathbf{z}, S)\big)\big]$ (Müller et al., 2022). If the adaptation function is expressive enough, the global optimum recovers exact Bayesian prediction for every support set and query embedding that $\pi$ produces with positive probability. This view connects episodic meta-learning to in-context learning. Transformers trained on task distributions perform implicit Bayesian inference within their forward pass (Müller et al., 2022; von Oswald et al., 2023).

## A.2 Episode complexity of the two constructions

This section compares the explicit and the implicit construction by the number of tasks each of them can sample. Recall from Section 3 that a task is the conditional distribution $p(S, Q \mid \tau)$ of an episode given the task-level latent. The number of tasks therefore measures the diversity of the episodes. The number of latent values measures only the size of the space on which the prior is written. Both numbers are counted below for $M_1$ training generators, $M_0$ real-image domains, and a fixed number $K$ of support images per class.

**Step 1. Explicit latent values.** Under the explicit construction of Section 3.1, the latent is the ordered tuple $V = (V_1, \ldots, V_{N_C})$ of distinct classes from $\{0, 1, \ldots, M_1\}$, drawn from the uniform prior (6). Its support holds

$$\big|\operatorname{supp} p(V)\big| = \frac{(M_1 + 1)!}{(M_1 + 1 - N_C)!}$$

latent values.

**Step 2. Explicit tasks.** Given $V$, the pairs of slot $k$ are drawn from the class kernels $p_\psi(\mathbf{z} \mid V_k)$ and $p_\psi(\mathbf{z} \mid V_k, S_k)$ of (7). Every pair carries its label $y = V_k$. Two tuples with the same classes in a different order produce the same labelled pairs and differ only in the order of the slots. The explicit tasks therefore correspond to the sets of $N_C$ distinct classes, and their number is

$$T_{\text{exp}} = \binom{M_1 + 1}{N_C}.$$

For a fixed $N_C$, this is a polynomial of degree $N_C$ in $M_1$. In the binary regime $N_C$=2, this gives $T_{\text{exp}} = (M_1 + 1)M_1/2$. Only $M_1$ of these tasks contain the real class and therefore pose a real-versus-AI-generated problem. The other $M_1(M_1 - 1)/2$ tasks separate two generators.

**Step 3. The structure of an explicit task.** Both the support and the query of slot $k$ are drawn from the same class $V_k$. For an AI-generated class this class is one generator. Every query image of an explicit episode is therefore produced by a generator that is also present in the support.

**Step 4. Values of the subsets and domains.** Under the implicit construction of Section 3.2 with the empirical prior, an episode draws four kinds of values. These are the fixed tuple $V = (0, 1)$, the disjoint non-empty generator subsets $G_S$ and $G_Q$, the non-empty real-domain subset $R$, and the four domain latents of (12). These are the support generator $\alpha_S^1 = d_S^1 \in G_S$, the query generator $\alpha_Q^1 = d_Q^1 \in G_Q$, and the support and query real domains $\alpha_S^0 = d_S^0 \in R$ and $\alpha_Q^0 = d_Q^0 \in R$, drawn independently. We assume that the priors of the subsets put mass on every admissible choice. The values are then counted in two factors. For the generators, there are $M_1(M_1 - 1)$ ordered pairs $d_S^1 \neq d_Q^1$. Every other generator then lies in $G_S$, in $G_Q$, or in neither, which gives $3^{M_1 - 2}$ choices. For the real domains, the number of pairs $(R, (d_S^0, d_Q^0))$ with $d_S^0, d_Q^0 \in R$ is $\sum_R |R|^2 = M_0(M_0 + 1)\, 2^{M_0 - 2}$. Together,

$$\big|\operatorname{supp} p(R, G_S, G_Q, \alpha_S, \alpha_Q)\big| = M_1(M_1 - 1)\, 3^{M_1 - 2} \cdot M_0(M_0 + 1)\, 2^{M_0 - 2},$$

which grows exponentially in $M_1$ and $M_0$.

**Step 5. Implicit tasks.** By (9), (10), and (11), the conditional distribution of an episode depends only on the four domain latents $(d_S^0, d_S^1, d_Q^0, d_Q^1)$. The subsets $G_S$, $G_Q$, and $R$ only restrict the set from which the indices are drawn. They do not enter the kernels. Every tuple with $d_S^1 \neq d_Q^1$ is reachable, for example through $G_S = \{d_S^1\}$, $G_Q = \{d_Q^1\}$, and $R = \{d_S^0, d_Q^0\}$. Different tuples define different kernels. The number of implicit tasks is therefore

$$T_{\text{imp}} = M_0^2 \, M_1(M_1 - 1),$$

a polynomial of degree two in $M_1$ and of degree two in $M_0$. The factor $M_0^2$ arises because the implicit construction splits the real class into its domains, which the explicit construction pools. The exponential number of values of Step 4 collapses onto this polynomial number of tasks. A task with $d_S^0 \neq d_Q^0$ is reached by $3^{M_1-2}\, 2^{M_0-2}$ values, and a task with $d_S^0 = d_Q^0$ by $3^{M_1-2}\, 2^{M_0-1}$. The subsets therefore change how often each task is sampled, but they do not create new tasks.

**Step 6. The structure of an implicit task.** Under the disjoint subsets of Section 3.2, the support generator $d_S^1$ and the query generator $d_Q^1$ always differ. Every query image of an implicit episode is therefore produced by a generator that is absent from the support. By Step 3, no explicit task has this property, and by this step, every implicit task has it. The two sets of tasks are thus disjoint.

**Step 7. Cost of sampling one episode.** An explicit episode draws $N_C$ classes and then $N_C K$ support and $N_Q$ query embeddings. This takes a number of operations proportional to $N_C K + N_Q$. An implicit episode draws three subsets and four indices in a number of operations proportional to $M_1 + M_0$. It then draws $2K$ support and $N_Q$ query embeddings. Both costs are linear, so neither construction is more expensive to sample.

**Conclusions.** (i) In the number of tasks, both constructions are polynomial. For $N_C$=2, the explicit construction has $M_1$ real-versus-AI-generated tasks and $M_1(M_1 - 1)/2$ tasks that separate two generators. The implicit construction has $M_0^2 M_1(M_1 - 1)$ real-versus-AI-generated tasks. (ii) The exponential count of Step 4 describes values of the subsets and domains and not tasks. It must not be read as an exponential number of tasks. Figure 3 draws both task counts up to the thirty-six generators of AI-GenBench. (iii) The decisive difference is the kind of task and not the number of tasks. Every implicit task poses a real-versus-AI-generated problem, whereas most explicit tasks separate two generators. Under the disjoint subsets, every implicit query also comes from a generator absent from the support set.

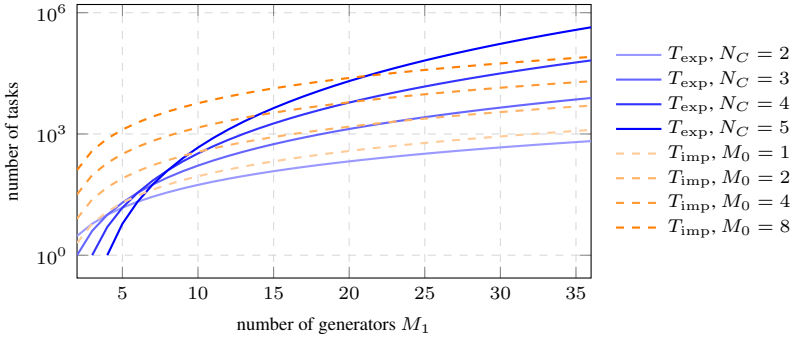



Figure 3: *Number of tasks of the two episode constructions.* The explicit count $T_{\text{exp}} = \binom{M_1+1}{N_C}$ is drawn in blue for four numbers of ways, and the implicit count $T_{\text{imp}} = M_0^2 M_1(M_1 - 1)$ in orange for four numbers of real-image domains. Darker shades mark larger $N_C$ and larger $M_0$. Both counts are polynomial in the number of generators $M_1$, which reaches thirty-six on AI-GenBench.

## B Episodic Meta-Learning Algorithms

This appendix gives pseudocode for the two episode constructions of Sections 3.1 and 3.2. Algorithm 1 states the standard $N_C$-way $K$-shot episodic loop of the explicit construction. With the nearest-prototype rule as $f_\phi$, it is the ProtoNet training procedure (Snell et al., 2017). Algorithm 2 states

the implicit construction. The class-slot tuple $V$ is fixed to the binary label space $\mathcal{Y} = \{0, 1\}$, and the per-class auxiliary variables $\alpha_S, \alpha_Q$ of Section 3.2 supply the diversity of the episodes. One pseudocode covers both priors. Under the empirical prior, $\alpha^y$ is a domain index drawn from $G_S$, $G_Q$, or $R$. Under the synthetic prior, $\alpha^y$ is the parameter set of a Gaussian mixture in $\mathbb{R}^D$.

### B.1 Explicit construction

**Algorithm 1** Explicit episodic meta-learning ($N_C$-way $K$-shot)

**Require:** Dataset $\mathcal{D}$, slot prior $p(V)$, ways $N_C$, shots $K$, queries $N_Q$, outer-loop learning rate $\eta_{\text{out}}$, episodes $T_{\text{ep}}$
**Ensure:** Trained parameters $\theta = (\psi, \phi)$

**for** $t = 1, \ldots, T_{\text{ep}}$ **do**
  Sample slot tuple $V \sim p(V)$ (6)
  **for** $k = 1, \ldots, N_C$ **do**
    Sample $K$ images $\{\mathbf{x}^S_{k,i}\}_{i=1}^{K}$ from $\mathcal{D}_{V_k}$ for the support
    Sample $N_Q/N_C$ images $\{\mathbf{x}^Q_{k,j}\}_{j=1}^{N_Q/N_C}$ from $\mathcal{D}_{V_k} \setminus \{\mathbf{x}^S_{k,i}\}_i$ for the query (7)
    Compute embeddings $\mathbf{z}^S_{k,i} = h_\psi(\mathbf{x}^S_{k,i})$ and $\mathbf{z}^Q_{k,j} = h_\psi(\mathbf{x}^Q_{k,j})$
  **end for**
  Form support $S = \{(\mathbf{z}^S_{k,i}, V_k)\}$ and query $Q = \{(\mathbf{z}^Q_{k,j}, V_k)\}$
  Compute loss $\ell = -\frac{1}{N_Q} \sum_{(\mathbf{z},y)\in Q} \log q_\phi(y \mid \mathbf{z}, S)$ (4)
  Update $\theta \leftarrow \theta - \eta_{\text{out}} \nabla_\theta \ell$
**end for**

### B.2 Implicit construction

**Algorithm 2** Implicit episodic meta-learning with $N_C$=2 ways and $K$ shots under the empirical or the synthetic prior

**Require:** Prior $p(\alpha_S, \alpha_Q \mid V)$ with $V = (V_1, V_2) = (0, 1)$ held fixed, shots $K$, queries $N_Q$, outer-loop learning rate $\eta_{\text{out}}$, episodes $T_{\text{ep}}$, and under the empirical prior a labelled dataset $\mathcal{D}$
**Ensure:** Trained parameters $\theta = (\psi, \phi)$ under the empirical prior; only $\phi$ under the synthetic prior

**for** $t = 1, \ldots, T_{\text{ep}}$ **do**
  Fix the binary class slots $V = (V_1, V_2) = (0, 1)$
  Sample per-class auxiliaries $\alpha_S = (\alpha^0_S, \alpha^1_S)$ and $\alpha_Q = (\alpha^0_Q, \alpha^1_Q)$ from $p(\alpha_S, \alpha_Q \mid V)$:
    - *Empirical prior:* draw disjoint non-empty generator subsets $(G_S, G_Q) \sim p(G_S, G_Q)$ in $\{1, \ldots, M_1\}$ and a non-empty real-domain subset $R \subseteq \{1, \ldots, M_0\}$, then set $\alpha^1_S = d^1_S \sim \text{Uniform}(G_S)$, $\alpha^1_Q = d^1_Q \sim \text{Uniform}(G_Q)$, and $\alpha^0_S = d^0_S \sim \text{Uniform}(R)$, $\alpha^0_Q = d^0_Q \sim \text{Uniform}(R)$ independently
    - *Synthetic prior:* draw $\alpha^y_S$ as Gaussian-mixture parameters $(m, \boldsymbol{\eta})$ and set $\alpha^y_Q = \alpha^y_S$ for each $y \in \{0, 1\}$ (13)
  **for** $k = 1, 2$ **do**
    Sample $K$ support embeddings $\{\mathbf{z}^S_{k,i}\}_{i=1}^{K}$ i.i.d. from $p(\mathbf{z} \mid V_k, \alpha^{V_k}_S)$ (9)
    Sample $N_Q/2$ query embeddings $\{\mathbf{z}^Q_{k,j}\}_{j=1}^{N_Q/2}$ i.i.d. from $p(\mathbf{z} \mid V_k, \alpha^{V_k}_Q, S_k)$ (10)
  **end for**
  Form support $S = \{(\mathbf{z}^S_{k,i}, V_k)\}$ and query $Q = \{(\mathbf{z}^Q_{k,j}, V_k)\}$
  Compute loss $\ell = -\frac{1}{N_Q} \sum_{(\mathbf{z},y)\in Q} \log q_\phi(y \mid \mathbf{z}, S)$ (4)
  Update $\theta \leftarrow \theta - \eta_{\text{out}} \nabla_\theta \ell$ (empirical) or $\phi \leftarrow \phi - \eta_{\text{out}} \nabla_\phi \ell$ (synthetic)
**end for**

Under the empirical prior, sampling $\mathbf{z} \sim p_\psi(\mathbf{z} \mid V_k, \alpha^{V_k}, \cdot)$ draws an image $\mathbf{x}$ uniformly with replacement from $\mathcal{D}_{V_k, \alpha^{V_k}}$ and sets $\mathbf{z} = h_\psi(\mathbf{x})$. Here $\alpha^{V_k}$ is the domain index of the class and the side of the episode, that is $d^0_S$, $d^0_Q$, $d^1_S$, or $d^1_Q$. Under the synthetic prior the embedding is drawn

directly from the Gaussian mixture (13) in $\mathbb{R}^D$. The feature extractor $h_\psi$ is then bypassed and only $\phi$ is updated.

### B.3 Prediction at test time

At test time, a support set $\mathcal{S}^*$ of images from a new deployment context and a query set $\mathcal{Q}^*$ are given. The support embeddings are $S^* = \{(h_\psi(\mathbf{x}_i), y_i) : (\mathbf{x}_i, y_i) \in \mathcal{S}^*\}$. For each query image $\mathbf{x} \in \mathcal{Q}^*$ with $\mathbf{z} = h_\psi(\mathbf{x})$, the trained model evaluates $q_\phi(y \mid \mathbf{z}, S^*)$ in a single forward pass without parameter updates.

## C Synthetic Prior Hyperparameters

The synthetic prior (13) of Section 3.2 parameterises each latent $\alpha^y = (m, \boldsymbol{\eta}) = (m, \{w_j, \mu_j, \Sigma_j\}_{j=1}^m)$ as a finite Gaussian mixture in $\mathbb{R}^D$. Its hyperparameters are set independently for each class $y \in \{0, 1\}$.

- The Poisson rate $\rho_y > 0$ controls the expected number of mixture components $m$.
- The normal-inverse-Wishart distribution has a prior mean $\mu_{0,y} \in \mathbb{R}^D$, a scaling factor $\lambda_y > 0$ of the mean precision, a positive-definite scale matrix $\Lambda_y \in \mathbb{R}^{D \times D}$, and $\nu_y > D - 1$ degrees of freedom.
- The Dirichlet concentration $\beta_y > 0$ sets the symmetric prior $\mathrm{Dir}(\beta_y \mathbf{1}_m)$ over the $m$ mixture weights. The value $\beta_y = 1$ gives uniform weights, $\beta_y < 1$ favours sparse mixtures, and $\beta_y > 1$ favours balanced mixtures.

## D Datasets

### D.1 AI-GenBench

AI-GenBench (Pellegrini et al., 2025) contains thirty-six generators, each with 4 000 AI-generated training images and 1 000 AI-generated test images, together with 144 000 real training images and 36 000 real test images pooled from COCO, ImageNet, RAISE, and LAION. We follow its official train and test split. Table 2 groups the generators into the eight families used in Section 5 and marks the two generators of each training family that enter the training pool. The training pool holds 16 000 AI-generated images spread evenly over its eight training generators and 16 000 real images. Token-AR has only two generators, so it has no unseen sibling.

### D.2 Chameleon

Chameleon (Yan et al., 2025a) is a test-only benchmark of real and AI-generated images collected from the internet. It has no per-image generator annotation, so all of its AI-generated images form a single test generator, which is an unseen family. The support and the query images are drawn from disjoint halves of its test split, and every query is scored with eight support sets whose predictions are averaged.

### D.3 Cozzolino

Cozzolino (Cozzolino et al., 2024) contains twenty generators in its test split, with 4 000 real and 20 000 AI-generated images. We use only this test split, in the emergence curves of Experiment 2. The support and the query images are drawn from disjoint halves of each generator.

### D.4 Licences

According to their repositories, AI-GenBench is licensed under BSD-3-Clause, Cozzolino under the Apache License 2.0, and Chameleon for academic research only. AI-GenBench does not redistribute its real images, which must be downloaded from the original datasets.

Table 2: *Generator families of AI-GenBench (Pellegrini et al., 2025).* Generators marked with $*$ are the training generators. The other generators of a training family are its unseen siblings.

| Family | Mechanism | Generators |
|---|---|---|
| GAN-U | unconditional GANs | ProGAN$^*$, StyleGAN3$^*$, CIPS, GANformer, ProjectedGAN, StyleGAN1, StyleGAN2 |
| GAN-C | conditional GANs | GauGAN, StarGAN, BigGAN, CycleGAN |
| GAN-Diff | adversarial diffusion | DD-GAN, DiffGAN-StyGAN2, DiffGAN-ProjGAN |
| Diff-Px | pixel-space diffusion | ADM$^*$, Glide$^*$, DDPM, DeepFloyd IF |
| Diff-Lat | latent diffusion | DALL·E 3, Latent Diffusion, Midjourney, SD 1.4, SD 1.5, SD 2.1, SDXL |
| Flow | rectified flow | FLUX Dev, FLUX Schnell |
| Token-AR | token-based autoregressive | VQ-Diffusion$^*$, VQGAN$^*$ |
| Inpaint | inpainting and layout-conditioned | FaceSynthetics$^*$, LaMa$^*$, CRN, IMLE, MAT, Palette, SN-PatchGAN |

# E Additional Details on Compared Models

This appendix briefly describes each model of Section 5. There are seven context-free detectors (UnivFD-ZS, UnivFD-FT, PatchCraft, BSA, AIDE, MLEP, OSD) and four domain-aware detectors (CORAL, IRM, GroupDRO, TENT). There are twelve meta-learners of the four paradigms of Section 4, the parameter-free NearestCentroid, and ICD-Causal, which is pre-trained on the synthetic prior in Experiment 2. Context-free and domain-aware detectors read no support set at test time and are static detectors (1). Table 3 lists the model-specific hyperparameters. All other hyperparameters take the common values of Appendix F.3.

## E.1 Context-free detectors

**UnivFD (Ojha et al., 2023).** UnivFD argues that a detector should not learn the artefacts of particular generators but should rely on a feature space that has never seen AI-generated images (Ojha et al., 2023). It therefore freezes a CLIP ViT-L/14 image encoder and trains only a linear classifier on its [CLS] embedding. UnivFD-ZS is this linear head fitted once on the training pool, and UnivFD-FT fine-tunes it on the $2K$ labelled images that a meta-learner receives as its support set, for at most 100 epochs with early stopping on the remaining support images.

**BSA (Zheng et al., 2024).** BSA attributes the failure of detectors across image scenes to semantic artefacts that both real and AI-generated images carry (Zheng et al., 2024). It tiles every image into non-overlapping $32\times32$ patches, shuffles them to remove the global semantic structure, and classifies the result with a patch-based convolutional network trained end-to-end. We use 18 convolutional blocks and the learning rate $10^{-4}$.

**MLEP (Yuan et al., 2025).** MLEP describes an image by the entropy of its local pixel values, which carries traces of the generation process and little semantic content (Yuan et al., 2025). It computes entropy maps of $2\times2$ neighbourhoods of a patch-shuffled image at three scales and classifies their concatenation with a ResNet-50 whose first convolution takes the entropy channels as input. We use patches of side 4, the learning rate $10^{-4}$, and 500 warm-up steps.

**PatchCraft (Zhong et al., 2024).** PatchCraft assumes that AI-generated images reveal themselves by unnatural texture statistics rather than by their semantic content (Zhong et al., 2024). It crops $3N^2$ random patches per image and ranks them by texture diversity. It reassembles one image from the $N^2$ poorest and one from the $N^2$ richest patches. Both are filtered with learnt high-pass filters, and a convolutional network scores the contrast between the two. We use the original five-group network with $N{=}3$, image side 256, and the learning rate $10^{-4}$.

**AIDE (Yan et al., 2025a).** AIDE combines the high-level semantics of CLIP with low-level frequency cues (Yan et al., 2025a). It upsamples the two highest- and the two lowest-frequency $32\times32$ patches of the discrete cosine transform, encodes each frequency band with its own ImageNet-pre-trained ResNet-50, and classifies the averaged ResNet features together with frozen CLIP ViT-B/32 features by a two-layer perceptron. As in the original recipe, the CLIP encoder and both ResNet-50s stay fixed. We use patch side 256 and the learning rate $10^{-4}$.

**OSD (Yan et al., 2025b).** OSD (orthogonal subspace decomposition) separates the weights of a pre-trained encoder into a semantic part that is kept and a forgery part that is learnt (Yan et al., 2025b). It decomposes the query and value projections of every attention layer of CLIP by a singular value decomposition, freezes the leading singular components, trains only the last $r$ of them, and keeps

the trained components orthonormal by an auxiliary penalty. We use CLIP ViT-B/32 with rank $r{=}8$, penalty weight 1, the learning rate $10^{-3}$, and 500 warm-up steps.

### E.2 Domain-aware detectors

All four domain-aware detectors share one architecture on the CLIP embeddings, a hidden layer of width 256 followed by a linear classifier, and treat every training generator as a domain.

**CORAL (Sun & Saenko, 2016).** Deep CORAL learns features whose statistics agree across domains, so that a classifier fitted on one domain transfers to another (Sun & Saenko, 2016). It adds to the cross-entropy a penalty on the distance between the feature covariance matrices of two domains. We penalise the differences of the feature means and covariances between every pair of domains with weight 1.

**IRM (Arjovsky et al., 2019).** Invariant risk minimisation seeks features on which one classifier is optimal for every domain, which excludes features whose relation to the label changes across domains (Arjovsky et al., 2019). Its IRMv1 form adds to the loss the squared gradient of the risk of every domain with respect to a fixed scaling of the classifier output. We weight this penalty by 1 from the first training step.

**GroupDRO (Sagawa et al., 2020).** Group distributionally robust optimisation minimises the worst expected loss over groups of the training data instead of the average loss (Sagawa et al., 2020). It keeps a weight for every group and raises the weights of the groups with the highest current loss by an exponentiated-gradient update. The groups are the domains, and the step size of the update is 0.01.

**TENT (Wang et al., 2021).** TENT adapts a trained classifier to a shifted test distribution without labels by minimising the entropy of its predictions on the test inputs (Wang et al., 2021). It updates only the affine parameters of the normalisation layers and keeps all other weights fixed. We apply one entropy-minimisation step with the learning rate $10^{-3}$ per batch of queries and never use the support labels.

### E.3 Metric-based meta-learners

**MatchNet (Vinyals et al., 2016).** Matching Networks predict the query label by a soft nearest-neighbour vote over the support set (Vinyals et al., 2016). The predictive distribution is $\sum_{i=1}^{N_S} \mathrm{att}(\mathbf{z}, \mathbf{z}_i)\, \mathbb{1}[y_i{=}y]$, where the attention weights are a softmax over the similarities between the query and the support embeddings. A *full-context embedding* (FCE) re-encodes the support set with a bidirectional LSTM and the query with an attention LSTM that reads the support set. Both embeddings then depend on the whole episode. We use the squared Euclidean distance instead of the cosine similarity, as in ProtoNet, and apply the FCE refinement before the vote.

**ProtoNet (Snell et al., 2017).** Prototypical Networks summarise each class by its mean support embedding $\mathbf{c}_y = \frac{1}{K}\sum_{i:y_i=y} \mathbf{z}_i$, called its prototype (Snell et al., 2017). A query is assigned to the nearest prototype in squared Euclidean distance. The resulting softmax classifier is the Bayes posterior of a model with spherical Gaussian classes of shared variance, whose means are estimated by the prototypes. In the binary case, the two prototypes are the mean real and the mean AI-generated support embedding.

**FEAT (Ye et al., 2020).** FEAT adds to ProtoNet a set-to-set Transformer that adapts the prototypes to the episode (Ye et al., 2020). The Transformer takes the $N_C$ prototypes as tokens and returns adapted prototypes, which the nearest-prototype rule then uses. Prototypes that are easily confused are pushed apart by attending to each other. A contrastive auxiliary loss on the adapted embeddings is added to the episodic loss, which the original work finds important for stable training.

**AFRT (Song & Huang, 2025).** AFRT (Adaptive Feature Recalibration Transformer) addresses the bias of query embeddings relative to the prototypes when the feature extractor was pre-trained on another domain (Song & Huang, 2025). A Transformer block learns a linear correction of the query embedding that depends on the support set. The corrected query then enters a nearest-prototype classifier. The original work reports gains on cross-domain few-shot benchmarks, which resemble our detection of new generators.

**NearestCentroid.** NearestCentroid computes the mean support embedding of each class and assigns a query to the class whose mean is nearer in squared Euclidean distance. It fits no parameter, so it shows how much of the detection is already contained in the CLIP embeddings and the support set.

### E.4 Optimisation-based meta-learners

**MAML (Finn et al., 2017).** Model-Agnostic Meta-Learning trains an initialisation from which a few gradient steps adapt the model to a task (Finn et al., 2017). An inner loop takes $T$ SGD steps on the support loss, starting from the meta-parameters. The outer loop differentiates the query loss after adaptation through the inner steps. We use the second-order variant, which propagates second derivatives through the inner steps, and adapt all parameters of $f_\phi$.

**Reptile (Nichol et al., 2018).** Reptile is a first-order counterpart of MAML (Nichol et al., 2018). It adapts the parameters to one task with $T$ SGD steps and then moves the meta-parameters towards the adapted ones by $\theta \leftarrow \theta + \gamma(\theta^{\text{adapt}} - \theta)$. This update behaves like MAML without second derivatives. It is therefore the cheaper optimisation-based meta-learner of our comparison.

**GAP (Kang et al., 2023).** GAP (Geometry-Adaptive Preconditioner) extends MAML by a meta-learnt preconditioner of the inner-loop gradient (Kang et al., 2023). The preconditioner of each layer combines the singular vectors of the current gradient with a meta-learnt positive diagonal. It therefore adapts to the task and to the inner-loop step. We use the full preconditioner and not its diagonal approximation.

### E.5 Amortisation-based meta-learners

**CNAPs (Requeima et al., 2019).** CNAPs (Conditional Neural Adaptive Processes) treats each task as a draw from a distribution over predictive functions (Requeima et al., 2019). A permutation-invariant encoder maps the labelled support set to a task vector $\zeta$. The task vector scales and shifts the features of every block of the network. The support set is read once, and every query is scored in a single forward pass without gradient steps.

**CAML (Fifty et al., 2024).** CAML (Context-Aware Meta-Learning) is a non-causal Transformer over the labelled support sequence followed by a query token (Fifty et al., 2024). The prediction is read at the output position of the query, without prototypes or task vectors. The visual encoder is frozen, so the Transformer learns to classify from the labels in its input alone. CAML is thus the closest counterpart of in-context learning in large language models among the amortisation-based meta-learners.

### E.6 In-context learners

**ICL-Causal (Chan et al., 2022).** ICL-Causal is a small causal Transformer trained from scratch on sequences of $(\mathbf{z}, y)$ pairs (Chan et al., 2022). The support set forms a sequence of example and label tokens, followed by the unlabelled query. The output at the query position is the predicted label. On linear-regression tasks, the optimum of this objective implements one gradient step on the support loss (von Oswald et al., 2023). It is the simplest in-context learner of our comparison, with causal masking as its only architectural prior.

**ICD-Causal.** ICD-Causal is the causal Transformer that we pre-train on the synthetic prior of Section 3.2 in Experiment 2. It differs from ICL-Causal in its tokens. ICL-Causal alternates example tokens and label tokens. ICD-Causal concatenates each projected support embedding with a learnt label embedding of dimension $d_\ell$ into one token, and the query token carries an extra label embedding that marks it as unlabelled. The sequence is thus half as long. The rest of the architecture follows ICL-Causal. It is a pre-norm causal Transformer with sinusoidal positions that classifies from the last position. We use $d_h$=512, $L$=6, $H$=8, $d_\ell$=64, dropout 0.1, label smoothing 0.1, the learning rate $5\times10^{-5}$, and 30 000 pre-training iterations with the synthetic prior of Appendix C.

**ICL-GP (Kirsch et al., 2022).** ICL-GP (General-Purpose In-Context Learning) follows the causal-Transformer recipe of ICL-Causal with the tokens of (Kirsch et al., 2022). Each token concatenates a projected embedding with the one-hot label of the *previous* position, so the model predicts every label from its causal prefix. With enough task diversity, this construction yields an in-context learner

that transfers across tasks. We train ICL-GP from scratch on our episodes as the deeper counterpart of ICL-Causal.

**TabPFN (Müller et al., 2022).** TabPFN is a prior-fitted network, a Transformer trained on millions of synthetic tabular classification tasks (Müller et al., 2022). The tasks are drawn from a prior over structural causal models, and the training objective approximates the Bayesian posterior predictive of each query. With infinite synthetic data, the trained model approaches exact Bayesian inference under this prior. We apply TabPFN to the frozen CLIP embeddings and treat each episode as a table with $N_S$ training rows and $N_Q$ test rows. It handles at most ten classes.

# F Hyperparameter Settings

A common set of hyperparameters is shared by every model and every experiment.

## F.1 Feature extractor

In every experiment except Experiment 5, the feature extractor $h_\psi$ of Section 3 is a frozen CLIP ViT-L/14 (Radford et al., 2021) image encoder. Experiment 5 replaces it by CLIP ViT-B/32 and DINOv2-L (Appendix K). CLIP ViT-L/14 maps support and query images to embeddings in $\mathbb{R}^D$ with $D$=768. Freezing $h_\psi$ leaves all adaptation to $f_\phi$ and separates the effect of meta-learning from that of representation learning. CLIP features suit AI-generated image detection (Ojha et al., 2023; Cozzolino et al., 2024).

