INFORMED PLASTIC ATTRACTORS FOR ADAPTIVE FEW-SHOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Hebbian plasticity enables gradient-free adaptation at deployment, making it attractive for few-shot and continual learning, but designing plasticity rules typically requires black-box optimization because their weight dynamics are not well understood. We introduce *Informed Plastic Attractors* (IPA) that analytically define Hebbian plasticity matrices ($\mathbf{ABCD}$) in correspondence with the Information Bottleneck principle, resulting in unique local weight attractors. From K labeled shots, these attractors adapt the head on a stream of single unlabeled data without gradients, replay or a query batch. We formalise a unique, locally contractive fixed point of the expected centroid mode and sample-wise dissipation of pre-activation energy. Under matched unlabeled-data access, IPA matches or improves nearest-class-mean classifiers, with gains where prototypes are weak, while batch-transductive methods win only with large, balanced query pools; the same plastic head meta-trains bodies that transfer across domains and heads; and in few-shot class-incremental learning with a frozen ResNet-12 it outperforms NCM-r, TENT, TIM, TEEN, ER, and the published NC-FSCIL result on CIFAR-100 without replay or gradient updates, is never below the prototype baseline, and with a frozen DINO backbone ends at 84.04% final-session accuracy on MiniImageNet against 78.17% for ER.

1 INTRODUCTION

Gradient-based optimization has enabled neural networks (NNs) to achieve remarkable performance across a wide range of tasks (He et al., 2016; Finn et al., 2017). However, models trained via gradient descent typically require large amounts of labeled data and converge to static parameter configurations that struggle to adapt to distribution shifts or novel tasks at deployment time (Finn et al., 2017). This is in contrast to biological agents that can continuously integrate new information without explicit supervision or replay.

Hebbian learning provides a biologically plausible model of synaptic plasticity based on co-activation patterns of connected neurons (Hebb, 1949). Most algorithms employ variations of the same rule, differing in how correlations are interpreted and translated into weight updates (Oja, 1982; Soltoggio et al., 2007; van Diggelen et al., 2025). In this generalized form, synaptic strength changes as a function of correlated pre- and post-synaptic activity (Bienenstock et al., 1982), enabling online adaptation through purely local signals without backpropagation. Despite its conceptual simplicity, designing Hebbian plasticity rules remains challenging as online weight dynamics are not well understood, ill-defined adaptation can lead to unbounded weight growth (Oja, 1982; Najarro & Risi, 2020), or result in misalignment with changing task goals.

A general description of Hebbian plasticity can be formulated by a parametric function of presynaptic and postsynaptic neural activation (Niv et al., 2001; Soltoggio et al., 2007):

$$\Delta w_{ij} = \eta \left(A_{ij} i_j + B_{ij} o_i + C_{ij} i_j o_i + D_{ij} \right), \tag{1}$$

where input i_j and output o_i represent the pre- and postsynaptic activations, η denotes the learning rate, and the scalar parameters $\{A, B, C, D\}_{ij}$ control the contribution of different correlation terms between pre- and postsynaptic activation. These parameters are typically distinct for each connection and are optimized using black-box methods, such as evolutionary algorithms (Niv et al., 2001; Soltoggio et al., 2007; Najarro & Risi, 2020; van Diggelen et al., 2025). The use of black-box optimization is due to a limited understanding of how rules affect weight dynamics online.

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.

We propose the design of Hebbian plasticity rules in correspondence with the Information Bottleneck principle (IB) (Tishby et al., 2000; Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017). The IB defines two objectives based on mutual information ($\mathcal{I}$): (1) maximizing $\mathcal{I}(O;Y)$ for task alignment between model output (O) and target (Y), and (2) minimizing $\mathcal{I}(X;O)$ for compression. We use the IB as a design principle for local plasticity, where the IB defines the adaptation rules from K labeled shots, while the Hebbian dynamics provide the label-free mechanism that converges to it on an unlabeled stream. Unlike black-box ABCD optimization, we prove a unique locally stable equilibrium for the expected centroid mode and sample-wise homeostatic dissipation of pre-activation energy. IPA interprets the test-time adaptation of a NN as a dynamical system to define a discriminative force $\mathbf{AD}$, while the activity-dependent $\mathbf{BC}$ terms provide negative feedback to stabilize its magnitude into an attractor.

Our proposed method *Informed Plastic Attractors* (IPA, see Figure 1) defines plasticity rules ($\mathbf{ABCD}$), using K-shot labeled data, that govern weight updates ($\Delta\mathbf{W}$) to align with the IB at deployment from unlabeled data. The expected dynamics converge weight rows towards class-specific input centroids while contrasting against the centroids of competing classes (maximizing $\mathcal{I}(O;Y)$). A self-normalizing term, inspired by the regularization component of the Oja rule (Oja, 1982), suppresses weight growth proportionally to a neuron's own alignment with its class centroid, bounding weight magnitudes and decorrelating redundant input dimensions (minimizing an upper bound on $\mathcal{I}(X;O)$). IPA requires only K-shot labeled data to define $\mathbf{ABCD}$, uses no gradients at deployment, and remains adaptive throughout deployment; the same unmodified rule is applied to an unlabeled stream one sample at a time in all three settings we evaluate.

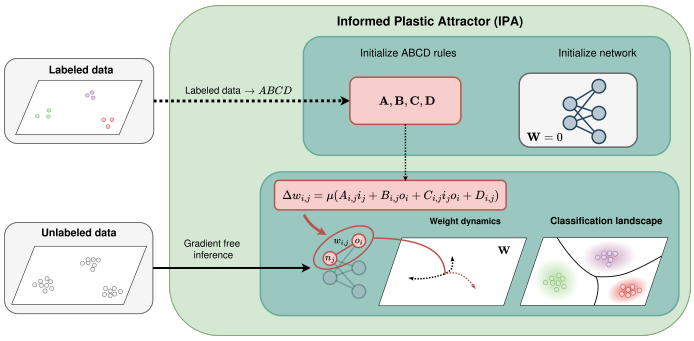


Figure 1: Informed Plastic Attractors (IPA) for few-shot tasks.

Our central contribution is a recipe for designing local Hebbian plasticity rules *analytically*, instead of searching for them with black-box optimization; the evaluations below are evidence that this design principle holds across settings. Concretely, this paper contributes:

- **Hebbian rules from the IB principle, with a local stability guarantee.** We define plasticity matrices ($\mathbf{ABCD}$) from a tractable IB surrogate using K labeled shots, yielding a gradient-free adaptive head for any frozen feature extractor, and prove that the projected expected dynamics along the class-centroid mode have a unique, locally contractive fixed point and that the $\mathbf{BC}$ term dissipates pre-activation energy (Proposition 3.1, Corollary A.1).
- **IPA dynamics in few-shot learning (Section 4.1).** On a 3-Gaussian toy problem, β shapes the information-plane trajectory rather than accuracy; on MNIST and CIFAR-FS under matched unlabeled-data access, IPA matches or outperforms NCM, with gains where prototypes are weak, whereas batch-transductive methods win only with large, balanced query pools.
- **Compatibility with meta-learned representations (Section 4.2).** The same plastic head can be meta-trained through; the resulting bodies transfer across domains (23/32 pairs against ANILr, $p < 0.001$) and across heads (11/16).
- **Test-time adaptation in FSCIL (Section 4.3).** On CIFAR-100, IPA-Approx outperforms NC-FSCIL, ER, and NCM-r without a replay buffer or gradient updates at deployment, and under the identical unlabeled stream also TENT, TIM, and TEEN; with a frozen DINO-ViT-S/8 backbone IPA-DINO reaches AA 86.55% on MiniImageNet without FSCIL-stage training.

2 RELATED WORK

Hebbian learning. Plastic networks adapt at deployment without backpropagation (Najarro & Risi, 2020; Miconi et al., 2018); the central design difficulty is the dynamics induced by plasticity (Bienenstock et al., 1982). Existing approaches either optimize the plasticity rules with evolutionary algorithms (Niv et al., 2001; Soltoggio et al., 2007; Najarro & Risi, 2020; van Diggelen et al., 2025), or train per-connection plasticity by gradient descent (Miconi et al., 2018). These approaches offer limited theoretical insight into synaptic dynamics; we define plasticity rules on the basis of the Information Bottleneck.

Information Bottleneck. The IB principle characterizes representation learning in information-theoretic terms (Tishby et al., 2000; Tishby & Zaslavsky, 2015). The two phenomena — maximizing $\mathcal{I}(O;Y)$ and minimizing $\mathcal{I}(X;O)$ — have been described as a *fitting* phase followed by a *compression* phase (Shwartz-Ziv & Tishby, 2017; Tishby & Zaslavsky, 2015); this compression phase is not universal and depends on the activation function (Saxe et al., 2019). Most closely related to our work, Ma et al. (2020) replace the intractable mutual-information terms with kernel-based surrogates; these estimators require batched data (the empirical HSIC estimator is identically zero at batch size one), whereas IPA's update consumes a single unlabeled sample plus the stored centroids, operating on a 1-sample stream without any labels after the K shots. Daruwalla & Lipasti (2024) derive an IB-based three-factor Hebbian rule whose global signal is accumulated across samples by an auxiliary memory network; IPA instead fixes the adaptation target from the K labeled shots and needs neither an auxiliary network nor a memory of past samples.

Few-shot and class-incremental adaptation. Meta-learning separates a global initialization (MAML; Finn et al., 2017) from a static reusable body with an adaptive head (Raghu et al., 2019; Snell et al., 2017), the structure Meta-IPA adopts. Unlabeled episode data is exploited by transductive FSL over the full query batch (Boudiaf et al., 2020; Ziko et al., 2020), semi-supervised prototype refinement (Ren et al., 2018), test-time entropy minimization (Wang et al., 2021), training-free caches (Zhang et al., 2022), and prototype rectification in FSL and FSCIL (Liu et al., 2020; Wang et al., 2023); FSCIL methods additionally train the base model by gradient descent (Tao et al., 2020; Yang et al., 2023; Zhuang et al., 2023). IPA is closest to the rectification family but refines in *weight space* through a local per-sample update with an explicit contraction condition, and needs no query batch.

3 METHODS

3.1 INFORMED PLASTIC ATTRACTOR WEIGHT DYNAMICS

We consider IPA for a classification task. Hebbian learning is considered between two fully connected layers defined by a weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$, an input vector $I = [i_1, \ldots, i_n]^\top \in \mathbb{R}^n$, and an output vector $O = [o_1, \ldots, o_m]^\top = h(\mathbf{W}I) \in \mathbb{R}^m$, where $h(\cdot)$ is a non-linear, monotonic, and Lipschitz-continuous activation function. Following the formulation of Equation (1) (Niv et al., 2001; Soltoggio et al., 2007), we can express the layer-wise update rule in full matrix form $\Delta\mathbf{W}$. Specifically, we define $\mathbf{I} = [I, \ldots, I]^\top \in \mathbb{R}^{m\times n}$ as the row-wise tiling of the input, and $\mathbf{O} = [O, \ldots, O] \in \mathbb{R}^{m\times n}$ as the column-wise tiling of the output. The complete update rule for all weights $\mathbf{W}$ is given by:

$$\Delta\mathbf{W} = \eta(\mathbf{A} \circ \mathbf{I} + \mathbf{B} \circ \mathbf{O} + \mathbf{C} \circ \mathbf{I} \circ \mathbf{O} + \mathbf{D}), \tag{2}$$

where $\circ$ denotes the Hadamard product. At the core of IPA is the design of $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \in \mathbb{R}^{m\times n}$ aligned with the Information Bottleneck (IB) objective: $\min_{\mathbf{W}} \mathcal{I}(X;O) - \beta\mathcal{I}(O;Y)$, which trades compression against task alignment via the scalar β. For an m-way classification task, the output that maximizes $\mathcal{I}(O;Y)$ under a one-hot proxy satisfies $o_c = 1$, $o_{i\neq c} = 0$ when the input belongs to class c. Achieving this expected behaviour requires aligning each weight row $\mathbf{W}_i$ with the class-conditional input mean $\bar{X}_{c=i}$ while suppressing alignment with $\bar{X}_{c\neq i}$. Conversely, minimizing $\mathcal{I}(X;O)$ requires bounding weight magnitudes; we achieve this through a self-normalizing term inspired by the homeostatic component of the Oja rule (Oja, 1982).

We define the plasticity matrices as follows. The class-discriminative term uses $\mathbf{A}_i = -\mathbf{1}^\top$ and $\mathbf{D}_i = \bar{X}_{c=i}^\top(1 + 1/m) \in \mathbb{R}^{1\times n}$, where $\bar{X}_{c=i}$ denotes the mean input vector for class i (in practice the normalized centroid μ_i of Section 3.3). The compressive term is $\mathbf{B}_i = \mathbf{C}_i = -\beta\,\mathbf{W}_i\bar{X}_{c=i}\,\mathbf{1}^\top$, which

depends only on the alignment $\mathbf{W}_i \bar{X}_{c=i}$ between row $\mathbf{W}_i$ and its assigned class centroid: this is a self-normalizing (homeostatic) gain rather than classical lateral inhibition between neurons. Under a balanced class distribution, the resulting expected updates are (full derivation in Section A.1):

$$\mathbb{E}[\Delta \mathbf{W}_i^{\mathbf{AD}}] = \eta\big(\bar{X}_{c=i}^\top - \tfrac{m-1}{m} \bar{X}_{c\neq i}^\top\big), \quad \mathbb{E}[\Delta \mathbf{W}_i^{\mathbf{BC}}] = -\eta\beta\, \mathbf{W}_i \bar{X}_{c=i}\, \mathbb{E}[(\mathbf{1}^\top + I^\top) o_i]. \tag{3}$$

The $\mathbf{AD}$ term drives each row $\mathbf{W}_i$ towards its class centroid while contrasting against centroids of competing classes, increasing the discriminability of O given Y, a proxy motivated by the $\mathcal{I}(O;Y)$ term. The $\mathbf{BC}$ term contracts $\mathbf{W}_i$ proportionally to its own activity-weighted alignment, bounding weight growth and inducing sparsity along directions of high input variance — a winner-take-all dynamic across redundant inputs to a single output neuron (Oja, 1982). This is the IB-inspired compression proxy for $\mathcal{I}(X;O)$. The two terms have distinct, provable roles (Section 3.2): the expected $\mathbf{AD}$ update is the exact descent direction of a contrastive centroid-alignment loss and $\mathbf{BC}$ dissipates the pre-activation energy of a class-aligned row. The IB motivates combining the two (Section B); IPA does not optimize an IB objective, and the benchmark results depend little on β.

3.2 Informed Plastic Attractors stability analysis

We analyze the expected dynamics along the direction that the labeled construction selects (full derivation in Section A). The instantaneous Jacobian of $\Delta \mathbf{W}_i$ has rank one, but the sign of its eigenvalue depends on the sample (Section A.3). With inputs de-meaned by the class-balanced global mean, $\mathbb{E}[I] = 0$ and $\mathbb{E}[\Delta \mathbf{W}_i^{\mathbf{AD}}] = \eta(\mathbf{D}_i - \mathbb{E}[I]^\top) = \eta(1 + 1/m)\, \mu_i^\top$ exactly: the labeled shots fix the direction of adaptation, and projecting the expected update onto the centroid ray $\mathbf{W}_i = a\, \mu_i^\top$ (unit-norm centroid μ_i) gives scalar dynamics for the magnitude a. Individual samples need not be contractive; the guarantee concerns the expected centroid mode.

Proposition 3.1 (Stable centroid mode)**.** *Let classes be balanced and features be layer-normalized (so $\mathbf{1}^\top I = 0$) and de-meaned by the class-balanced global mean, with unit-norm centroids μ_i (hence $\mathbf{1}^\top \mu_i = 0$). Let h be differentiable, monotone, and sign preserving ($z\,h(z) \geq 0$). Projecting the expected IPA dynamics onto the centroid ray $\mathbf{W}_i = a\, \mu_i^\top$, $a \geq 0$, gives*

$$a_{t+1} = a_t + \eta\big[\kappa - \beta\, a_t\, r_i(a_t)\big], \qquad r_i(a) = \mathbb{E}\big[z\, h(az)\big], \quad z = \mu_i^\top I, \quad \kappa = 1 + 1/m. \tag{4}$$

Suppose $r_i(a) > 0$, $r_i(a) + a\, r_i'(a) > 0$ for $a > 0$, and $a\, r_i(a) \to \infty$ as $a \to \infty$. Then for every $\beta > 0$ there is a unique positive fixed point $a^\star$ with $\beta a^\star r_i(a^\star) = \kappa$, which is locally contractive whenever $0 < \eta\beta\,[r_i(a^\star) + a^\star r_i'(a^\star)] < 2$. For $\beta = 0$ no finite fixed point exists and $a_t = a_0 + t\eta\kappa$ grows linearly.

These conditions hold for Tanh and the identity whenever z is non-degenerate (ReLU: $P(z > 0) > 0$); if the expected $\mathbf{BC}$ component orthogonal to μ_i vanishes, e.g., under symmetry about the centroid ray, $a^\star$ is also a fixed point of the full expected dynamics (Section A.4). Under the same normalization the $\mathbf{BC}$ term also decreases the pre-activation energy $E_I = \frac{1}{2}(\mathbf{W}_i I)^2$ of a class-aligned row ($\mathbf{W}_i \mu_i > 0$) to first order, $\Delta E_I^{\mathbf{BC}} = -\eta\beta\, (\mathbf{W}_i \mu_i)(\mathbf{W}_i I)\, h(\mathbf{W}_i I)\, \|I\|^2 \leq 0$ (Corollary A.1). The support examples thus do two jobs: their centroids determine the discriminative direction through $\mathbf{AD}$, while $\mathbf{BC}$ provides class-conditioned negative feedback on the magnitude of that response, replacing unbounded growth ($\beta = 0$) by a finite stable operating point ($\beta > 0$); for $\beta = 0$ the stream enters every row identically ($\mathbf{W}_i^{(T)} = T\eta\kappa\, \mu_i^\top - \eta \sum_t I_t^\top$) and cancels from all pairwise score differences, leaving a scaled prototype classifier; this is why removing $\mathbf{BC}$ leaves accuracy almost unchanged but removes the attractor and compression behaviour (Section 4.1). The result is local (Section C.14), and under standard bounded-variance stochastic-approximation assumptions a constant η yields an $\mathcal{O}(\sqrt{\eta})$ stationary neighbourhood of $a^\star$.

3.3 Experimental setup

IPA is applied as follows: given K labeled shots of task $\mathcal{T}$, we compute the raw class means $\bar{X}_{\mathcal{T}_i}$, their mean $\bar{X} = m^{-1} \sum_i \bar{X}_{\mathcal{T}_i}$, and the unit-normalized rule centroids $\mu_i = (\bar{X}_{\mathcal{T}_i} - \bar{X})/\|\bar{X}_{\mathcal{T}_i} - \bar{X}\|$; inputs are $I = f_{\theta_b}(x) - \bar{X}$, and we set $\mathbf{A}_i{=}{-}\mathbf{1}^\top$, $\mathbf{D}_i{=}\mu_i^\top(1 + 1/m)$; at deployment we update $\mathbf{B}_i{=}\mathbf{C}_i{=}{-}\beta\, \mathbf{W}_i \mu_i \mathbf{1}^\top$ and apply Equation (2) to unlabeled inputs (full pseudocode in Algorithm 1). Updates are strictly per sample: each update consumes one unlabeled sample plus the stored centroids, at $\mathcal{O}(mn)$ cost per update (the same order as the head's forward pass), with no optimizer state, replay

buffer, or backward graph. Adaptation always completes before evaluation, after which the weights are frozen; in all experiments the adaptation data is disjoint from the evaluation data. Features are layer-normalized (raw input only in the 2-D toy problem). We evaluate IPA in three stages.

(i) IPA dynamics in few-shot learning (Section 4.1). A toy Gaussian classification problem sheds light on the effects of β and output activations on weight dynamics. Mutual information is estimated following Kolchinsky & Tracey (2017) (Section C). We then evaluate IPA with several static feature extractors f_{θ_b} on MNIST (LeCun et al., 1998) and CIFAR-FS, the few-shot split of CIFAR-100 (Krizhevsky, 2009): random MLPs, random CNNs, pre-trained AlexNet (Krizhevsky et al., 2012), ResNet-18 (He et al., 2016), and DINO-ViT-S/8 (Caron et al., 2021). All methods (NCM, TENT, LaplacianShot, TIM) receive the same frozen features and the same unlabeled pool.

(ii) Meta-IPA feature analysis (Section 4.2). Because IPA is fully differentiable, f_{θ_b} (a CNN) is optimized via a meta-learning loop to produce embeddings well-suited for Hebbian attractor estimation, and compared against MAML (Finn et al., 2017) and ANILr, a variant of ANIL (Raghu et al., 2019) that reinitializes the head at every inner loop (Algorithm 5).

(iii) Continual few-shot learning (Section 4.3). A class-incremental extension of IPA on CIFAR-100 and MiniImageNet evaluates gradient-free adaptation over $M = 8$ incremental sessions of new classes (5-shot), following the FSCIL protocol of Tao et al. (2020), with a frozen ResNet-12 pretrained on the base classes and a frozen DINO-ViT-S/8 without any FSCIL-stage fine-tuning (IPA-DINO); baselines receive the identical unlabeled stream.

4 Experimental Results & Discussion

4.1 IPA dynamics in few-shot learning

On a 3-class 2D Gaussian problem, all β settings reach 99.99% accuracy: when class centroids are well separated, β primarily shapes the information-plane trajectory rather than final accuracy (Figures 6 and 7). Varying β from 0 to 1 transitions IPA from a nearest-class-mean classifier (β=0, equivalent to a prototypical classifier, or NCM on the same centroids) to one exhibiting IB compression: the $\mathbf{BC}$ self-normalizing term suppresses co-activation between classes, producing curvature in the information plane characteristic of the fitting-then-compression dynamic (Shwartz-Ziv & Tishby, 2017; Tishby & Zaslavsky, 2015). Figure 2 shows the dynamics for the setting used in the remainder of this work ($\beta = 1$, Tanh). Figure 6 shows that the output distribution spreads more evenly with lower $\beta = 0$, i.e., more co-activity between classes. This same compressive effect shows a positive effect on MNIST accuracy with AlexNet features — $\beta = 1$ marginally improves accuracy over $\beta = 0.1$ at every shot count (Table 5). We therefore use $\beta = 1$ in all subsequent experiments; an $\mathbf{ABCD}$ rule evolved with CMA-ES reaches the same accuracy at twice the information cost (Section C.17).

With a frozen feature extractor f_{θ_b}, the class centroid $\bar{X}_{\mathcal{T}_i} = f_{\theta_b}(\mathcal{D}_{\mathcal{T}_i})$ defines the IPA rules in feature space, similar to NCM/prototypical networks classifier that obtain centroids for classification (Mensink et al., 2013; Snell et al., 2017), but governs an online weight update rather than a fixed similarity metric. We evaluate $K \in \{5, 10, 50\}$ on MNIST (LeCun et al., 1998) (10-way; random MLP_{L2W512} and Conv_{L5W512}, AlexNet (Krizhevsky et al., 2012)) and CIFAR-FS (5-way; AlexNet, ResNet-18 (He et al., 2016), DINO-ViT-S/8 (Caron et al., 2021)) with $\eta = 10^{-5}$, $\beta = 1$, and $N = 30$ episodes, where NCM, TENT, LaplacianShot, and TIM receive the same frozen features and the same balanced unlabeled pool, disjoint from the evaluated samples (Figure 8; backbone and augmentation ablations in Tables 2 and 3, learning curves in Figure 9). IPA outperforms NCM on the random Conv backbone at every K and on AlexNet at low K ($p < 0.05$ in 7/18 settings, by at most 0.8 points), and matches NCM on ResNet-18 and DINO: Hebbian adaptation refines weak prototypes and is neutral when prototypes are strong. TIM and LaplacianShot are up to 9 and 4 points more accurate than NCM on balanced query pools, as they infer jointly over the full batch; under a 5:1 pool imbalance TIM's margin shrinks to $+2.9$ (ResNet-18) and $+1.2$ (DINO), and at 25 unlabeled samples its margin over IPA is $+1.0$ and $+0.3$ (Figure 14), while IPA is flat across pool budgets as it consumes one sample at a time.

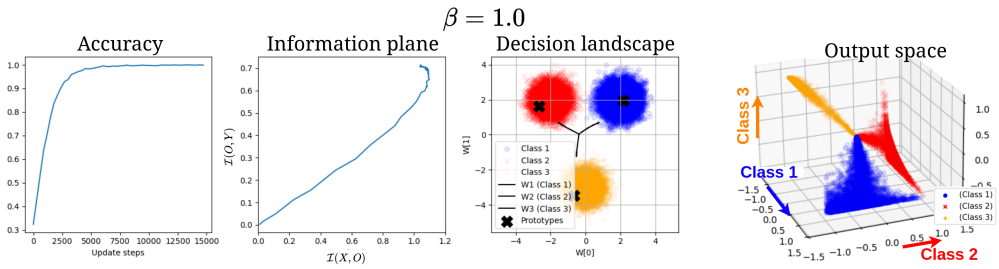


Figure 2: Weight dynamics of IPA during the 2D-Gaussians-3 task with the setting used throughout this work ($\beta = 1$, Tanh). First column: accuracy during optimization. Second column: information plane progression. Third column: input space X with class distributions, one-shot prototype $\bar{X}_i$, and weight trajectory $\mathbf{W}_i$. Fourth column: output space Y; arrows indicate per-class output Y_i. Full β sweep in Figure 6.