## F.2 Image preprocessing

Every image is decoded to RGB and transformed as follows, depending on the split.

- *Training.* The shorter side is resized to 256 pixels. A Gaussian blur with kernel size 5 and $\sigma \sim \mathrm{U}[0.001, 3]$ and a JPEG re-encoding with quality $\sim \mathrm{U}\{30, \ldots, 100\}$ are each applied with probability 0.1. A random crop of 224×224 pixels and a horizontal flip with probability 0.5 follow.
- *Validation.* The same transformations are applied, except that a central crop replaces the random crop and the flip is omitted.
- *Test.* Only a central crop of 224×224 pixels is applied.

In all three splits, the image is finally normalised by the CLIP mean $(0.4815, 0.4578, 0.4082)$ and standard deviation $(0.2686, 0.2613, 0.2758)$.

## F.3 Optimisation

Every model is trained with AdamW, weight decay 0.01, $\epsilon$=$10^{-8}$, gradient-norm clipping at 1.0, and no warm-up unless stated otherwise. Models on cached embeddings use the learning rate $10^{-3}$ and are trained for three epochs, and models on images are trained for six epochs with the learning rates of Table 3. The schedule is fixed and the last epoch is kept, because a validation split drawn from the training generators saturates within the first epoch and cannot select a checkpoint. Unless stated otherwise, both constructions read support sets of $N_S$=16 images, split as described in Section 5. Table 3 lists the model-specific hyperparameters, and its caption defines all symbols.

## F.4 Random seeds

Each configuration is trained with three training seeds. A seed controls the weight initialisation and all random operations during training, namely the mini-batches, the episodes, and the data augmentation. Every checkpoint is evaluated with one sampling of support sets, because the spread over samplings is at most 0.01 and far below the spread over training seeds. All reported values are means over the three training seeds, and standard deviations are taken over the same seeds.

## F.5 Compute

All models on the frozen CLIP ViT-L/14 feature extractor use embeddings that are computed once and cached in ten augmented copies per image. The adaptation function is thus trained entirely on

Table 3: *Model-specific hyperparameters.* $d_h$ is the hidden dimension of the adaptation function, $L$ the number of Transformer layers, $H$ the number of attention heads, and $p_d$ the dropout probability. $\eta_{\text{in}}$ is the inner-loop learning rate and $T$ the number of inner-loop steps. *2nd* marks second-order MAML. $d_\varsigma$ is the dimension of the task vector of CNAPs, $d_\ell$ the dimension of the label embedding of CAML and ICD-Causal, $T_{\text{FCE}}$ the number of FCE refinement steps of MatchNet, and $E$ the number of estimators of TabPFN. GAP uses the full preconditioner of (Kang et al., 2023). PatchCraft, BSA, AIDE, MLEP, and OSD take images as input and are trained end-to-end with the learning rate $\eta_{\text{out}}$ of the *extra* column. All other models operate on cached CLIP ViT-L/14 embeddings.

| Model | $d_h$ | $L$ | $H$ | $p_d$ | $\eta_{\text{in}}$ | $T$ | 2nd / aux | extra |
|---|---|---|---|---|---|---|---|---|
| UnivFD-ZS | – | – | – | – | – | – | – | linear head, CLIP-L/14 |
| UnivFD-FT | – | – | – | – | – | – | – | linear head, $\leq$ 100 ep. on the support set |
| PatchCraft | – | – | – | – | – | – | – | CNN of five groups, $\eta_{\text{out}}{=}10^{-4}$, $N{=}3$ |
| BSA | – | – | – | – | – | – | – | patch 32, 18 conv. blocks, $\eta_{\text{out}}{=}10^{-4}$ |
| AIDE | – | – | – | – | – | – | – | CLIP-B/32+ResNet-50, $\eta_{\text{out}}{=}10^{-4}$ |
| MLEP | – | – | – | – | – | – | – | ResNet-50, patch 4, $\eta_{\text{out}}{=}10^{-4}$, 500 warm-up steps |
| OSD | – | – | – | – | – | – | – | CLIP-B/32, $r{=}8$, penalty 1, $\eta_{\text{out}}{=}10^{-3}$ |
| CORAL | 256 | – | – | – | – | – | – | penalty weight 1 |
| IRM | 256 | – | – | – | – | – | – | penalty weight 1 |
| GroupDRO | 256 | – | – | – | – | – | – | group step size 0.01 |
| TENT | 256 | – | – | – | $10^{-3}$ | 1 | – | test-time entropy minimisation |
| MatchNet | 256 | – | – | – | – | – | – | FCE, $T_{\text{FCE}}{=}20$ |
| MAML | 256 | – | – | – | 0.1 | 50 | yes | – |
| ProtoNet | 256 | – | – | – | – | – | – | – |
| Reptile | 256 | – | – | – | 0.1 | 50 | – | – |
| CNAPs | 256 | – | – | – | – | – | – | $d_\varsigma{=}128$ |
| FEAT | 256 | 1 | 1 | 0.1 | – | – | – | out-drop 0.5 |
| ICL-Causal | 256 | 1 | 8 | 0.1 | – | – | – | causal, interleaved |
| ICL-GP | 256 | 3 | 8 | 0.1 | – | – | – | causal |
| TabPFN | 128 | – | – | – | – | – | – | $E{=}1$ |
| GAP | 256 | – | – | – | 0.1 | 50 | – | full preconditioner |
| CAML | – | 1 | 8 | 0.1 | – | – | – | $d_\ell{=}256$, MLP 1024 |
| AFRT | 256 | – | – | – | – | – | – | adaptive rectification |
| NearestCentroid | – | – | – | – | – | – | – | no trained parameters |
| ICD-Causal | 512 | 6 | 8 | 0.1 | – | – | – | causal, one token per pair, $d_\ell{=}64$ |

cached embeddings. This keeps the GPU memory low. All meta-learners can be trained on a CPU or on a single commodity GPU. Context-free detectors that process raw images (PatchCraft, BSA, AIDE, MLEP, OSD) were trained on NVIDIA V100 or A100 GPUs. Exact per-run wall-clock times and total-compute estimates were not tracked systematically across all experiments.

# G Experiment 1: Implicit and Explicit Episode Construction

This appendix complements Section 5 with the accuracy per family under each support policy, Chameleon, the remaining metrics, and the per-generator tables.

## G.1 Experiment description

Every detector is trained on the training pool under our implicit and under the explicit construction. Detectors that read no support set are trained under the label space of each construction as well. Under the explicit construction, a static detector is thus a classifier with one class for real images and one class per generator. Each checkpoint is evaluated on all thirty-six generators of AI-GenBench under the sibling, the matched, and the disjoint support policy, and on Chameleon under the matched policy. Besides ACC, we report the area under the ROC curve (AUC), the average precision (AP), the adaptive calibration error (ACE, lower is better), and the false-negative rate at a false-positive rate of 1% (FNR@1%FPR, lower is better) under all three policies. The per-generator tables report the mean and the standard deviation over three training seeds.

## G.2 Support policies

Tables 4–6 split the averages of Table 1 by family and test group, for the sibling, the matched, and the disjoint policy. Each table has our implicit construction on the left and the explicit construction on the right. Under the sibling policy (Table 4), the context-aware detectors reach the highest average, 0.88 for MAML, Reptile, FEAT, and GAP. Context-free detectors trained end-to-end on images stay far behind, at most 0.74. The test groups do not order the difficulty of the generators. The unseen siblings of Diff-Px are the hardest columns, where no detector exceeds 0.74. The unseen families GAN-C and GAN-Diff, in contrast, are detected nearly as well as seen generators. Under

the matched policy (Table 5), the support set contains images of the query's own generator. All context-aware detectors gain, and the explicit ones gain most (MAML, ProtoNet, Reptile, FEAT, and GAP reach 0.92). NearestCentroid reaches 0.90 and matches the best implicit detectors, which shows how much information a support set from the query's generator carries by itself. Under the disjoint policy (Table 6), the explicit context-aware detectors fall to 0.79–0.83, while the implicit ones except TabPFN keep 0.85–0.87, and NearestCentroid falls to 0.73.

### G.3 CHAMELEON

Table 15 repeats the Chameleon column of Table 1 with the differences from UnivFD-FT and from NearestCentroid. Only TabPFN exceeds NearestCentroid, by 0.01. Only TabPFN, NearestCentroid, FEAT, and AFRT exceed UnivFD-FT.

### G.4 OTHER METRICS

Tables 16–19 report the remaining metrics under the sibling policy. Under our implicit construction, the context-aware detectors are the best calibrated. MAML, Reptile, GAP, and AFRT reach the lowest ACE (0.07), followed by ProtoNet (0.08) and the domain-aware detectors (0.09–0.10). ProtoNet, CNAPs, and FEAT share the highest AUC (0.94) with the domain-aware detectors. MAML, ProtoNet, Reptile, CNAPs, ICL-Causal, and GAP share the highest AP (0.93) with them. UnivFD-ZS reaches an AUC of 0.92 although its ACC is only 0.84, so a part of its deficit in ACC comes from its threshold and not from its ranking of the images. At a false-positive rate of 1%, ICL-Causal misses the fewest AI-generated images (FNR 0.45), followed by the domain-aware detectors (0.46–0.48), CNAPs, and ICL-GP (0.49). FEAT (0.88) and NearestCentroid (0.97) miss most of them, although both are among the most accurate detectors at the default threshold. Under the explicit construction, MAML (0.44), GAP (0.45), Reptile, and TENT (0.46) reach the lowest FNR. The unseen siblings of Diff-Px and the unseen family Diff-Lat are the columns where most AI-generated images pass undetected. Tables 20–24 give every metric per generator, each on a page of its own.

Tables 7–14 report the same four metrics under the matched and the disjoint policy. Under the matched policy, the context-aware detectors lead on every metric. FEAT reaches the highest AUC (0.96) and AP (0.95) under our construction, and MAML, Reptile, and GAP the lowest ACE (0.04). Under the disjoint policy, our construction keeps these metrics close to their values under the sibling policy. ProtoNet and CNAPs share the highest AUC (0.94) with the domain-aware detectors, six context-aware detectors share the lowest ACE (0.09) with CORAL, and ICL-Causal keeps the lowest FNR (0.45). The explicit construction degrades more. From the matched to the disjoint policy, the AUC of FEAT falls from 0.97 to 0.88 under the explicit construction but only from 0.96 to 0.93 under ours.

Table 4: *Experiment 1 – Sibling support policy – ACC.* Accuracy averaged over the test generators of each family and test group, for our implicit construction (left) and the explicit construction with an oracle label space (right). *Average* is the mean over all thirty-six generators. Top-3 entries per column are coloured. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.55 | 0.64 | 0.60 | 0.53 | 0.60 | 0.52 | 0.50 | 0.63 | 0.64 | 0.56 | 0.70 | 0.58 | 0.48 | 0.73 | 0.59 | 0.51 | 0.54 | 0.52 | 0.51 | 0.66 | 0.57 | 0.57 | 0.79 | 0.58 |
| UnivFD-FT (2023) | 0.91 | 0.80 | 0.91 | 0.84 | 0.90 | 0.70 | 0.76 | 0.90 | 0.91 | 0.77 | 0.90 | 0.84 | 0.94 | 0.80 | 0.91 | 0.88 | 0.89 | 0.70 | 0.71 | 0.90 | 0.93 | 0.75 | 0.94 | 0.83 |
| UnivFD-ZS (2023) | 0.90 | 0.84 | 0.90 | 0.86 | 0.90 | 0.71 | 0.72 | 0.90 | 0.90 | 0.79 | 0.90 | 0.84 | 0.93 | 0.85 | 0.91 | 0.88 | 0.90 | 0.71 | 0.68 | 0.90 | 0.92 | 0.75 | 0.89 | 0.83 |
| BSA (2024) | 0.60 | 0.60 | 0.62 | 0.59 | 0.68 | 0.67 | 0.54 | 0.57 | 0.70 | 0.66 | 0.84 | 0.64 | 0.59 | 0.52 | 0.67 | 0.72 | 0.66 | 0.38 | 0.56 | 0.59 | 0.67 | 0.47 | 0.70 | 0.58 |
| AIDE (2025) | 0.85 | 0.78 | 0.79 | 0.76 | 0.79 | 0.68 | 0.58 | 0.75 | 0.82 | 0.69 | 0.85 | 0.74 | 0.89 | 0.80 | 0.78 | 0.78 | 0.80 | 0.69 | 0.58 | 0.73 | 0.84 | 0.65 | 0.85 | 0.73 |
| MLEP (2025) | 0.57 | 0.68 | 0.45 | 0.42 | 0.49 | **0.74** | 0.47 | 0.46 | 0.50 | 0.64 | 0.70 | 0.55 | 0.60 | 0.72 | 0.52 | 0.48 | 0.54 | 0.70 | 0.47 | 0.51 | 0.56 | 0.61 | 0.67 | 0.57 |
| OSD (2025) | 0.90 | 0.88 | 0.87 | 0.79 | 0.86 | 0.71 | 0.68 | 0.87 | 0.88 | 0.72 | 0.85 | 0.80 | 0.86 | 0.89 | 0.83 | 0.72 | 0.79 | 0.73 | 0.56 | 0.77 | 0.74 | 0.70 | 0.90 | 0.75 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | 0.93 | 0.90 | 0.93 | 0.90 | 0.93 | **0.74** | 0.82 | 0.93 | 0.93 | **0.80** | 0.90 | 0.87 | 0.95 | 0.91 | 0.94 | **0.91** | **0.93** | **0.74** | 0.72 | 0.92 | 0.94 | 0.75 | 0.85 | 0.85 |
| IRM (2019) | 0.93 | 0.90 | 0.92 | 0.90 | 0.92 | **0.74** | 0.82 | 0.92 | 0.92 | 0.79 | 0.88 | 0.87 | 0.75 | 0.71 | 0.76 | 0.74 | 0.75 | 0.64 | 0.64 | 0.76 | 0.75 | 0.69 | 0.75 | 0.72 |
| GroupDRO (2020) | 0.93 | **0.91** | 0.92 | 0.90 | 0.92 | **0.74** | 0.82 | 0.93 | 0.93 | 0.79 | 0.87 | 0.87 | 0.84 | 0.84 | 0.84 | 0.83 | 0.84 | 0.70 | 0.69 | 0.84 | 0.84 | **0.76** | 0.83 | 0.80 |
| TENT (2021) | **0.95** | **0.91** | 0.93 | **0.91** | **0.94** | **0.74** | 0.81 | 0.94 | 0.94 | 0.78 | 0.86 | 0.87 | 0.95 | **0.92** | 0.94 | **0.91** | **0.93** | **0.74** | 0.72 | 0.93 | **0.95** | 0.75 | 0.86 | 0.85 |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.91 | 0.85 | 0.91 | 0.87 | 0.91 | 0.72 | 0.76 | 0.91 | 0.91 | 0.79 | 0.88 | 0.85 | 0.94 | 0.84 | 0.94 | 0.85 | 0.92 | 0.71 | 0.74 | **0.94** | 0.92 | 0.74 | 0.82 | 0.84 |
| MAML (2017) | **0.95** | 0.88 | **0.94** | 0.90 | **0.94** | 0.72 | 0.81 | 0.94 | 0.95 | 0.79 | 0.89 | **0.88** | 0.95 | 0.78 | **0.95** | **0.91** | 0.92 | 0.68 | **0.82** | 0.93 | 0.94 | 0.75 | 0.96 | **0.86** |
| ProtoNet (2017) | **0.95** | 0.88 | **0.94** | 0.90 | **0.94** | 0.71 | **0.83** | 0.93 | **0.96** | 0.79 | 0.86 | 0.87 | 0.95 | 0.79 | 0.94 | 0.88 | 0.92 | 0.68 | 0.80 | 0.92 | 0.91 | **0.76** | 0.96 | 0.85 |
| Reptile (2018) | **0.95** | 0.89 | **0.94** | 0.90 | **0.94** | 0.72 | 0.82 | 0.94 | 0.95 | 0.79 | 0.88 | **0.88** | 0.95 | 0.80 | 0.94 | 0.89 | 0.92 | 0.68 | **0.82** | 0.92 | 0.93 | **0.76** | 0.96 | **0.86** |
| CNAPs (2019) | **0.95** | 0.89 | **0.94** | 0.90 | **0.94** | 0.73 | 0.78 | 0.93 | 0.95 | 0.77 | 0.85 | 0.86 | 0.95 | 0.79 | 0.94 | 0.88 | **0.93** | 0.69 | 0.75 | 0.92 | 0.86 | **0.76** | 0.90 | 0.84 |
| FEAT (2020) | **0.95** | 0.85 | **0.94** | 0.90 | **0.94** | 0.73 | 0.81 | **0.95** | 0.95 | 0.79 | 0.93 | **0.88** | **0.96** | 0.77 | 0.93 | 0.89 | **0.93** | 0.70 | **0.82** | **0.94** | 0.93 | 0.75 | 0.96 | **0.86** |
| ICL-Causal (2022) | 0.94 | 0.89 | 0.93 | 0.89 | 0.93 | 0.73 | 0.80 | 0.94 | 0.93 | 0.77 | 0.85 | 0.86 | **0.96** | 0.87 | 0.94 | **0.91** | 0.91 | 0.73 | 0.71 | 0.91 | **0.95** | 0.71 | 0.82 | 0.84 |
| ICL-GP (2022) | 0.93 | 0.87 | 0.92 | 0.88 | 0.92 | 0.73 | 0.79 | 0.92 | 0.92 | 0.78 | 0.88 | 0.86 | 0.94 | 0.85 | 0.92 | 0.90 | 0.88 | 0.72 | 0.70 | 0.89 | 0.93 | 0.71 | 0.86 | 0.82 |
| TabPFN (2022) | 0.75 | 0.61 | 0.76 | 0.74 | 0.77 | 0.60 | 0.73 | 0.76 | 0.72 | 0.64 | 0.96 | 0.72 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | **0.95** | 0.88 | **0.94** | 0.90 | **0.94** | 0.72 | 0.82 | 0.94 | 0.95 | 0.79 | 0.89 | **0.88** | 0.95 | 0.78 | **0.95** | **0.91** | 0.92 | 0.67 | **0.82** | 0.93 | 0.94 | 0.75 | 0.96 | **0.86** |
| CAML (2024) | 0.91 | 0.85 | 0.91 | 0.87 | 0.91 | 0.72 | 0.77 | 0.91 | 0.91 | 0.78 | 0.88 | 0.85 | 0.95 | 0.77 | 0.93 | 0.88 | 0.90 | 0.72 | 0.72 | 0.91 | 0.94 | 0.63 | 0.70 | 0.80 |
| AFRT (2025) | 0.94 | 0.82 | 0.93 | 0.86 | 0.93 | 0.70 | 0.79 | 0.93 | 0.94 | 0.78 | 0.90 | 0.86 | 0.95 | 0.77 | 0.91 | 0.86 | 0.92 | 0.69 | 0.79 | **0.94** | 0.93 | 0.75 | 0.95 | 0.85 |
| NearestCentroid | 0.81 | 0.64 | 0.82 | 0.79 | 0.82 | 0.61 | 0.76 | 0.82 | 0.70 | 0.72 | **0.98** | 0.77 | 0.78 | 0.63 | 0.82 | 0.82 | 0.79 | 0.62 | 0.80 | 0.82 | 0.80 | 0.67 | **0.98** | 0.77 |

Table 5: *Experiment 1 – Matched support policy – ACC.* Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.55 | 0.64 | 0.60 | 0.53 | 0.60 | 0.52 | 0.50 | 0.63 | 0.64 | 0.56 | 0.70 | 0.58 | 0.48 | 0.73 | 0.59 | 0.51 | 0.54 | 0.52 | 0.51 | 0.66 | 0.57 | 0.57 | 0.79 | 0.58 |
| UnivFD-FT (2023) | 0.91 | 0.83 | 0.91 | 0.89 | 0.91 | 0.78 | 0.87 | 0.92 | 0.92 | 0.80 | 0.90 | 0.87 | 0.96 | 0.89 | 0.92 | 0.92 | 0.92 | 0.76 | 0.87 | 0.90 | 0.93 | 0.78 | 0.95 | 0.88 |
| UnivFD-ZS (2023) | 0.90 | 0.84 | 0.90 | 0.86 | 0.90 | 0.71 | 0.72 | 0.90 | 0.90 | 0.79 | 0.90 | 0.84 | 0.93 | 0.85 | 0.91 | 0.88 | 0.90 | 0.71 | 0.68 | 0.90 | 0.92 | 0.75 | 0.89 | 0.83 |
| BSA (2024) | 0.60 | 0.60 | 0.62 | 0.59 | 0.68 | 0.67 | 0.54 | 0.57 | 0.70 | 0.66 | 0.84 | 0.64 | 0.59 | 0.52 | 0.67 | 0.72 | 0.66 | 0.38 | 0.56 | 0.59 | 0.67 | 0.47 | 0.70 | 0.58 |
| AIDE (2025) | 0.85 | 0.78 | 0.79 | 0.76 | 0.79 | 0.68 | 0.58 | 0.75 | 0.82 | 0.69 | 0.85 | 0.74 | 0.89 | 0.80 | 0.78 | 0.78 | 0.80 | 0.69 | 0.58 | 0.73 | 0.84 | 0.65 | 0.85 | 0.73 |
| MLEP (2025) | 0.57 | 0.68 | 0.45 | 0.42 | 0.49 | 0.74 | 0.47 | 0.46 | 0.50 | 0.64 | 0.70 | 0.55 | 0.60 | 0.72 | 0.52 | 0.48 | 0.54 | 0.70 | 0.47 | 0.52 | 0.56 | 0.61 | 0.67 | 0.57 |
| OSD (2025) | 0.90 | 0.88 | 0.87 | 0.79 | 0.86 | 0.71 | 0.68 | 0.87 | 0.88 | 0.72 | 0.85 | 0.80 | 0.86 | 0.89 | 0.83 | 0.72 | 0.79 | 0.73 | 0.56 | 0.77 | 0.74 | 0.70 | 0.90 | 0.75 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | 0.93 | 0.90 | 0.93 | 0.90 | 0.93 | 0.74 | 0.82 | 0.93 | 0.93 | 0.80 | 0.90 | 0.87 | 0.95 | 0.91 | 0.94 | 0.91 | 0.93 | 0.74 | 0.72 | 0.92 | 0.94 | 0.75 | 0.85 | 0.85 |
| IRM (2019) | 0.93 | 0.90 | 0.92 | 0.90 | 0.92 | 0.74 | 0.82 | 0.92 | 0.92 | 0.79 | 0.88 | 0.87 | 0.75 | 0.71 | 0.76 | 0.74 | 0.75 | 0.64 | 0.64 | 0.76 | 0.75 | 0.69 | 0.75 | 0.72 |
| GroupDRO (2020) | 0.93 | 0.91 | 0.92 | 0.90 | 0.92 | 0.74 | 0.82 | 0.93 | 0.93 | 0.79 | 0.87 | 0.87 | 0.84 | 0.84 | 0.84 | 0.83 | 0.84 | 0.70 | 0.69 | 0.84 | 0.84 | 0.76 | 0.83 | 0.80 |
| TENT (2021) | **0.95** | 0.91 | 0.93 | 0.91 | 0.94 | 0.74 | 0.81 | 0.94 | 0.94 | 0.78 | 0.86 | 0.87 | 0.95 | 0.92 | 0.94 | 0.91 | 0.93 | 0.74 | 0.72 | 0.93 | 0.95 | 0.75 | 0.86 | 0.85 |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.91 | 0.85 | 0.91 | 0.87 | 0.91 | 0.72 | 0.76 | 0.91 | 0.91 | 0.79 | 0.88 | 0.85 | 0.95 | 0.86 | 0.94 | 0.90 | 0.93 | 0.71 | 0.74 | 0.94 | 0.95 | 0.74 | 0.82 | 0.85 |
| MAML (2017) | **0.95** | 0.91 | **0.94** | 0.92 | 0.94 | 0.81 | 0.90 | **0.95** | 0.95 | **0.83** | 0.89 | 0.90 | **0.99** | **0.94** | **0.96** | **0.97** | **0.95** | **0.86** | 0.95 | 0.96 | 0.96 | 0.81 | 0.96 | **0.92** |
| ProtoNet (2017) | **0.95** | **0.92** | **0.94** | 0.91 | 0.94 | 0.79 | 0.90 | **0.95** | **0.96** | 0.81 | 0.88 | 0.90 | **0.99** | **0.94** | 0.95 | 0.96 | 0.94 | 0.84 | 0.93 | 0.96 | 0.96 | 0.81 | 0.96 | **0.92** |
| Reptile (2018) | **0.95** | **0.92** | **0.94** | 0.91 | **0.95** | 0.80 | 0.90 | 0.94 | 0.95 | 0.82 | 0.89 | 0.90 | 0.98 | **0.94** | 0.95 | 0.96 | **0.95** | 0.85 | 0.94 | 0.96 | 0.96 | 0.81 | 0.96 | **0.92** |
| CNAPs (2019) | **0.95** | 0.91 | **0.94** | 0.90 | 0.94 | 0.73 | 0.79 | 0.93 | 0.94 | 0.77 | 0.85 | 0.87 | **0.99** | **0.94** | 0.95 | 0.95 | 0.93 | 0.78 | 0.81 | 0.94 | 0.95 | 0.77 | 0.91 | 0.88 |
| FEAT (2020) | **0.95** | 0.88 | **0.94** | 0.92 | **0.95** | 0.81 | 0.93 | **0.95** | 0.95 | 0.82 | 0.93 | **0.91** | 0.98 | 0.92 | 0.95 | **0.97** | **0.95** | **0.86** | **0.96** | **0.97** | **0.97** | 0.80 | 0.97 | **0.92** |
| ICL-Causal (2022) | 0.94 | 0.89 | 0.93 | 0.89 | 0.93 | 0.73 | 0.80 | 0.94 | 0.93 | 0.77 | 0.84 | 0.86 | 0.97 | 0.91 | 0.94 | 0.93 | 0.93 | 0.74 | 0.72 | 0.92 | 0.95 | 0.72 | 0.83 | 0.85 |
| ICL-GP (2022) | 0.93 | 0.87 | 0.92 | 0.87 | 0.92 | 0.73 | 0.79 | 0.92 | 0.92 | 0.78 | 0.88 | 0.86 | 0.96 | 0.91 | 0.93 | 0.92 | 0.90 | 0.73 | 0.70 | 0.89 | 0.92 | 0.72 | 0.86 | 0.83 |
| TabPFN (2022) | 0.88 | 0.77 | 0.80 | 0.91 | 0.89 | 0.82 | 0.95 | 0.92 | 0.91 | 0.73 | **0.98** | 0.86 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | **0.95** | **0.92** | **0.94** | 0.92 | **0.95** | 0.81 | 0.90 | 0.94 | 0.95 | 0.82 | 0.89 | 0.90 | 0.98 | **0.94** | 0.95 | 0.96 | **0.95** | **0.86** | 0.94 | 0.96 | 0.96 | **0.82** | 0.96 | **0.92** |
| CAML (2024) | 0.91 | 0.85 | 0.91 | 0.87 | 0.91 | 0.72 | 0.75 | 0.91 | 0.91 | 0.79 | 0.88 | 0.85 | 0.97 | 0.91 | 0.93 | 0.95 | 0.88 | 0.71 | 0.68 | 0.89 | 0.93 | 0.62 | 0.71 | 0.80 |
| AFRT (2025) | 0.94 | 0.87 | **0.94** | 0.88 | 0.93 | 0.75 | 0.88 | **0.95** | 0.95 | 0.80 | 0.91 | 0.89 | 0.98 | 0.91 | 0.94 | 0.96 | 0.93 | 0.85 | 0.94 | 0.96 | 0.96 | 0.80 | 0.96 | 0.91 |
| NearestCentroid | 0.92 | 0.80 | 0.87 | **0.94** | 0.90 | **0.83** | **0.96** | **0.95** | 0.93 | 0.80 | **0.98** | 0.90 | 0.92 | 0.80 | 0.87 | 0.94 | 0.90 | 0.83 | **0.96** | 0.95 | 0.93 | 0.80 | **0.98** | 0.90 |