4.2 IPA META-LEARNED FEATURE ANALYSIS

As an extension, we meta-train a CNN body f_{θ_b} so that its embeddings are well-suited for IPA attractor estimation. At every meta-step we re-initialize a fresh IPA head ($\mathbf{W} = 0$) and run $n_{\text{inner}} = 100$ Hebbian steps with $lr_{\text{inner}} = 10^{-3}$. Unlike MAML and ANILr, we do not back-propagate through the inner loop: the meta-gradient is computed only at the end of the inner trajectory, which keeps the computational graph memory-efficient and decouples body training from inner-loop optimization (Algorithm 3). During meta-training, the inner Hebbian loop adapts on unlabeled episode images drawn separately from the evaluated queries (no labels are accessed); at evaluation, adaptation uses the support embeddings only, so no evaluated sample is ever used for adaptation. The protocol is therefore semi-supervised rather than transductive (Ren et al., 2018): compared to MAML and ANILr, which adapt only on the labeled support set, the remaining asymmetry is access to extra unlabeled episode data during meta-training. Bodies are trained on MNIST 5-way, Omniglot 20-way (Lake et al., 2015), CUB200 20-way (Wah et al., 2011), and MiniImageNet 5-way (Vinyals et al., 2016), and cross-validated on the same datasets plus TieredImageNet (Ren et al., 2018). Based on the hyperparameter sweep in Figure 10 we use a four-layer CNN with $(64, 3)$ feature maps and $lr_{\text{meta}} = 10^{-2}$; Meta-IPA uses 100 inner and 100 meta-steps against 5 and 10,000 for MAML and ANILr (Table 6), i.e., each method's own protocol. Table 7 reports the same-domain results on Omniglot, CUB200, and MiniImageNet (MNIST 5-way $\geq 99.9\%$ for all methods). IPA outperforms both MAML and ANILr in $4/6$ reported settings, including MiniImageNet 1- and 5-shot tasks. The two losses (Omniglot 20-way 5-shot; CUB200 20-way 1-shot) are consistent with IPA's centroid-quality dependence: at high K on Omniglot characters, ANILr's gradient inner-loop only slightly edges Hebbian adaptation, while at $K = 1$ on CUB200 the centroid estimate $\bar{X}_{\mathcal{T}_i}$ is too noisy to drive the attractor effectively, which might result in overfitting (see also Figure 11). A soft-k-means ProtoNet (Ren et al., 2018) given strictly more unlabeled data remains well below Meta-IPA (Section C.16).

Out-of-distribution adaptation. Both IPA and ANILr reset their heads at every inner loop while keeping their bodies meta-trained, which makes them directly comparable across tasks of varying class counts. Vanilla ANIL and MAML cannot participate because their meta-trained heads have a fixed output dimensionality; an adapted MAML comparison that rescales the source task to match the target appears in Figure 12. Concretely, we transfer (e.g.) a model trained on MNIST 5-way to Omniglot 20-way by re-initializing a head of size 20. Figure 3 (left) reports the resulting transfer matrix at $K \in \{1, 5\}$. IPA significantly outperforms ANILr in 23/32 source-target pairs and is outperformed in 5/32; the aggregated effect is significant at $p < 0.001$, below the Bonferroni-corrected threshold $0.05/32$. The exceptions cluster on MNIST-trained sources, where noisy inner loops reduce the meta-learning signal (Figure 11), limiting the final performance of the backbone. The MiniImageNet-trained IPA backbone is the most domain-robust (last row of Figure 3, left), and although TieredImageNet targets are the hardest for all methods, IPA re-adapts significantly better than ANILr.

Generalizing feature adaptation. To separate the contribution of the head from the body, we cross-swap meta-trained heads f_h and bodies f_{θ_b} (Figure 3, right). Here a *head* denotes the adaptation strategy rather than a weight matrix: for MAML the meta-trained head weights, for ANILr the gradient

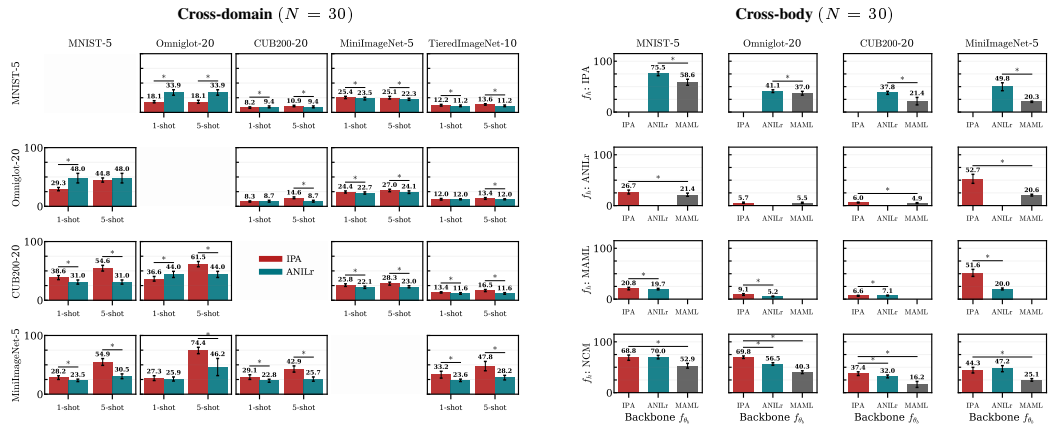


Figure 3: Out-of-distribution generalization of IPA vs. MAML and ANILr ($N = 30$, * denotes $p < 0.05$). **Left (Cross-domain):** accuracy after transferring a meta-trained model from the source dataset (row) to the target dataset (column), at 1- and 5-shot. **Right (Cross-body):** accuracy when each meta-trained head f_h (rows) is paired with each meta-trained body f_{θ_b} (columns) at 5-shot.

inner loop from a random re-initialization, and for IPA the $\mathbf{ABCD}$ constructor — the fixed mapping from support centroids to plasticity matrices, together with its meta-learned inner hyperparameters. The resulting $\mathbf{ABCD}$ values are episode-conditional, but the constructor is episode-independent; per episode it is instantiated from the support centroids in the paired body's feature space, with $\mathbf{W} = 0$. The swap therefore tests the compatibility of an adaptation mechanism with a foreign feature extractor. MAML backbones — trained jointly with their head — generalize poorly when paired with foreign heads, re-emphasizing the distinction between learning a global initialization and a reusable feature extractor (Raghu et al., 2019). ANILr heads transfer worst across foreign backbones, consistent with gradient inner-loops adapting tightly to body-specific feature statistics. The IPA head transfers significantly better across all backbones, while the IPA backbone is the most reusable: pairing any head with the IPA body significantly outperforms pairing the same head with other backbones in 11/16 settings ($p < 0.05$). Together, these results suggest that IPA's contractive head selects features that are class-discriminative rather than head-specific, making the meta-trained body more transferable.

4.3 IPA TEST-TIME ADAPTATION FOR FSCIL

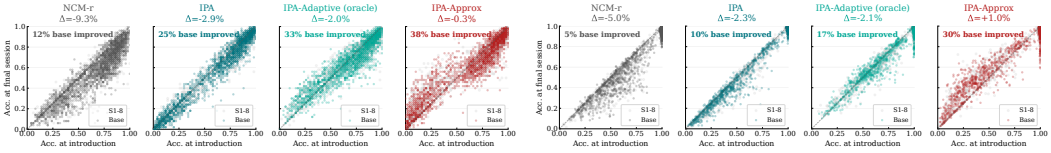
We test three IPA variants: **IPA** uses the same Hebbian adaptation as before, based on K-shot estimates of class prototypes $\bar{X}_{\mathcal{T}_i} = f_{\theta_b}(\mathcal{D}_{\mathcal{T}_i})$; **IPA-Approx** integrates its prediction to refine its prototype at test time; **IPA-Adaptive** is an oracle method that also updates its prototypes based on true labels (an upper bound on IPA-Approx performance). IPA-Approx updates every class mean $\bar{X}_{\mathcal{T}_c}$ online as a confidence-weighted running mean, with weights $p = \mathrm{softmax}(\mathbf{W}I/\tau)$ from the pre-update head ($\tau = 0.25$; Algorithm 2). At each session, 50% of the new test images per class are streamed (unlabeled) to IPA, while the remaining 50% is used to evaluate accuracy on *all* classes seen so far. This split simulates real-world deployment with K-shot labeled data to estimate $\bar{X}_{\mathcal{T}_i}$ and Hebbian adaptation with unlabeled data at test time. Different split variants (25/75; held-out *training* data as the stream, condition C, so that no test image is ever seen during adaptation; evaluating during the updates) change AA by at most 2.1 points for IPA-Approx (Section C.12 and Table 14), so the gains do not rely on test-distribution access; we report condition C alongside the main protocol. We additionally present IPA-Approx on a DINO-ViT-S/8 backbone without fine-tuning (IPA-DINO).

The class-incremental setting is as follows: we use CIFAR-100 and MiniImageNet and train a 60-class base session followed by $M = 8$ incremental sessions of 5 new classes each (5-shot), following the FSCIL protocol of Tao et al. (2020). A ResNet-12 backbone is pretrained on the base classes with cross-entropy (linear warmup + cosine schedule) and frozen throughout all sessions. Initial- and final-session accuracy (S0, S8), the average accuracy across sessions (AA), and the performance drop PD = S0 − S8 are reported in Table 1. We compare against gradient-free NCM and NCM-r, which applies the same running-mean refinement to the same stream, so that IPA-Approx vs. NCM-r isolates the plastic head; and gradient-based baselines Finetune, EWC (Kirkpatrick et al., 2017),

Table 1: FSCIL on CIFAR-100 (left) and MiniImageNet (right) with a frozen **ResNet-12** backbone (frozen **DINO-ViT-S/8** for IPA-DINO). S0, S8: accuracy (%) at the base and final sessions; AA: mean over sessions; PD=S0−S8. (o) oracle upper bounds; bold: significantly best non-oracle method (paired t-tests, $p < 0.05$); † gradient-based baselines; ‡ adaptation methods run under the identical unlabeled stream, frozen features, and adaptation budget as IPA-Approx (complete matched-access grid in Table 15); (C) adaptation on held-out training data instead of the test stream.

CIFAR-100	S0	S8	AA	PD
IPA	81.69	56.71	67.63	24.97
IPA-Approx	81.68	58.01	**68.27**	23.67
IPA-Approx (C)	81.68	58.51	68.61	23.18
IPA-Adaptive (o)	81.70	64.05	71.76	17.65
NCM	81.59	56.13	66.78	25.46
NCM-r	81.05	53.14	63.66	27.91
TENT‡	76.00	39.52	55.52	36.48
TIM‡	81.02	54.63	66.29	26.38
TEEN‡	82.15	55.56	66.66	26.59
Finetune†	83.17	3.95	14.66	79.22
EWC†	83.17	3.94	14.66	79.23
ER†	83.17	56.44	67.81	26.73
Joint† (o)	83.17	66.92	72.65	16.25
IPA-DINO$_{(Approx)}$	76.43	64.74	69.65	11.69

MiniImageNet	S0	S8	AA	PD
IPA	92.06	69.02	79.48	23.04
IPA-Approx	91.95	70.39	**80.22**	21.56
IPA-Approx (C)	91.92	70.39	80.21	21.54
IPA-Adaptive (o)	92.04	74.99	82.91	17.05
NCM	92.21	69.68	80.10	22.52
NCM-r	92.01	68.71	79.63	23.30
TENT‡	86.26	48.49	66.42	37.77
TIM‡	91.61	69.08	80.03	22.53
TEEN‡	92.45	69.49	80.19	22.96
Finetune†	83.84	4.26	14.83	79.58
EWC†	83.84	4.26	14.85	79.58
ER†	83.84	67.33	77.35	16.51
Joint† (o)	83.84	92.29	93.24	-8.45
IPA-DINO$_{(Approx)}$	89.78	84.04	86.55	5.74



(a) CIFAR-100, ResNet-12 backbone

(b) MiniImageNet, ResNet-12 backbone

Figure 4: Per-class accuracy at class introduction (x-axis) versus the final session (y-axis), ResNet-12. Points above the diagonal are classes whose accuracy *improves* during deployment (positive backward transfer): IPA-Approx improves 38% (CIFAR-100) and 30% (MiniImageNet) of base classes and tracks the IPA-Adaptive oracle, whereas NCM-r improves only 12%. The effect is absent in the gradient-based continual learners.

ER (Chaudhry et al., 2019), and Joint (upper bound). Under the identical 50% unlabeled stream we also run TENT (Wang et al., 2021), TIM (Boudiaf et al., 2020), and TEEN (Wang et al., 2023). Published numbers for NC-FSCIL (Yang et al., 2023), TOPIC (Tao et al., 2020), and GKEAL (Zhuang et al., 2023) on the same protocol are listed in Table 12, and full S0–S8 AA±95% CI in Section C.10.

Matched-access results. On CIFAR-100, IPA-Approx achieves 68.27% AA, above ER (20-shot, 67.81%) without any replay buffer and above the published NC-FSCIL result (67.50%, ResNet-18 backbone), while IPA-Adaptive (oracle) approaches the Joint upper bound (71.76% vs. 72.65%). The performance drop is also smaller for IPA-Approx (23.67%) than for ER (26.73%) and NCM (25.46%), indicating better class retention. Under the identical unlabeled stream, IPA-Approx also outperforms TEEN (66.66%), TIM (66.29%), and TENT (55.52%), is the only adaptation method never below NCM (Table 15), and gives the same result when adapting on held-out training data only (C). On MiniImageNet, IPA-Approx also leads the non-oracle methods with AA 80.22%, statistically tied with TEEN ($p = 0.57$) and only marginally ahead of NCM (+0.12, $p = 0.016$). IPA-DINO reaches AA 69.65% (CIFAR-100) and 86.55% (MiniImageNet) without any FSCIL-stage backbone fine-tuning; on CIFAR-100 it exceeds ER, while on MiniImageNet ER (replay buffer and gradient updates) achieves higher AA (87.67%) but ends at 78.17% final-session accuracy against 84.04% for IPA-DINO. DINO is self-supervised on ImageNet-1k, of which MiniImageNet is a subset, so absolute DINO numbers are an off-the-shelf feature study; all methods share the same frozen backbone (Section C.11). The initialization–adaptation grid in Section C.14 (Table 16) separates the information supplied by the K-shot construction from that acquired online: prototype initialization is already strong (67.63%) and IPA consistently improves it (68.25%), whereas IPA recovers near-prototype accuracy from $\mathbf{W} = 0$ (66.52%) but not from a random initialization (10.41%), in line with the local, centroid-aligned contraction of Section 3.2.

Deployment cost. A single streaming pass (predict, update once, never revisit) matches the multi-pass protocol exactly (68.26 ± 0.21 AA in both), so the gains need no iterative optimization over a stored pool; one update costs 619 µs on CPU with 250 KiB of state against 23.4 MiB for the ER buffer (Section C.15).

Positive backward transfer. Interestingly, Figure 4 shows per-class accuracy at class introduction versus the final session, where 38% (CIFAR-100, left) and 30% (MiniImageNet, right) of base classes *individually improve* under IPA-Approx (red), with an average $\Delta = -0.5\%, +1.0\%$. This is in contrast to the accuracy decline typically seen during FSCIL as classes are added, e.g., in NCM-r (gray, average accuracy drop of $\Delta = -9.3\%, -5.0\%$). This *positive backward transfer* emerges from the Hebbian centroid update: every unlabeled inference step refines $\bar{X}_{\mathcal{T}_i}$, which allows for better classification when new classes are introduced. We identify three dynamics, 1) when not integrating unlabeled data $\bar{X}_{\mathcal{T}_i}$ is sub-optimal but does not degrade with new sessions (IPA condition dark teal); 2) With perfect integration of unlabeled data $\bar{X}_{\mathcal{T}_i}$ is optimal and Hebbian adaptation maintains old-class accuracy with new sessions (IPA-Adaptive, light teal); 3) Most interestingly, when integration of unlabeled data is approximate $\bar{X}_{\mathcal{T}_i}$ is sub-optimal and Hebbian adaptation can improve the estimate with new sessions (IPA-Approx, red).

Robustness under imbalance, noise, and drift. IPA-Approx degrades gracefully when the balanced-class and stationarity assumptions are violated (Table 17). A 10:1 imbalanced adaptation stream costs 0.5 AA and a class-blockwise non-stationary stream 0.03; with a single support shot per new class the margin over NCM holds at $+1.3$, and under prototype noise at half the feature standard deviation it grows to $+1.9$, as adaptation partially repairs corrupted centroids (Section C.18). Under drifting class means (Figure 5) the confidence-weighted running mean tracks the drift where the static prototype classifier collapses (86.1% vs. 49.3% after a 60° rotation), and an EMA of the prototypes restores near-complete tracking (98.7%).

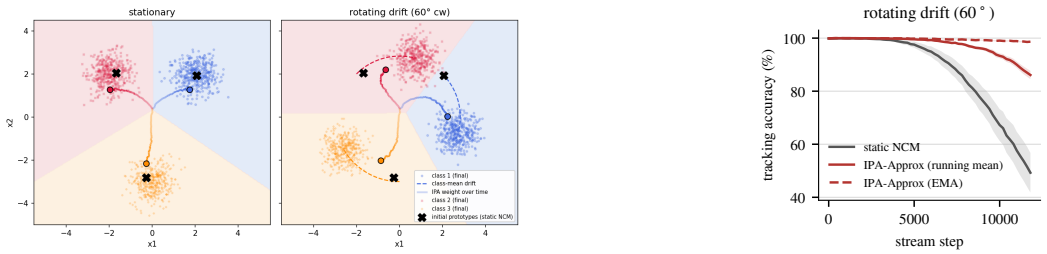


Figure 5: IPA-Approx under class-mean drift on the 3-Gaussian toy problem. Left: weight-row trajectories (from $\mathbf{W} = 0$) and final decision regions for a stationary stream and a 60° rotation of the class means (dashed: class-mean path; crosses: initial prototypes). Right: accuracy under the rotating drift (20 seeds, mean ± 95% CI) with a running mean or an EMA ($\alpha = 0.01$) of the prototypes.

5 Conclusion

Informed Plastic Attractors (IPA) define Hebbian plasticity matrices that result in a gradient-free adaptive head, based on a tractable IB-inspired surrogate: the homeostatic term shapes the dynamics, while the performance gains arise from support-conditioned plastic refinement. IPA matches or outperforms prototype classifiers under matched unlabeled-data access, produces more transferable bodies than MAML and ANILr under its own meta-training protocol, and on FSCIL (ResNet-12 backbone) outperforms NC-FSCIL, ER, TENT, TIM, TEEN, and NCM with refinement on CIFAR-100 without any replay buffer or gradient updates at deployment. IPA-DINO additionally reaches AA 86.55% on MiniImageNet without backbone training, ending at 84.04% against 78.17% for ER. IPA depends on accurate class centroid estimates, batch-transductive inference remains more accurate when a large balanced query batch is available, and regression tasks are future work.

AI use statement

Generative AI tools were used for writing, experimental data management, literature search, and drafting and editing manuscript text. The authors checked the final claims, mathematical statements,

reported numbers, and citations against the implementation, underlying experiments, and cited sources, and take responsibility for the final submission.

Reproducibility statement

The IPA update is given in closed form in Equation (2) with the plasticity matrices of Section 3.1, and as pseudocode in Algorithm 1 (Meta-IPA, MAML, and ANILr in Algorithms 3 to 5). The assumptions and full derivations behind the stability results are given in Sections A and B. Experimental settings are stated in the experimental setup and in Section C (toy problems, random backbones, augmentation), Table 6 (meta-learning hyperparameters), and Sections C.10, C.12 and C.13 (FSCIL protocol, data splits, and matched baselines); all reported numbers are averages over 30 repetitions or episodes with the stated dispersion. Source code and the scripts that regenerate every table and figure will be released upon publication. Codebase, checkpoints and data will be made available after acceptance.

References

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

Elie L Bienenstock, Leon N Cooper, and Paul W Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2 (1):32–48, 1982.

Malik Boudiaf, Imtiaz Ziko, Jérôme Rony, Jose Dolz, Pablo Piantanida, and Ismail Ben Ayed. Information maximization for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2445–2457, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021. URL `https://arxiv.org/abs/2104.14294`.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. In *ICML Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019.

Kyle Daruwalla and Mikko Lipasti. Information bottleneck-based Hebbian learning rule naturally ties working memory and synaptic updates. *Frontiers in Computational Neuroscience*, 18:1240348, 2024.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. John Wiley & Sons, New York, 1949.

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

Artemy Kolchinsky and Brendan D Tracey. Estimating mixture entropy with pairwise distances. *Entropy*, 19(7):361, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. In *European Conference on Computer Vision*, pp. 741–756, 2020.

Wan-Duo Kurt Ma, JP Lewis, and W Bastiaan Kleijn. The hsic bottleneck: Deep learning without back-propagation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5085–5092, 2020.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2624–2637, 2013. doi: 10.1109/TPAMI.2013.83.

Thomas Miconi, Kenneth Stanley, and Jeff Clune. Differentiable plasticity: training plastic neural networks with backpropagation. In *International Conference on Machine Learning*, pp. 3559–3568. PMLR, 2018.

Elias Najarro and Sebastian Risi. Meta-learning through hebbian plasticity in random networks. *Advances in Neural Information Processing Systems*, 33:20719–20731, 2020.

Yael Niv, Daphna Joel, Isaac Meilijson, and Eytan Ruppin. Evolution of reinforcement learning in uncertain environments: Emergence of risk-aversion and matching. In *European Conference on Artificial Life*, pp. 252–261. Springer, 2001.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations (ICLR)*, 2018.

Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Andrea Soltoggio, Peter Durr, Claudio Mattiussi, and Dario Floreano. Evolving neuromodulatory topologies for reinforcement learning-like problems. In *2007 IEEE Congress on evolutionary computation*, pp. 2471–2478. IEEE, 2007.

Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12183–12192, 2020.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pp. 1–5. Ieee, 2015.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Fuda van Diggelen, Tugay Alperen Karagüzel, Andres Garcia Rincon, AE Eiben, Dario Floreano, and Eliseo Ferrante. Emergent heterogeneous swarm control through hebbian learning. *arXiv preprint arXiv:2507.11566*, 2025.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, and Koray Kavukcuoglu. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200 dataset. Technical report, California Institute of Technology, 2011.

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*, 2021.

Qi-Wei Wang, Da-Wei Zhou, Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. Few-shot class-incremental learning via training-free prototype calibration. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision*, pp. 493–510, 2022.

Huiping Zhuang, Zhenyu Weng, Run He, Zhiping Lin, and Ziqian Zeng. GKEAL: Gaussian kernel embedded analytic learning for few-shot class incremental task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. Laplacian regularized few-shot learning. In *International Conference on Machine Learning*, pp. 11660–11670, 2020.

A FORMAL ANALYSIS OF ABCD

Let us recall our task setting. We consider a neural network with Hebbian adaptation that predicts m classes using n inputs on an input vector $I = [i_1, \ldots, i_n]^\top \in \mathbb{R}^n$, fully connected by a weight matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$ with output vector $O = [o_1, \ldots, o_m]^\top = h(\mathbf{W}I) \in \mathbb{R}^m$. The adaptation rule (Equation (1)) is defined for all weights $\mathbf{W}_i \in \mathbb{R}^n$ connecting to a single output neuron o_i as:

$$\Delta \mathbf{W}_i = \eta \left(\mathbf{A}_i \circ I^\top + \mathbf{B}_i o_i + \mathbf{C}_i \circ I^\top o_i + \mathbf{D}_i \right) \tag{5}$$

where η is the learning rate and $\circ$ the Hadamard product.

A.1 EXPECTED VALUE OF THE UPDATE RULE

Under a balanced class distribution ($P(y = i) = 1/m$), the expected update for row i of the weight matrix decomposes as

$$\begin{aligned}
\mathbb{E}[\Delta \mathbf{W}_i^{\mathbf{AD}}] &= \eta \left(\mathbf{D}_i + \mathbf{A}_i \circ \mathbb{E}[I^\top] \right) \\
&= \eta \left(\bar{X}_{c=i}^\top \Big(1 + \tfrac{1}{m} \Big) - \left(\tfrac{1}{m} \bar{X}_{c=i}^\top + \tfrac{m-1}{m} \bar{X}_{c \neq i}^\top \right) \right) \\
&= \eta \left(\bar{X}_{c=i}^\top - \tfrac{m-1}{m} \bar{X}_{c \neq i}^\top \right), \\
\mathbb{E}[\Delta \mathbf{W}_i^{\mathbf{BC}}] &= \eta \, \mathbb{E}\big[\mathbf{B}_i o_i + \mathbf{C}_i \circ I^\top o_i \big] \\
&= -\eta \beta \, \mathbf{W}_i \bar{X}_{c=i} \, \mathbb{E}\big[(\mathbf{1}^\top + I^\top) o_i \big],
\end{aligned} \tag{6}$$

where the second line uses $\mathbf{B}_i = \mathbf{C}_i = -\beta \, \mathbf{W}_i \bar{X}_{c=i} \, \mathbf{1}^\top$ (a scalar gain $\mathbf{W}_i \bar{X}_{c=i} \in \mathbb{R}$ multiplied by the row vector $\mathbf{1}^\top$). Each row $\mathbf{W}_i$ is therefore expected to drift towards a contrastive class-centroid direction $\bar{X}_{c=i}^\top - \frac{m-1}{m} \bar{X}_{c \neq i}^\top$, while the $\mathbf{BC}$ term applies a self-normalizing pull-back proportional to the scalar alignment $\mathbf{W}_i \bar{X}_{c=i}$ and to the activity o_i. As weights start at $\mathbf{W} = 0$, the $\mathbf{BC}$ term vanishes initially and grows monotonically with alignment, producing the fitting-then-compression dynamic visible in the information plane (Section 4.1). In the implementation the rule uses the de-meaned, unit-normalized centroids μ_i ($\mathbf{D}_i = \kappa \mu_i^\top$, $\kappa = 1 + 1/m$) and inputs de-meaned by the class-balanced global mean, so $\mathbb{E}[I] = 0$ and directly $\mathbb{E}[\Delta \mathbf{W}_i^{\mathbf{AD}}] = \eta(\mathbf{D}_i + \mathbf{A}_i \circ \mathbb{E}[I^\top]) = \eta \kappa \, \mu_i^\top$: the expected drift is exactly along the class-centroid ray, without requiring the normalized centroids to sum to zero.