Table 6: *Experiment 1 – Disjoint support policy – ACC.* Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.55 | 0.64 | 0.60 | 0.53 | 0.60 | 0.52 | 0.50 | 0.63 | 0.64 | 0.56 | 0.70 | 0.58 | 0.48 | 0.73 | 0.59 | 0.51 | 0.54 | 0.52 | 0.51 | 0.66 | 0.57 | 0.57 | 0.79 | 0.58 |
| UnivFD-FT (2023) | 0.90 | 0.79 | 0.90 | 0.86 | 0.90 | 0.72 | 0.72 | 0.89 | 0.87 | 0.78 | **0.90** | 0.83 | 0.92 | 0.83 | 0.90 | 0.87 | 0.90 | 0.71 | 0.68 | 0.89 | 0.91 | 0.70 | 0.83 | 0.81 |
| UnivFD-ZS (2023) | 0.90 | 0.84 | 0.90 | 0.86 | 0.90 | 0.71 | 0.72 | 0.90 | 0.90 | 0.79 | **0.90** | 0.84 | 0.93 | 0.85 | 0.91 | 0.88 | 0.90 | 0.71 | 0.68 | 0.90 | 0.92 | 0.75 | 0.89 | 0.83 |
| BSA (2024) | 0.60 | 0.60 | 0.62 | 0.59 | 0.68 | 0.67 | 0.54 | 0.57 | 0.70 | 0.66 | 0.84 | 0.64 | 0.59 | 0.52 | 0.67 | 0.72 | 0.66 | 0.38 | 0.56 | 0.59 | 0.67 | 0.47 | 0.70 | 0.58 |
| AIDE (2025) | 0.85 | 0.78 | 0.79 | 0.76 | 0.79 | 0.68 | 0.58 | 0.75 | 0.82 | 0.69 | 0.85 | 0.74 | 0.89 | 0.80 | 0.78 | 0.78 | 0.80 | 0.69 | 0.58 | 0.73 | 0.84 | 0.65 | 0.85 | 0.73 |
| MLEP (2025) | 0.57 | 0.68 | 0.45 | 0.42 | 0.50 | 0.74 | 0.47 | 0.46 | 0.49 | 0.64 | 0.70 | 0.55 | 0.60 | 0.72 | 0.52 | 0.48 | 0.54 | 0.70 | 0.47 | 0.51 | 0.56 | 0.61 | 0.67 | 0.57 |
| OSD (2025) | 0.90 | 0.88 | 0.87 | 0.79 | 0.86 | 0.71 | 0.68 | 0.87 | 0.88 | 0.72 | 0.85 | 0.80 | 0.86 | 0.89 | 0.83 | 0.72 | 0.79 | 0.73 | 0.56 | 0.77 | 0.74 | 0.70 | **0.90** | 0.75 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | 0.93 | 0.90 | 0.93 | 0.90 | 0.93 | 0.74 | 0.82 | 0.93 | 0.93 | **0.80** | **0.90** | **0.87** | **0.95** | 0.91 | **0.94** | **0.91** | **0.93** | **0.74** | 0.72 | 0.92 | 0.94 | 0.75 | 0.85 | **0.85** |
| IRM (2019) | 0.93 | 0.90 | 0.92 | 0.90 | 0.92 | 0.74 | 0.82 | 0.92 | 0.92 | 0.79 | 0.88 | **0.87** | 0.75 | 0.71 | 0.76 | 0.74 | 0.75 | 0.64 | 0.64 | 0.76 | 0.75 | 0.69 | 0.75 | 0.72 |
| GroupDRO (2020) | 0.93 | **0.91** | 0.92 | 0.90 | 0.92 | 0.74 | 0.82 | 0.93 | 0.93 | 0.79 | 0.87 | **0.87** | 0.84 | 0.84 | 0.84 | 0.83 | 0.84 | 0.70 | 0.69 | 0.84 | 0.84 | **0.76** | 0.83 | 0.80 |
| TENT (2021) | **0.95** | **0.91** | 0.93 | **0.91** | **0.94** | 0.74 | 0.81 | **0.94** | **0.94** | 0.78 | 0.86 | **0.87** | **0.95** | **0.92** | **0.94** | **0.91** | **0.93** | **0.74** | 0.72 | 0.93 | **0.95** | 0.75 | 0.86 | **0.85** |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.91 | 0.85 | 0.91 | 0.87 | 0.91 | 0.72 | 0.76 | 0.91 | 0.91 | 0.79 | 0.88 | 0.85 | 0.94 | 0.83 | **0.94** | 0.80 | 0.92 | 0.70 | 0.70 | **0.94** | 0.93 | 0.74 | 0.79 | 0.83 |
| MAML (2017) | 0.93 | **0.91** | 0.92 | 0.90 | 0.93 | **0.75** | 0.80 | 0.93 | 0.93 | 0.77 | 0.85 | 0.86 | 0.89 | 0.80 | 0.88 | 0.82 | 0.88 | 0.73 | 0.73 | 0.88 | 0.88 | 0.73 | 0.83 | 0.81 |
| ProtoNet (2017) | 0.93 | **0.91** | 0.92 | 0.90 | 0.93 | 0.74 | **0.83** | 0.92 | 0.93 | 0.77 | 0.83 | 0.86 | 0.89 | 0.84 | 0.88 | 0.77 | 0.88 | 0.73 | 0.71 | 0.88 | 0.89 | 0.72 | 0.79 | 0.81 |
| Reptile (2018) | 0.93 | **0.91** | 0.93 | 0.90 | 0.93 | **0.75** | 0.81 | 0.93 | 0.93 | 0.78 | 0.86 | **0.87** | 0.89 | 0.81 | 0.88 | 0.77 | 0.87 | 0.72 | 0.71 | 0.88 | 0.88 | 0.73 | 0.82 | 0.80 |
| CNAPs (2019) | 0.94 | **0.91** | **0.94** | 0.90 | **0.94** | 0.74 | 0.78 | 0.93 | **0.94** | 0.77 | 0.85 | 0.86 | 0.91 | 0.84 | 0.91 | 0.81 | 0.90 | 0.72 | 0.70 | 0.90 | 0.90 | 0.73 | 0.76 | 0.81 |
| FEAT (2020) | 0.94 | 0.86 | 0.93 | 0.89 | 0.93 | 0.74 | 0.76 | **0.94** | 0.93 | 0.76 | 0.88 | 0.86 | 0.89 | 0.79 | 0.88 | 0.79 | 0.87 | 0.70 | 0.72 | 0.88 | 0.89 | 0.69 | 0.70 | 0.79 |
| ICL-Causal (2022) | 0.94 | 0.89 | 0.93 | 0.89 | 0.93 | 0.73 | 0.80 | **0.94** | 0.93 | 0.77 | 0.84 | 0.86 | **0.95** | 0.87 | 0.92 | 0.89 | 0.92 | 0.72 | 0.69 | 0.91 | 0.94 | 0.71 | 0.76 | 0.83 |
| ICL-GP (2022) | 0.93 | 0.87 | 0.92 | 0.88 | 0.92 | 0.73 | 0.79 | 0.92 | 0.92 | 0.79 | 0.88 | 0.86 | 0.93 | 0.83 | 0.91 | 0.86 | 0.89 | 0.71 | 0.67 | 0.88 | 0.92 | 0.70 | 0.81 | 0.81 |
| TabPFN (2022) | 0.72 | 0.64 | 0.70 | 0.71 | 0.71 | 0.62 | 0.69 | 0.72 | 0.70 | 0.57 | 0.61 | 0.67 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | 0.93 | **0.91** | 0.93 | 0.90 | 0.93 | **0.75** | 0.81 | 0.93 | 0.93 | 0.77 | 0.85 | 0.86 | 0.89 | 0.80 | 0.87 | 0.82 | 0.88 | 0.73 | **0.74** | 0.88 | 0.88 | 0.74 | 0.84 | 0.82 |
| CAML (2024) | 0.91 | 0.86 | 0.91 | 0.87 | 0.91 | 0.72 | 0.77 | 0.91 | 0.91 | 0.79 | 0.88 | 0.85 | 0.93 | 0.76 | 0.90 | 0.85 | 0.88 | 0.69 | 0.67 | 0.89 | 0.92 | 0.69 | 0.74 | 0.80 |
| AFRT (2025) | 0.92 | 0.87 | 0.91 | 0.87 | 0.91 | 0.73 | 0.77 | 0.91 | 0.91 | 0.77 | 0.86 | 0.85 | 0.87 | 0.78 | 0.86 | 0.77 | 0.87 | 0.71 | 0.71 | 0.87 | 0.87 | 0.71 | 0.73 | 0.79 |
| NearestCentroid | 0.79 | 0.71 | 0.77 | 0.76 | 0.76 | 0.60 | 0.76 | 0.80 | 0.74 | 0.65 | 0.72 | 0.73 | 0.73 | 0.69 | 0.72 | 0.73 | 0.70 | 0.59 | 0.69 | 0.71 | 0.71 | 0.63 | 0.68 | 0.69 |

Table 7: *Experiment 1 – Matched support policy – AUC.* Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.57 | 0.71 | 0.64 | 0.56 | 0.62 | 0.51 | 0.46 | 0.69 | 0.67 | 0.59 | 0.79 | 0.61 | 0.51 | 0.79 | 0.65 | 0.56 | 0.62 | 0.53 | 0.50 | 0.71 | 0.66 | 0.60 | 0.88 | 0.62 |
| UnivFD-FT (2023) | **0.99** | 0.92 | 0.97 | 0.95 | 0.98 | 0.83 | 0.95 | **0.99** | **0.99** | **0.90** | 0.99 | 0.95 | **1.00** | 0.96 | 0.97 | 0.98 | 0.97 | 0.83 | 0.94 | 0.97 | 0.98 | 0.88 | 0.98 | 0.94 |
| UnivFD-ZS (2023) | **0.99** | 0.91 | 0.97 | 0.94 | 0.97 | 0.76 | 0.83 | 0.98 | 0.98 | 0.87 | 0.96 | 0.92 | 0.99 | 0.92 | 0.97 | 0.96 | 0.96 | 0.75 | 0.79 | 0.96 | 0.98 | 0.84 | 0.95 | 0.90 |
| BSA (2024) | 0.66 | 0.66 | 0.68 | 0.67 | 0.74 | 0.71 | 0.60 | 0.62 | 0.76 | 0.72 | 0.98 | 0.70 | 0.63 | 0.54 | 0.72 | 0.78 | 0.72 | 0.38 | 0.58 | 0.62 | 0.74 | 0.47 | 0.66 | 0.61 |
| AIDE (2025) | 0.92 | 0.86 | 0.88 | 0.85 | 0.87 | 0.72 | 0.66 | 0.85 | 0.89 | 0.77 | 0.93 | 0.82 | 0.95 | 0.88 | 0.88 | 0.85 | 0.88 | 0.72 | 0.60 | 0.83 | 0.91 | 0.75 | 0.92 | 0.81 |
| MLEP (2025) | 0.62 | 0.75 | 0.47 | 0.45 | 0.55 | 0.82 | 0.51 | 0.50 | 0.55 | 0.71 | 0.80 | 0.60 | 0.63 | 0.79 | 0.51 | 0.48 | 0.57 | 0.81 | 0.45 | 0.52 | 0.58 | 0.71 | 0.79 | 0.61 |
| OSD (2025) | 0.97 | 0.96 | 0.94 | 0.87 | 0.94 | 0.76 | 0.80 | 0.94 | 0.95 | 0.83 | 0.93 | 0.89 | 0.95 | 0.95 | 0.93 | 0.82 | 0.89 | 0.76 | 0.75 | 0.89 | 0.89 | 0.81 | 0.96 | 0.86 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | **0.99** | 0.96 | **0.99** | **0.97** | **0.99** | 0.80 | 0.92 | **0.99** | **0.99** | 0.89 | 0.95 | 0.94 | 0.99 | 0.96 | 0.98 | 0.97 | 0.98 | 0.80 | 0.87 | 0.98 | **0.99** | 0.87 | 0.93 | 0.93 |
| IRM (2019) | **0.99** | 0.96 | **0.99** | **0.97** | **0.99** | 0.79 | 0.91 | **0.99** | **0.99** | 0.87 | 0.95 | 0.94 | 0.96 | 0.88 | 0.96 | 0.93 | 0.93 | 0.73 | 0.77 | 0.95 | 0.94 | 0.84 | 0.92 | 0.88 |
| GroupDRO (2020) | **0.99** | **0.97** | **0.99** | **0.97** | **0.99** | 0.79 | 0.92 | **0.99** | **0.99** | 0.88 | 0.94 | 0.94 | 0.98 | 0.94 | 0.98 | 0.94 | 0.96 | 0.78 | 0.78 | 0.96 | 0.97 | 0.85 | 0.92 | 0.90 |
| TENT (2021) | **0.99** | **0.97** | **0.99** | **0.97** | **0.99** | 0.79 | 0.92 | **0.99** | **0.99** | 0.87 | 0.94 | 0.94 | **1.00** | 0.97 | **0.99** | 0.97 | 0.98 | 0.80 | 0.86 | 0.98 | **0.99** | 0.86 | 0.94 | 0.93 |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.98 | 0.93 | 0.97 | 0.94 | 0.97 | 0.77 | 0.84 | 0.98 | 0.98 | 0.85 | 0.94 | 0.92 | 0.99 | 0.94 | 0.98 | 0.97 | 0.96 | 0.75 | 0.90 | 0.98 | 0.98 | 0.78 | 0.91 | 0.91 |
| MAML (2017) | **0.99** | **0.97** | 0.98 | **0.97** | 0.98 | 0.85 | 0.96 | **0.99** | **0.99** | 0.89 | 0.95 | 0.95 | **1.00** | **0.99** | **0.99** | 0.99 | **0.99** | 0.90 | 0.98 | 0.99 | **0.99** | 0.89 | 0.99 | 0.96 |
| ProtoNet (2017) | **0.99** | **0.97** | 0.98 | **0.97** | 0.98 | 0.82 | 0.95 | 0.98 | **0.99** | 0.88 | 0.94 | 0.95 | **1.00** | 0.98 | **0.99** | 0.99 | 0.98 | 0.88 | 0.97 | 0.99 | **0.99** | 0.88 | 0.99 | 0.96 |
| Reptile (2018) | **0.99** | **0.97** | 0.98 | **0.97** | 0.98 | 0.84 | 0.95 | **0.99** | **0.99** | 0.89 | 0.94 | 0.95 | **1.00** | **0.99** | **0.99** | 0.99 | **0.99** | 0.89 | 0.98 | 0.99 | **0.99** | 0.88 | 0.99 | 0.96 |
| CNAPs (2019) | **0.99** | **0.97** | **0.99** | **0.97** | **0.99** | 0.77 | 0.90 | 0.98 | **0.99** | 0.87 | 0.94 | 0.93 | **1.00** | **0.99** | **0.99** | 0.99 | 0.98 | 0.79 | 0.92 | 0.99 | **0.99** | 0.87 | 0.97 | 0.94 |
| FEAT (2020) | 0.98 | 0.95 | 0.98 | **0.97** | **0.99** | **0.86** | 0.98 | **0.99** | **0.99** | **0.90** | 0.98 | **0.96** | **1.00** | 0.97 | **0.99** | **1.00** | **0.99** | **0.91** | **0.99** | **1.00** | **0.99** | **0.90** | **1.00** | **0.97** |
| ICL-Causal (2022) | **0.99** | 0.96 | 0.98 | 0.96 | 0.98 | 0.79 | 0.90 | **0.99** | **0.99** | 0.87 | 0.92 | 0.93 | **1.00** | 0.98 | **0.99** | 0.99 | 0.98 | 0.81 | 0.89 | 0.98 | **0.99** | 0.86 | 0.94 | 0.93 |
| ICL-GP (2022) | **0.99** | 0.94 | 0.98 | 0.96 | 0.98 | 0.78 | 0.89 | **0.99** | **0.99** | 0.87 | 0.95 | 0.93 | **1.00** | 0.97 | 0.98 | 0.98 | 0.96 | 0.79 | 0.86 | 0.97 | 0.98 | 0.85 | 0.94 | 0.92 |
| TabPFN (2022) | 0.95 | 0.84 | 0.88 | 0.96 | 0.94 | **0.86** | 0.98 | 0.97 | 0.96 | 0.80 | **1.00** | 0.91 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | **0.99** | **0.97** | 0.98 | **0.97** | 0.98 | 0.84 | 0.96 | **0.99** | **0.99** | 0.89 | 0.95 | 0.95 | **1.00** | **0.99** | **0.99** | 0.99 | **0.99** | 0.90 | 0.98 | 0.99 | **0.99** | 0.89 | 0.99 | **0.97** |
| CAML (2024) | **0.99** | 0.92 | 0.97 | 0.95 | 0.97 | 0.77 | 0.85 | 0.98 | 0.98 | 0.86 | 0.95 | 0.92 | **1.00** | 0.97 | 0.98 | 0.99 | 0.97 | 0.76 | 0.80 | 0.98 | 0.98 | 0.84 | 0.92 | 0.91 |
| AFRT (2025) | **0.99** | 0.94 | 0.98 | 0.95 | 0.98 | 0.80 | 0.96 | **0.99** | **0.99** | 0.88 | 0.98 | 0.94 | **1.00** | 0.97 | 0.98 | 0.99 | 0.98 | 0.90 | 0.98 | 0.99 | **0.99** | 0.88 | 0.99 | 0.96 |
| NearestCentroid | 0.97 | 0.87 | 0.94 | **0.97** | 0.95 | **0.86** | **0.99** | 0.98 | 0.96 | 0.87 | 0.99 | 0.94 | 0.97 | 0.87 | 0.94 | 0.97 | 0.95 | 0.86 | **0.99** | 0.98 | 0.96 | 0.87 | 0.99 | 0.94 |

Table 8: *Experiment 1 – Matched support policy – AP.* Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.53 | 0.68 | 0.61 | 0.51 | 0.57 | 0.49 | 0.49 | 0.66 | 0.62 | 0.58 | 0.75 | 0.58 | 0.49 | 0.77 | 0.64 | 0.53 | 0.57 | 0.50 | 0.51 | 0.71 | 0.60 | 0.59 | 0.85 | 0.60 |
| UnivFD-FT (2023) | **0.99** | 0.91 | 0.97 | 0.95 | **0.98** | 0.82 | 0.94 | 0.98 | **0.99** | **0.88** | 0.99 | 0.94 | **1.00** | 0.95 | 0.96 | 0.98 | 0.96 | 0.80 | 0.93 | 0.96 | 0.97 | 0.85 | 0.98 | 0.93 |
| UnivFD-ZS (2023) | **0.99** | 0.90 | 0.97 | 0.94 | 0.97 | 0.74 | 0.79 | 0.97 | 0.98 | 0.84 | 0.94 | 0.90 | 0.99 | 0.91 | 0.95 | 0.96 | 0.96 | 0.73 | 0.74 | 0.96 | 0.97 | 0.81 | 0.93 | 0.89 |
| BSA (2024) | 0.61 | 0.62 | 0.63 | 0.58 | 0.67 | 0.66 | 0.55 | 0.58 | 0.66 | 0.69 | 0.98 | 0.65 | 0.59 | 0.50 | 0.66 | 0.71 | 0.65 | 0.41 | 0.58 | 0.59 | 0.68 | 0.46 | 0.55 | 0.57 |
| AIDE (2025) | 0.90 | 0.84 | 0.85 | 0.82 | 0.84 | 0.69 | 0.62 | 0.82 | 0.86 | 0.73 | 0.90 | 0.79 | 0.94 | 0.86 | 0.85 | 0.83 | 0.85 | 0.70 | 0.59 | 0.79 | 0.88 | 0.71 | 0.89 | 0.78 |
| MLEP (2025) | 0.62 | 0.73 | 0.46 | 0.43 | 0.54 | 0.79 | 0.47 | 0.49 | 0.55 | 0.68 | 0.80 | 0.59 | 0.65 | 0.80 | 0.52 | 0.47 | 0.57 | 0.78 | 0.45 | 0.52 | 0.60 | 0.69 | 0.77 | 0.61 |
| OSD (2025) | 0.96 | 0.96 | 0.94 | 0.84 | 0.94 | 0.75 | 0.73 | 0.93 | 0.94 | 0.82 | 0.92 | 0.87 | 0.95 | 0.96 | 0.93 | 0.78 | 0.89 | 0.75 | 0.67 | 0.87 | 0.86 | 0.81 | 0.96 | 0.84 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | **0.99** | 0.96 | **0.98** | 0.96 | **0.98** | 0.78 | 0.89 | 0.98 | **0.99** | 0.87 | 0.94 | 0.93 | 0.99 | 0.96 | 0.97 | 0.97 | 0.97 | 0.78 | 0.81 | 0.97 | 0.98 | 0.84 | 0.92 | 0.91 |
| IRM (2019) | **0.99** | 0.96 | **0.98** | 0.96 | **0.98** | 0.77 | 0.88 | 0.98 | **0.99** | 0.86 | 0.94 | 0.93 | 0.94 | 0.84 | 0.93 | 0.90 | 0.90 | 0.69 | 0.72 | 0.93 | 0.90 | 0.81 | 0.87 | 0.85 |
| GroupDRO (2020) | **0.99** | 0.96 | **0.98** | 0.96 | **0.98** | 0.77 | 0.89 | 0.98 | **0.99** | 0.86 | 0.93 | 0.93 | 0.97 | 0.92 | 0.97 | 0.92 | 0.95 | 0.75 | 0.72 | 0.95 | 0.96 | 0.82 | 0.89 | 0.88 |
| TENT (2021) | **0.99** | 0.96 | **0.98** | **0.97** | **0.98** | 0.77 | 0.89 | 0.98 | **0.99** | 0.86 | 0.93 | 0.93 | 0.99 | 0.97 | 0.98 | 0.97 | 0.98 | 0.78 | 0.81 | 0.98 | **0.99** | 0.85 | 0.93 | 0.92 |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.98 | 0.92 | 0.97 | 0.94 | 0.97 | 0.76 | 0.82 | 0.97 | 0.98 | 0.84 | 0.94 | 0.91 | 0.99 | 0.93 | 0.97 | 0.97 | 0.96 | 0.75 | 0.88 | 0.98 | 0.97 | 0.79 | 0.91 | 0.91 |
| MAML (2017) | 0.98 | 0.96 | **0.98** | 0.96 | **0.98** | 0.83 | 0.95 | 0.98 | 0.98 | **0.88** | 0.94 | 0.94 | **1.00** | **0.99** | **0.99** | 0.99 | **0.99** | 0.89 | 0.98 | **0.99** | **0.99** | 0.88 | 0.99 | 0.96 |
| ProtoNet (2017) | **0.99** | **0.97** | **0.98** | 0.96 | **0.98** | 0.81 | 0.93 | 0.98 | **0.99** | 0.86 | 0.92 | 0.94 | **1.00** | **0.99** | 0.98 | 0.99 | 0.98 | 0.87 | 0.96 | **0.99** | **0.99** | 0.87 | 0.99 | 0.95 |
| Reptile (2018) | **0.99** | 0.96 | **0.98** | 0.96 | **0.98** | 0.81 | 0.95 | 0.98 | 0.98 | **0.88** | 0.94 | 0.94 | **1.00** | **0.99** | 0.98 | 0.99 | **0.99** | 0.88 | 0.98 | **0.99** | **0.99** | 0.88 | 0.99 | 0.96 |
| CNAPs (2019) | **0.99** | 0.96 | **0.98** | **0.97** | **0.98** | 0.75 | 0.87 | 0.98 | **0.99** | 0.86 | 0.92 | 0.92 | **1.00** | **0.99** | **0.99** | 0.99 | 0.98 | 0.77 | 0.90 | 0.98 | **0.99** | 0.86 | 0.97 | 0.94 |
| FEAT (2020) | 0.97 | 0.94 | 0.97 | **0.97** | **0.98** | 0.83 | **0.98** | 0.98 | 0.98 | **0.88** | 0.98 | **0.95** | **1.00** | 0.97 | 0.98 | **1.00** | **0.99** | **0.90** | **0.99** | **0.99** | **0.99** | **0.89** | **1.00** | **0.97** |
| ICL-Causal (2022) | **0.99** | 0.95 | **0.98** | 0.96 | **0.98** | 0.77 | 0.89 | **0.99** | **0.99** | 0.86 | 0.92 | 0.93 | **1.00** | 0.98 | 0.98 | 0.99 | 0.98 | 0.79 | 0.85 | 0.98 | **0.99** | 0.85 | 0.94 | 0.93 |
| ICL-GP (2022) | **0.99** | 0.93 | **0.98** | 0.96 | **0.98** | 0.76 | 0.87 | 0.98 | **0.99** | 0.86 | 0.94 | 0.92 | 0.99 | 0.97 | 0.98 | 0.98 | 0.96 | 0.77 | 0.81 | 0.96 | 0.98 | 0.84 | 0.94 | 0.91 |
| TabPFN (2022) | 0.96 | 0.84 | 0.88 | 0.96 | 0.95 | **0.86** | **0.98** | 0.97 | 0.97 | 0.80 | **1.00** | 0.92 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | 0.98 | 0.96 | **0.98** | 0.96 | **0.98** | 0.82 | 0.95 | 0.98 | 0.98 | **0.88** | 0.94 | 0.94 | **1.00** | **0.99** | **0.99** | 0.99 | **0.99** | 0.89 | 0.98 | **0.99** | **0.99** | 0.88 | 0.99 | 0.96 |
| CAML (2024) | 0.98 | 0.91 | 0.96 | 0.94 | 0.96 | 0.74 | 0.82 | 0.97 | 0.97 | 0.84 | 0.93 | 0.90 | 0.99 | 0.97 | 0.97 | 0.99 | 0.97 | 0.75 | 0.78 | 0.98 | 0.98 | 0.82 | 0.91 | 0.90 |
| AFRT (2025) | 0.98 | 0.93 | 0.97 | 0.95 | **0.98** | 0.78 | 0.94 | 0.98 | **0.99** | 0.86 | 0.97 | 0.94 | **1.00** | 0.97 | 0.98 | 0.99 | 0.98 | 0.89 | 0.98 | **0.99** | **0.99** | 0.87 | 0.99 | 0.96 |
| NearestCentroid | 0.96 | 0.84 | 0.92 | **0.97** | 0.94 | 0.84 | **0.98** | 0.97 | 0.95 | 0.83 | 0.99 | 0.92 | 0.96 | 0.84 | 0.92 | 0.97 | 0.94 | 0.84 | 0.98 | 0.97 | 0.95 | 0.83 | 0.99 | 0.92 |

Table 9: *Experiment 1 – Matched support policy – ACE* (lower is better). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.20 | 0.16 | 0.16 | 0.21 | 0.18 | 0.25 | 0.33 | 0.19 | 0.16 | 0.20 | 0.15 | 0.21 | 0.29 | 0.09 | 0.19 | 0.27 | 0.23 | 0.28 | 0.31 | 0.19 | 0.22 | 0.22 | 0.11 | 0.23 |
| UnivFD-FT (2023) | 0.09 | 0.08 | 0.08 | 0.06 | 0.08 | 0.15 | 0.09 | 0.09 | 0.08 | 0.14 | 0.12 | 0.10 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.18 | 0.09 | 0.05 | 0.06 | 0.11 | 0.04 | 0.07 |
| UnivFD-ZS (2023) | 0.11 | 0.05 | 0.09 | 0.08 | 0.08 | 0.20 | 0.18 | 0.09 | 0.10 | 0.08 | 0.07 | 0.10 | 0.08 | 0.04 | 0.05 | 0.07 | 0.05 | 0.20 | 0.22 | 0.05 | 0.06 | 0.13 | 0.04 | 0.10 |
| BSA (2024) | 0.16 | 0.16 | 0.13 | 0.18 | 0.10 | 0.13 | 0.24 | 0.22 | 0.10 | 0.12 | 0.16 | 0.16 | 0.27 | 0.31 | 0.20 | 0.18 | 0.24 | 0.46 | 0.36 | 0.29 | 0.22 | 0.37 | 0.18 | 0.29 |
| AIDE (2025) | 0.07 | 0.10 | 0.09 | 0.14 | 0.10 | 0.22 | 0.31 | 0.15 | 0.07 | 0.18 | 0.08 | 0.15 | 0.04 | 0.10 | 0.11 | 0.16 | 0.11 | 0.23 | 0.34 | 0.18 | 0.07 | 0.24 | 0.04 | 0.17 |
| MLEP (2025) | 0.20 | 0.10 | 0.33 | 0.35 | 0.27 | 0.09 | 0.32 | 0.31 | 0.28 | 0.13 | 0.10 | 0.23 | 0.22 | 0.11 | 0.32 | 0.37 | 0.30 | 0.14 | 0.40 | 0.32 | 0.29 | 0.21 | 0.16 | 0.27 |
| OSD (2025) | 0.05 | 0.08 | 0.07 | 0.14 | 0.09 | 0.25 | 0.24 | 0.08 | 0.07 | 0.20 | 0.07 | 0.14 | 0.08 | 0.07 | 0.10 | 0.22 | 0.15 | 0.23 | 0.36 | 0.16 | 0.18 | 0.23 | 0.05 | 0.19 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | 0.07 | **0.03** | 0.07 | 0.06 | 0.06 | 0.21 | 0.11 | 0.06 | 0.07 | 0.12 | 0.04 | 0.09 | 0.05 | **0.02** | 0.03 | 0.05 | 0.03 | 0.21 | 0.21 | 0.04 | 0.04 | 0.16 | 0.09 | 0.10 |
| IRM (2019) | 0.08 | 0.06 | 0.08 | 0.08 | 0.07 | 0.23 | 0.13 | 0.07 | 0.08 | 0.15 | 0.07 | 0.10 | 0.16 | 0.13 | 0.15 | 0.15 | 0.14 | 0.25 | 0.25 | 0.15 | 0.15 | 0.15 | 0.17 | 0.17 |
| GroupDRO (2020) | 0.07 | 0.05 | 0.07 | 0.07 | 0.06 | 0.23 | 0.13 | 0.06 | 0.07 | 0.16 | 0.07 | 0.10 | 0.17 | 0.15 | 0.17 | 0.14 | 0.15 | 0.24 | 0.22 | 0.15 | 0.16 | 0.15 | 0.14 | 0.17 |
| TENT (2021) | 0.05 | 0.05 | 0.06 | 0.07 | 0.05 | 0.23 | 0.15 | 0.05 | 0.06 | 0.17 | 0.09 | 0.10 | 0.04 | 0.03 | 0.03 | 0.06 | 0.04 | 0.23 | 0.23 | 0.04 | 0.04 | 0.19 | 0.09 | 0.11 |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.13 | 0.08 | 0.12 | 0.11 | 0.12 | 0.19 | 0.15 | 0.12 | 0.13 | 0.08 | 0.08 | 0.12 | 0.19 | 0.18 | 0.16 | 0.20 | 0.17 | 0.15 | 0.22 | 0.18 | 0.18 | 0.14 | 0.16 | 0.17 |
| MAML (2017) | **0.04** | **0.03** | **0.03** | **0.04** | **0.03** | **0.04** | 0.05 | **0.03** | 0.04 | **0.06** | 0.04 | **0.04** | **0.01** | 0.03 | **0.02** | **0.02** | **0.02** | 0.04 | 0.03 | 0.02 | **0.02** | 0.07 | 0.02 | **0.03** |
| ProtoNet (2017) | **0.04** | 0.04 | **0.03** | 0.05 | **0.03** | 0.06 | 0.05 | **0.03** | **0.03** | 0.07 | 0.04 | 0.05 | **0.01** | 0.03 | **0.02** | **0.02** | 0.03 | **0.03** | 0.07 | **0.01** | **0.02** | **0.06** | **0.01** | **0.03** |
| Reptile (2018) | **0.04** | **0.03** | **0.03** | **0.04** | **0.03** | 0.05 | 0.05 | **0.03** | 0.04 | **0.06** | 0.04 | **0.04** | **0.01** | 0.03 | **0.02** | **0.02** | **0.02** | 0.04 | 0.03 | 0.02 | **0.02** | 0.08 | 0.02 | **0.03** |
| CNAPs (2019) | 0.05 | 0.06 | 0.04 | 0.07 | 0.04 | 0.15 | 0.14 | 0.05 | 0.05 | 0.16 | 0.10 | 0.09 | **0.01** | 0.03 | **0.02** | 0.03 | 0.04 | 0.11 | 0.14 | 0.04 | 0.03 | 0.16 | 0.07 | 0.08 |
| FEAT (2020) | 0.05 | 0.09 | 0.04 | 0.07 | 0.04 | 0.14 | 0.06 | 0.04 | 0.04 | 0.14 | 0.06 | 0.07 | **0.01** | 0.05 | 0.04 | **0.02** | 0.04 | 0.09 | **0.02** | 0.02 | **0.02** | 0.15 | **0.01** | 0.05 |
| ICL-Causal (2022) | 0.06 | 0.04 | 0.05 | 0.07 | 0.05 | 0.22 | 0.14 | 0.05 | 0.06 | 0.15 | 0.09 | 0.10 | 0.03 | 0.05 | **0.02** | 0.04 | 0.04 | 0.22 | 0.23 | 0.06 | 0.03 | 0.23 | 0.14 | 0.12 |
| ICL-GP (2022) | 0.08 | 0.05 | 0.06 | 0.09 | 0.06 | 0.22 | 0.14 | 0.06 | 0.07 | 0.13 | 0.06 | 0.10 | 0.03 | 0.06 | 0.05 | 0.05 | 0.06 | 0.22 | 0.24 | 0.07 | 0.05 | 0.22 | 0.09 | 0.12 |
| TabPFN (2022) | 0.19 | 0.16 | 0.17 | 0.17 | 0.18 | 0.12 | 0.15 | 0.19 | 0.16 | 0.12 | 0.17 | 0.16 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | **0.04** | **0.03** | **0.03** | **0.04** | **0.03** | 0.05 | 0.05 | **0.03** | 0.04 | **0.06** | 0.04 | **0.04** | **0.01** | 0.03 | **0.02** | **0.02** | **0.02** | 0.04 | 0.04 | 0.02 | **0.02** | 0.07 | 0.03 | **0.03** |
| CAML (2024) | 0.09 | 0.06 | 0.08 | 0.09 | 0.07 | 0.21 | 0.17 | 0.07 | 0.08 | 0.12 | 0.06 | 0.11 | 0.03 | 0.05 | 0.05 | 0.03 | 0.10 | 0.28 | 0.30 | 0.09 | 0.06 | 0.35 | 0.27 | 0.18 |
| AFRT (2025) | 0.05 | **0.03** | 0.04 | 0.06 | 0.05 | 0.10 | 0.09 | 0.05 | 0.05 | **0.06** | 0.08 | 0.06 | **0.01** | 0.05 | 0.03 | **0.02** | 0.03 | 0.04 | 0.04 | **0.01** | **0.02** | 0.08 | 0.02 | 0.04 |
| NearestCentroid | 0.07 | 0.17 | 0.10 | 0.05 | 0.08 | 0.16 | **0.03** | **0.03** | 0.06 | 0.17 | **0.02** | 0.09 | 0.07 | 0.17 | 0.10 | 0.05 | 0.08 | 0.16 | 0.03 | 0.03 | 0.06 | 0.17 | 0.02 | 0.09 |