A.2 JACOBIAN DERIVATION

We examine the *instantaneous* Jacobian $\mathbf{J} = \partial \Delta \mathbf{W}_i / \partial \mathbf{W}_i$ at a fixed input I. Because $\mathbf{A}_i$ and $\mathbf{D}_i$ do not depend on $\mathbf{W}_i$, the Jacobian is governed by the $\mathbf{BC}$ term and the activation h. With $\mathbf{B}_i = \mathbf{C}_i = -\beta \, \mathbf{W}_i \bar{X}_{c=i} \, \mathbf{1}^\top$ (an explicit linear function of $\mathbf{W}_i$), $\mathbf{B}_i + \mathbf{C}_i \circ I^\top = -\beta \, \mathbf{W}_i \bar{X}_{c=i} \, (\mathbf{1}^\top + I^\top)$. Substituting and applying the chain rule with h' denoting the derivative of h:

$$\mathbf{J} = -\eta \beta \Big[h'(\mathbf{W}_i I) \, (\mathbf{W}_i \bar{X}_{c=i}) \, I \otimes (\mathbf{1}^\top + I^\top) + h(\mathbf{W}_i I) \, \bar{X}_{c=i} \otimes (\mathbf{1}^\top + I^\top) \Big], \tag{7}$$

where $\mathbf{u} \otimes \mathbf{v}^\top = \mathbf{u}\mathbf{v}^\top$ is the outer product. Both terms share the factor $(\mathbf{1}^\top + I^\top)$ on the right; the formulation above is regular (no singularity at $\mathbf{W}_i \bar{X}_{c=i} = 0$) and reduces to the trace expression in Equation (8) below.

A.3 EIGENVALUE ANALYSIS AND STABILITY

The two summands of Equation (7) are both outer products of the form $\mathbf{u} \otimes (\mathbf{1}^\top + I^\top)$, so the instantaneous Jacobian $\mathbf{J}$ is a rank-1 matrix and has a single non-zero eigenvalue λ_i equal to its trace:

$$\lambda_i = \mathrm{Tr}(\mathbf{J}) = -\eta \beta \left[h'(\mathbf{W}_i I) \, (\mathbf{W}_i \bar{X}_{c=i}) \left(\sum_{j=1}^{n} I_j + \sum_{j=1}^{n} I_j^2 \right) + h(\mathbf{W}_i I) \left(\sum_{j=1}^{n} \bar{X}_{c=i,j} + \sum_{j=1}^{n} I_j \bar{X}_{c=i,j} \right) \right]. \tag{8}$$

Sign of λ_i. The first summand is non-positive whenever $h' \geq 0$, $\mathbf{W}_i \bar{X}_{c=i} \geq 0$, and $\sum_j I_j + \sum_j I_j^2 \geq 0$; the variance term dominates unless $\mathbb{E}[I] < 0$ and $\|I\|$ is small simultaneously, which de-meaning prevents. The second summand is not sign-definite: for a sample anti-correlated with its centroid ($\sum_j I_j \bar{X}_{c=i,j} < 0$) while $h(\mathbf{W}_i I) > 0$ it is positive and can dominate. A concrete instance with Tanh, $\bar{X}_{c=i} = (1, -1, 0)$, $I = (-1, 0, 1)$, and $\mathbf{W}_i = (1, 0, 3)$ satisfies $\mathbf{W}_i \bar{X}_{c=i} = 1 > 0$ and $\mathbf{1}^\top \bar{X}_{c=i} = \mathbf{1}^\top I = 0$, yet gives $\lambda_i = +0.82\,\eta\beta$. Alignment with the centroid therefore does not imply sample-wise contraction, and we make no such claim; the stability statement of Proposition 3.1 concerns the expected dynamics along the centroid ray (Section A.4).

Discrete-time stability. For the update map $\mathbf{W}_{t+1} = \mathbf{W}_t + \Delta\mathbf{W}_t$, with $\Delta\mathbf{W}_t = \eta(\cdots)$ as in Equation (2), the Jacobian of the map is $\mathbf{I} + \mathbf{J}$ with $\mathbf{J} = \partial(\Delta\mathbf{W})/\partial\mathbf{W}$ of Equation (7), so local stability along the rank-1 image direction requires

$$|1 + \lambda_i| < 1 \quad \Longleftrightarrow \quad -2 < \lambda_i < 0, \tag{9}$$

where $\lambda_i = \eta\tilde{\lambda}_i$ contains the learning rate exactly once (Equation (8)). Whenever $\lambda_i < 0$, it suffices to choose η small enough that $|\lambda_i| = \eta|\tilde{\lambda}_i| < 2$; bounding $|\tilde{\lambda}_i|$ via $|h|, |h'| \leq L_h$, $\|I\| \leq R_I$, and a working bound $\|\mathbf{W}_i \bar{X}_{c=i}\| \leq M_W$ yields the sufficient step size $\eta < 2/\big(\beta L_h (M_W (R_I + R_I^2) + R_I + R_I^2)\big)$, which reduces to $\eta < 10^{-3}$ for unit-normalized inputs with Tanh, $\beta = 1$, and typical M_W in our experiments.

A.4 Centroid-mode reduction

Let μ_i denote the unit-norm centroid and consider $\mathbf{W}_i = a\,\mu_i^\top$ with $a \geq 0$, so that $\mathbf{W}_i I = az$ with $z = \mu_i^\top I$ and $\mathbf{W}_i \mu_i = a$. Projecting the expected update of Section A.1 onto μ_i gives $\mu_i^\top \mathbb{E}[\Delta\mathbf{W}_i^{\mathbf{AD}\top}] = \eta\kappa$ with $\kappa = 1 + 1/m$, and

$$\mu_i^\top \mathbb{E}[\Delta\mathbf{W}_i^{\mathbf{BC}\top}] = -\eta\beta\, a\, \mathbb{E}\big[h(az)\, \mu_i^\top (\mathbf{1} + I)\big] = -\eta\beta\, a\, \mathbb{E}\big[z\, h(az)\big] = -\eta\beta\, a\, r_i(a) \tag{10}$$

when $\mathbf{1}^\top \mu_i = 0$, which yields Equation (4). The component of $\Delta\mathbf{W}_i^{\mathbf{BC}}$ orthogonal to μ_i, along $(\mathbf{1} + I) - z\mu_i$, is not part of this reduction; its expectation vanishes when the class-conditional distribution of $I - z\mu_i$ is symmetric about the centroid ray, and otherwise it is the higher-order term that makes the result local.

Monotonicity and uniqueness. $\frac{d}{da}\big[a\, r_i(a)\big] = r_i(a) + a\, r_i'(a) = \mathbb{E}\big[z\, h(az) + az^2 h'(az)\big]$; for sign-preserving h both summands are non-negative, and the sum is strictly positive for Tanh and the identity whenever z is non-degenerate, and for ReLU whenever $P(z > 0) > 0$ (for z supported on $z < 0$, ReLU gives $r_i \equiv 0$). Moreover $a\, r_i(a) \to 0$ as $a \to 0$ and $a\, r_i(a) \to \infty$ as $a \to \infty$ (for Tanh $r_i(a) \to \mathbb{E}|z|$; for ReLU and the identity $a\, r_i(a)$ grows quadratically), so $\beta a\, r_i(a) = \kappa$ has exactly one positive root $a^\star$ for every $\beta > 0$. Linearizing Equation (4) at $a^\star$ gives the multiplier $1 - \eta\beta\,[r_i(a^\star) + a^\star r_i'(a^\star)]$, whence the contraction condition of Proposition 3.1. For $\beta = 0$ the bracket in Equation (4) is the constant κ and a_t grows linearly while the direction μ_i is unchanged, which is why β alters the information-plane trajectory far more than classification accuracy (Section 4.1).

Corollary A.1 (Homeostatic dissipation)**.** *Under the normalization of Proposition 3.1, the* $\mathbf{BC}$ *term decreases the pre-activation energy* $E_I = \frac{1}{2}(\mathbf{W}_i I)^2$ *to first order whenever* $\mathbf{W}_i \mu_i > 0$*:* $\Delta E_I^{\mathbf{BC}} = -\eta\beta\, (\mathbf{W}_i \mu_i)\, (\mathbf{W}_i I)\, h(\mathbf{W}_i I)\, \|I\|^2 + \mathcal{O}(\eta^2) \leq 0$.

Proof. With $E_I = \frac{1}{2}(\mathbf{W}_i I)^2$, $\nabla_{\mathbf{W}_i} E_I = (\mathbf{W}_i I)\, I^\top$ and $\Delta\mathbf{W}_i^{\mathbf{BC}} = -\eta\beta\, (\mathbf{W}_i \mu_i)\, h(\mathbf{W}_i I)\, (\mathbf{1}^\top + I^\top)$, so $\Delta E_I^{\mathbf{BC}} = -\eta\beta\, (\mathbf{W}_i \mu_i)(\mathbf{W}_i I)\, h(\mathbf{W}_i I)\, I^\top (\mathbf{1} + I) + \mathcal{O}(\eta^2)$, and $I^\top (\mathbf{1} + I) = \|I\|^2$ when $\mathbf{1}^\top I = 0$. The sign follows from $\mathbf{W}_i \mu_i > 0$ and $(\mathbf{W}_i I)\, h(\mathbf{W}_i I) \geq 0$. This is a direct homeostasis statement and does not rely on the surrogate of Section B.

Normalization. The head inputs are layer-normalized (mean-subtracted across feature dimensions), so $\mathbf{1}^\top I = 0$ exactly; the centroids are de-meaned averages of such vectors, scaled to unit norm, so $\mathbf{1}^\top \mu_i = 0$ as well. Without these conditions $\mathbf{1}^\top \mu_i + z$ replaces z in r_i and $\sum_j I_j + \|I\|^2$ replaces $\|I\|^2$, and the conclusions hold whenever these factors keep the sign of z and stay positive.

Stochastic updates. The per-sample updates fluctuate around the expected field; under standard bounded-variance stochastic-approximation assumptions, a constant step η yields an $\mathcal{O}(\sqrt{\eta})$ stationary neighbourhood of $a^\star$ rather than exact convergence, which suffices for the finite streaming budgets used in our experiments.

B Formal Motivation: IPA as Surrogate Gradient Descent

In this section, we provide a formal motivation for the specific structure of the IPA plasticity rules. We do not claim to optimize the exact variational IB bound (which often requires intractable integral estimations); we relate the two terms of the IPA rule to differentiable proxies of the two IB terms. The surrogate replaces each mutual-information term with such a proxy; this is a motivating heuristic rather than a rigorous bound derivation or a gradient-flow statement.

Recall the IB optimization problem:

$$\min_{\mathbf{W}} \mathcal{L}_{IB} = \underbrace{-\mathcal{I}(O;Y)}_{\text{Fitting}} + \beta \underbrace{\mathcal{I}(X;O)}_{\text{Compression}} \tag{11}$$

We define a surrogate loss function, $\tilde{\mathcal{L}}(\mathbf{W})$, which substitutes each mutual information term with differentiable proxies commonly used in representation learning:

$$\tilde{\mathcal{L}}(\mathbf{W}) = \mathcal{L}_{align}(\mathbf{W}) + \lambda \mathcal{L}_{reg}(\mathbf{W}) \tag{12}$$

AD maximizing $\mathcal{I}(O;Y)$: Maximizing mutual information $\mathcal{I}(O;Y)$ requires the output to be discriminative: high response for the target class $Y_{c=i}$ and low response for non-target classes $Y_{c\neq i}$. We effectively minimize the overlap between the class-conditional output distributions. In practice, this is often approximated in classification models by maximizing a discriminative alignment between outputs and class-conditionals, e.g., class means, that serves as a tractable proxy for likelihood maximization.