Table 10: *Experiment 1 – Matched support policy – FNR@1%FPR* (lower is better). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.99 | 0.90 | 0.96 | 1.00 | 0.98 | 1.00 | 1.00 | 0.94 | 0.96 | 0.96 | 0.91 | 0.97 | 0.99 | 0.87 | 0.95 | 0.99 | 0.97 | 1.00 | 0.97 | 0.88 | 0.96 | 0.97 | 0.76 | 0.95 |
| UnivFD-FT (2023) | 0.26 | 0.72 | 0.62 | 0.34 | 0.34 | **0.52** | 0.33 | 0.29 | 0.24 | 0.80 | 0.14 | **0.44** | 0.09 | 0.54 | 0.62 | 0.23 | 0.51 | 0.52 | 0.34 | 0.43 | 0.49 | 0.86 | 0.39 | 0.51 |
| UnivFD-ZS (2023) | 0.28 | 0.74 | 0.63 | 0.51 | 0.50 | 0.81 | 0.72 | 0.42 | 0.41 | 0.88 | 0.59 | 0.62 | 0.30 | 0.78 | 0.68 | 0.36 | 0.56 | 0.85 | 0.73 | 0.45 | 0.46 | 0.90 | 0.73 | 0.65 |
| BSA (2024) | 0.97 | 0.98 | 0.99 | 1.00 | 0.98 | 0.97 | 1.00 | 0.96 | 1.00 | 0.94 | 0.43 | 0.94 | 0.99 | 0.99 | 1.00 | 0.99 | 1.00 | 1.00 | 0.99 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 |
| AIDE (2025) | 0.79 | 0.84 | 0.85 | 0.84 | 0.89 | 0.93 | 0.95 | 0.82 | 0.88 | 0.95 | 0.85 | 0.89 | 0.68 | 0.85 | 0.87 | 0.61 | 0.88 | 0.93 | 0.97 | 0.92 | 0.87 | 0.95 | 0.90 | 0.88 |
| MLEP (2025) | 0.93 | 0.94 | 0.99 | 1.00 | 0.97 | 0.86 | 1.00 | 0.99 | 0.96 | 0.93 | 0.82 | 0.95 | 0.91 | 0.80 | 0.98 | 0.99 | 0.95 | 0.79 | 1.00 | 0.99 | 0.92 | 0.92 | 0.87 | 0.93 |
| OSD (2025) | 0.43 | **0.38** | 0.55 | 0.88 | 0.54 | 0.69 | 0.94 | 0.62 | 0.52 | **0.75** | 0.67 | 0.66 | 0.45 | 0.43 | 0.59 | 0.70 | 0.70 | 0.80 | 0.99 | 0.77 | 0.68 | 0.79 | 0.50 | 0.72 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | 0.15 | 0.52 | 0.36 | 0.39 | **0.22** | 0.62 | 0.63 | 0.30 | 0.21 | 0.78 | 0.69 | 0.47 | 0.29 | 0.62 | 0.45 | 0.33 | 0.40 | 0.70 | 0.72 | 0.44 | 0.37 | 0.86 | 0.77 | 0.58 |
| IRM (2019) | 0.15 | 0.48 | 0.34 | 0.38 | 0.30 | 0.67 | 0.65 | 0.27 | 0.23 | 0.78 | 0.57 | 0.48 | 0.60 | 0.89 | 0.78 | 0.57 | 0.75 | 0.96 | 0.86 | 0.66 | 0.78 | 0.91 | 0.63 | 0.78 |
| GroupDRO (2020) | 0.16 | 0.42 | 0.30 | 0.36 | 0.26 | 0.66 | 0.62 | 0.26 | 0.19 | 0.76 | 0.65 | 0.46 | 0.48 | 0.78 | 0.47 | 0.77 | 0.66 | 0.74 | 0.86 | 0.61 | 0.54 | 0.88 | 0.86 | 0.72 |
| TENT (2021) | 0.13 | 0.46 | 0.31 | 0.36 | 0.25 | 0.70 | 0.65 | 0.26 | 0.20 | 0.77 | 0.62 | 0.46 | 0.13 | 0.44 | 0.32 | 0.26 | 0.26 | 0.56 | 0.67 | 0.29 | 0.20 | 0.79 | 0.67 | 0.46 |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.29 | 0.69 | 0.53 | 0.53 | 0.37 | 0.68 | 0.71 | 0.39 | 0.37 | 0.84 | 0.61 | 0.57 | 0.17 | 0.67 | 0.55 | 0.29 | 0.51 | 0.71 | 0.59 | 0.32 | 0.41 | 0.89 | 0.60 | 0.56 |
| MAML (2017) | 0.33 | 0.51 | 0.40 | 0.45 | 0.37 | 0.68 | 0.55 | 0.33 | 0.36 | 0.80 | 0.61 | 0.52 | 0.02 | 0.18 | **0.26** | 0.13 | **0.16** | **0.45** | 0.21 | 0.13 | **0.09** | 0.75 | 0.14 | 0.29 |
| ProtoNet (2017) | 0.27 | 0.42 | **0.28** | 0.44 | 0.26 | 0.59 | 0.58 | 0.30 | 0.25 | 0.79 | 0.72 | 0.48 | 0.03 | 0.21 | 0.37 | 0.13 | 0.28 | 0.46 | 0.37 | 0.20 | 0.14 | 0.81 | 0.16 | 0.36 |
| Reptile (2018) | 0.27 | 0.55 | 0.40 | 0.41 | 0.33 | 0.66 | 0.53 | 0.30 | 0.33 | 0.79 | 0.60 | 0.50 | 0.02 | 0.19 | 0.31 | 0.16 | 0.19 | 0.47 | 0.22 | 0.16 | 0.11 | 0.76 | 0.18 | 0.31 |
| CNAPs (2019) | 0.13 | 0.48 | 0.35 | **0.31** | 0.24 | 0.62 | 0.64 | 0.29 | 0.25 | 0.79 | 0.66 | 0.47 | **0.01** | 0.25 | 0.28 | 0.15 | 0.26 | 0.53 | 0.57 | 0.24 | 0.17 | 0.77 | 0.44 | 0.40 |
| FEAT (2020) | 1.00 | 0.69 | 1.00 | 0.59 | 0.72 | 0.83 | 0.48 | 0.85 | 1.00 | 1.00 | 0.85 | 0.82 | 0.02 | 0.35 | 0.39 | **0.07** | 0.18 | 0.48 | **0.11** | **0.09** | **0.09** | 0.76 | **0.08** | **0.28** |
| ICL-Causal (2022) | **0.11** | 0.57 | 0.36 | 0.33 | 0.24 | 0.67 | 0.59 | **0.25** | **0.17** | 0.76 | 0.60 | 0.45 | 0.04 | 0.33 | 0.35 | 0.18 | 0.27 | 0.57 | 0.64 | 0.31 | 0.15 | 0.78 | 0.56 | 0.44 |
| ICL-GP (2022) | 0.14 | 0.66 | 0.44 | 0.36 | 0.31 | 0.69 | 0.62 | 0.29 | 0.24 | 0.78 | 0.55 | 0.49 | 0.09 | 0.35 | 0.35 | 0.24 | 0.41 | 0.62 | 0.69 | 0.45 | 0.32 | 0.82 | 0.48 | 0.51 |
| TabPFN (2022) | 0.36 | 0.85 | 0.73 | 0.33 | 0.31 | 0.54 | **0.23** | 0.29 | 0.25 | 0.83 | **0.01** | 0.45 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | 0.31 | 0.51 | 0.40 | 0.45 | 0.34 | 0.70 | 0.54 | 0.35 | 0.35 | 0.80 | 0.65 | 0.51 | 0.02 | **0.17** | 0.30 | 0.13 | 0.19 | 0.46 | 0.22 | 0.14 | 0.10 | **0.74** | 0.14 | 0.29 |
| CAML (2024) | 0.30 | 0.79 | 0.59 | 0.44 | 0.49 | 0.89 | 0.72 | 0.41 | 0.50 | 0.86 | 0.64 | 0.63 | 0.04 | 0.42 | 0.49 | 0.15 | 0.38 | 0.68 | 0.66 | 0.33 | 0.25 | 0.87 | 0.64 | 0.51 |
| AFRT (2025) | 0.34 | 0.66 | 0.43 | 0.44 | 0.34 | 0.64 | 0.32 | 0.32 | 0.21 | 0.81 | 0.41 | 0.47 | 0.04 | 0.35 | 0.45 | 0.10 | 0.28 | 0.46 | 0.20 | 0.19 | 0.13 | 0.80 | 0.22 | 0.34 |
| NearestCentroid | 1.00 | 1.00 | 1.00 | 0.50 | 0.82 | 1.00 | 0.40 | 0.52 | 0.67 | 1.00 | 0.50 | 0.76 | 1.00 | 1.00 | 1.00 | 0.50 | 0.82 | 1.00 | 0.40 | 0.52 | 0.67 | 1.00 | 0.50 | 0.76 |

Table 11: *Experiment 1 – Disjoint support policy – AUC.* Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.57 | 0.71 | 0.64 | 0.56 | 0.62 | 0.51 | 0.46 | 0.69 | 0.67 | 0.59 | 0.79 | 0.61 | 0.51 | 0.79 | 0.65 | 0.56 | 0.62 | 0.53 | 0.50 | 0.71 | 0.66 | 0.60 | 0.88 | 0.62 |
| UnivFD-FT (2023) | 0.98 | 0.91 | 0.97 | 0.94 | 0.97 | 0.79 | 0.83 | 0.98 | 0.97 | 0.87 | 0.96 | 0.92 | 0.98 | 0.91 | 0.96 | 0.96 | 0.96 | 0.78 | 0.78 | 0.96 | 0.97 | 0.83 | 0.92 | 0.90 |
| UnivFD-ZS (2023) | **0.99** | 0.91 | 0.97 | 0.94 | 0.97 | 0.76 | 0.83 | 0.98 | 0.98 | 0.87 | 0.96 | 0.92 | 0.99 | 0.92 | 0.97 | 0.96 | 0.96 | 0.75 | 0.79 | 0.96 | 0.98 | 0.84 | 0.95 | 0.90 |
| BSA (2024) | 0.66 | 0.66 | 0.68 | 0.67 | 0.74 | 0.71 | 0.60 | 0.62 | 0.76 | 0.72 | **0.98** | 0.70 | 0.63 | 0.54 | 0.72 | 0.78 | 0.72 | 0.38 | 0.58 | 0.62 | 0.74 | 0.47 | 0.66 | 0.61 |
| AIDE (2025) | 0.92 | 0.86 | 0.88 | 0.85 | 0.87 | 0.72 | 0.66 | 0.85 | 0.89 | 0.77 | 0.93 | 0.82 | 0.95 | 0.88 | 0.88 | 0.85 | 0.88 | 0.72 | 0.60 | 0.83 | 0.91 | 0.75 | 0.92 | 0.81 |
| MLEP (2025) | 0.62 | 0.75 | 0.47 | 0.45 | 0.55 | **0.82** | 0.51 | 0.50 | 0.55 | 0.71 | 0.80 | 0.60 | 0.63 | 0.79 | 0.51 | 0.48 | 0.57 | **0.81** | 0.45 | 0.52 | 0.58 | 0.71 | 0.79 | 0.61 |
| OSD (2025) | 0.97 | 0.96 | 0.94 | 0.87 | 0.94 | 0.76 | 0.80 | 0.94 | 0.95 | 0.83 | 0.93 | 0.89 | 0.95 | 0.95 | 0.93 | 0.82 | 0.89 | 0.76 | 0.75 | 0.89 | 0.89 | 0.81 | **0.96** | 0.86 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | **0.99** | 0.96 | **0.99** | **0.97** | **0.99** | 0.80 | **0.92** | **0.99** | **0.99** | **0.89** | 0.95 | **0.94** | 0.99 | 0.96 | 0.98 | **0.97** | **0.98** | 0.80 | **0.87** | **0.98** | **0.99** | **0.87** | 0.93 | **0.93** |
| IRM (2019) | **0.99** | 0.96 | **0.99** | **0.97** | **0.99** | 0.79 | 0.91 | **0.99** | **0.99** | 0.87 | 0.95 | **0.94** | 0.96 | 0.88 | 0.96 | 0.93 | 0.93 | 0.73 | 0.77 | 0.95 | 0.94 | 0.84 | 0.92 | 0.88 |
| GroupDRO (2020) | **0.99** | **0.97** | **0.99** | **0.97** | **0.99** | 0.79 | **0.92** | **0.99** | **0.99** | 0.88 | 0.94 | **0.94** | 0.98 | 0.94 | 0.98 | 0.94 | 0.96 | 0.78 | 0.78 | 0.96 | 0.97 | 0.85 | 0.92 | 0.90 |
| TENT (2021) | **0.99** | **0.97** | **0.99** | **0.97** | **0.99** | 0.79 | **0.92** | **0.99** | **0.99** | 0.87 | 0.94 | **0.94** | **1.00** | **0.97** | **0.99** | **0.97** | **0.98** | 0.80 | 0.86 | **0.98** | **0.99** | 0.86 | 0.94 | **0.93** |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.98 | 0.93 | 0.97 | 0.94 | 0.97 | 0.77 | 0.85 | 0.98 | 0.98 | 0.86 | 0.94 | 0.92 | 0.97 | 0.89 | 0.97 | 0.86 | 0.95 | 0.75 | 0.76 | 0.97 | 0.96 | 0.78 | 0.81 | 0.87 |
| MAML (2017) | 0.98 | 0.96 | 0.98 | 0.96 | 0.98 | 0.80 | 0.89 | 0.98 | 0.98 | 0.86 | 0.92 | 0.93 | 0.98 | 0.89 | 0.96 | 0.91 | 0.96 | 0.78 | 0.81 | 0.96 | 0.97 | 0.83 | 0.91 | 0.90 |
| ProtoNet (2017) | **0.99** | **0.97** | 0.98 | 0.96 | 0.98 | 0.80 | **0.92** | 0.98 | **0.99** | 0.87 | 0.92 | **0.94** | 0.97 | 0.91 | 0.96 | 0.85 | 0.95 | 0.78 | 0.79 | 0.96 | 0.96 | 0.81 | 0.87 | 0.88 |
| Reptile (2018) | 0.98 | 0.96 | 0.98 | 0.96 | 0.98 | 0.80 | 0.90 | 0.98 | 0.98 | 0.86 | 0.92 | 0.93 | 0.97 | 0.90 | 0.95 | 0.86 | 0.95 | 0.77 | 0.79 | 0.95 | 0.96 | 0.84 | 0.89 | 0.89 |
| CNAPs (2019) | **0.99** | **0.97** | 0.98 | **0.97** | 0.98 | 0.80 | 0.90 | 0.98 | **0.99** | 0.88 | 0.94 | **0.94** | 0.97 | 0.93 | 0.97 | 0.90 | 0.96 | 0.79 | 0.81 | 0.96 | 0.96 | 0.84 | 0.87 | 0.90 |
| FEAT (2020) | 0.98 | 0.94 | 0.98 | 0.95 | 0.98 | 0.80 | 0.88 | 0.98 | 0.98 | 0.86 | 0.94 | 0.93 | 0.96 | 0.88 | 0.95 | 0.88 | 0.95 | 0.77 | 0.81 | 0.96 | 0.96 | 0.81 | 0.81 | 0.88 |
| ICL-Causal (2022) | **0.99** | 0.96 | 0.98 | **0.97** | 0.98 | 0.79 | 0.90 | **0.99** | **0.99** | 0.87 | 0.92 | 0.93 | 0.99 | 0.96 | 0.98 | **0.97** | 0.97 | 0.80 | 0.85 | **0.98** | **0.99** | 0.85 | 0.90 | 0.92 |
| ICL-GP (2022) | **0.99** | 0.94 | 0.98 | 0.96 | 0.98 | 0.78 | 0.89 | **0.99** | **0.99** | 0.87 | 0.95 | 0.93 | 0.99 | 0.94 | 0.97 | 0.95 | 0.96 | 0.77 | 0.82 | 0.96 | 0.97 | 0.84 | 0.92 | 0.91 |
| TabPFN (2022) | 0.79 | 0.70 | 0.76 | 0.78 | 0.77 | 0.65 | 0.76 | 0.78 | 0.77 | 0.62 | 0.65 | 0.72 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | 0.98 | 0.96 | 0.98 | 0.96 | 0.98 | 0.80 | 0.90 | 0.98 | 0.98 | 0.86 | 0.92 | 0.93 | 0.98 | 0.89 | 0.96 | 0.90 | 0.96 | 0.78 | 0.82 | 0.96 | 0.97 | 0.84 | 0.92 | 0.90 |
| CAML (2024) | **0.99** | 0.93 | 0.97 | 0.95 | 0.97 | 0.77 | 0.86 | 0.98 | 0.98 | 0.86 | 0.95 | 0.92 | 0.98 | 0.91 | 0.96 | 0.93 | 0.95 | 0.76 | 0.74 | 0.96 | 0.97 | 0.82 | 0.88 | 0.88 |
| AFRT (2025) | 0.98 | 0.94 | 0.97 | 0.94 | 0.97 | 0.78 | 0.88 | 0.97 | 0.98 | 0.86 | 0.94 | 0.92 | 0.96 | 0.87 | 0.93 | 0.86 | 0.95 | 0.76 | 0.79 | 0.95 | 0.95 | 0.81 | 0.83 | 0.87 |
| NearestCentroid | 0.86 | 0.78 | 0.84 | 0.84 | 0.83 | 0.64 | 0.82 | 0.86 | 0.80 | 0.71 | 0.80 | 0.79 | 0.80 | 0.75 | 0.77 | 0.79 | 0.76 | 0.62 | 0.75 | 0.77 | 0.77 | 0.67 | 0.73 | 0.74 |

Table 12: *Experiment 1 – Disjoint support policy – AP.* Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.53 | 0.68 | 0.61 | 0.51 | 0.57 | 0.49 | 0.49 | 0.66 | 0.62 | 0.58 | 0.75 | 0.58 | 0.49 | 0.77 | 0.64 | 0.53 | 0.57 | 0.50 | 0.51 | 0.71 | 0.60 | 0.59 | 0.85 | 0.60 |
| UnivFD-FT (2023) | 0.98 | 0.89 | 0.96 | 0.94 | 0.96 | 0.76 | 0.79 | 0.97 | 0.97 | 0.85 | 0.94 | 0.90 | 0.98 | 0.89 | 0.95 | 0.95 | 0.96 | 0.75 | 0.74 | 0.95 | 0.96 | 0.80 | 0.89 | 0.88 |
| UnivFD-ZS (2023) | **0.99** | 0.90 | 0.97 | 0.94 | 0.97 | 0.74 | 0.79 | 0.97 | 0.98 | 0.84 | 0.94 | 0.90 | **0.99** | 0.91 | 0.95 | 0.96 | 0.96 | 0.73 | 0.74 | 0.96 | 0.97 | 0.81 | 0.93 | 0.89 |
| BSA (2024) | 0.61 | 0.62 | 0.63 | 0.58 | 0.67 | 0.66 | 0.55 | 0.58 | 0.66 | 0.69 | **0.98** | 0.65 | 0.59 | 0.50 | 0.66 | 0.71 | 0.65 | 0.41 | 0.58 | 0.59 | 0.68 | 0.46 | 0.55 | 0.57 |
| AIDE (2025) | 0.90 | 0.84 | 0.85 | 0.82 | 0.84 | 0.69 | 0.62 | 0.82 | 0.86 | 0.73 | 0.90 | 0.79 | 0.94 | 0.86 | 0.85 | 0.83 | 0.85 | 0.70 | 0.59 | 0.79 | 0.88 | 0.71 | 0.89 | 0.78 |
| MLEP (2025) | 0.62 | 0.73 | 0.46 | 0.43 | 0.54 | **0.79** | 0.47 | 0.49 | 0.55 | 0.68 | 0.80 | 0.59 | 0.65 | 0.80 | 0.52 | 0.47 | 0.57 | **0.78** | 0.45 | 0.52 | 0.60 | 0.69 | 0.77 | 0.61 |
| OSD (2025) | 0.96 | **0.96** | 0.94 | 0.84 | 0.94 | 0.75 | 0.73 | 0.93 | 0.94 | 0.82 | 0.92 | 0.87 | 0.95 | 0.96 | 0.93 | 0.78 | 0.89 | 0.75 | 0.67 | 0.87 | 0.86 | 0.81 | **0.96** | 0.84 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | **0.99** | **0.96** | **0.98** | 0.96 | **0.98** | 0.78 | 0.89 | 0.98 | **0.99** | **0.87** | 0.94 | **0.93** | **0.99** | 0.96 | 0.97 | **0.97** | 0.97 | **0.78** | **0.81** | 0.97 | 0.98 | 0.84 | 0.92 | 0.91 |
| IRM (2019) | **0.99** | **0.96** | **0.98** | 0.96 | **0.98** | 0.77 | 0.88 | 0.98 | **0.99** | 0.86 | 0.94 | **0.93** | 0.94 | 0.84 | 0.93 | 0.90 | 0.90 | 0.69 | 0.72 | 0.93 | 0.90 | 0.81 | 0.87 | 0.85 |
| GroupDRO (2020) | **0.99** | **0.96** | **0.98** | 0.96 | **0.98** | 0.77 | 0.89 | 0.98 | **0.99** | 0.86 | 0.93 | **0.93** | 0.97 | 0.92 | 0.97 | 0.92 | 0.95 | 0.75 | 0.72 | 0.95 | 0.96 | 0.82 | 0.89 | 0.88 |
| TENT (2021) | **0.99** | **0.96** | **0.98** | **0.97** | **0.98** | 0.77 | 0.89 | 0.98 | **0.99** | 0.86 | 0.93 | **0.93** | **0.99** | **0.97** | **0.98** | **0.97** | **0.98** | **0.78** | **0.81** | **0.98** | **0.99** | **0.85** | 0.93 | **0.92** |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.98 | 0.92 | 0.96 | 0.94 | 0.97 | 0.75 | 0.82 | 0.97 | 0.97 | 0.85 | 0.93 | 0.91 | 0.96 | 0.87 | 0.97 | 0.80 | 0.95 | 0.73 | 0.74 | 0.96 | 0.94 | 0.79 | 0.81 | 0.86 |
| MAML (2017) | 0.98 | **0.96** | **0.98** | 0.95 | 0.97 | **0.79** | 0.87 | 0.98 | 0.98 | 0.85 | 0.91 | 0.92 | 0.98 | 0.88 | 0.95 | 0.91 | 0.96 | 0.77 | 0.80 | 0.96 | 0.97 | 0.82 | 0.90 | 0.89 |
| ProtoNet (2017) | **0.99** | **0.96** | **0.98** | 0.96 | **0.98** | 0.78 | **0.90** | 0.98 | 0.98 | 0.85 | 0.90 | **0.93** | 0.97 | 0.91 | 0.96 | 0.85 | 0.95 | 0.77 | 0.77 | 0.95 | 0.96 | 0.80 | 0.86 | 0.88 |
| Reptile (2018) | 0.98 | 0.95 | **0.98** | 0.96 | **0.98** | 0.78 | 0.88 | 0.98 | 0.98 | 0.85 | 0.92 | 0.92 | 0.97 | 0.89 | 0.95 | 0.87 | 0.95 | 0.76 | 0.77 | 0.95 | 0.96 | 0.83 | 0.89 | 0.88 |
| CNAPs (2019) | **0.99** | **0.96** | **0.98** | **0.97** | **0.98** | 0.78 | 0.87 | 0.98 | **0.99** | 0.86 | 0.92 | **0.93** | 0.97 | 0.92 | 0.96 | 0.89 | 0.96 | 0.77 | 0.78 | 0.96 | 0.96 | 0.82 | 0.85 | 0.89 |
| FEAT (2020) | 0.98 | 0.93 | 0.97 | 0.95 | 0.97 | 0.78 | 0.84 | 0.97 | 0.97 | 0.84 | 0.92 | 0.91 | 0.96 | 0.88 | 0.95 | 0.88 | 0.95 | 0.76 | 0.79 | 0.95 | 0.96 | 0.80 | 0.79 | 0.87 |
| ICL-Causal (2022) | **0.99** | 0.95 | **0.98** | 0.96 | **0.98** | 0.77 | 0.89 | **0.99** | **0.99** | 0.86 | 0.92 | **0.93** | **0.99** | 0.95 | 0.97 | **0.97** | 0.97 | **0.78** | 0.80 | 0.97 | **0.99** | 0.84 | 0.90 | 0.91 |
| ICL-GP (2022) | **0.99** | 0.93 | **0.98** | 0.96 | **0.98** | 0.76 | 0.87 | 0.98 | **0.99** | 0.86 | 0.94 | 0.92 | 0.98 | 0.93 | 0.96 | 0.95 | 0.96 | 0.76 | 0.77 | 0.96 | 0.97 | 0.83 | 0.92 | 0.89 |
| TabPFN (2022) | 0.79 | 0.70 | 0.75 | 0.76 | 0.77 | 0.65 | 0.75 | 0.79 | 0.77 | 0.61 | 0.65 | 0.72 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | 0.98 | **0.96** | **0.98** | 0.96 | 0.97 | **0.79** | 0.88 | 0.97 | 0.98 | 0.85 | 0.91 | 0.92 | 0.98 | 0.89 | 0.95 | 0.91 | 0.96 | 0.77 | 0.80 | 0.96 | 0.97 | 0.83 | 0.91 | 0.89 |
| CAML (2024) | 0.98 | 0.91 | 0.96 | 0.95 | 0.96 | 0.75 | 0.83 | 0.97 | 0.97 | 0.84 | 0.94 | 0.91 | 0.98 | 0.89 | 0.95 | 0.93 | 0.94 | 0.74 | 0.73 | 0.96 | 0.97 | 0.80 | 0.87 | 0.87 |
| AFRT (2025) | 0.98 | 0.92 | 0.96 | 0.94 | 0.96 | 0.76 | 0.84 | 0.96 | 0.97 | 0.84 | 0.92 | 0.91 | 0.96 | 0.87 | 0.93 | 0.86 | 0.95 | 0.76 | 0.77 | 0.95 | 0.95 | 0.79 | 0.81 | 0.87 |
| NearestCentroid | 0.84 | 0.76 | 0.81 | 0.81 | 0.81 | 0.63 | 0.78 | 0.84 | 0.79 | 0.69 | 0.76 | 0.77 | 0.74 | 0.72 | 0.71 | 0.73 | 0.72 | 0.60 | 0.70 | 0.72 | 0.73 | 0.64 | 0.69 | 0.70 |

Table 13: *Experiment 1 – Disjoint support policy – ACE* (lower is better). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.20 | 0.16 | 0.16 | 0.21 | 0.18 | 0.25 | 0.33 | 0.19 | 0.16 | 0.20 | 0.15 | 0.21 | 0.29 | 0.09 | 0.19 | 0.27 | 0.23 | 0.28 | 0.31 | 0.19 | 0.22 | 0.22 | 0.11 | 0.23 |
| UnivFD-FT (2023) | 0.10 | 0.14 | 0.08 | 0.07 | 0.06 | 0.18 | 0.18 | 0.09 | 0.12 | 0.11 | 0.05 | 0.11 | 0.08 | 0.06 | 0.06 | 0.09 | 0.05 | 0.20 | 0.24 | 0.05 | 0.05 | 0.20 | 0.10 | 0.12 |
| UnivFD-ZS (2023) | 0.11 | 0.05 | 0.09 | 0.08 | 0.08 | 0.20 | 0.18 | 0.09 | 0.10 | **0.08** | 0.07 | 0.10 | 0.08 | 0.04 | 0.05 | 0.07 | 0.05 | 0.20 | 0.22 | 0.05 | 0.06 | 0.13 | **0.04** | **0.10** |
| BSA (2024) | 0.16 | 0.16 | 0.13 | 0.18 | 0.10 | 0.13 | 0.24 | 0.22 | 0.10 | 0.12 | 0.16 | 0.16 | 0.27 | 0.31 | 0.20 | 0.18 | 0.24 | 0.46 | 0.36 | 0.29 | 0.22 | 0.37 | 0.18 | 0.29 |
| AIDE (2025) | 0.07 | 0.10 | 0.09 | 0.14 | 0.10 | 0.22 | 0.31 | 0.15 | 0.07 | 0.18 | 0.08 | 0.15 | 0.04 | 0.10 | 0.11 | 0.16 | 0.11 | 0.23 | 0.34 | 0.18 | 0.07 | 0.24 | **0.04** | 0.17 |
| MLEP (2025) | 0.20 | 0.09 | 0.33 | 0.35 | 0.27 | 0.09 | 0.32 | 0.32 | 0.28 | 0.13 | 0.10 | 0.23 | 0.22 | 0.11 | 0.32 | 0.37 | 0.30 | **0.14** | 0.40 | 0.32 | 0.29 | 0.21 | 0.16 | 0.27 |
| OSD (2025) | **0.05** | 0.08 | 0.07 | 0.14 | 0.09 | 0.25 | 0.24 | 0.08 | 0.07 | 0.20 | 0.07 | 0.14 | 0.08 | 0.07 | 0.10 | 0.22 | 0.15 | 0.23 | 0.36 | 0.16 | 0.18 | 0.23 | 0.05 | 0.19 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | 0.07 | 0.03 | 0.07 | 0.06 | 0.06 | 0.21 | **0.11** | 0.06 | 0.07 | 0.12 | **0.04** | **0.09** | 0.05 | **0.02** | **0.03** | **0.05** | **0.03** | 0.21 | 0.21 | **0.04** | 0.04 | 0.16 | 0.09 | **0.10** |
| IRM (2019) | 0.08 | 0.06 | 0.08 | 0.08 | 0.07 | 0.23 | 0.13 | 0.07 | 0.08 | 0.15 | 0.07 | 0.10 | 0.16 | 0.13 | 0.15 | 0.15 | 0.14 | 0.25 | 0.25 | 0.15 | 0.15 | 0.15 | 0.17 | 0.17 |
| GroupDRO (2020) | 0.07 | 0.05 | 0.07 | 0.07 | 0.06 | 0.23 | 0.13 | 0.06 | 0.07 | 0.16 | 0.07 | 0.10 | 0.17 | 0.15 | 0.17 | 0.14 | 0.15 | 0.24 | 0.22 | 0.15 | 0.16 | 0.15 | 0.14 | 0.17 |
| TENT (2021) | **0.05** | 0.05 | 0.06 | 0.07 | 0.05 | 0.23 | 0.15 | **0.05** | 0.06 | 0.17 | 0.09 | 0.10 | 0.04 | 0.03 | **0.03** | 0.06 | 0.04 | 0.23 | 0.23 | **0.04** | 0.04 | 0.19 | 0.09 | 0.11 |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.13 | 0.08 | 0.12 | 0.11 | 0.11 | 0.21 | 0.16 | 0.12 | 0.13 | **0.08** | 0.08 | 0.12 | 0.15 | 0.12 | 0.16 | 0.16 | 0.14 | 0.22 | 0.19 | 0.16 | 0.15 | **0.11** | 0.11 | 0.15 |
| MAML (2017) | 0.07 | 0.03 | 0.06 | **0.05** | 0.06 | 0.20 | 0.12 | **0.05** | 0.06 | 0.14 | 0.06 | **0.09** | 0.10 | 0.08 | 0.07 | 0.07 | 0.07 | 0.19 | **0.15** | 0.07 | 0.08 | 0.16 | 0.05 | 0.11 |
| ProtoNet (2017) | 0.07 | 0.04 | 0.06 | 0.06 | 0.06 | 0.22 | 0.12 | **0.05** | 0.07 | 0.17 | 0.11 | 0.10 | 0.10 | 0.05 | 0.08 | 0.10 | 0.08 | 0.19 | 0.17 | 0.08 | 0.08 | 0.15 | 0.08 | 0.11 |
| Reptile (2018) | 0.07 | 0.03 | 0.06 | **0.05** | 0.06 | 0.20 | 0.12 | **0.05** | 0.06 | 0.14 | 0.06 | **0.09** | 0.07 | 0.08 | 0.05 | 0.11 | 0.06 | 0.19 | 0.18 | 0.05 | 0.07 | 0.17 | 0.07 | 0.11 |
| CNAPs (2019) | **0.05** | 0.05 | **0.04** | 0.07 | **0.04** | 0.23 | 0.18 | **0.05** | **0.05** | 0.18 | 0.11 | 0.11 | 0.05 | 0.09 | 0.05 | 0.12 | 0.05 | 0.22 | 0.22 | 0.06 | 0.05 | 0.20 | 0.15 | 0.12 |
| FEAT (2020) | 0.06 | 0.12 | 0.05 | 0.09 | 0.06 | 0.23 | 0.21 | **0.05** | 0.06 | 0.21 | 0.10 | 0.12 | 0.08 | 0.16 | 0.08 | 0.15 | 0.09 | 0.25 | 0.23 | 0.08 | 0.07 | 0.26 | 0.24 | 0.16 |
| ICL-Causal (2022) | 0.06 | 0.04 | 0.05 | 0.07 | 0.05 | 0.22 | 0.14 | **0.05** | 0.06 | 0.15 | 0.09 | 0.10 | **0.03** | 0.10 | 0.04 | 0.09 | 0.05 | 0.24 | 0.26 | 0.06 | **0.03** | 0.24 | 0.19 | 0.14 |
| ICL-GP (2022) | 0.07 | 0.05 | 0.06 | 0.08 | 0.06 | 0.22 | 0.14 | 0.06 | 0.07 | 0.13 | 0.06 | **0.09** | 0.04 | 0.12 | 0.04 | 0.11 | 0.06 | 0.25 | 0.28 | 0.07 | 0.04 | 0.24 | 0.14 | 0.14 |
| TabPFN (2022) | 0.12 | 0.07 | 0.11 | 0.13 | 0.11 | **0.08** | **0.11** | 0.12 | 0.10 | 0.09 | 0.07 | 0.10 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | 0.07 | **0.02** | 0.06 | **0.05** | 0.05 | 0.20 | 0.12 | 0.06 | 0.06 | 0.14 | 0.06 | **0.09** | 0.10 | 0.07 | 0.07 | 0.07 | 0.08 | 0.19 | **0.15** | 0.07 | 0.08 | 0.15 | 0.05 | **0.10** |
| CAML (2024) | 0.08 | 0.06 | 0.08 | 0.09 | 0.07 | 0.23 | 0.16 | 0.07 | 0.08 | 0.13 | 0.06 | 0.11 | **0.03** | 0.20 | 0.07 | 0.11 | 0.08 | 0.27 | 0.28 | 0.07 | 0.04 | 0.25 | 0.21 | 0.16 |
| AFRT (2025) | 0.07 | 0.04 | 0.07 | 0.06 | 0.06 | 0.19 | 0.14 | 0.06 | 0.08 | 0.12 | 0.05 | **0.09** | 0.08 | 0.10 | 0.05 | 0.10 | 0.06 | 0.17 | 0.18 | 0.06 | 0.06 | 0.17 | 0.13 | 0.11 |
| NearestCentroid | 0.17 | 0.25 | 0.19 | 0.21 | 0.21 | 0.37 | 0.21 | 0.18 | 0.22 | 0.32 | 0.25 | 0.24 | 0.24 | 0.28 | 0.26 | 0.25 | 0.28 | 0.38 | 0.27 | 0.25 | 0.26 | 0.34 | 0.28 | 0.29 |

Table 14: *Experiment 1 – Disjoint support policy – FNR@1%FPR* (lower is better). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.99 | 0.90 | 0.96 | 1.00 | 0.98 | 1.00 | 1.00 | 0.94 | 0.96 | 0.96 | 0.91 | 0.97 | 0.99 | 0.87 | 0.95 | 0.99 | 0.97 | 1.00 | 0.97 | 0.88 | 0.96 | 0.97 | 0.76 | 0.95 |
| UnivFD-FT (2023) | 0.31 | 0.74 | 0.59 | 0.49 | 0.52 | 0.84 | 0.72 | 0.43 | 0.48 | 0.87 | 0.54 | 0.63 | 0.32 | 0.81 | 0.68 | 0.37 | 0.57 | 0.89 | 0.77 | 0.52 | 0.58 | 0.92 | 0.81 | 0.69 |
| UnivFD-ZS (2023) | 0.28 | 0.74 | 0.63 | 0.51 | 0.50 | 0.81 | 0.72 | 0.42 | 0.41 | 0.88 | 0.59 | 0.62 | 0.30 | 0.78 | 0.68 | 0.36 | 0.56 | 0.85 | 0.73 | 0.45 | 0.46 | 0.90 | 0.73 | 0.65 |
| BSA (2024) | 0.97 | 0.98 | 0.99 | 1.00 | 0.98 | 0.97 | 1.00 | 0.97 | 1.00 | 0.94 | **0.43** | 0.95 | 1.00 | 0.99 | 1.00 | 0.99 | 1.00 | 1.00 | 0.99 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 |
| AIDE (2025) | 0.79 | 0.84 | 0.85 | 0.84 | 0.89 | 0.93 | 0.95 | 0.82 | 0.88 | 0.95 | 0.85 | 0.89 | 0.68 | 0.85 | 0.87 | 0.61 | 0.88 | 0.93 | 0.97 | 0.92 | 0.87 | 0.95 | 0.90 | 0.88 |
| MLEP (2025) | 0.93 | 0.93 | 0.99 | 1.00 | 0.97 | 0.84 | 1.00 | 0.98 | 0.96 | 0.94 | 0.82 | 0.95 | 0.91 | 0.80 | 0.98 | 0.99 | 0.95 | 0.79 | 1.00 | 0.99 | 0.92 | 0.92 | 0.87 | 0.93 |
| OSD (2025) | 0.43 | **0.38** | 0.55 | 0.88 | 0.54 | 0.69 | 0.94 | 0.62 | 0.52 | **0.75** | 0.67 | 0.66 | 0.45 | **0.43** | 0.59 | 0.70 | 0.70 | 0.80 | 0.99 | 0.77 | 0.68 | **0.79** | **0.50** | 0.72 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | 0.15 | 0.52 | 0.36 | 0.39 | **0.22** | **0.62** | 0.63 | 0.30 | 0.21 | 0.78 | 0.69 | 0.47 | 0.29 | 0.62 | 0.45 | 0.33 | 0.40 | 0.70 | 0.72 | 0.44 | 0.37 | 0.86 | 0.77 | 0.58 |
| IRM (2019) | 0.15 | 0.48 | 0.34 | 0.38 | 0.30 | 0.67 | 0.65 | 0.27 | 0.23 | 0.78 | 0.57 | 0.48 | 0.60 | 0.89 | 0.78 | 0.57 | 0.75 | 0.96 | 0.86 | 0.66 | 0.78 | 0.91 | 0.63 | 0.78 |
| GroupDRO (2020) | 0.16 | 0.42 | **0.30** | 0.36 | 0.26 | 0.66 | 0.62 | **0.26** | 0.19 | 0.76 | 0.65 | 0.46 | 0.48 | 0.78 | 0.47 | 0.77 | 0.66 | 0.74 | 0.86 | 0.61 | 0.54 | 0.88 | 0.86 | 0.72 |
| TENT (2021) | 0.13 | 0.46 | 0.31 | 0.36 | 0.25 | 0.70 | 0.65 | **0.26** | 0.20 | 0.77 | 0.62 | 0.46 | **0.13** | 0.44 | **0.32** | **0.26** | **0.26** | **0.56** | **0.67** | **0.29** | **0.20** | **0.79** | 0.67 | **0.46** |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.38 | 0.62 | 0.56 | 0.54 | 0.44 | 0.70 | 0.73 | 0.46 | 0.42 | 0.81 | 0.70 | 0.60 | 0.60 | 0.82 | 0.60 | 0.93 | 0.59 | 0.88 | 0.90 | 0.50 | 0.65 | 0.86 | 0.89 | 0.75 |
| MAML (2017) | 0.42 | 0.54 | 0.45 | 0.55 | 0.45 | 0.73 | 0.72 | 0.41 | 0.44 | 0.80 | 0.68 | 0.59 | 0.26 | 0.71 | 0.53 | 0.53 | 0.39 | 0.67 | 0.75 | 0.38 | 0.32 | 0.82 | 0.68 | 0.57 |
| ProtoNet (2017) | 0.28 | 0.41 | 0.41 | 0.45 | 0.31 | 0.63 | 0.65 | 0.36 | 0.34 | 0.80 | 0.78 | 0.52 | 0.41 | 0.64 | 0.44 | 0.77 | 0.45 | 0.67 | 0.78 | 0.47 | 0.46 | 0.83 | 0.80 | 0.63 |
| Reptile (2018) | 0.42 | 0.56 | 0.42 | 0.52 | 0.43 | 0.71 | 0.67 | 0.42 | 0.48 | 0.80 | 0.66 | 0.58 | 0.33 | 0.72 | 0.54 | 0.71 | 0.45 | 0.69 | 0.75 | 0.48 | 0.39 | 0.84 | 0.73 | 0.62 |
| CNAPs (2019) | 0.23 | 0.49 | 0.34 | **0.31** | 0.28 | 0.66 | 0.63 | 0.29 | 0.25 | 0.80 | 0.67 | 0.49 | 0.40 | 0.65 | 0.52 | 0.71 | 0.51 | 0.69 | 0.82 | 0.48 | 0.48 | 0.85 | 0.83 | 0.65 |
| FEAT (2020) | 1.00 | 0.91 | 1.00 | 0.75 | 0.71 | 0.89 | 1.00 | 0.80 | 0.93 | 0.97 | 0.88 | 0.89 | 0.45 | 0.80 | 0.51 | 0.70 | 0.50 | 0.82 | 0.75 | 0.51 | 0.46 | 0.86 | 0.90 | 0.67 |
| ICL-Causal (2022) | **0.12** | 0.57 | 0.36 | 0.33 | 0.25 | 0.66 | **0.58** | **0.26** | **0.17** | 0.76 | 0.60 | **0.45** | 0.17 | 0.57 | 0.44 | 0.29 | 0.35 | 0.64 | 0.73 | 0.36 | 0.24 | 0.81 | 0.70 | 0.52 |
| ICL-GP (2022) | 0.14 | 0.64 | 0.45 | 0.36 | 0.31 | 0.68 | 0.62 | 0.29 | 0.23 | 0.77 | 0.55 | 0.49 | 0.19 | 0.69 | 0.55 | 0.36 | 0.47 | 0.72 | 0.79 | 0.47 | 0.44 | 0.85 | 0.58 | 0.60 |
| TabPFN (2022) | 0.87 | 0.93 | 0.93 | 0.91 | 0.90 | 0.94 | 0.93 | 0.88 | 0.88 | 0.97 | 0.95 | 0.92 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | 0.45 | 0.56 | 0.46 | 0.53 | 0.44 | 0.70 | 0.71 | 0.44 | 0.45 | 0.79 | 0.71 | 0.59 | 0.29 | 0.73 | 0.55 | 0.56 | 0.39 | 0.69 | 0.74 | 0.41 | 0.37 | 0.82 | 0.69 | 0.59 |
| CAML (2024) | 0.33 | 0.74 | 0.58 | 0.40 | 0.50 | 0.85 | 0.71 | 0.44 | 0.44 | 0.86 | 0.64 | 0.62 | 0.39 | 0.90 | 0.66 | 0.48 | 0.61 | 0.77 | 0.83 | 0.52 | 0.48 | 0.90 | 0.78 | 0.69 |
| AFRT (2025) | 0.33 | 0.68 | 0.46 | 0.54 | 0.49 | 0.77 | 0.71 | 0.49 | 0.46 | 0.83 | 0.73 | 0.62 | 0.42 | 0.71 | 0.62 | 0.79 | 0.50 | 0.76 | 0.78 | 0.50 | 0.43 | 0.89 | 0.87 | 0.67 |
| NearestCentroid | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Table 15: *Experiment 1 – Chameleon – ACC.* Detectors trained under the implicit construction, matched support policy. The last two columns are the differences from UnivFD-FT and from NearestCentroid.

| Model | Wild (unseen family) | Average | Δ vs UnivFD-FT (2023) | Δ vs NearestCentroid |
|---|---|---|---|---|
| | Chameleon | | | |
| *Context-free* | | | | |
| PatchCraft (2023) | 0.58 | 0.58 | -0.17 | -0.27 |
| UnivFD-FT (2023) | 0.75 | 0.75 | +0.00 | -0.10 |
| UnivFD-ZS (2023) | 0.63 | 0.63 | -0.12 | -0.22 |
| BSA (2024) | 0.51 | 0.51 | -0.24 | -0.34 |
| AIDE (2025) | 0.62 | 0.62 | -0.13 | -0.23 |
| MLEP (2025) | 0.54 | 0.54 | -0.21 | -0.31 |
| OSD (2025) | 0.66 | 0.66 | -0.09 | -0.19 |
| *Domain-aware* | | | | |
| CORAL (2016) | 0.61 | 0.61 | -0.14 | -0.24 |
| IRM (2019) | 0.61 | 0.61 | -0.14 | -0.24 |
| GroupDRO (2020) | 0.62 | 0.62 | -0.13 | -0.23 |
| TENT (2021) | 0.62 | 0.62 | -0.13 | -0.23 |
| *Context-aware* | | | | |
| MatchNet (2016) | 0.63 | 0.63 | -0.12 | -0.22 |
| MAML (2017) | 0.73 | 0.73 | -0.02 | -0.12 |
| ProtoNet (2017) | 0.67 | 0.67 | -0.08 | -0.18 |
| Reptile (2018) | 0.73 | 0.73 | -0.02 | -0.12 |
| CNAPs (2019) | 0.60 | 0.60 | -0.15 | -0.25 |
| FEAT (2020) | 0.80 | 0.80 | +0.05 | -0.05 |
| ICL-Causal (2022) | 0.61 | 0.61 | -0.14 | -0.24 |
| ICL-GP (2022) | 0.65 | 0.65 | -0.10 | -0.20 |
| TabPFN (2022) | **0.86** | **0.86** | **+0.11** | **+0.01** |
| GAP (2023) | 0.73 | 0.73 | -0.02 | -0.12 |
| CAML (2024) | 0.63 | 0.63 | -0.12 | -0.22 |
| AFRT (2025) | 0.76 | 0.76 | +0.01 | -0.09 |
| NearestCentroid | 0.85 | 0.85 | +0.10 | +0.00 |

Table 16: *Experiment 1 – Sibling support policy – AUC.* Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.57 | 0.71 | 0.64 | 0.56 | 0.62 | 0.51 | 0.46 | 0.69 | 0.67 | 0.59 | 0.79 | 0.61 | 0.51 | 0.79 | 0.65 | 0.56 | 0.62 | 0.53 | 0.50 | 0.71 | 0.66 | 0.60 | 0.88 | 0.62 |
| UnivFD-FT (2023) | 0.98 | 0.90 | 0.97 | 0.94 | 0.97 | 0.76 | 0.89 | 0.98 | 0.98 | 0.88 | 0.99 | 0.93 | 0.98 | 0.90 | 0.96 | 0.96 | 0.96 | 0.75 | 0.83 | 0.97 | 0.98 | 0.85 | 0.98 | 0.91 |
| UnivFD-ZS (2023) | **0.99** | 0.91 | 0.97 | 0.94 | 0.97 | 0.76 | 0.83 | 0.98 | 0.98 | 0.87 | 0.96 | 0.92 | 0.99 | 0.92 | 0.97 | 0.96 | 0.96 | 0.75 | 0.79 | 0.96 | 0.98 | 0.84 | 0.95 | 0.90 |
| BSA (2024) | 0.66 | 0.66 | 0.68 | 0.67 | 0.74 | 0.71 | 0.60 | 0.62 | 0.76 | 0.72 | 0.98 | 0.70 | 0.63 | 0.54 | 0.72 | 0.78 | 0.72 | 0.38 | 0.58 | 0.62 | 0.74 | 0.47 | 0.66 | 0.61 |
| AIDE (2025) | 0.92 | 0.86 | 0.88 | 0.85 | 0.87 | 0.72 | 0.66 | 0.85 | 0.89 | 0.77 | 0.93 | 0.82 | 0.95 | 0.88 | 0.88 | 0.85 | 0.88 | 0.72 | 0.60 | 0.83 | 0.91 | 0.75 | 0.92 | 0.81 |
| MLEP (2025) | 0.62 | 0.75 | 0.47 | 0.45 | 0.55 | **0.82** | 0.51 | 0.50 | 0.55 | 0.71 | 0.80 | 0.60 | 0.63 | 0.79 | 0.51 | 0.48 | 0.57 | **0.81** | 0.45 | 0.52 | 0.58 | 0.71 | 0.79 | 0.61 |
| OSD (2025) | 0.97 | 0.96 | 0.94 | 0.87 | 0.94 | 0.76 | 0.80 | 0.94 | 0.95 | 0.83 | 0.93 | 0.89 | 0.95 | 0.95 | 0.93 | 0.82 | 0.89 | 0.76 | 0.75 | 0.89 | 0.89 | 0.81 | 0.96 | 0.86 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | **0.99** | 0.96 | **0.99** | **0.97** | **0.99** | 0.80 | **0.92** | **0.99** | **0.99** | **0.89** | 0.95 | **0.94** | 0.99 | 0.96 | 0.98 | 0.97 | **0.98** | 0.80 | 0.87 | **0.98** | **0.99** | **0.87** | 0.93 | **0.93** |
| IRM (2019) | **0.99** | 0.96 | **0.99** | **0.97** | **0.99** | 0.79 | 0.91 | **0.99** | **0.99** | 0.87 | 0.95 | **0.94** | 0.96 | 0.88 | 0.96 | 0.93 | 0.93 | 0.73 | 0.77 | 0.95 | 0.94 | 0.84 | 0.92 | 0.88 |
| GroupDRO (2020) | **0.99** | **0.97** | **0.99** | **0.97** | **0.99** | 0.79 | **0.92** | **0.99** | **0.99** | 0.88 | 0.94 | **0.94** | 0.98 | 0.94 | 0.98 | 0.94 | 0.96 | 0.78 | 0.78 | 0.96 | 0.97 | 0.85 | 0.92 | 0.90 |
| TENT (2021) | **0.99** | **0.97** | **0.99** | **0.97** | **0.99** | 0.79 | **0.92** | **0.99** | **0.99** | 0.87 | 0.94 | **0.94** | **1.00** | **0.97** | **0.99** | 0.97 | **0.98** | 0.80 | 0.86 | **0.98** | **0.99** | 0.86 | 0.94 | **0.93** |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.98 | 0.92 | 0.98 | 0.94 | 0.97 | 0.77 | 0.84 | 0.98 | 0.98 | 0.85 | 0.94 | 0.92 | 0.98 | 0.87 | 0.97 | 0.91 | 0.97 | 0.76 | 0.83 | **0.98** | 0.95 | 0.77 | 0.92 | 0.89 |
| MAML (2017) | **0.99** | 0.95 | 0.98 | 0.96 | 0.98 | 0.80 | 0.91 | 0.98 | 0.98 | 0.86 | 0.94 | 0.93 | **1.00** | 0.90 | **0.99** | 0.97 | **0.98** | 0.80 | **0.90** | **0.98** | **0.99** | 0.83 | **0.99** | **0.93** |
| ProtoNet (2017) | **0.99** | 0.95 | 0.98 | 0.96 | 0.98 | 0.80 | **0.92** | 0.98 | **0.99** | 0.86 | 0.94 | **0.94** | 0.99 | 0.90 | 0.98 | 0.95 | 0.97 | 0.80 | 0.87 | **0.98** | 0.97 | 0.84 | **0.99** | 0.92 |
| Reptile (2018) | **0.99** | 0.95 | 0.98 | 0.96 | 0.98 | 0.80 | 0.91 | 0.98 | **0.99** | 0.86 | 0.94 | 0.93 | 0.99 | 0.91 | 0.98 | 0.95 | **0.98** | 0.79 | **0.90** | **0.98** | 0.98 | 0.84 | **0.99** | **0.93** |
| CNAPs (2019) | **0.99** | 0.95 | 0.98 | 0.96 | **0.99** | 0.80 | 0.90 | 0.98 | **0.99** | 0.87 | 0.94 | **0.94** | 0.99 | 0.88 | **0.99** | 0.95 | **0.98** | 0.79 | 0.84 | **0.98** | 0.96 | 0.85 | 0.97 | 0.92 |
| FEAT (2020) | 0.98 | 0.93 | 0.98 | 0.96 | 0.98 | 0.80 | 0.90 | 0.98 | 0.98 | 0.87 | 0.98 | **0.94** | 0.99 | 0.85 | 0.98 | 0.95 | **0.98** | 0.78 | **0.90** | **0.98** | 0.98 | 0.83 | **0.99** | 0.92 |
| ICL-Causal (2022) | **0.99** | 0.96 | 0.98 | 0.96 | 0.98 | 0.79 | 0.90 | **0.99** | **0.99** | 0.87 | 0.92 | 0.93 | **1.00** | 0.95 | 0.98 | **0.98** | **0.98** | 0.78 | 0.87 | **0.98** | **0.99** | 0.86 | 0.94 | **0.93** |
| ICL-GP (2022) | **0.99** | 0.94 | 0.98 | 0.96 | 0.98 | 0.78 | 0.89 | **0.99** | **0.99** | 0.87 | 0.95 | 0.93 | 0.99 | 0.93 | 0.97 | 0.97 | 0.96 | 0.78 | 0.86 | 0.97 | 0.98 | 0.85 | 0.95 | 0.92 |
| TabPFN (2022) | 0.82 | 0.66 | 0.85 | 0.80 | 0.86 | 0.62 | 0.81 | 0.87 | 0.79 | 0.69 | **1.00** | 0.79 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | **0.99** | 0.95 | 0.98 | 0.96 | 0.98 | 0.80 | 0.91 | 0.98 | **0.99** | 0.86 | 0.94 | 0.93 | **1.00** | 0.89 | **0.99** | 0.97 | **0.98** | 0.80 | **0.90** | **0.98** | 0.98 | 0.83 | **0.99** | **0.93** |
| CAML (2024) | 0.98 | 0.93 | 0.98 | 0.95 | 0.97 | 0.77 | 0.86 | 0.98 | 0.98 | 0.86 | 0.95 | 0.92 | 0.99 | 0.88 | 0.97 | 0.96 | 0.97 | 0.77 | 0.80 | **0.98** | 0.98 | 0.84 | 0.91 | 0.90 |
| AFRT (2025) | **0.99** | 0.90 | 0.98 | 0.94 | 0.98 | 0.77 | 0.88 | 0.98 | 0.98 | 0.86 | 0.97 | 0.92 | 0.99 | 0.87 | 0.98 | 0.94 | **0.98** | 0.78 | 0.88 | **0.98** | 0.98 | 0.83 | **0.99** | 0.92 |
| NearestCentroid | 0.89 | 0.69 | 0.90 | 0.85 | 0.89 | 0.63 | 0.83 | 0.89 | 0.75 | 0.78 | 0.99 | 0.83 | 0.85 | 0.69 | 0.90 | 0.88 | 0.86 | 0.65 | 0.87 | 0.89 | 0.86 | 0.73 | **0.99** | 0.83 |

Table 17: *Experiment 1 – Sibling support policy – AP.* Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.53 | 0.68 | 0.61 | 0.51 | 0.57 | 0.49 | 0.49 | 0.66 | 0.62 | 0.58 | 0.75 | 0.58 | 0.49 | 0.77 | 0.64 | 0.53 | 0.57 | 0.50 | 0.51 | 0.71 | 0.60 | 0.59 | 0.85 | 0.60 |
| UnivFD-FT (2023) | 0.98 | 0.89 | 0.97 | 0.94 | 0.96 | 0.74 | 0.85 | 0.97 | 0.97 | 0.85 | 0.99 | 0.91 | 0.98 | 0.88 | 0.94 | 0.95 | 0.95 | 0.73 | 0.79 | 0.96 | 0.97 | 0.82 | 0.97 | 0.89 |
| UnivFD-ZS (2023) | **0.99** | 0.90 | 0.97 | 0.94 | 0.97 | 0.74 | 0.79 | 0.97 | 0.98 | 0.84 | 0.94 | 0.90 | 0.99 | 0.91 | 0.95 | 0.96 | 0.96 | 0.73 | 0.74 | 0.96 | 0.97 | 0.81 | 0.93 | 0.89 |
| BSA (2024) | 0.61 | 0.62 | 0.63 | 0.58 | 0.67 | 0.66 | 0.55 | 0.58 | 0.66 | 0.69 | 0.98 | 0.65 | 0.59 | 0.50 | 0.66 | 0.71 | 0.65 | 0.41 | 0.58 | 0.59 | 0.68 | 0.46 | 0.55 | 0.57 |
| AIDE (2025) | 0.90 | 0.84 | 0.85 | 0.82 | 0.84 | 0.69 | 0.62 | 0.82 | 0.86 | 0.73 | 0.90 | 0.79 | 0.94 | 0.86 | 0.85 | 0.83 | 0.85 | 0.70 | 0.59 | 0.79 | 0.88 | 0.71 | 0.89 | 0.78 |
| MLEP (2025) | 0.62 | 0.73 | 0.46 | 0.43 | 0.54 | **0.79** | 0.47 | 0.49 | 0.55 | 0.68 | 0.80 | 0.59 | 0.65 | 0.80 | 0.52 | 0.47 | 0.57 | 0.78 | 0.45 | 0.52 | 0.60 | 0.69 | 0.77 | 0.61 |
| OSD (2025) | 0.96 | **0.96** | 0.94 | 0.84 | 0.94 | 0.75 | 0.73 | 0.93 | 0.94 | 0.82 | 0.92 | 0.87 | 0.95 | 0.96 | 0.93 | 0.78 | 0.89 | 0.75 | 0.67 | 0.87 | 0.86 | 0.81 | 0.96 | 0.84 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | **0.99** | **0.96** | **0.98** | 0.96 | **0.98** | 0.78 | 0.89 | 0.98 | **0.99** | **0.87** | 0.94 | **0.93** | 0.99 | 0.96 | 0.97 | 0.97 | 0.97 | 0.78 | 0.81 | 0.97 | 0.98 | 0.84 | 0.92 | 0.91 |
| IRM (2019) | **0.99** | **0.96** | **0.98** | 0.96 | **0.98** | 0.77 | 0.88 | 0.98 | **0.99** | 0.86 | 0.94 | **0.93** | 0.94 | 0.84 | 0.93 | 0.90 | 0.90 | 0.69 | 0.72 | 0.93 | 0.90 | 0.81 | 0.87 | 0.85 |
| GroupDRO (2020) | **0.99** | **0.96** | **0.98** | 0.96 | **0.98** | 0.77 | 0.89 | 0.98 | **0.99** | 0.86 | 0.93 | **0.93** | 0.97 | 0.92 | 0.97 | 0.92 | 0.95 | 0.75 | 0.72 | 0.95 | 0.96 | 0.82 | 0.89 | 0.88 |
| TENT (2021) | **0.99** | **0.96** | **0.98** | **0.97** | **0.98** | 0.77 | 0.89 | 0.98 | **0.99** | 0.86 | 0.93 | **0.93** | 0.99 | **0.97** | **0.98** | 0.97 | **0.98** | 0.78 | 0.81 | **0.98** | **0.99** | **0.85** | 0.93 | **0.92** |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.98 | 0.91 | 0.97 | 0.93 | 0.97 | 0.75 | 0.82 | 0.97 | 0.98 | 0.84 | 0.93 | 0.91 | 0.97 | 0.88 | 0.96 | 0.88 | 0.96 | 0.74 | 0.81 | 0.97 | 0.92 | 0.79 | 0.91 | 0.88 |
| MAML (2017) | 0.98 | 0.95 | **0.98** | 0.96 | **0.98** | **0.79** | 0.89 | 0.98 | 0.98 | 0.85 | 0.93 | **0.93** | **1.00** | 0.91 | **0.98** | **0.98** | **0.98** | **0.79** | 0.90 | **0.98** | 0.98 | 0.81 | **0.99** | **0.92** |
| ProtoNet (2017) | **0.99** | 0.95 | **0.98** | 0.96 | **0.98** | 0.78 | **0.90** | 0.98 | **0.99** | 0.85 | 0.92 | **0.93** | 0.99 | 0.92 | **0.98** | 0.96 | **0.98** | 0.78 | 0.86 | 0.97 | 0.97 | 0.83 | **0.99** | **0.92** |
| Reptile (2018) | **0.99** | 0.95 | **0.98** | 0.96 | **0.98** | 0.78 | 0.89 | 0.98 | 0.98 | 0.85 | 0.93 | **0.93** | 0.99 | 0.92 | **0.98** | 0.96 | **0.98** | 0.78 | 0.90 | **0.98** | 0.98 | 0.82 | **0.99** | **0.92** |
| CNAPs (2019) | **0.99** | 0.95 | **0.98** | 0.96 | **0.98** | 0.78 | 0.87 | 0.98 | **0.99** | 0.86 | 0.92 | **0.93** | 0.99 | 0.86 | **0.98** | 0.95 | **0.98** | 0.77 | 0.82 | 0.97 | 0.95 | 0.83 | 0.97 | 0.91 |
| FEAT (2020) | 0.97 | 0.92 | 0.96 | 0.95 | 0.97 | 0.78 | 0.89 | 0.97 | 0.97 | 0.85 | 0.97 | 0.92 | 0.99 | 0.86 | **0.98** | 0.96 | **0.98** | 0.76 | **0.91** | **0.98** | 0.98 | 0.81 | **0.99** | **0.92** |
| ICL-Causal (2022) | **0.99** | 0.95 | **0.98** | 0.96 | **0.98** | 0.77 | 0.89 | **0.99** | **0.99** | 0.86 | 0.92 | **0.93** | 0.99 | 0.94 | **0.98** | **0.98** | **0.98** | 0.76 | 0.82 | **0.98** | **0.99** | 0.84 | 0.94 | **0.92** |
| ICL-GP (2022) | **0.99** | 0.93 | **0.98** | 0.96 | **0.98** | 0.76 | 0.87 | 0.98 | **0.99** | 0.86 | 0.94 | 0.92 | 0.99 | 0.92 | 0.96 | 0.97 | 0.96 | 0.76 | 0.81 | 0.96 | 0.97 | 0.83 | 0.94 | 0.90 |
| TabPFN (2022) | 0.84 | 0.67 | 0.85 | 0.83 | 0.87 | 0.65 | 0.84 | 0.88 | 0.83 | 0.68 | **1.00** | 0.80 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | **0.99** | 0.95 | **0.98** | 0.96 | **0.98** | 0.78 | 0.89 | 0.98 | 0.98 | 0.85 | 0.93 | **0.93** | 0.99 | 0.91 | **0.98** | **0.98** | **0.98** | **0.79** | 0.90 | **0.98** | 0.98 | 0.82 | **0.99** | **0.92** |
| CAML (2024) | 0.98 | 0.91 | 0.97 | 0.94 | 0.96 | 0.75 | 0.83 | 0.97 | 0.97 | 0.84 | 0.94 | 0.91 | 0.98 | 0.86 | 0.94 | 0.96 | 0.97 | 0.76 | 0.79 | 0.97 | 0.98 | 0.82 | 0.91 | 0.89 |
| AFRT (2025) | **0.99** | 0.89 | **0.98** | 0.94 | 0.97 | 0.75 | 0.86 | 0.98 | 0.98 | 0.84 | 0.96 | 0.91 | 0.99 | 0.88 | 0.97 | 0.95 | 0.97 | 0.77 | 0.89 | 0.97 | 0.98 | 0.81 | 0.98 | **0.92** |
| NearestCentroid | 0.87 | 0.69 | 0.88 | 0.87 | 0.88 | 0.64 | 0.85 | 0.90 | 0.77 | 0.74 | 0.99 | 0.82 | 0.80 | 0.65 | 0.88 | 0.85 | 0.81 | 0.62 | 0.85 | 0.84 | 0.84 | 0.67 | **0.99** | 0.79 |