We consider a simplified linear discriminant model in which class scores are given by $s_i(x) = \langle \mathbf{W}_i, x \rangle$ and higher alignment corresponds to higher posterior probability. Here the probability of the correct class is proportional to the alignment between the weight vector $\mathbf{W}_{i=c}$ and the class-conditional mean input $\overline{X}_{i=c}$, we define the alignment as a contrastive loss as:

$$\mathcal{L}_{align}(\mathbf{W}_i) = \underbrace{-\langle \mathbf{W}_i, \overline{X}_{c=i} \rangle}_{\text{Maximize } s_{c=i}} + \gamma \underbrace{\langle \mathbf{W}_i, \overline{X}_{c\neq i} \rangle}_{\text{Minimize } s_{c\neq i}} \tag{13}$$

where γ balances the positive and negative distinctiveness. Taking the gradient with respect to W_i:

$$\nabla_{\mathbf{W}_i} \mathcal{L}_{align} \approx -\bar{X}_{c=i} + \gamma \bar{X}_{c\neq i} \tag{14}$$

This corresponds directly to the behavior of $\mathbf{AD}$ with $\gamma = \frac{m-1}{m}$ (Equation (3)). Thus, the additive Hebbian term acts as a first-order gradient step maximizing the alignment between the output neuron and the class centroid in addition to feature decorrelation, directly minimizing the conditional entropy $H(Y|O)$ and maximizing $\mathcal{I}(O;Y)$.

BC minimizing $\mathcal{I}(X;O)$: Minimizing $\mathcal{I}(X;O)$ corresponds to compressing the representation; an upper bound is given by the channel capacity, which can be controlled by penalizing the output energy. We define the regularization proxy as

$$\mathcal{L}_{reg}(\mathbf{W}_i) = \tfrac{1}{2}\, \sigma(o_i)\, (\mathbf{W}_i x)^2, \tag{15}$$

where the neuron's own activation gating $\sigma(o_i)$ is treated as a stop-gradient (i.e., it modulates the regularizer but is not differentiated through). The gradient of Equation (15) with respect to $\mathbf{W}_i \in \mathbb{R}^{1\times n}$ is then

$$\nabla_{\mathbf{W}_i} \mathcal{L}_{reg} = \sigma(o_i)\, (\mathbf{W}_i x)\, x^\top. \tag{16}$$

This penalizes weight magnitude along input directions of high variance where the neuron is active. From Equation (3), the IPA $\mathbf{BC}$ update is proportional to $-\mathbf{W}_i \bar{X}_{c=i}\, (\mathbf{1}^\top + I^\top)\, o_i$. Approximating the (population) covariance $\mathbb{E}[xx^\top]$ by the rank-1 outer product $\bar{X}_{c=i}\, (\mathbf{1}^\top + I^\top)$ recovers the gradient direction up to the same activation gating, so the $\mathbf{BC}$ update is a stochastic approximation to the gradient of the energy regularizer and therefore minimizes the upper bound on $\mathcal{I}(X;O)$.

B.1 Conclusion

Combining these derivations, the IPA update is aligned with the descent directions of the surrogate IB objective

$$\Delta \mathbf{W}_i \;\approx\; \underbrace{\bar{X}_{c=i}^\top - \gamma\, \bar{X}_{c\neq i}^\top}_{\text{maximize } \mathcal{I}(O;Y) \text{ proxy}} \;-\; \beta \underbrace{\nabla_{\mathbf{W}_i} \mathcal{L}_{reg}}_{\text{minimize } \mathcal{I}(X;O) \text{ upper bound}}. \tag{17}$$

This motivates the IPA–IB connection: the $\mathbf{AD}$ term is the exact gradient of $\mathcal{L}_{align}$, while the $\mathbf{BC}$ vector field $(\mathbf{W}_i \bar{X}_{c=i})\, h(\mathbf{W}_i x)\, (\mathbf{1}^\top + x^\top)$ is in general not conservative, so no scalar objective has it as its Euclidean gradient; its provable property is the energy dissipation of Corollary A.1, which does not require the IB. We do not claim optimization of the exact variational IB bound, nor that $\mathcal{L}_{align}$ is a formally valid lower bound on mutual information. Moreover, because the activation gating is frozen during differentiation (stop-gradient), the true dynamics are not the exact gradient flow of a fixed objective; empirically the surrogate energy is not stepwise monotone along IPA trajectories, and we therefore make no descent claim for the surrogate; the contraction results of Section A are proved on the update dynamics and do not use $\tilde{\mathcal{L}}$.

C IPA ABLATION DYNAMICS

We analyze the weight dynamics under varying IPA formulations, investigating the effect of β and different output activations.

C.1 IPA DYNAMICS: 2D GAUSSIAN

Figure 6 shows the results on the toy 2D Gaussian classification problem. The three classes are sampled around $(-2, 2)$, $(0, -3)$, and $(2, 2)$ with a variance of 0.5 for $15{,}000$ steps per class. $\bar{X}_i$ is estimated with one sample per class and is also used to de-mean the input I. On this task, we test three different variations of IPA-layer with Tanh activation, where we vary the relative size of the weight updates $\Delta\mathbf{W}^{\mathbf{BC}}$ using different scaling factors $\beta \in \{0.0, 0.5, 1.0\}$. The scaling factor β is designed to affect compression through the self-normalizing $\mathbf{BC}$ terms, similarly to the IB objective (Tishby et al., 2000). Mutual information is estimated as described in Section C.

With $\beta = 0$, the weights follow a near-linear increase driven by $\Delta\mathbf{W}^{\mathbf{AD}}$, effectively turning IPA into a dot-product similarity model comparable to a nearest-class-mean or prototypical classifier. Introducing $\beta > 0$ changes the weight trajectory (third column): early training remains approximately linear, but the $\mathbf{BC}$ self-normalizing term increasingly induces compression — it bounds weight growth as a function of the neuron's own alignment with its class centroid and reduces co-activation between similar classes — producing curvature in the information plane. This compression distinguishes IPA from standard centroid-based approaches. In practice, the final-accuracy effect is small when class centroids are well separated, and we use $\beta = 1$ in all subsequent experiments. The reduction in co-activation is visible in the output space (fourth column): starting from the origin (output is zero because $\mathbf{W} = 0$), class samples move toward high target activity while saturating at the corner points and suppressing activation in non-target neurons.

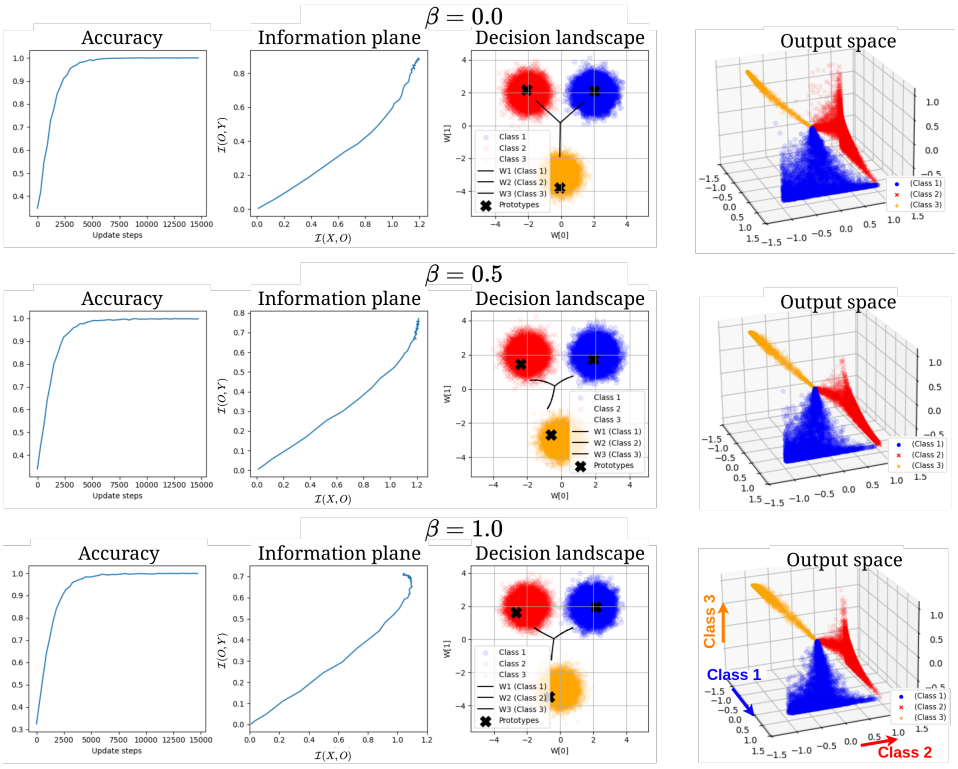


Figure 6: Weight dynamics of IPA during 2D-Gaussians-3 task with varying β scaling factors. First column: accuracy during optimization. Second column: information plane progression. Third column: input space X with class distributions, one-shot prototype $\bar{X}_i$, and weight trajectory $\mathbf{W}_i$. Fourth column: output space Y; arrows indicate per-class output Y_i.

Figure 7 shows the weight dynamics for different IPA output functions and β values on the same 3-Gaussian classification problem. When $\beta = 0$, all trajectories follow a similar progression ($\Delta \mathbf{W}^{\mathbf{AD}}$ depends only on the input). As β increases, different weight progressions arise.

C.2 Output function & β

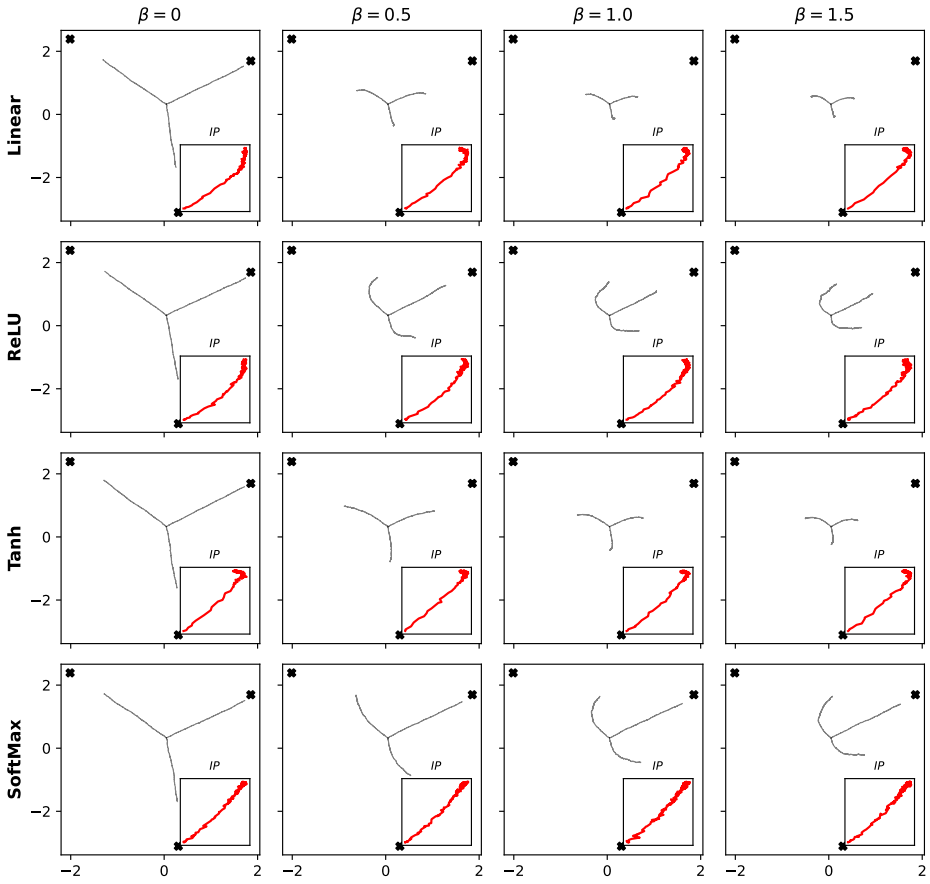


Figure 7: Weight dynamics and information plane (IP), under different output functions and compression terms β. Each experiment consists of 20000 steps with $\eta = 10^{-5}$, with each class providing 1 sample (prototype).

C.3 IPA few-shot ablation

Table 2: MNIST n-way few-shot classification accuracy (%, mean ± STD $N = 30$) using random MLP (left) and CNN (right) reservoirs with varying depth L and width W. Results are reported for 5-, 10-, and 50-shot settings with data augmentation.