Table 18: *Experiment 1 – Sibling support policy – ACE* (lower is better). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.20 | 0.16 | 0.16 | 0.21 | 0.18 | 0.25 | 0.33 | 0.19 | 0.16 | 0.20 | 0.15 | 0.21 | 0.29 | 0.09 | 0.19 | 0.27 | 0.23 | 0.28 | 0.31 | 0.19 | 0.22 | 0.22 | 0.11 | 0.23 |
| UnivFD-FT (2023) | 0.09 | 0.10 | 0.07 | 0.11 | 0.06 | 0.23 | 0.17 | 0.08 | 0.07 | 0.14 | 0.11 | 0.12 | 0.05 | 0.10 | 0.04 | 0.08 | 0.06 | 0.22 | 0.22 | 0.04 | 0.06 | 0.13 | 0.05 | 0.10 |
| UnivFD-ZS (2023) | 0.11 | 0.05 | 0.09 | 0.08 | 0.08 | 0.20 | 0.18 | 0.09 | 0.10 | 0.08 | 0.07 | 0.10 | 0.08 | 0.04 | 0.05 | 0.07 | 0.05 | 0.20 | 0.22 | 0.05 | 0.06 | 0.13 | 0.04 | 0.10 |
| BSA (2024) | 0.16 | 0.16 | 0.13 | 0.18 | 0.10 | 0.13 | 0.24 | 0.22 | 0.10 | 0.12 | 0.16 | 0.16 | 0.27 | 0.31 | 0.20 | 0.18 | 0.24 | 0.46 | 0.36 | 0.29 | 0.22 | 0.37 | 0.18 | 0.29 |
| AIDE (2025) | 0.07 | 0.10 | 0.09 | 0.14 | 0.10 | 0.22 | 0.31 | 0.15 | 0.07 | 0.18 | 0.08 | 0.15 | 0.04 | 0.10 | 0.11 | 0.16 | 0.11 | 0.23 | 0.34 | 0.18 | 0.07 | 0.24 | 0.04 | 0.17 |
| MLEP (2025) | 0.20 | 0.10 | 0.33 | 0.35 | 0.27 | **0.09** | 0.32 | 0.31 | 0.28 | 0.13 | 0.10 | 0.23 | 0.22 | 0.11 | 0.32 | 0.37 | 0.30 | **0.14** | 0.40 | 0.32 | 0.29 | 0.21 | 0.16 | 0.27 |
| OSD (2025) | 0.05 | 0.08 | 0.07 | 0.14 | 0.09 | 0.25 | 0.24 | 0.08 | 0.07 | 0.20 | 0.07 | 0.14 | 0.08 | 0.07 | 0.10 | 0.22 | 0.15 | 0.23 | 0.36 | 0.16 | 0.18 | 0.23 | 0.05 | 0.19 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | 0.07 | **0.03** | 0.07 | 0.06 | 0.06 | 0.21 | **0.11** | 0.06 | 0.07 | 0.12 | 0.04 | 0.09 | 0.05 | **0.02** | **0.03** | **0.05** | **0.03** | 0.21 | 0.21 | 0.04 | 0.04 | 0.16 | 0.09 | 0.10 |
| IRM (2019) | 0.08 | 0.06 | 0.08 | 0.08 | 0.07 | 0.23 | 0.13 | 0.07 | 0.08 | 0.15 | 0.07 | 0.10 | 0.16 | 0.13 | 0.15 | 0.15 | 0.14 | 0.25 | 0.25 | 0.15 | 0.15 | 0.15 | 0.17 | 0.17 |
| GroupDRO (2020) | 0.07 | 0.05 | 0.07 | 0.07 | 0.06 | 0.23 | 0.13 | 0.06 | 0.07 | 0.16 | 0.07 | 0.10 | 0.17 | 0.15 | 0.17 | 0.14 | 0.15 | 0.24 | 0.22 | 0.15 | 0.16 | 0.15 | 0.14 | 0.17 |
| TENT (2021) | 0.05 | 0.05 | 0.06 | 0.07 | 0.05 | 0.23 | 0.15 | 0.05 | 0.06 | 0.17 | 0.09 | 0.10 | 0.04 | 0.03 | **0.03** | 0.06 | 0.04 | 0.23 | 0.23 | 0.04 | 0.04 | 0.19 | 0.09 | 0.11 |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.14 | 0.06 | 0.13 | 0.10 | 0.12 | 0.20 | 0.16 | 0.12 | 0.13 | 0.08 | 0.08 | 0.12 | 0.14 | 0.13 | 0.17 | 0.15 | 0.13 | 0.26 | 0.18 | 0.14 | 0.19 | 0.11 | 0.17 | 0.15 |
| MAML (2017) | **0.04** | 0.05 | 0.04 | **0.05** | **0.03** | 0.23 | **0.11** | **0.04** | 0.04 | 0.09 | 0.06 | **0.07** | 0.04 | 0.12 | **0.03** | 0.07 | 0.05 | 0.29 | 0.10 | 0.04 | **0.02** | **0.09** | **0.02** | 0.08 |
| ProtoNet (2017) | **0.04** | 0.06 | **0.03** | 0.06 | 0.04 | 0.24 | **0.11** | 0.05 | **0.03** | 0.11 | 0.06 | 0.08 | 0.04 | 0.12 | **0.03** | 0.09 | 0.05 | 0.28 | 0.11 | 0.05 | 0.06 | 0.10 | 0.03 | 0.08 |
| Reptile (2018) | **0.04** | 0.06 | 0.04 | **0.05** | **0.03** | 0.24 | **0.11** | **0.04** | 0.04 | 0.09 | 0.07 | **0.07** | 0.03 | 0.12 | 0.04 | **0.05** | 0.05 | 0.29 | 0.10 | 0.04 | 0.03 | **0.09** | **0.02** | **0.07** |
| CNAPs (2019) | 0.05 | 0.05 | 0.04 | 0.07 | 0.04 | 0.23 | 0.17 | 0.06 | 0.05 | 0.16 | 0.10 | 0.10 | 0.03 | 0.09 | **0.03** | **0.05** | 0.05 | 0.29 | 0.18 | 0.05 | 0.11 | 0.14 | 0.08 | 0.10 |
| FEAT (2020) | **0.04** | 0.12 | 0.04 | 0.08 | 0.05 | 0.24 | 0.17 | **0.04** | 0.04 | 0.17 | 0.06 | 0.10 | 0.03 | 0.17 | 0.06 | 0.07 | 0.05 | 0.26 | 0.15 | 0.04 | 0.04 | 0.17 | 0.03 | 0.10 |
| ICL-Causal (2022) | 0.06 | 0.04 | 0.05 | 0.07 | 0.05 | 0.22 | 0.14 | 0.05 | 0.06 | 0.15 | 0.09 | 0.10 | **0.02** | 0.07 | **0.03** | 0.07 | 0.06 | 0.23 | 0.25 | 0.06 | **0.02** | 0.24 | 0.14 | 0.13 |
| ICL-GP (2022) | 0.07 | 0.05 | 0.06 | 0.08 | 0.06 | 0.22 | 0.14 | 0.06 | 0.07 | 0.13 | 0.06 | 0.10 | 0.03 | 0.08 | 0.04 | 0.08 | 0.08 | 0.23 | 0.24 | 0.07 | 0.03 | 0.23 | 0.09 | 0.13 |
| TabPFN (2022) | 0.12 | 0.08 | 0.15 | 0.14 | 0.14 | **0.09** | 0.12 | 0.15 | 0.13 | 0.08 | 0.19 | 0.12 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | **0.04** | 0.05 | **0.03** | **0.05** | **0.03** | 0.23 | **0.11** | **0.04** | 0.04 | 0.09 | 0.06 | **0.07** | 0.05 | 0.12 | **0.03** | 0.07 | 0.05 | 0.29 | **0.09** | 0.04 | **0.02** | **0.09** | **0.02** | 0.08 |
| CAML (2024) | 0.09 | 0.06 | 0.08 | 0.09 | 0.08 | 0.22 | 0.17 | 0.07 | 0.08 | 0.13 | 0.06 | 0.11 | **0.02** | 0.16 | 0.04 | 0.10 | 0.06 | 0.23 | 0.24 | 0.06 | 0.03 | 0.31 | 0.28 | 0.16 |
| AFRT (2025) | 0.06 | 0.04 | 0.04 | 0.06 | 0.04 | 0.19 | 0.12 | **0.04** | 0.04 | **0.07** | 0.07 | **0.07** | 0.03 | 0.11 | 0.06 | 0.07 | 0.04 | 0.23 | 0.11 | **0.02** | 0.03 | **0.09** | 0.03 | **0.07** |
| NearestCentroid | 0.15 | 0.33 | 0.16 | 0.18 | 0.16 | 0.37 | 0.22 | 0.17 | 0.28 | 0.26 | **0.02** | 0.21 | 0.18 | 0.33 | 0.16 | 0.17 | 0.19 | 0.36 | 0.16 | 0.15 | 0.18 | 0.30 | **0.02** | 0.21 |

Table 19: *Experiment 1 – Sibling support policy – FNR@1%FPR* (lower is better). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | | | | | | | | | | | Explicit | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average | GAN-U (seen) | Diff-Px (seen) | Token-AR (seen) | Inpaint (seen) | GAN-U (unseen sibling) | Diff-Px (unseen sibling) | Inpaint (unseen sibling) | GAN-C (unseen family) | GAN-Diff (unseen family) | Diff-Lat (unseen family) | Flow (unseen family) | Average |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.99 | 0.90 | 0.96 | 1.00 | 0.98 | 1.00 | 1.00 | 0.94 | 0.96 | 0.96 | 0.91 | 0.97 | 0.99 | 0.87 | 0.95 | 0.99 | 0.97 | 1.00 | 0.97 | 0.88 | 0.96 | 0.97 | 0.76 | 0.95 |
| UnivFD-FT (2023) | 0.32 | 0.74 | 0.63 | 0.43 | 0.48 | 0.83 | 0.64 | 0.40 | 0.41 | 0.87 | 0.13 | 0.57 | 0.32 | 0.82 | 0.78 | 0.36 | 0.53 | 0.88 | 0.64 | 0.44 | 0.47 | 0.91 | 0.45 | 0.63 |
| UnivFD-ZS (2023) | 0.28 | 0.74 | 0.63 | 0.51 | 0.50 | 0.81 | 0.72 | 0.42 | 0.41 | 0.88 | 0.59 | 0.62 | 0.30 | 0.78 | 0.68 | 0.36 | 0.56 | 0.85 | 0.73 | 0.45 | 0.46 | 0.90 | 0.73 | 0.65 |
| BSA (2024) | 0.97 | 0.98 | 0.99 | 1.00 | 0.98 | 0.97 | 1.00 | 0.96 | 1.00 | 0.94 | 0.43 | 0.94 | 1.00 | 0.99 | 1.00 | 0.99 | 1.00 | 1.00 | 0.99 | 0.99 | 0.99 | 1.00 | 1.00 | 1.00 |
| AIDE (2025) | 0.79 | 0.84 | 0.85 | 0.84 | 0.89 | 0.93 | 0.95 | 0.82 | 0.88 | 0.95 | 0.85 | 0.89 | 0.68 | 0.85 | 0.87 | 0.61 | 0.88 | 0.93 | 0.97 | 0.92 | 0.87 | 0.95 | 0.90 | 0.88 |
| MLEP (2025) | 0.93 | 0.94 | 0.99 | 1.00 | 0.97 | 0.86 | 1.00 | 0.99 | 0.96 | 0.93 | 0.82 | 0.95 | 0.91 | 0.80 | 0.98 | 0.99 | 0.95 | 0.79 | 1.00 | 0.99 | 0.92 | 0.92 | 0.87 | 0.93 |
| OSD (2025) | 0.43 | **0.38** | 0.55 | 0.88 | 0.54 | 0.69 | 0.94 | 0.62 | 0.52 | **0.75** | 0.67 | 0.66 | 0.45 | **0.43** | 0.59 | 0.70 | 0.70 | 0.80 | 0.99 | 0.77 | 0.68 | **0.79** | 0.50 | 0.72 |
| *Domain-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| CORAL (2016) | 0.15 | 0.52 | 0.36 | 0.39 | **0.22** | **0.62** | 0.63 | 0.30 | 0.21 | 0.78 | 0.69 | 0.47 | 0.29 | 0.62 | 0.45 | 0.33 | 0.40 | 0.70 | 0.72 | 0.44 | 0.37 | 0.86 | 0.77 | 0.58 |
| IRM (2019) | 0.15 | 0.48 | 0.34 | 0.38 | 0.30 | 0.67 | 0.65 | 0.27 | 0.23 | 0.78 | 0.57 | 0.48 | 0.60 | 0.89 | 0.78 | 0.57 | 0.75 | 0.96 | 0.86 | 0.66 | 0.78 | 0.91 | 0.63 | 0.78 |
| GroupDRO (2020) | 0.16 | 0.42 | **0.30** | 0.36 | 0.26 | 0.66 | 0.62 | 0.26 | 0.19 | 0.76 | 0.65 | 0.46 | 0.48 | 0.78 | 0.47 | 0.77 | 0.66 | 0.74 | 0.86 | 0.61 | 0.54 | 0.88 | 0.86 | 0.72 |
| TENT (2021) | 0.13 | 0.46 | 0.31 | 0.36 | 0.25 | 0.70 | 0.65 | 0.26 | 0.20 | 0.77 | 0.62 | 0.46 | 0.13 | 0.44 | **0.32** | 0.26 | 0.26 | **0.56** | 0.67 | 0.29 | **0.20** | **0.79** | 0.67 | 0.46 |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.30 | 0.69 | 0.42 | 0.67 | 0.38 | 0.81 | 0.77 | 0.37 | 0.39 | 0.84 | 0.63 | 0.59 | 0.52 | 0.80 | 0.60 | 0.84 | 0.50 | 0.84 | 0.76 | 0.45 | 0.75 | 0.87 | 0.67 | 0.69 |
| MAML (2017) | 0.30 | 0.53 | 0.45 | 0.51 | 0.38 | 0.72 | 0.68 | 0.35 | 0.39 | 0.82 | 0.68 | 0.55 | **0.07** | 0.56 | 0.35 | 0.25 | **0.24** | 0.61 | 0.56 | 0.29 | 0.28 | 0.84 | 0.19 | **0.44** |
| ProtoNet (2017) | 0.28 | 0.53 | 0.37 | 0.50 | 0.24 | 0.69 | 0.67 | 0.35 | 0.26 | 0.81 | 0.73 | 0.52 | 0.12 | 0.55 | 0.43 | 0.33 | 0.29 | 0.62 | 0.66 | 0.40 | 0.44 | 0.83 | 0.17 | 0.50 |
| Reptile (2018) | 0.25 | 0.56 | 0.43 | 0.46 | 0.32 | 0.79 | 0.67 | 0.31 | 0.33 | 0.83 | 0.70 | 0.54 | 0.10 | 0.51 | 0.42 | 0.29 | 0.25 | 0.59 | 0.56 | 0.38 | 0.30 | 0.86 | 0.18 | 0.46 |
| CNAPs (2019) | 0.18 | 0.57 | 0.36 | 0.35 | 0.25 | 0.68 | 0.66 | 0.29 | 0.23 | 0.81 | 0.67 | 0.49 | 0.21 | 0.80 | 0.35 | 0.52 | 0.31 | 0.81 | 0.71 | 0.42 | 0.50 | 0.84 | 0.42 | 0.57 |
| FEAT (2020) | 1.00 | 0.73 | 1.00 | 0.61 | 0.95 | 0.92 | 0.64 | 0.93 | 1.00 | 0.96 | 0.89 | 0.88 | 0.19 | 0.76 | 0.43 | 0.40 | 0.27 | 0.82 | **0.46** | **0.24** | 0.27 | 0.88 | **0.14** | 0.47 |
| ICL-Causal (2022) | **0.11** | 0.56 | 0.36 | **0.33** | 0.24 | 0.66 | **0.58** | **0.25** | **0.17** | 0.76 | 0.60 | **0.45** | 0.08 | 0.57 | 0.45 | **0.20** | 0.28 | 0.61 | 0.65 | 0.31 | 0.29 | 0.80 | 0.55 | 0.48 |
| ICL-GP (2022) | 0.13 | 0.63 | 0.44 | 0.36 | 0.31 | 0.69 | 0.62 | 0.28 | 0.23 | 0.77 | 0.55 | 0.49 | 0.18 | 0.68 | 0.55 | 0.26 | 0.42 | 0.65 | 0.69 | 0.41 | 0.46 | 0.83 | 0.45 | 0.55 |
| TabPFN (2022) | 0.76 | 0.95 | 0.78 | 0.77 | 0.67 | 0.92 | 0.65 | 0.65 | 0.70 | 0.95 | **0.05** | 0.73 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| GAP (2023) | 0.31 | 0.55 | 0.41 | 0.52 | 0.36 | 0.72 | 0.68 | 0.36 | 0.36 | 0.81 | 0.71 | 0.55 | 0.08 | 0.56 | 0.37 | 0.27 | 0.26 | 0.61 | 0.56 | 0.31 | 0.28 | 0.84 | 0.20 | 0.45 |
| CAML (2024) | 0.32 | 0.72 | 0.55 | 0.46 | 0.51 | 0.80 | 0.73 | 0.43 | 0.47 | 0.88 | 0.61 | 0.62 | 0.22 | 0.84 | 0.81 | 0.31 | 0.44 | 0.64 | 0.65 | 0.39 | 0.41 | 0.86 | 0.70 | 0.59 |
| AFRT (2025) | 0.28 | 0.78 | 0.46 | 0.58 | 0.38 | 0.84 | 0.71 | 0.37 | 0.34 | 0.86 | 0.54 | 0.58 | 0.23 | 0.70 | 0.45 | 0.39 | 0.35 | 0.77 | 0.51 | 0.39 | 0.29 | 0.88 | 0.27 | 0.51 |
| NearestCentroid | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.98 | 1.00 | 1.00 | 1.00 | 0.52 | 0.97 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.52 | 0.97 |

Table 20: *Experiment 1 – Sibling support policy – ACC per generator* (mean ± std over three training seeds). Columns are grouped by family and test group, implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | |

Table 21: *Experiment 1 – Sibling support policy – AUC per generator* (mean ± std over three training seeds). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | |

Table 22: *Experiment 1 – Sibling support policy – AP per generator* (mean ± std over three training seeds). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | |

Table 23: *Experiment 1 – Sibling support policy – ACE per generator* (mean ± std over three training seeds, lower is better). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | |

Table 24: *Experiment 1 – Sibling support policy – FNR@1% FPR per generator* (mean ± std over three training seeds, lower is better). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | |

Table 25: *Experiment 1 – Matched support policy – ACC per generator* (mean ± std over three training seeds). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | |

Table 26: *Experiment 1 – Disjoint support policy – ACC per generator* (mean ± std over three training seeds). Implicit construction on the left and explicit construction on the right. N/A marks TabPFN under the explicit construction, whose label space exceeds the ten classes TabPFN handles.

| | Implicit | | |

# H EXPERIMENT 2: SYNTHETIC PRIOR

## H.1 EXPERIMENT DESCRIPTION

We pre-train ICD-Causal (Appendix E) with batch size 32 for 30 000 iterations on episodes of the synthetic prior of Section 3.2, with the hyperparameters of Appendix C. The final checkpoint is evaluated at $K{=}32$ with CLIP ViT-L/14 embeddings. AI-GenBench is read under the sibling policy and Chameleon under the matched policy. The references are UnivFD-ZS, UnivFD-FT, and ICL-Causal, all trained on the training pool, and NearestCentroid. The curves of Figure 4 are computed during training on at most 200 queries per generator. They are read every 200 iterations for ICD-Causal and every 10 iterations for ICL-Causal, so their values differ slightly from the final evaluation. For these curves, ICL-Causal is trained a second time with the learning rate $10^{-5}$ instead of $10^{-3}$. At $10^{-3}$, it exceeds an accuracy of 0.8 within 20 iterations, and its curve cannot be resolved. For Figure 2(d) and Figure 5, ICD-Causal is pre-trained on the same prior with the learning rate $10^{-4}$ for 6 000 iterations and three seeds. Its accuracy is read every 200 iterations, on AI-GenBench under the sibling policy and on Cozzolino and Chameleon under the matched policy. The flat lines on Chameleon are the evaluations of Table 27. On Cozzolino, they are UnivFD-ZS and ICL-Causal at $K{=}32$ under the matched policy.

## H.2 RESULTS

Table 28 shows that on AI-GenBench the synthetic prior stays below the detectors trained on AI-GenBench images for almost every generator. Its accuracy is highest on the two Flow generators (0.84), where the classes are well separated in CLIP space. It is lowest on DeepFloyd IF and SN-PatchGAN, where every detector struggles. Table 27 shows the reversed order on Chameleon, where ICD-Causal exceeds both UnivFD-ZS and ICL-Causal. NearestCentroid, which fits no parameters, is the most accurate detector of the table (0.86). Figure 4 shows the emergence of ICD-Causal on AI-GenBench, where it stays below ICL-Causal trained on the training pool. With the learning rate $10^{-4}$ (Figure 5 and Figure 2(d)), ICD-Causal stays at chance for about 600 iterations on all three benchmarks. It then reaches about 0.80 on Cozzolino and Chameleon and 0.73 on AI-GenBench.

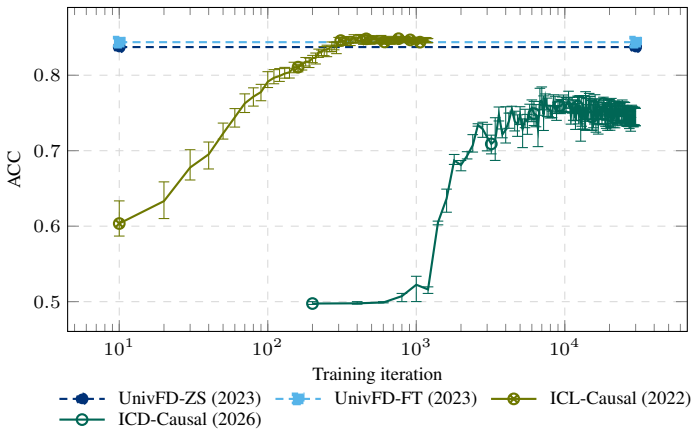
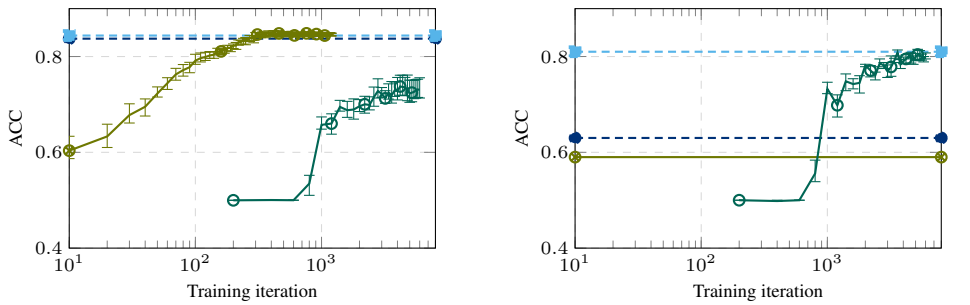



Figure 4: *Experiment 2 – AI-GenBench – ACC* as a function of the training iterations of ICD-Causal under the synthetic prior and of ICL-Causal on the training pool.




(a) AI-GenBench, sibling policy

(b) Chameleon, matched policy

Figure 5: *Experiment 2 – ACC* as a function of the training iterations of ICD-Causal under the synthetic prior.

Table 27: *Experiment 2 – Chameleon – ACC* at $K=32$. ICD-Causal is pre-trained on the synthetic prior only, NearestCentroid fits no parameters, and the other models are trained on the training pool. The last two columns are the differences from UnivFD-FT and from NearestCentroid.

| Model | Wild (unseen family) | Average | Δ vs UnivFD-FT (2023) | Δ vs NearestCentroid |
|---|---|---|---|---|
| | Chameleon | | | |
| *Context-free* | | | | |
| UnivFD-FT (2023) | 0.81 | 0.81 | +0.00 | -0.05 |
| UnivFD-ZS (2023) | 0.63 | 0.63 | -0.18 | -0.23 |
| *Context-aware* | | | | |
| ICL-Causal (2022) | 0.59 | 0.59 | -0.22 | -0.27 |
| ICD-Causal (2026) | 0.75 | 0.75 | -0.06 | -0.11 |
| NearestCentroid | **0.86** | **0.86** | **+0.05** | **+0.00** |

Table 28: *Experiment 2 – AI-GenBench, sibling support policy – ACC per generator* at $K$=32 (mean ± std over three training seeds). ICD-Causal is pre-trained on the synthetic prior only, NearestCentroid fits no parameters, and the other models are trained on the training pool.

| | GAN-U (seen) | | Diff-Px (seen) | | Token-AR (seen) | | Inpaint (seen) | | GAN-U (unseen sibling) | | | | | Diff-Px (unseen sibling) | | Inpaint (unseen sibling) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | ProGAN (2017) | StyleGAN3 (2021) | ADM (2021) | Glide (2021) | VQ-Diffusion (2021) | VQGAN (2020) | FaceSynth (2021) | LaMa (2021) | CIPS (2020) | GANformer (2021) | ProjectedGAN (2021) | StyleGAN1 (2018) | StyleGAN2 (2019) | DDPM (2020) | DeepFloyd IF (2023) | CRN (2017) | IMLE (2018) | MAT (2022) | Palette (2021) | SN-PatchGAN (2018) |
| *Context-free* | | | | | | | | | | | | | | | | | | | | |
| UnivFD-FT (2023) | 0.90±0.02 | 0.93±0.01 | 0.79±0.02 | 0.79±0.03 | 0.91±0.00 | 0.91±0.00 | 0.92±0.04 | 0.76±0.03 | 0.88±0.04 | 0.94±0.01 | 0.89±0.01 | 0.92±0.01 | 0.89±0.01 | 0.85±0.04 | 0.52±0.01 | 0.76±0.20 | 0.65±0.18 | 0.86±0.02 | 0.90±0.04 | 0.56±0.04 |
| UnivFD-ZS (2023) | 0.90±0.00 | 0.90±0.00 | 0.83±0.00 | 0.85±0.00 | 0.91±0.00 | 0.90±0.01 | 0.91±0.00 | 0.81±0.01 | 0.90±0.01 | 0.92±0.00 | 0.89±0.00 | 0.91±0.00 | 0.87±0.00 | 0.91±0.00 | 0.52±0.01 | 0.50±0.02 | 0.68±0.04 | 0.87±0.00 | 0.90±0.00 | 0.65±0.02 |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | |
| ICL-Causal (2022) | **0.93±0.00** | **0.94±0.00** | **0.88±0.01** | **0.88±0.01** | **0.93±0.01** | **0.92±0.01** | **0.94±0.00** | **0.84±0.01** | **0.92±0.01** | **0.95±0.00** | **0.91±0.00** | **0.94±0.01** | **0.90±0.01** | **0.94±0.00** | 0.52±0.01 | 0.67±0.04 | **0.85±0.04** | **0.89±0.01** | **0.93±0.00** | **0.72±0.01** |
| ICD-Causal (2026) | 0.74±0.00 | 0.78±0.03 | 0.62±0.01 | 0.59±0.02 | 0.73±0.02 | 0.75±0.01 | 0.84±0.02 | 0.66±0.02 | 0.78±0.03 | 0.82±0.02 | 0.78±0.03 | 0.78±0.01 | 0.72±0.05 | 0.66±0.01 | **0.53±0.02** | 0.79±0.09 | 0.79±0.07 | 0.70±0.02 | 0.76±0.05 | 0.54±0.02 |
| NearestCentroid | 0.88±0.00 | 0.90±0.00 | 0.66±0.00 | 0.68±0.00 | 0.83±0.00 | 0.89±0.00 | **0.94±0.00** | 0.70±0.00 | 0.91±0.00 | 0.90±0.00 | 0.86±0.00 | 0.86±0.00 | 0.77±0.00 | 0.77±0.00 | **0.53±0.00** | **0.84±0.00** | **0.85±0.00** | 0.76±0.00 | 0.83±0.00 | 0.60±0.00 |

| | GAN-U (unseen sibling) | | GAN-C (unseen family) | | | | GAN-Diff (unseen family) | | | Diff-Lat (unseen family) | | | | | | | Flow (unseen family) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | Palette (2021) | SN-PatchGAN (2018) | BigGAN (2018) | CycleGAN (2017) | GauGAN (2019) | StarGAN (2017) | DD-GAN (2021) | DiffGAN-ProjGAN (2022) | DiffGAN-StyGAN2 (2022) | DALL-E 3 (2023) | LatDiff (2021) | Midjourney (2022) | SD1.4 (2022) | SD1.5 (2022) | SD2.1 (2022) | SDXL (2023) | FLUXDev (2024) | FLUXSch (2024) | Average | Δ vs UnivFD-FT (2023) | Δ vs NearestCentroid |
| *Context-free* | | | | | | | | | | | | | | | | | | | | | |
| UnivFD-FT (2023) | 0.90±0.04 | 0.56±0.04 | 0.92±0.02 | 0.89±0.06 | 0.91±0.02 | 0.92±0.02 | **0.93±0.02** | 0.91±0.03 | 0.91±0.01 | **0.80±0.02** | 0.80±0.01 | 0.80±0.02 | **0.79±0.01** | 0.78±0.01 | **0.79±0.03** | **0.81±0.02** | 0.94±0.01 | 0.93±0.01 | 0.84±0.03 | +0.00 | +0.04 |
| UnivFD-ZS (2023) | 0.90±0.00 | 0.65±0.02 | 0.90±0.00 | 0.91±0.01 | 0.90±0.00 | 0.91±0.00 | 0.90±0.00 | 0.90±0.00 | 0.90±0.00 | 0.79±0.01 | 0.85±0.01 | 0.83±0.00 | 0.76±0.01 | 0.78±0.01 | 0.74±0.00 | 0.78±0.00 | 0.88±0.00 | 0.91±0.00 | 0.84±0.01 | +0.00 | +0.04 |
| *Context-aware* | | | | | | | | | | | | | | | | | | | | | |
| ICL-Causal (2022) | **0.92±0.00** | **0.72±0.01** | **0.93±0.00** | **0.92±0.01** | **0.93±0.00** | **0.93±0.00** | **0.93±0.00** | **0.93±0.01** | **0.92±0.00** | 0.63±0.03 | **0.89±0.01** | **0.85±0.00** | 0.76±0.00 | **0.80±0.01** | 0.72±0.01 | 0.77±0.01 | 0.82±0.02 | 0.89±0.01 | **0.86±0.01** | **+0.02** | **+0.06** |
| ICD-Causal (2026) | 0.68±0.02 | 0.54±0.02 | 0.68±0.02 | 0.75±0.05 | 0.72±0.04 | 0.67±0.04 | 0.64±0.04 | 0.67±0.05 | 0.71±0.06 | 0.68±0.05 | 0.60±0.03 | 0.71±0.03 | 0.67±0.02 | 0.68±0.01 | 0.70±0.03 | 0.67±0.02 | 0.84±0.03 | 0.84±0.02 | 0.71±0.03 | -0.13 | -0.09 |
| NearestCentroid | 0.83±0.00 | 0.60±0.00 | 0.82±0.00 | 0.86±0.00 | 0.87±0.00 | 0.76±0.00 | 0.55±0.00 | 0.76±0.00 | 0.81±0.00 | 0.79±0.00 | 0.60±0.00 | 0.82±0.00 | 0.74±0.00 | 0.79±0.00 | 0.78±0.00 | 0.80±0.00 | **0.99±0.00** | **0.99±0.00** | 0.80±0.00 | -0.04 | +0.00 |

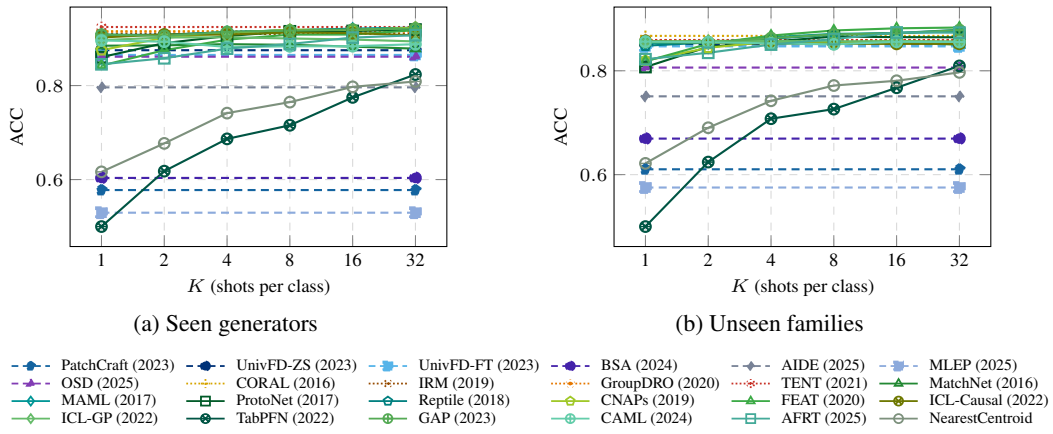
# I EXPERIMENT 3: SUPPORT-SET SIZE

## I.1 EXPERIMENT DESCRIPTION

Every context-aware detector is trained on the training pool under the implicit construction once for each $K \in \{1, 2, 4, 8, 16, 32\}$ and evaluated at the same $K$. Detectors that read no support set are trained once at $K{=}8$ and drawn as horizontal lines at their mean accuracy. AI-GenBench is read under the sibling policy and Chameleon under the matched policy.

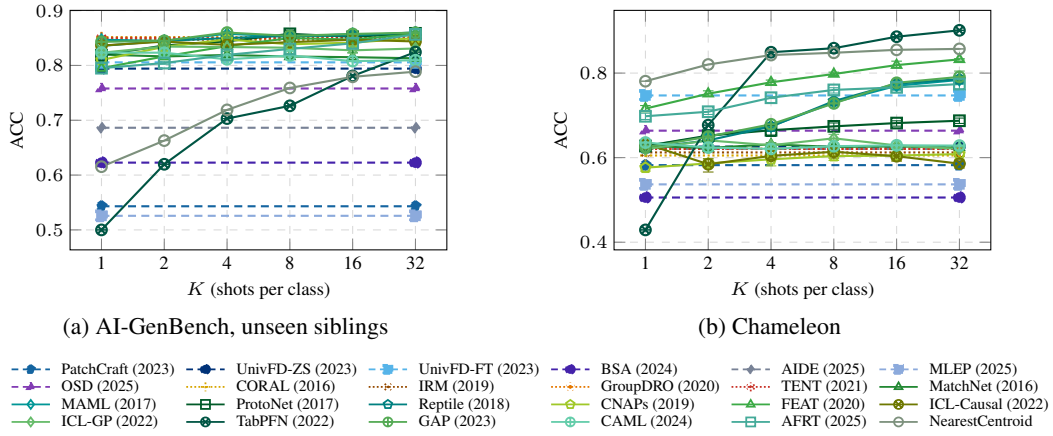
## I.2 RESULTS

Figure 7 repeats Figure 2(a, b) with every detector in colour. Figure 6 adds the seen generators and the unseen families. On the seen generators the context-aware detectors gain about as much from a larger support set as on the unseen ones (FEAT from 0.84 to 0.92), and the domain-aware detectors reach 0.91–0.92, so the best detectors of both categories end at the same accuracy. On the unseen families, FEAT, Reptile, and GAP reach 0.88 at $K{=}32$ and exceed the best domain-aware detector (0.87) slightly. TabPFN starts at chance at $K{=}1$ and improves steadily with $K$ on every benchmark.




(a) Seen generators (b) Unseen families

Figure 6: *Experiment 3 – AI-GenBench – ACC* as a function of $K$ on the seen generators and on the unseen families, under the sibling policy.




(a) AI-GenBench, unseen siblings (b) Chameleon

Figure 7: *Experiment 3 – ACC* as a function of $K$ for every detector, on the unseen siblings of AI-GenBench under the sibling policy and on Chameleon under the matched policy.

# J EXPERIMENT 4: DIVERSITY OF TRAINING FAMILIES

## J.1 EXPERIMENT DESCRIPTION

Each point of Figure 2(c) and Figure 8 averages two pools of eight generators drawn from $F \in \{1, 2, 3\}$ of the four training families, each trained with one seed and with the budget of 16 000 real and 16 000 AI-generated images. For $F{=}1$, a family with fewer than eight generators contributes all of them. The point $F{=}4$ is the training pool. Models that fit nothing, or that are fitted only on the support set, are left out, because the number of training families does not affect them.

## J.2 RESULTS

Figure 8 repeats Figure 2(c) with every detector in colour and adds Chameleon. On Chameleon, the accuracy of most detectors does not grow with $F$, and several have their minimum at $F{=}3$. The ordering is stable, however. FEAT is the most accurate detector for every $F$ (0.80–0.84), followed by AFRT, Reptile, GAP, and MAML. No static detector exceeds 0.67. On in-the-wild images, a more diverse training pool therefore does not replace the support set.

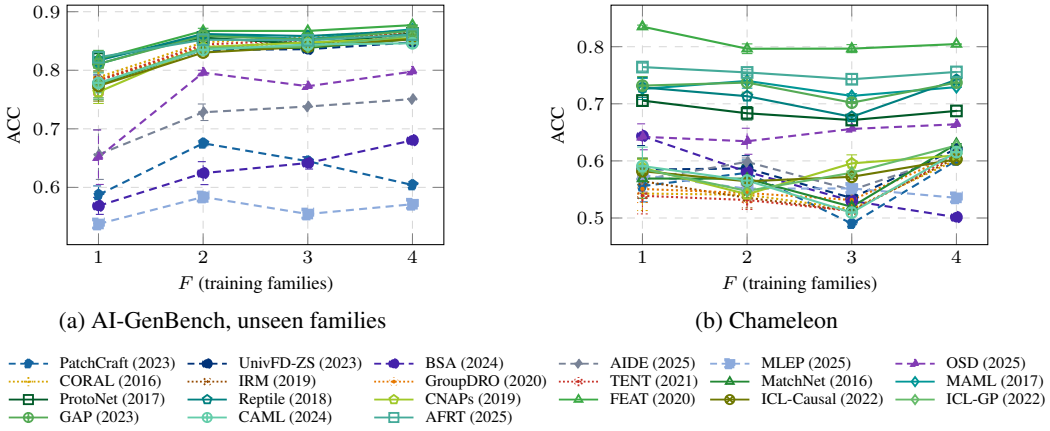



Figure 8: *Experiment 4 – ACC* as a function of the number $F$ of training families for every detector, on the unseen families of AI-GenBench under the sibling policy and on Chameleon under the matched policy.

# K EXPERIMENT 5: FEATURE SPACE

## K.1 EXPERIMENT DESCRIPTION

The advantage of the context-aware detectors could be a property of the frozen CLIP ViT-L/14 embeddings rather than of the formulation. We therefore repeat the training and the evaluation of Experiment 1 with two other frozen feature extractors. CLIP ViT-B/32 (Radford et al., 2021) has a lower capacity. DINOv2-L (Oquab et al., 2023) is self-supervised and not aligned with language. Detectors that read raw images do not use the feature extractor and have a single entry. All other settings follow Experiment 1 under the implicit construction.

## K.2 RESULTS

Table 29 shows that CLIP ViT-L/14 is the best feature space for every model on AI-GenBench, while on Chameleon CLIP ViT-B/32 is about as good. On AI-GenBench, the best context-aware and the best domain-aware detectors stay within 0.02 of each other for every feature extractor, which repeats the small margin of Experiment 1. On Chameleon, the ranking does not depend on the feature extractor. TabPFN, NearestCentroid, and FEAT lead for all three feature extractors. NearestCentroid keeps 0.81 on DINOv2, while UnivFD-FT drops to 0.65. The advantage of the support set on Chameleon is therefore a property of the formulation and not of one feature space.

Table 29: *Experiment 5 – ACC per feature extractor* on AI-GenBench under the sibling policy (left) and on Chameleon under the matched policy (right). N/A marks detectors that read raw images.

| Model | clip_l14 | clip_b32 | dinov2 |
|---|---|---|---|
| *Context-free* | | | |
| PatchCraft (2023) | 0.581 | N/A | N/A |
| UnivFD-FT (2023) | 0.837 | 0.748 | 0.820 |
| UnivFD-ZS (2023) | 0.837 | 0.744 | 0.821 |
| BSA (2024) | 0.639 | N/A | N/A |
| AIDE (2025) | 0.739 | N/A | N/A |
| MLEP (2025) | 0.549 | N/A | N/A |
| OSD (2025) | 0.802 | N/A | N/A |
| *Domain-aware* | | | |
| CORAL (2016) | 0.873 | 0.758 | 0.836 |
| IRM (2019) | 0.866 | 0.758 | 0.833 |
| GroupDRO (2020) | 0.868 | 0.755 | 0.832 |
| TENT (2021) | 0.871 | 0.752 | 0.830 |
| *Context-aware* | | | |
| MatchNet (2016) | 0.849 | 0.741 | 0.819 |
| MAML (2017) | **0.875** | 0.775 | 0.831 |
| ProtoNet (2017) | 0.874 | 0.774 | 0.833 |
| Reptile (2018) | **0.875** | 0.772 | **0.837** |
| CNAPs (2019) | 0.863 | 0.749 | 0.826 |
| FEAT (2020) | **0.875** | 0.767 | 0.817 |
| ICL-Causal (2022) | 0.862 | 0.742 | 0.825 |
| ICL-GP (2022) | 0.858 | 0.741 | 0.826 |
| TabPFN (2022) | 0.724 | 0.696 | 0.632 |
| GAP (2023) | **0.875** | **0.776** | 0.833 |
| CAML (2024) | 0.847 | 0.748 | 0.828 |
| AFRT (2025) | 0.857 | 0.730 | 0.798 |
| NearestCentroid | 0.766 | 0.705 | 0.667 |

| Model | clip_l14 | clip_b32 | dinov2 |
|---|---|---|---|
| *Context-free* | | | |
| PatchCraft (2023) | 0.582 | N/A | N/A |
| UnivFD-FT (2023) | 0.747 | 0.734 | 0.650 |
| UnivFD-ZS (2023) | 0.626 | 0.628 | 0.588 |
| BSA (2024) | 0.506 | N/A | N/A |
| AIDE (2025) | 0.621 | N/A | N/A |
| MLEP (2025) | 0.537 | N/A | N/A |
| OSD (2025) | 0.664 | N/A | N/A |
| *Domain-aware* | | | |
| CORAL (2016) | 0.605 | 0.610 | 0.597 |
| IRM (2019) | 0.612 | 0.619 | 0.573 |
| GroupDRO (2020) | 0.621 | 0.628 | 0.579 |
| TENT (2021) | 0.621 | 0.645 | 0.583 |
| *Context-aware* | | | |
| MatchNet (2016) | 0.625 | 0.628 | 0.586 |
| MAML (2017) | 0.731 | 0.673 | 0.660 |
| ProtoNet (2017) | 0.674 | 0.655 | 0.618 |
| Reptile (2018) | 0.734 | 0.658 | 0.666 |
| CNAPs (2019) | 0.604 | 0.619 | 0.585 |
| FEAT (2020) | 0.798 | 0.803 | 0.693 |
| ICL-Causal (2022) | 0.614 | 0.629 | 0.581 |
| ICL-GP (2022) | 0.646 | 0.621 | 0.585 |
| TabPFN (2022) | **0.859** | **0.846** | 0.770 |
| GAP (2023) | 0.728 | 0.679 | 0.656 |
| CAML (2024) | 0.627 | 0.614 | 0.583 |
| AFRT (2025) | 0.760 | 0.713 | 0.677 |
| NearestCentroid | 0.848 | 0.828 | **0.809** |

# L Experiment 6: Decomposition of the Gain

## L.1 Experiment description

Experiment 6 asks which choice in the training of a context-aware detector causes its accuracy. It compares four variants of every context-aware detector. The variants share the architecture, the feature extractor, the hyperparameters, and the training pool. All four are evaluated as in Experiment 1, under the sibling policy on AI-GenBench and under the matched policy on Chameleon. They differ only in how the training episodes are built.

- *R0* shows the same support set in every training episode. This support set is drawn once from the training pool. At test time, one support set is drawn once from the test split and used for every query. The detector reads a support set, but it never sees the support set change.
- *R1* draws a new support set for every episode, as our implicit construction does. Its AI-generated support images, however, may come from the query's own generator.
- *R2* is the explicit construction of Experiment 1.
- *R3* is our implicit construction. Its AI-generated support images never come from the query's generator.

Each difference between two variants isolates one choice. R1−R0 measures the effect of drawing a new support set for every episode. R3−R1 measures the effect of keeping the support and the query generators apart. R3−R2 compares our construction with the explicit one.

## L.2 Results

Tables 30–32 show that on AI-GenBench the implicit construction is the only ingredient with a consistent effect. The mean difference R3−R2 is positive for all three column types and grows from the seen generators (+0.012) to the unseen siblings (+0.018) and the unseen families (+0.023), so the implicit construction helps most where generalisation is needed. Drawing a new support set for every episode (R1−R0) changes the mean accuracy by at most 0.015, and the generator-disjoint constraint (R3−R1) by at most 0.001. Unexpectedly, the disjoint constraint therefore does not matter at this training budget. On Chameleon (Table 33), drawing a new support set for every episode raises the mean accuracy by 0.014, with the largest gains for AFRT (+0.05) and Reptile (+0.04).

Table 30: *Experiment 6 – AI-GenBench, seen generators – ACC* of the four variants and their differences. The columns *conditioning*, *disjointness*, and *implicit vs explicit* are R1−R0, R3−R1, and R3−R2. N/A marks the explicit variant R2 of TabPFN, whose label space exceeds the ten classes TabPFN handles.

| Model | R0 static | R1 support non-disjoint | R2 explicit N-way | R3 implicit disjoint | Δ R1−R0 conditioning | Δ R3−R1 disjointness | Δ R3−R2 implicit vs explicit |
|---|---|---|---|---|---|---|---|
| *Context-aware* | | | | | | | |
| MatchNet (2016) | 0.882 | 0.887 | 0.893 | 0.887 | 0.004 | 0.000 | -0.006 |
| MAML (2017) | 0.919 | **0.918** | 0.898 | 0.917 | -0.001 | -0.001 | 0.019 |
| ProtoNet (2017) | 0.917 | 0.914 | 0.888 | 0.915 | -0.003 | **0.002** | **0.027** |
| Reptile (2018) | 0.870 | **0.918** | 0.896 | 0.918 | 0.048 | 0.000 | 0.022 |
| CNAPs (2019) | **0.926** | **0.918** | 0.893 | **0.919** | -0.008 | 0.001 | 0.025 |
| FEAT (2020) | 0.891 | 0.907 | 0.885 | 0.908 | 0.016 | 0.001 | 0.024 |
| ICL-Causal (2022) | 0.911 | 0.913 | **0.922** | 0.913 | 0.002 | 0.000 | -0.008 |
| ICL-GP (2022) | 0.893 | 0.901 | 0.904 | 0.900 | 0.008 | -0.001 | -0.004 |
| TabPFN (2022) | 0.697 | 0.716 | N/A | 0.716 | 0.019 | 0.000 | N/A |
| GAP (2023) | 0.917 | **0.918** | 0.896 | 0.917 | 0.000 | -0.000 | 0.021 |
| CAML (2024) | 0.861 | 0.888 | 0.881 | 0.884 | 0.027 | -0.004 | 0.003 |
| AFRT (2025) | 0.886 | 0.888 | 0.875 | 0.888 | 0.002 | 0.001 | 0.013 |
| NearestCentroid | 0.689 | 0.765 | 0.763 | 0.765 | **0.076** | 0.000 | 0.002 |
| Mean | 0.866 | 0.881 | 0.883 | 0.881 | 0.015 | -0.000 | 0.012 |

Table 31: *Experiment 6 – AI-GenBench, unseen siblings – ACC* of the four variants and their differences. N/A as in Table 30.

| Model | R0 static | R1 support non-disjoint | R2 explicit N-way | R3 implicit disjoint | Δ R1−R0 conditioning | Δ R3−R1 disjointness | Δ R3−R2 implicit vs explicit |
|---|---|---|---|---|---|---|---|
| *Context-aware* | | | | | | | |
| MatchNet (2016) | 0.810 | 0.820 | 0.810 | 0.816 | 0.009 | -0.003 | 0.006 |
| MAML (2017) | **0.864** | **0.859** | 0.841 | 0.852 | -0.005 | -0.007 | 0.011 |
| ProtoNet (2017) | **0.864** | 0.850 | 0.831 | **0.858** | -0.014 | **0.008** | 0.027 |
| Reptile (2018) | 0.853 | 0.853 | 0.840 | 0.854 | -0.000 | 0.001 | 0.014 |
| CNAPs (2019) | 0.851 | 0.852 | 0.812 | 0.839 | 0.001 | -0.013 | 0.027 |
| FEAT (2020) | 0.855 | 0.853 | **0.844** | 0.850 | -0.002 | -0.003 | 0.006 |
| ICL-Causal (2022) | 0.839 | 0.842 | 0.798 | 0.842 | 0.003 | 0.000 | 0.044 |
| ICL-GP (2022) | 0.827 | 0.827 | 0.781 | 0.832 | 0.000 | 0.005 | **0.051** |
| TabPFN (2022) | 0.821 | 0.726 | N/A | 0.726 | -0.095 | 0.000 | N/A |
| GAP (2023) | **0.864** | 0.857 | 0.840 | 0.854 | -0.007 | -0.003 | 0.014 |
| CAML (2024) | 0.787 | 0.815 | 0.795 | 0.818 | **0.028** | 0.004 | 0.023 |
| AFRT (2025) | 0.830 | 0.838 | 0.829 | 0.830 | 0.008 | -0.008 | 0.002 |
| NearestCentroid | 0.827 | 0.759 | 0.765 | 0.759 | -0.068 | 0.000 | -0.006 |
| Mean | 0.838 | 0.827 | 0.816 | 0.825 | -0.011 | -0.001 | 0.018 |

# M Experiment 7: Support-Set Counterfactuals

## M.1 Experiment description

Three probes evaluate the checkpoints of Experiment 1 under modified support sets. (a) The *family matrix* scores the queries of every family with support sets drawn from each family in turn, never from the query's own generator. The diagonal corresponds to the sibling policy and the off-diagonal entries to the disjoint policy. It is computed for the three detectors with the highest average accuracy in Table 1. (b) *Label inversion* swaps the real and AI-generated labels of the support set while keeping its images. A detector that conditions on the support labels must invert its predictions and fall below chance. (c) *Prediction shift* measures the standard deviation and the range of the predicted probability of the AI-generated class over resampled support sets, and the rate at which the decision flips. All three quantities are zero for a static detector.

## M.2 Results

Table 34 and Figure 9 show that sibling support is better than disjoint support by only 0.01–0.02. The rows of the matrices are nearly identical, so the family of the query decides the accuracy much more than the family of the support set. The only visible diagonal effects are Flow, Diff-Lat, and Inpaint, whose queries profit from support images of their own family. Table 35 shows that label inversion drives MatchNet, MAML, ProtoNet, Reptile, CNAPs, FEAT, GAP, AFRT, and NearestCentroid to 0.12–0.23 on AI-GenBench, so these detectors follow the support labels. ICL-Causal, ICL-GP, and CAML are unchanged, so they have learned a static decision boundary despite their architecture. Table 36 shows that predictions depend on the support set much more on Chameleon than on AI-

Table 32: *Experiment 6 – AI-GenBench, unseen families – ACC* of the four variants and their differences. N/A as in Table 30.

| Model | R0 static | R1 support non-disjoint | R2 explicit N-way | R3 implicit disjoint | Δ R1−R0 conditioning | Δ R3−R1 disjointness | Δ R3−R2 implicit vs explicit |
|---|---|---|---|---|---|---|---|
| *Context-aware* | | | | | | | |
| MatchNet (2016) | 0.850 | 0.856 | 0.832 | 0.855 | 0.006 | -0.002 | 0.023 |
| MAML (2017) | 0.872 | 0.872 | 0.856 | 0.871 | 0.000 | -0.001 | 0.015 |
| ProtoNet (2017) | 0.860 | 0.863 | 0.853 | 0.866 | 0.003 | **0.003** | 0.013 |
| Reptile (2018) | 0.855 | 0.869 | 0.857 | 0.869 | **0.015** | 0.000 | 0.013 |
| CNAPs (2019) | 0.861 | 0.859 | 0.836 | 0.854 | -0.002 | -0.005 | 0.018 |
| FEAT (2020) | **0.877** | **0.879** | **0.858** | **0.878** | 0.002 | -0.001 | 0.020 |
| ICL-Causal (2022) | 0.850 | 0.856 | 0.820 | 0.852 | 0.006 | -0.004 | 0.032 |
| ICL-GP (2022) | 0.857 | 0.858 | 0.816 | 0.857 | 0.001 | -0.001 | 0.041 |
| TabPFN (2022) | 0.776 | 0.726 | N/A | 0.726 | -0.050 | 0.000 | N/A |
| GAP (2023) | 0.871 | 0.873 | 0.857 | 0.870 | 0.001 | -0.002 | 0.013 |
| CAML (2024) | 0.838 | 0.853 | 0.770 | 0.851 | **0.015** | -0.002 | **0.081** |
| AFRT (2025) | 0.856 | 0.863 | 0.856 | 0.861 | 0.007 | -0.002 | 0.006 |
| NearestCentroid | 0.816 | 0.772 | 0.772 | 0.772 | -0.044 | 0.000 | -0.001 |
| Mean | 0.849 | 0.846 | 0.832 | 0.845 | -0.003 | -0.001 | 0.023 |

Table 33: *Experiment 6 – Chameleon – ACC* of the implicit variants and their differences. The explicit variant R2 needs generator labels and is not defined on Chameleon.

| Model | R0 static | R1 support non-disjoint | R3 implicit disjoint | Δ R1−R0 conditioning | Δ R3−R1 disjointness |
|---|---|---|---|---|---|
| *Context-aware* | | | | | |
| MatchNet (2016) | 0.625 | 0.624 | 0.625 | -0.001 | 0.002 |
| MAML (2017) | 0.727 | 0.738 | 0.731 | 0.012 | -0.008 |
| ProtoNet (2017) | 0.663 | 0.675 | 0.674 | 0.012 | -0.001 |
| Reptile (2018) | 0.681 | 0.720 | 0.734 | 0.039 | **0.013** |
| CNAPs (2019) | 0.605 | 0.604 | 0.604 | -0.001 | -0.001 |
| FEAT (2020) | 0.805 | 0.815 | 0.798 | 0.010 | -0.017 |
| ICL-Causal (2022) | 0.597 | 0.606 | 0.614 | 0.009 | 0.008 |
| ICL-GP (2022) | 0.622 | 0.634 | 0.646 | 0.012 | 0.011 |
| TabPFN (2022) | 0.831 | **0.859** | **0.859** | 0.028 | 0.000 |
| GAP (2023) | 0.744 | 0.737 | 0.728 | -0.007 | -0.008 |
| CAML (2024) | 0.610 | 0.632 | 0.627 | 0.022 | -0.005 |
| AFRT (2025) | 0.701 | 0.752 | 0.760 | **0.051** | 0.008 |
| NearestCentroid | **0.856** | 0.848 | 0.848 | -0.008 | 0.000 |
| Mean | 0.698 | 0.711 | 0.711 | 0.014 | 0.000 |

GenBench. For MAML, the decision-flip rate grows from 0.14 to 0.62, and for FEAT from 0.15 to 0.51.

Table 34: *Experiment 7(a) – Family matrix on AI-GenBench – ACC*, averaged over the diagonal (sibling support) and the off-diagonal (disjoint support), with the least accurate pair of support and query family.

| Model | Sibling support | Disjoint support | Δ sibling − disjoint | Worst family pair |
|---|---|---|---|---|
| *Context-aware* | | | | |
| MAML (2017) | 0.886 | 0.873 | 0.013 | 0.765 |
| Reptile (2018) | 0.886 | **0.876** | 0.010 | **0.767** |
| FEAT (2020) | **0.890** | 0.866 | **0.024** | 0.742 |

# N Experiment 8: Support-Set Corruption

## N.1 Experiment description

Two probes evaluate the checkpoints of Experiment 1 with corrupted support labels. (a) *Label noise* replaces a fraction $\varepsilon \in \{0.125, 0.25, 0.5\}$ of the support labels in every episode by the other label, under both constructions. The *breakdown point* is the interval of $\varepsilon$ in which the accuracy of a detector falls below that of UnivFD-ZS. (b) *Pseudo-labelled support* replaces the ground-truth support labels by the predictions of UnivFD-ZS on the same images, which simulates a deployment in which only unlabelled support images are available. The values at $\varepsilon$=0 are those of Experiment 1.