MNIST 10-way	5-shot	10-shot	50-shot
MLP_{L1W32}	48.7 ± 1.5	54.3 ± 1.8	61.6 ± 1.1
MLP_{L1W128}	64.1 ± 1.4	70.6 ± 0.9	78.1 ± 0.6
MLP_{L2W64}	50.8 ± 1.3	57.1 ± 1.3	65.8 ± 0.9
MLP_{L2W128}	58.5 ± 1.4	63.8 ± 1.2	71.7 ± 0.9
MLP_{L2W256}	63.7 ± 1.6	70.3 ± 1.0	78.2 ± 0.9
MLP_{L2W512}	68.3 ± 1.3	74.7 ± 1.4	80.8 ± 0.6
MLP_{L3W64}	44.7 ± 1.4	49.8 ± 1.4	58.6 ± 0.9
MLP_{L3W128}	53.0 ± 1.3	60.1 ± 1.1	69.4 ± 0.8
MLP_{L3W256}	61.5 ± 1.2	68.0 ± 1.0	75.2 ± 0.9
MLP_{L3W512}	65.3 ± 1.2	71.1 ± 1.2	79.7 ± 0.7
$\text{MLP}_{L3W1024}$	68.6 ± 1.3	74.6 ± 1.0	81.2 ± 0.5
$\text{MLP}_{L10W256}$	17.6 ± 0.8	20.1 ± 0.9	28.8 ± 0.9

MNIST 10-way	5-shot	10-shot	50-shot
Conv_{L1W32}	65.7 ± 3.4	71.8 ± 3.1	78.1 ± 1.8
Conv_{L1W128}	68.5 ± 3.5	73.1 ± 2.9	79.5 ± 1.6
Conv_{L3W64}	71.7 ± 3.5	80.0 ± 3.4	84.9 ± 2.1
Conv_{L3W128}	73.4 ± 4.0	79.8 ± 3.4	86.3 ± 1.4
Conv_{L3W256}	73.0 ± 5.2	80.0 ± 3.0	86.5 ± 1.5
Conv_{L3W512}	73.9 ± 4.5	81.0 ± 3.4	87.3 ± 1.3
Conv_{L5W64}	75.1 ± 3.8	82.1 ± 2.3	86.7 ± 1.5
Conv_{L5W128}	73.6 ± 4.3	79.7 ± 3.1	87.9 ± 1.2
Conv_{L5W256}	73.1 ± 4.0	82.4 ± 2.2	88.8 ± 1.2
Conv_{L5W512}	72.4 ± 5.1	81.6 ± 2.6	$\mathbf{89.8 \pm 1.1}$
$\text{Conv}_{L5W1024}$	69.2 ± 4.5	79.7 ± 2.9	89.6 ± 1.3
Conv_{L7W256}	73.1 ± 4.8	79.0 ± 2.9	84.2 ± 1.4

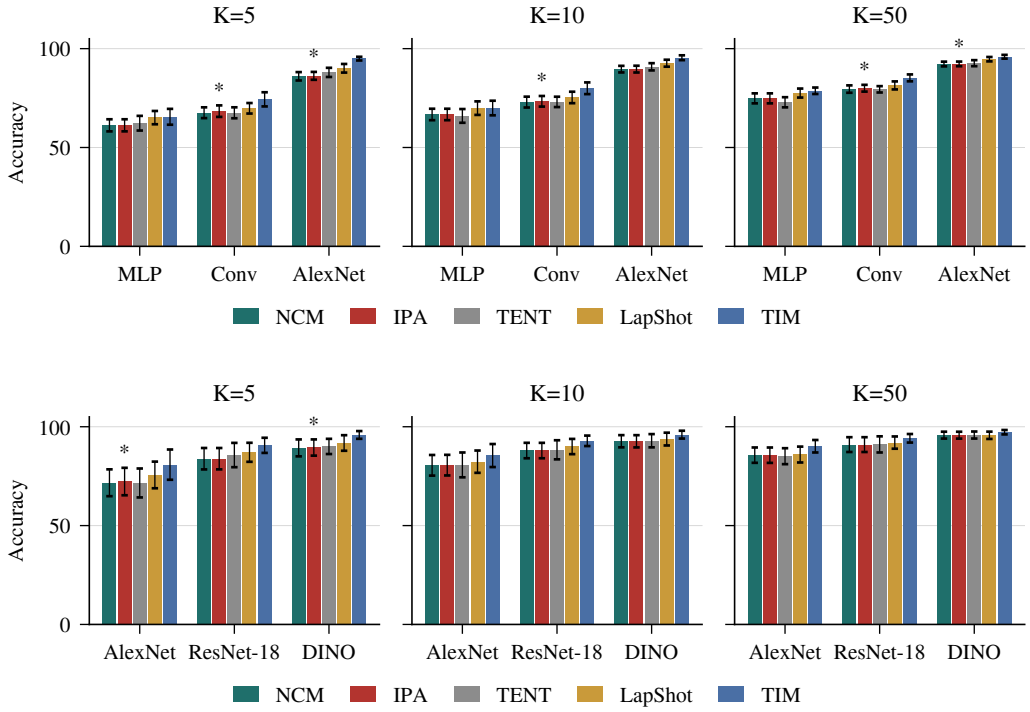


Figure 8: Few-shot accuracy under matched unlabeled-data access (single evaluation harness, disjoint evaluation, $N = 30$ episodes, mean $\pm$ std) at $K \in \{5, 10, 50\}$ shots. Top: MNIST 10-way with random MLP_{L2W512}, random Conv_{L5W512}, and pretrained AlexNet. Bottom: CIFAR-FS 5-way with pretrained AlexNet, ResNet-18, and DINO-ViT-S/8. NCM uses the labeled shots only; IPA, TENT (Wang et al., 2021), LaplacianShot (Ziko et al., 2020), and TIM (Boudiaf et al., 2020) receive the same balanced unlabeled pool. * denotes $p < 0.05$ for IPA vs. NCM (paired two-sided t-test over episodes).

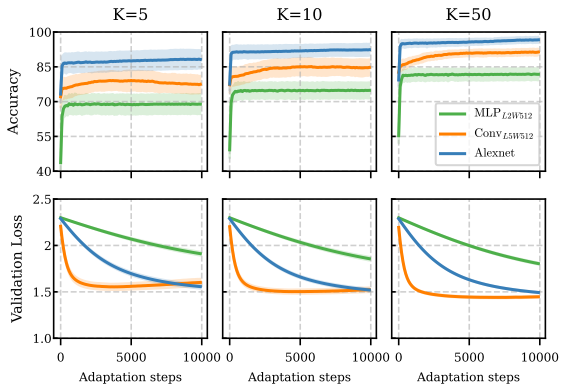


Figure 9: Training and validation curves for the best IPA configurations on MNIST 10-way at $K \in \{5, 10, 50\}$ (N=30, ±std): MLP$_{L2W512}$, Conv$_{L5W512}$, AlexNet. The matched-access comparison appears in Figure 8; backpropagation and prototypical baselines on the random backbones are listed in Table 4.

C.4 Data augmentation

Augmentation experiments with MNIST 10-way.

Table 3: Number of augmented samples [0,1,10]. Augmentation: degrees=[-10.0, 10.0], translate=(0.025, 0.025), scale=(0.9, 1.1). $lr_{inner} = 10^{-5}$, $\beta = 1$, 30 repetition per experiment.

MNIST 10-way	5	10	50
AlexNet	42.8 ± 0.6	70.8 ± 0.5	83.5 ± 0.4
AlexNet:1	$\mathbf{82.3 \pm 0.3}$	87.1 ± 0.3	88.7 ± 0.4
AlexNet:10	79.2 ± 0.3	$\mathbf{87.7 \pm 0.3}$	$\mathbf{93.6 \pm 0.3}$
ResNet	45.2 ± 0.8	81.2 ± 0.4	88.8 ± 0.6
ResNet:1	72.6 ± 0.4	78.5 ± 0.5	84.9 ± 0.7
ResNet:10	55.6 ± 0.7	68.9 ± 0.6	81.7 ± 0.7

Table 4: Random backbones: for degrees=[-5.0, 5.0], translate=(0.025, 0.025). $\eta = 10^{-5}$, $\beta = 1$ ($N = 30 \pm std$).

10 aug.	5	10	50
Proto$_{MLP}$	62.9 ± 1.3	69.5 ± 1.1	76.9 ± 0.7
Proto$_{Conv}$	59.4 ± 0.9	62.8 ± 0.9	66.6 ± 0.8
Proto$_{Alex}$	$\mathbf{74.2 \pm 0.9}$	$\mathbf{79.0 \pm 0.9}$	$\mathbf{79.0 \pm 1.0}$
BP$_{MLP}$	65.9 ± 0.9	74.6 ± 0.9	84.6 ± 0.5
BP$_{Conv}$	51.5 ± 1.4	58.8 ± 1.1	70.0 ± 0.9
BP$_{Alex}$	$\mathbf{81.2 \pm 1.0}$	$\mathbf{88.5 \pm 0.7}$	$\mathbf{95.5 \pm 0.3}$

At benchmark scale the accuracy effect of β is small: on CIFAR-100 FSCIL (ResNet-12, 30 repetitions) IPA-Approx reaches 68.26 AA with $\beta = 1$ and 68.14 with $\beta = 0$, and on the CIFAR-FS episodes of Figure 8 the difference between $\beta = 0$ and $\beta = 1$ is at most 0.18 points. The visible effect of the $\mathbf{BC}$ term is the compression in the information plane (Figures 6 and 7), not episode accuracy.

Table 5: AlexNet β variation: degrees=[-5.0, 5.0], translate=(0.025, 0.025). $lr_{inner} = 10^{-5}$, $\beta = 1$, 30 repetition per experiment

10 aug.	5	10	50
AlexNet $\beta = 0.1$	84.2 ± 0.2	88.78 ± 0.1	92.5 ± 0.1
AlexNet $\beta = 0.5$	84.1 ± 0.3	88.1 ± 0.1	92.4 ± 0.1
AlexNet $\beta = 1.0$	$\mathbf{84.5 \pm 0.3}$	$\mathbf{90.3 \pm 0.1}$	$\mathbf{94.3 \pm 0.03}$

C.5 META-IPA HYPERPARAMETER SWEEP

Before conducting the meta-experiments, we perform a full hyperparameter sweep analyzing the effects of lr_{inner}, lr_{meta}, K-shots, and model capacity (results are presented in Figure 10). Notably, low values of lr_{inner} negatively impact performance. Best performance is achieved using $lr_{\text{meta}} = 10^{-2}$, $lr_{\text{inner}} = 10^{-3}$, and a four-layer network with 64 features. Although this performance is not significantly different from an $L4W128$ model with the same learning rates, we select the smaller architecture since it matches the standard MAML implementation and allows for direct body swapping. We compare this configuration with the original MAML and the adapted ANILr.

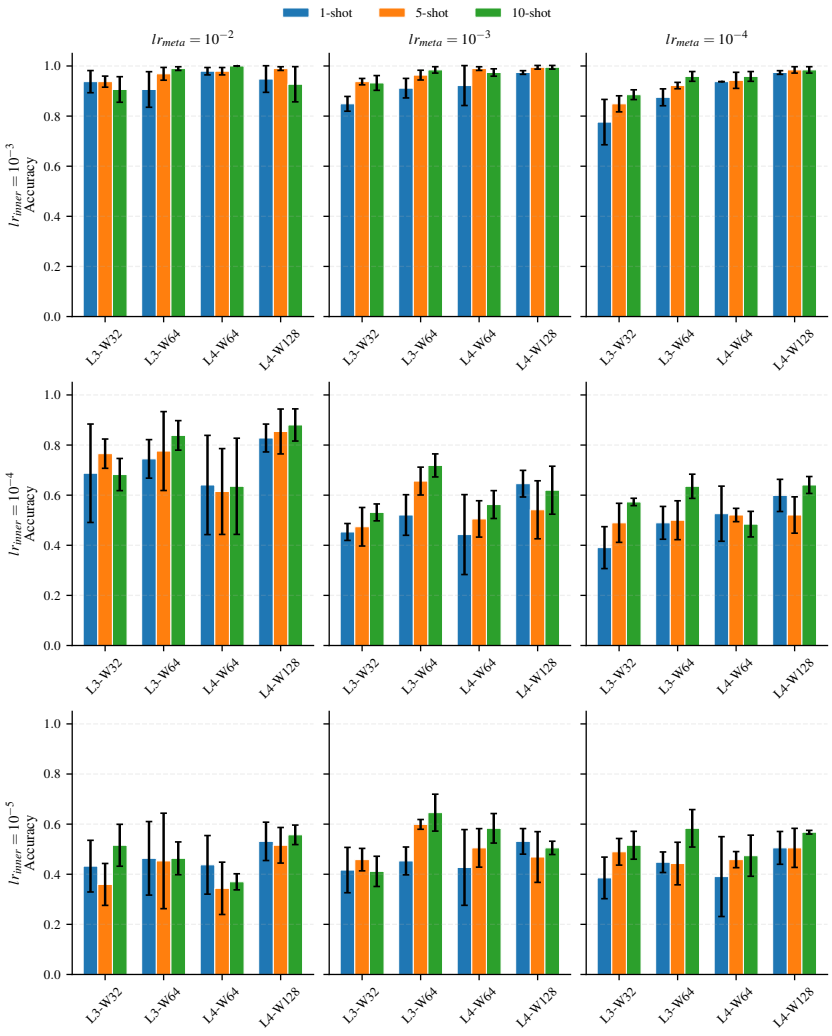


Figure 10: Hyper-parameter search of IPA meta-learning.