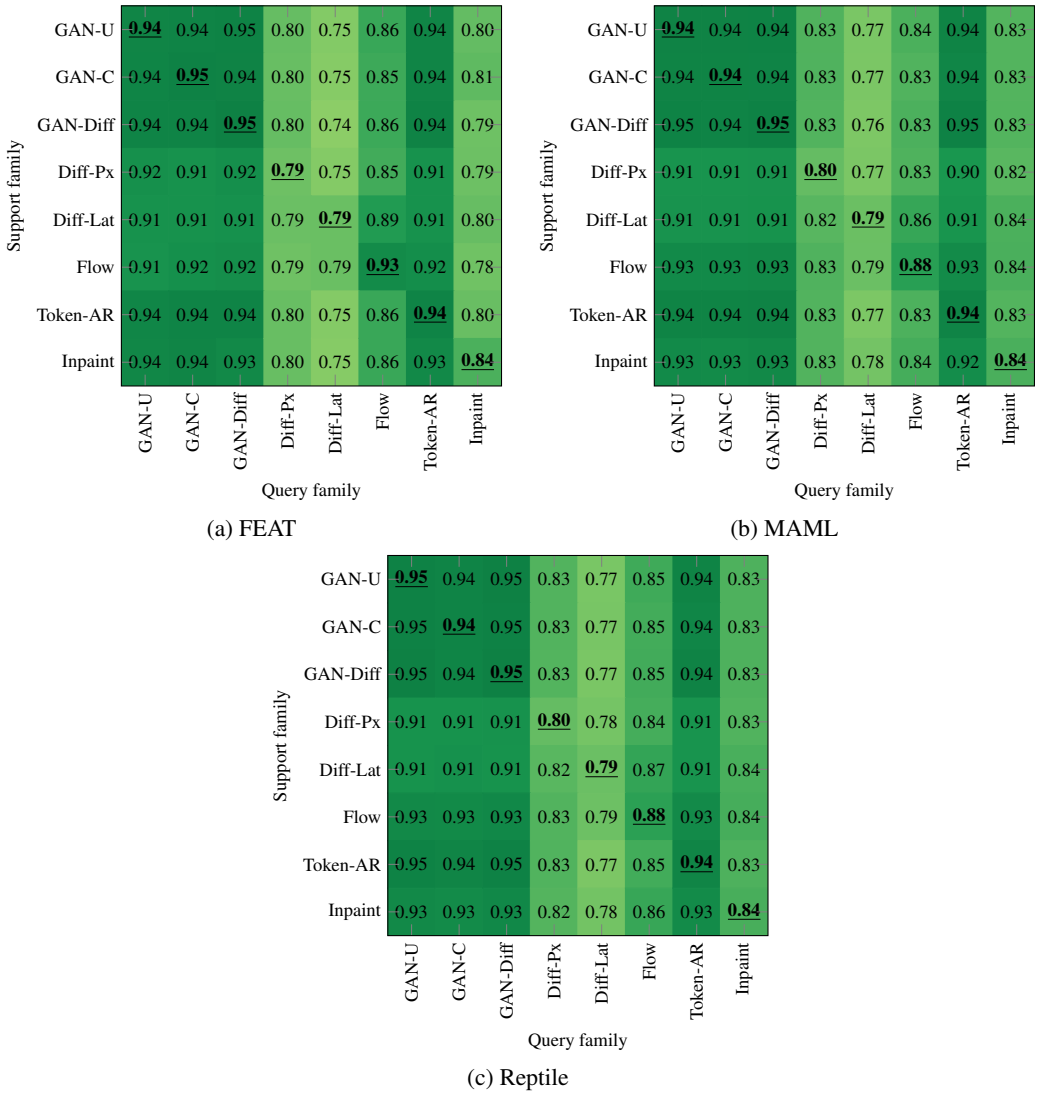



Figure 9: *Experiment 7(a) – Family matrices – ACC* of (a) FEAT, (b) MAML, and (c) Reptile. Rows are support families, columns are query families.

## N.2 Results

Table 37 shows that the implicit construction tolerates more label noise than the explicit one. ProtoNet, CNAPs, and FEAT stay above UnivFD-ZS up to $\varepsilon$=0.25 under the implicit construction but only up to $\varepsilon$=0.125 under the explicit one. MAML, Reptile, and GAP also break down one step earlier under the explicit construction. At $\varepsilon$=0.5 the binary support labels carry no information, and all detectors that follow them fall to chance under the implicit construction. On Chameleon, most context-aware detectors keep their lead over UnivFD-ZS up to $\varepsilon$=0.25 (FEAT 0.78, NearestCentroid 0.82). Table 39 shows that pseudo-labels work only when the zero-shot detector is already accurate. On AI-GenBench, UnivFD-ZS labels 84% of the support images correctly. ProtoNet, CNAPs, MAML, Reptile, and GAP then lose at most 0.006 and stay above both static baselines. On Chameleon, where it labels only 62% correctly, every detector that follows the labels falls below UnivFD-FT, and TabPFN drops from 0.86 to 0.47.

Table 35: *Experiment 7(b) – Label inversion – ACC* with the ground-truth and the inverted support labels, on AI-GenBench under the sibling policy (left) and on Chameleon under the matched policy (right).

| Model | Ground-truth support | Inverted support | Δ inverted − clean |
|---|---|---|---|
| *Context-aware* | | | |
| MatchNet (2016) | 0.849 | 0.151 | -0.698 |
| MAML (2017) | **0.875** | 0.124 | -0.751 |
| ProtoNet (2017) | 0.874 | 0.126 | -0.748 |
| Reptile (2018) | **0.875** | 0.125 | -0.750 |
| CNAPs (2019) | 0.863 | 0.137 | -0.726 |
| FEAT (2020) | **0.875** | 0.125 | -0.750 |
| ICL-Causal (2022) | 0.862 | **0.862** | 0.000 |
| ICL-GP (2022) | 0.858 | 0.859 | **0.001** |
| TabPFN (2022) | 0.724 | 0.268 | -0.456 |
| GAP (2023) | **0.875** | 0.124 | -0.751 |
| CAML (2024) | 0.847 | 0.840 | -0.007 |
| AFRT (2025) | 0.857 | 0.143 | -0.713 |
| NearestCentroid | 0.766 | 0.234 | -0.532 |

| Model | Ground-truth support | Inverted support | Δ inverted − clean |
|---|---|---|---|
| *Context-aware* | | | |
| MatchNet (2016) | 0.625 | 0.375 | -0.251 |
| MAML (2017) | 0.731 | 0.232 | -0.498 |
| ProtoNet (2017) | 0.674 | 0.326 | -0.348 |
| Reptile (2018) | 0.734 | 0.215 | -0.519 |
| CNAPs (2019) | 0.604 | 0.396 | -0.208 |
| FEAT (2020) | 0.798 | 0.202 | -0.596 |
| ICL-Causal (2022) | 0.614 | 0.614 | -0.000 |
| ICL-GP (2022) | 0.646 | **0.649** | 0.003 |
| TabPFN (2022) | **0.859** | 0.152 | -0.707 |
| GAP (2023) | 0.728 | 0.242 | -0.486 |
| CAML (2024) | 0.627 | 0.637 | **0.010** |
| AFRT (2025) | 0.760 | 0.240 | -0.520 |
| NearestCentroid | 0.848 | 0.152 | -0.696 |

Table 36: *Experiment 7(c) – Prediction shift* over resampled support sets, on AI-GenBench under the sibling policy (top) and on Chameleon under the matched policy (bottom).

| Model | P(fake) std over supports | P(fake) range over supports | Decision-flip rate | Accuracy |
|---|---|---|---|---|
| *Context-aware* | | | | |
| MatchNet (2016) | 0.011 | 0.034 | 0.005 | 0.848 |
| MAML (2017) | 0.062 | 0.183 | 0.144 | 0.875 |
| ProtoNet (2017) | 0.055 | 0.166 | 0.145 | 0.875 |
| Reptile (2018) | 0.059 | 0.172 | 0.126 | **0.876** |
| CNAPs (2019) | 0.019 | 0.058 | 0.027 | 0.863 |
| FEAT (2020) | 0.055 | 0.153 | 0.152 | 0.874 |
| ICL-Causal (2022) | 0.003 | 0.009 | 0.009 | 0.862 |
| ICL-GP (2022) | 0.008 | 0.024 | 0.023 | 0.857 |
| TabPFN (2022) | 0.113 | 0.352 | **0.707** | 0.726 |
| GAP (2023) | 0.061 | 0.181 | 0.143 | 0.875 |
| CAML (2024) | 0.027 | 0.085 | 0.073 | 0.847 |
| AFRT (2025) | 0.078 | 0.237 | 0.231 | 0.855 |
| NearestCentroid | **0.238** | **0.587** | 0.588 | 0.769 |

| Model | P(fake) std over supports | P(fake) range over supports | Decision-flip rate | Accuracy |
|---|---|---|---|---|
| *Context-aware* | | | | |
| MatchNet (2016) | 0.021 | 0.063 | 0.008 | 0.640 |
| MAML (2017) | 0.135 | 0.416 | 0.620 | 0.699 |
| ProtoNet (2017) | 0.124 | 0.385 | **0.746** | 0.622 |
| Reptile (2018) | 0.124 | 0.384 | 0.595 | 0.712 |
| CNAPs (2019) | 0.140 | 0.418 | 0.676 | 0.552 |
| FEAT (2020) | 0.180 | 0.502 | 0.514 | 0.765 |
| ICL-Causal (2022) | 0.004 | 0.013 | 0.019 | 0.603 |
| ICL-GP (2022) | 0.012 | 0.039 | 0.038 | 0.648 |
| TabPFN (2022) | 0.102 | 0.320 | 0.701 | 0.747 |
| GAP (2023) | 0.136 | 0.419 | 0.613 | 0.702 |
| CAML (2024) | 0.045 | 0.142 | 0.147 | 0.639 |
| AFRT (2025) | 0.122 | 0.377 | 0.627 | 0.705 |
| NearestCentroid | **0.216** | **0.536** | 0.535 | **0.785** |

# O Experiment 9: Semantic and Sub-Semantic Generators

## O.1 Experiment description

Frozen CLIP embeddings could limit the detection of generators whose images look like photographs, because their artefacts are not visible at the semantic level. We call GAN-U, GAN-C, and Inpaint the *sub-semantic* families, because their artefacts lie below the level of the image content. All other families are *semantic*. We regroup the evaluations of Experiment 1 under the sibling policy by this split within each test group. The advantage of a detector is its difference from UnivFD-ZS.

## O.2 Results

Table 40 shows that the advantage over UnivFD-ZS is concentrated on the sub-semantic families. On the unseen siblings, MAML, ProtoNet, Reptile, and GAP gain 0.07–0.08 on the sub-semantic families but at most 0.01 on the semantic ones. The domain-aware detectors gain about as much (0.06), so this advantage comes from a trained head on the CLIP embeddings and is not specific to the support set. The semantic unseen siblings, namely DDPM and DeepFloyd IF of Diff-Px, are the

Table 37: *Experiment 8(a) – Label noise – ACC* as a function of the fraction $\varepsilon$ of inverted support labels, with the breakdown point, the interval of $\varepsilon$ in which a detector falls below UnivFD-ZS. A breakdown point of 0 marks a detector that is below UnivFD-ZS already without noise. AI-GenBench under the implicit (top) and the explicit (bottom) construction. Table 38 shows Chameleon. N/A marks the static detectors, which are not evaluated with noisy support labels.

| Model | $\varepsilon = 0$ | $\varepsilon = 0.125$ | $\varepsilon = 0.25$ | $\varepsilon = 0.5$ | Breakdown $\varepsilon$ falls to zero-shot |
|---|---|---|---|---|---|
| *Context-free* | | | | | |
| PatchCraft (2023) | 0.581 | N/A | N/A | N/A | 0 |
| UnivFD-FT (2023) | 0.837 | N/A | N/A | N/A | 0 |
| UnivFD-ZS (2023) | 0.837 | N/A | N/A | N/A | 0 |
| BSA (2024) | 0.639 | N/A | N/A | N/A | 0 |
| AIDE (2025) | 0.739 | N/A | N/A | N/A | 0 |
| MLEP (2025) | 0.549 | N/A | N/A | N/A | 0 |
| OSD (2025) | 0.802 | N/A | N/A | N/A | 0 |
| *Domain-aware* | | | | | |
| CORAL (2016) | 0.873 | N/A | N/A | N/A | $> 0$ |
| IRM (2019) | 0.866 | N/A | N/A | N/A | $> 0$ |
| GroupDRO (2020) | 0.868 | N/A | N/A | N/A | $> 0$ |
| TENT (2021) | 0.871 | N/A | N/A | N/A | $> 0$ |
| *Context-aware* | | | | | |
| MatchNet (2016) | 0.849 | 0.850 | 0.821 | 0.504 | (0.125, 0.25] |
| MAML (2017) | **0.875** | 0.846 | 0.780 | 0.538 | (0.125, 0.25] |
| ProtoNet (2017) | 0.874 | **0.869** | 0.860 | 0.504 | **(0.25, 0.5]** |
| Reptile (2018) | **0.875** | 0.854 | 0.801 | 0.532 | (0.125, 0.25] |
| CNAPs (2019) | 0.863 | 0.861 | 0.849 | 0.499 | **(0.25, 0.5]** |
| FEAT (2020) | **0.875** | **0.869** | 0.850 | 0.500 | **(0.25, 0.5]** |
| ICL-Causal (2022) | 0.862 | 0.862 | **0.862** | **0.862** | $>$ **0.5** |
| ICL-GP (2022) | 0.858 | 0.858 | 0.859 | 0.859 | $>$ **0.5** |
| TabPFN (2022) | 0.724 | 0.656 | 0.591 | 0.501 | 0 |
| GAP (2023) | **0.875** | 0.846 | 0.777 | 0.542 | (0.125, 0.25] |
| CAML (2024) | 0.847 | 0.848 | 0.848 | 0.847 | $>$ **0.5** |
| AFRT (2025) | 0.857 | 0.826 | 0.767 | 0.500 | (0, 0.125] |
| NearestCentroid | 0.766 | 0.727 | 0.673 | 0.500 | 0 |

| Model | $\varepsilon = 0$ | $\varepsilon = 0.125$ | $\varepsilon = 0.25$ | $\varepsilon = 0.5$ | Breakdown $\varepsilon$ falls to zero-shot |
|---|---|---|---|---|---|
| *Context-free* | | | | | |
| PatchCraft (2023) | 0.579 | N/A | N/A | N/A | 0 |
| UnivFD-FT (2023) | 0.832 | N/A | N/A | N/A | $> 0$ |
| UnivFD-ZS (2023) | 0.830 | N/A | N/A | N/A | 0 |
| BSA (2024) | 0.583 | N/A | N/A | N/A | 0 |
| AIDE (2025) | 0.735 | N/A | N/A | N/A | 0 |
| MLEP (2025) | 0.568 | N/A | N/A | N/A | 0 |
| OSD (2025) | 0.746 | N/A | N/A | N/A | 0 |
| *Domain-aware* | | | | | |
| CORAL (2016) | 0.851 | N/A | N/A | N/A | $> 0$ |
| IRM (2019) | 0.716 | N/A | N/A | N/A | 0 |
| GroupDRO (2020) | 0.795 | N/A | N/A | N/A | 0 |
| TENT (2021) | 0.854 | N/A | N/A | N/A | $> 0$ |
| *Context-aware* | | | | | |
| MatchNet (2016) | 0.838 | 0.823 | 0.771 | 0.629 | (0, 0.125] |
| MAML (2017) | **0.860** | 0.818 | 0.767 | 0.629 | (0, 0.125] |
| ProtoNet (2017) | 0.854 | **0.840** | 0.810 | 0.644 | (0.125, 0.25] |
| Reptile (2018) | **0.860** | 0.819 | 0.769 | 0.642 | (0, 0.125] |
| CNAPs (2019) | 0.841 | 0.830 | 0.809 | 0.648 | (0.125, 0.25] |
| FEAT (2020) | 0.859 | 0.837 | 0.801 | 0.635 | (0.125, 0.25] |
| ICL-Causal (2022) | 0.835 | 0.835 | **0.836** | **0.836** | $>$ **0.5** |
| ICL-GP (2022) | 0.824 | 0.823 | 0.823 | 0.823 | 0 |
| GAP (2023) | **0.860** | 0.819 | 0.771 | 0.632 | (0, 0.125] |
| CAML (2024) | 0.803 | 0.802 | 0.800 | 0.797 | 0 |
| AFRT (2025) | 0.851 | 0.831 | 0.802 | 0.638 | (0.125, 0.25] |
| NearestCentroid | 0.768 | 0.713 | 0.678 | 0.571 | 0 |

hardest group for every detector on CLIP embeddings (0.60–0.74), so CLIP limits the detection of these semantic generators more than that of the sub-semantic ones.

Table 38: *Experiment 8(a) – Label noise on Chameleon – ACC* under the implicit construction, as in Table 37.

| Model | $\varepsilon = 0$ | $\varepsilon = 0.125$ | $\varepsilon = 0.25$ | $\varepsilon = 0.5$ | Breakdown $\varepsilon$ falls to zero-shot |
|---|---|---|---|---|---|
| *Context-free* | | | | | |
| PatchCraft (2023) | 0.582 | N/A | N/A | N/A | 0 |
| UnivFD-FT (2023) | 0.747 | N/A | N/A | N/A | $> 0$ |
| UnivFD-ZS (2023) | 0.626 | N/A | N/A | N/A | 0 |
| BSA (2024) | 0.506 | N/A | N/A | N/A | 0 |
| AIDE (2025) | 0.621 | N/A | N/A | N/A | 0 |
| MLEP (2025) | 0.537 | N/A | N/A | N/A | 0 |
| OSD (2025) | 0.664 | N/A | N/A | N/A | $> 0$ |
| *Domain-aware* | | | | | |
| CORAL (2016) | 0.605 | N/A | N/A | N/A | 0 |
| IRM (2019) | 0.612 | N/A | N/A | N/A | 0 |
| GroupDRO (2020) | 0.621 | N/A | N/A | N/A | 0 |
| TENT (2021) | 0.621 | N/A | N/A | N/A | 0 |
| *Context-aware* | | | | | |
| MatchNet (2016) | 0.625 | 0.626 | 0.626 | 0.499 | 0 |
| MAML (2017) | 0.731 | 0.718 | 0.683 | 0.474 | **(0.25, 0.5]** |
| ProtoNet (2017) | 0.674 | 0.671 | 0.659 | 0.500 | **(0.25, 0.5]** |
| Reptile (2018) | 0.734 | 0.728 | 0.702 | 0.532 | **(0.25, 0.5]** |
| CNAPs (2019) | 0.604 | 0.600 | 0.587 | 0.500 | 0 |
| FEAT (2020) | 0.798 | 0.795 | 0.783 | 0.499 | **(0.25, 0.5]** |
| ICL-Causal (2022) | 0.614 | 0.614 | 0.614 | 0.614 | 0 |
| ICL-GP (2022) | 0.646 | 0.646 | 0.646 | **0.647** | $> \mathbf{0.5}$ |
| TabPFN (2022) | **0.859** | 0.824 | 0.749 | 0.519 | **(0.25, 0.5]** |
| GAP (2023) | 0.728 | 0.716 | 0.682 | 0.483 | **(0.25, 0.5]** |
| CAML (2024) | 0.627 | 0.628 | 0.629 | 0.631 | $> \mathbf{0.5}$ |
| AFRT (2025) | 0.760 | 0.754 | 0.737 | 0.502 | **(0.25, 0.5]** |
| NearestCentroid | 0.848 | **0.840** | **0.820** | 0.502 | **(0.25, 0.5]** |

Table 39: *Experiment 8(b) – Pseudo-labelled support – ACC* with the ground-truth and the pseudo-labelled support set, together with the static baselines and the accuracy of the pseudo-labels. AI-GenBench under the implicit (top) and the explicit (middle) construction, and Chameleon under the implicit construction (bottom).

| Model | Labelled support | Pseudo-labelled support | $\Delta$ (pseudo − labelled) | Static zero-shot | Static fine-tuned | Pseudo-label accuracy |
|---|---|---|---|---|---|---|
| *Context-aware* | | | | | | |
| MatchNet (2016) | 0.849 | 0.808 | -0.041 | **0.837** | **0.837** | **0.838** |
| MAML (2017) | **0.875** | 0.869 | -0.006 | **0.837** | **0.837** | **0.838** |
| ProtoNet (2017) | 0.874 | **0.874** | -0.000 | **0.837** | **0.837** | **0.838** |
| Reptile (2018) | **0.875** | 0.870 | -0.005 | **0.837** | **0.837** | **0.838** |
| CNAPs (2019) | 0.863 | 0.864 | **0.001** | **0.837** | **0.837** | **0.838** |
| FEAT (2020) | **0.875** | 0.864 | -0.011 | **0.837** | **0.837** | **0.838** |
| ICL-Causal (2022) | 0.862 | 0.862 | 0.000 | **0.837** | **0.837** | **0.838** |
| ICL-GP (2022) | 0.858 | 0.858 | -0.000 | **0.837** | **0.837** | **0.838** |
| TabPFN (2022) | 0.724 | 0.669 | -0.055 | **0.837** | **0.837** | **0.838** |
| GAP (2023) | **0.875** | 0.870 | -0.005 | **0.837** | **0.837** | **0.838** |
| CAML (2024) | 0.847 | 0.848 | **0.001** | **0.837** | **0.837** | **0.838** |
| AFRT (2025) | 0.857 | 0.854 | -0.003 | **0.837** | **0.837** | **0.838** |
| NearestCentroid | 0.766 | 0.733 | -0.033 | **0.837** | **0.837** | **0.838** |

| Model | Labelled support | Pseudo-labelled support | $\Delta$ (pseudo − labelled) | Static zero-shot | Static fine-tuned | Pseudo-label accuracy |
|---|---|---|---|---|---|---|
| *Context-aware* | | | | | | |
| MatchNet (2016) | 0.838 | 0.751 | -0.087 | **0.830** | **0.832** | **0.838** |
| MAML (2017) | **0.860** | 0.846 | -0.014 | **0.830** | **0.832** | **0.838** |
| ProtoNet (2017) | 0.854 | 0.845 | -0.008 | **0.830** | **0.832** | **0.838** |
| Reptile (2018) | **0.860** | 0.838 | -0.022 | **0.830** | **0.832** | **0.838** |
| CNAPs (2019) | 0.841 | 0.828 | -0.013 | **0.830** | **0.832** | **0.838** |
| FEAT (2020) | 0.859 | 0.836 | -0.023 | **0.830** | **0.832** | **0.838** |
| ICL-Causal (2022) | 0.835 | 0.835 | 0.000 | **0.830** | **0.832** | **0.838** |
| ICL-GP (2022) | 0.824 | 0.823 | -0.001 | **0.830** | **0.832** | **0.838** |
| GAP (2023) | **0.860** | **0.847** | -0.013 | **0.830** | **0.832** | **0.838** |
| CAML (2024) | 0.803 | 0.808 | **0.005** | **0.830** | **0.832** | **0.838** |
| AFRT (2025) | 0.851 | 0.825 | -0.026 | **0.830** | **0.832** | **0.838** |
| NearestCentroid | 0.768 | 0.682 | -0.086 | **0.830** | **0.832** | **0.838** |

| Model | Labelled support | Pseudo-labelled support | $\Delta$ (pseudo − labelled) | Static zero-shot | Static fine-tuned | Pseudo-label accuracy |
|---|---|---|---|---|---|---|
| *Context-aware* | | | | | | |
| MatchNet (2016) | 0.625 | 0.500 | -0.126 | **0.626** | **0.747** | **0.624** |
| MAML (2017) | 0.731 | 0.598 | -0.132 | **0.626** | **0.747** | **0.624** |
| ProtoNet (2017) | 0.674 | 0.579 | -0.095 | **0.626** | **0.747** | **0.624** |
| Reptile (2018) | 0.734 | 0.587 | -0.147 | **0.626** | **0.747** | **0.624** |
| CNAPs (2019) | 0.604 | 0.604 | -0.000 | **0.626** | **0.747** | **0.624** |
| FEAT (2020) | 0.798 | **0.659** | -0.139 | **0.626** | **0.747** | **0.624** |
| ICL-Causal (2022) | 0.614 | 0.614 | -0.000 | **0.626** | **0.747** | **0.624** |
| ICL-GP (2022) | 0.646 | 0.646 | 0.000 | **0.626** | **0.747** | **0.624** |
| TabPFN (2022) | **0.859** | 0.472 | -0.387 | **0.626** | **0.747** | **0.624** |
| GAP (2023) | 0.728 | 0.600 | -0.129 | **0.626** | **0.747** | **0.624** |
| CAML (2024) | 0.627 | 0.631 | **0.004** | **0.626** | **0.747** | **0.624** |
| AFRT (2025) | 0.760 | 0.644 | -0.116 | **0.626** | **0.747** | **0.624** |
| NearestCentroid | 0.848 | 0.575 | -0.273 | **0.626** | **0.747** | **0.624** |

Table 40: *Experiment 9 – Semantic and sub-semantic generators – ACC* per test group, the advantage over UnivFD-ZS, and the difference between the two groups.

| Model | semantic (seen) | semantic (unseen sibling) | semantic (unseen family) | sub semantic (seen) | sub semantic (unseen sibling) | sub semantic (unseen family) | Adv. vs static semantic (seen) | Adv. vs static semantic (unseen sibling) | Adv. vs static semantic (unseen family) | Adv. vs static sub semantic (seen) | Adv. vs static sub semantic (unseen sibling) | Adv. vs static sub semantic (unseen family) | Δ sem − sub (seen) | Δ sem − sub (unseen sibling) | Δ sem − sub (unseen family) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Context-free* | | | | | | | | | | | | | | | |
| PatchCraft (2023) | 0.615 | 0.524 | 0.603 | 0.540 | 0.547 | 0.633 | -0.255 | -0.189 | -0.231 | -0.340 | -0.263 | -0.270 | **0.075** | -0.023 | -0.030 |
| UnivFD-FT (2023) | 0.856 | 0.695 | 0.829 | 0.875 | 0.828 | 0.901 | -0.015 | -0.018 | -0.004 | -0.005 | 0.017 | -0.002 | -0.019 | -0.132 | -0.072 |
| UnivFD-ZS (2023) | 0.870 | 0.713 | 0.833 | 0.880 | 0.810 | 0.903 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | -0.010 | -0.097 | -0.069 |
| BSA (2024) | 0.612 | 0.668 | 0.702 | 0.595 | 0.614 | 0.572 | -0.259 | -0.045 | -0.131 | -0.285 | -0.197 | -0.331 | 0.016 | 0.055 | 0.130 |
| AIDE (2025) | 0.786 | 0.680 | 0.751 | 0.806 | 0.688 | 0.752 | -0.084 | -0.033 | -0.083 | -0.074 | -0.123 | -0.151 | -0.020 | -0.007 | -0.001 |
| MLEP (2025) | 0.563 | **0.742** | 0.612 | 0.496 | 0.482 | 0.464 | -0.307 | **0.029** | -0.221 | -0.384 | -0.328 | -0.439 | 0.068 | **0.260** | **0.148** |
| OSD (2025) | 0.878 | 0.712 | 0.784 | 0.845 | 0.767 | 0.871 | 0.007 | -0.001 | -0.049 | -0.035 | -0.043 | -0.031 | 0.033 | -0.055 | -0.087 |
| *Domain-aware* | | | | | | | | | | | | | | | |
| CORAL (2016) | 0.914 | 0.739 | 0.845 | 0.917 | 0.874 | 0.933 | 0.044 | 0.026 | 0.012 | 0.037 | 0.064 | 0.030 | -0.003 | -0.135 | -0.088 |
| IRM (2019) | 0.909 | 0.735 | 0.837 | 0.911 | 0.869 | 0.924 | 0.039 | 0.022 | 0.004 | 0.031 | 0.059 | 0.021 | -0.002 | -0.134 | -0.086 |
| GroupDRO (2020) | 0.915 | 0.736 | 0.836 | 0.914 | 0.871 | 0.929 | 0.045 | 0.023 | 0.002 | 0.034 | 0.061 | 0.027 | 0.000 | -0.135 | -0.094 |
| TENT (2021) | **0.923** | 0.739 | 0.833 | **0.926** | 0.872 | 0.938 | **0.052** | 0.026 | -0.000 | **0.046** | 0.062 | 0.035 | -0.004 | -0.133 | -0.105 |
| *Context-aware* | | | | | | | | | | | | | | | |
| MatchNet (2016) | 0.882 | 0.724 | 0.835 | 0.891 | 0.835 | 0.913 | 0.012 | 0.011 | 0.002 | 0.011 | 0.024 | 0.010 | -0.009 | -0.111 | -0.077 |
| MAML (2017) | 0.913 | 0.722 | 0.847 | 0.922 | 0.878 | 0.942 | 0.042 | 0.009 | 0.014 | 0.042 | 0.068 | 0.039 | -0.009 | -0.157 | -0.095 |
| ProtoNet (2017) | 0.908 | 0.713 | 0.845 | 0.922 | **0.887** | 0.930 | 0.038 | 0.000 | 0.011 | 0.042 | **0.076** | 0.028 | -0.014 | -0.173 | -0.086 |
| Reptile (2018) | 0.914 | 0.718 | 0.846 | 0.923 | 0.881 | 0.941 | 0.043 | 0.005 | 0.012 | 0.043 | 0.071 | 0.038 | -0.009 | -0.163 | -0.096 |
| CNAPs (2019) | 0.916 | 0.727 | 0.829 | 0.922 | 0.861 | 0.927 | 0.045 | 0.014 | -0.004 | 0.042 | 0.050 | 0.025 | -0.006 | -0.134 | -0.098 |
| FEAT (2020) | 0.894 | 0.728 | **0.855** | 0.923 | 0.874 | **0.945** | 0.023 | 0.015 | **0.022** | 0.043 | 0.064 | **0.042** | -0.029 | -0.146 | -0.090 |
| ICL-Causal (2022) | 0.911 | 0.732 | 0.824 | 0.916 | 0.865 | 0.937 | 0.040 | 0.019 | -0.010 | 0.036 | 0.054 | 0.034 | -0.005 | -0.133 | -0.114 |
| ICL-GP (2022) | 0.896 | 0.726 | 0.835 | 0.904 | 0.854 | 0.924 | 0.026 | 0.013 | 0.001 | 0.024 | 0.043 | 0.021 | -0.008 | -0.127 | -0.089 |
| TabPFN (2022) | 0.685 | 0.599 | 0.715 | 0.746 | 0.752 | 0.760 | -0.183 | -0.111 | -0.116 | -0.129 | -0.063 | -0.138 | -0.062 | -0.153 | -0.045 |
| GAP (2023) | 0.911 | 0.720 | 0.847 | 0.923 | 0.881 | 0.941 | 0.040 | 0.007 | 0.013 | 0.043 | 0.070 | 0.038 | -0.013 | -0.160 | -0.094 |
| CAML (2024) | 0.878 | 0.723 | 0.832 | 0.889 | 0.837 | 0.909 | 0.008 | 0.010 | -0.002 | 0.009 | 0.027 | 0.007 | -0.011 | -0.114 | -0.078 |
| AFRT (2025) | 0.877 | 0.705 | 0.839 | 0.900 | 0.855 | 0.927 | 0.007 | -0.009 | 0.006 | 0.019 | 0.045 | 0.024 | -0.022 | -0.151 | -0.088 |
| NearestCentroid | 0.729 | 0.607 | 0.757 | 0.801 | 0.789 | 0.817 | -0.139 | -0.103 | -0.074 | -0.074 | -0.025 | -0.082 | -0.073 | -0.182 | -0.060 |