C.6 META LEARNING EXPERIMENTAL PARAMETERS

Table 6: Hyper-parameters used. Brackets indicate settings for memory heavy experiments (CUB200).

	IPA	MAML	ANILr
$Reset\ head$	Yes	No	Yes
BP_{inner}	-	h+b	h
lr_{inner}	0.001	0.5	0.1
n_{inner}	100	5	5
BP_{meta}	b	h+b	b
lr_{meta}	0.001	0.001	0.001
n_{meta}	100	10000	10000
$Batch\ size$	32	32(4)	32(4)

C.7 META LEARNING PSEUDO CODE

Below are all pseudo-codes used in our experiments.

Algorithm 1 Informed Plastic Attractors (IPA)

Require: Frozen feature extractor f_{θ_b} (f_{θ_b}=Id if no backbone)
Require: K-shot labeled data $\mathcal{D}_{\mathcal{T}_i}$ for each class $i \in \{1, \dots, m\}$
Require: Stream of unlabeled deployment data; learning rate η; compression β

$\mathbf{W} \leftarrow \mathbf{0}$ ▷ IPA head
for $i = 1, \dots, m$ **do**
  $\bar{X}_{\mathcal{T}_i} \leftarrow K^{-1} \sum_{k=1}^{K} f_{\theta_b}(\mathcal{D}_{\mathcal{T}_i,k})$ ▷ class centroid; f_{θ_b} is layer-normalized
end for
$\bar{X} \leftarrow m^{-1} \sum_{i=1}^{m} \bar{X}_{\mathcal{T}_i}$; $\bar{X}_{\mathcal{T}_i} \leftarrow (\bar{X}_{\mathcal{T}_i} - \bar{X})/\|\bar{X}_{\mathcal{T}_i} - \bar{X}\|$ for all i ▷ de-mean, unit norm
for $i = 1, \dots, m$ **do**
  $\mathbf{A}_i \leftarrow -\mathbf{1}^\top$; $\mathbf{D}_i \leftarrow \bar{X}_{\mathcal{T}_i}^\top (1 + 1/m)$
end for
while deployed **do**
  Receive unlabeled D; $I \leftarrow f_{\theta_b}(D) - \bar{X}$; $O \leftarrow h(\mathbf{W}I)$
  for $i = 1, \dots, m$ **do**
    $\mathbf{B}_i \leftarrow \mathbf{C}_i \leftarrow -\beta\,(\mathbf{W}_i \bar{X}_{\mathcal{T}_i})\,\mathbf{1}^\top$ ▷ $\mathbf{W}_i \bar{X}_{\mathcal{T}_i} \in \mathbb{R}$ scalar
  end for
  $\mathbf{I} \leftarrow [I, \dots, I]^\top$; $\mathbf{O} \leftarrow [O, \dots, O]$
  $\Delta\mathbf{W} \leftarrow \eta\,(\mathbf{A} \circ \mathbf{I} + \mathbf{B} \circ \mathbf{O} + \mathbf{C} \circ \mathbf{I} \circ \mathbf{O} + \mathbf{D})$ ▷ matches Equation (2)
  $\mathbf{W} \leftarrow \mathbf{W} + \Delta\mathbf{W}$ ▷ accumulate update in place
end while

Algorithm 2 IPA-Approx (confidence-weighted prototype refinement) and the NCM-r control

Require: Raw class means $\bar{X}_{\mathcal{T}_c}$ with counts n_c ($n_c = K$ for novel classes); temperature $\tau = 0.25$
Build $\bar{X}$, μ_c, $\mathbf{ABCD}$ as in Algorithm 1; initialize $\mathbf{W}_c \leftarrow \mu_c^\top$ ▷ prototype-row initialization
for each unlabeled stream sample x (one pass or repeated passes) **do**
  $I \leftarrow f_{\theta_b}(x) - \bar{X}$; $p \leftarrow \mathrm{softmax}(\mathbf{W}I/\tau)$ ▷ confidence from the pre-update head
  Hebbian update of $\mathbf{W}$ on I (Algorithm 1)
  for $c = 1, \dots, m$ **do**
    $n_c \leftarrow n_c + p_c$; $\bar{X}_{\mathcal{T}_c} \leftarrow \bar{X}_{\mathcal{T}_c} + p_c\,(f_{\theta_b}(x) - \bar{X}_{\mathcal{T}_c})/n_c$
  end for
  Recompute $\bar{X}$, μ_c, and $\mathbf{ABCD}$ from the updated means
end for
NCM-r: same stream and running-mean update with p replaced by the one-hot nearest-centroid assignment of $f_{\theta_b}(x)$; no plastic head; classification by nearest centroid

Algorithm 3 Meta-IPA

Require: Task distribution $p(\mathcal{T})$; inner/meta learning rates lr_{inner}, lr_{meta}; inner steps n_{inner}

Initialize body θ_b
while not converged **do**
  Sample task batch $\{\mathcal{T}_i\} \sim p(\mathcal{T})$
  for each task $\mathcal{T}_i$ **do**
    Sample disjoint support, query, and unlabeled sets $\mathcal{D}_i^{sup}, \mathcal{D}_i^{que}, \mathcal{D}_i^{unl}$
    Initialize $\mathbf{W}^{(i)} \leftarrow \mathbf{0}$ ▷ fresh head per task
    for $j = 1, \ldots, m$ **do**
      $\bar{X}_{\mathcal{T}_i,j} \leftarrow$ class-j mean of $f_{\theta_b}(\mathcal{D}_i^{sup})$, de-meaned and unit-normalized
      $\mathbf{A}_j^{(i)} \leftarrow -\mathbf{1}^\top; \quad \mathbf{D}_j^{(i)} \leftarrow \bar{X}_{\mathcal{T}_i,j}^\top (1 + 1/m)$
    end for
    for $t = 1, \ldots, n_{inner}$ **do**
      Sample $D_i^{(t)} \subset \mathcal{D}_i^{unl}$ *without labels*
      $I \leftarrow f_{\theta_b}(D_i^{(t)}); \quad O \leftarrow h(\mathbf{W}^{(i)} I)$
      Update $\mathbf{B}_j^{(i)} \leftarrow \mathbf{C}_j^{(i)} \leftarrow -\beta\,(\mathbf{W}_j^{(i)} \bar{X}_{\mathcal{T}_i,j})\,\mathbf{1}^\top$ for all j
      $\mathbf{W}^{(i)} \leftarrow \mathbf{W}^{(i)} + lr_{inner}\,(\mathbf{A}^{(i)} \circ \mathbf{I} + \mathbf{B}^{(i)} \circ \mathbf{O} + \mathbf{C}^{(i)} \circ \mathbf{I} \circ \mathbf{O} + \mathbf{D}^{(i)})$ ▷ $\Delta\mathbf{W}$ per Equation (2), η=lr_{inner}
    end for
  end for
  $\theta_b \leftarrow \theta_b - lr_{meta}\,\nabla_{\theta_b} \sum_i \mathcal{L}_{\mathcal{T}_i}\big(f_{\theta_b}, \mathbf{W}^{(i)};\, \mathcal{D}_i^{que}\big)$ ▷ meta-grad on θ_b only; no grad through inner loop
end while

Algorithm 4 MAML

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: inner and meta step sizes

Initialize parameters θ ▷ Initialize model parameters θ, including head f_h and body f_b
while not converged **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Sample datasets $\mathcal{D}_i^{sup}, \mathcal{D}_i^{que}$ from task $\mathcal{T}_i$
    Compute inner gradient on support set:
    $g_i = \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta, \mathcal{D}_i^{sup})$
    Compute adapted parameters: $\theta_i' = \theta - \alpha g_i$ ▷ Inner-update f_b and f_h
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}, \mathcal{D}_i^{que})$ ▷ Meta-update f_b and f_h using query sets
end while

Algorithm 5 ANILr (ANIL with head reinitialization)

Require: Task distribution $p(\mathcal{T})$; inner/meta step sizes α, γ

Initialize body θ_b
while not converged **do**
  Sample task batch $\{\mathcal{T}_i\} \sim p(\mathcal{T})$
  for each task $\mathcal{T}_i$ **do**
    Sample $\mathcal{D}_i^{sup}, \mathcal{D}_i^{que}$
    Re-initialize head $\theta_h^{(i)} \sim U[-1, 1]$ ▷ fresh head per task
    for $t = 1, \ldots, n_{inner}$ **do**
      $g_h \leftarrow \nabla_{\theta_h} \mathcal{L}_{\mathcal{T}_i}\big(f_{\theta_h^{(i)}} \circ f_{\theta_b};\, \mathcal{D}_i^{sup}\big)$ ▷ body θ_b frozen during inner loop
      $\theta_h^{(i)} \leftarrow \theta_h^{(i)} - \alpha\, g_h$
    end for
  end for
  $\theta_b \leftarrow \theta_b - \gamma\, \nabla_{\theta_b} \sum_i \mathcal{L}_{\mathcal{T}_i}\big(f_{\theta_h^{(i)}} \circ f_{\theta_b};\, \mathcal{D}_i^{que}\big)$ ▷ meta-update body only
end while

C.8 META-IPA MNIST 10-WAY LEARNING CURVES

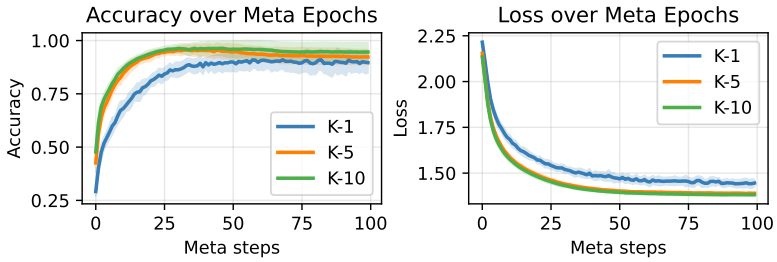


Figure 11: Learning curves for Meta-IPA on MNIST dataset containing 5 classes, for 1, 5, and 10 shot task. Solid lines indicate mean values over 30 repetitions with shaded area indicating $\pm std$.

C.9 ADAPTED CROSS-VALIDATION

The static nature of the original MAML implementation (Finn et al., 2017) prevents changing the model-output size for different numbers of tasks. We circumvent this issue by adjusting the target-domain task slightly to match the source. Specifically, we adapt cross-domain tasks to correspond with the original meta-task size by scaling the problem accordingly, e.g., MNIST-5 is tested on Omniglot-5 and CUB200-5. Conversely, Omniglot-20 uses only the first 5 outputs on the MNIST dataset. Note that rescaling the problem changes the task difficulty (e.g., CUB200-5 has an expected random-guessing accuracy of 20%).

Vanilla ANIL has the same fixed-head limitation as MAML for cross-domain evaluation with different class counts. ANILr addresses this by reinitializing the head at every inner loop, matching the head-reset protocol of IPA. We therefore use ANILr throughout the cross-domain comparisons; on same-domain tasks (Table 7) ANILr is also the relevant comparator since it shares the head-reset protocol with IPA and Meta-IPA.

Table 7: Same-domain meta-learning accuracy (%, mean ± std over $N = 30$ runs) on Omniglot 20-way, CUB200 20-way, and MiniImageNet 5-way at 1- and 5-shot. Bold: best per column. MNIST 5-way results in $\geq 99.9\%$ accuracy for all methods.

	Omniglot 20-way		**CUB200 20-way**		**Mini-Imagenet 5-way**	
	1	5	1	5	1	5
IPA (ours)	$\mathbf{98.98 \pm 0.93}$	95.71 ± 0.82	45.22 ± 1.37	$\mathbf{84.44 \pm 1.53}$	$\mathbf{57.60 \pm 1.09}$	$\mathbf{80.36 \pm 0.94}$
MAML	97.88 ± 0.16	97.92 ± 0.02	53.53 ± 1.20	62.85 ± 1.05	45.36 ± 0.85	64.22 ± 0.95
ANILr	96.16 ± 1.67	$\mathbf{98.34 \pm 0.61}$	$\mathbf{53.55 \pm 2.60}$	82.92 ± 1.19	54.48 ± 0.93	72.97 ± 0.97

C.10 FSCIL FULL SESSION PERFORMANCE

FSCIL performance (Average Accuracy, AA) per session (base:S0, end-session: S8) for all test conditions (ResNet-12 and DINO-ViT-S/8, on CIFAR-100 and MiniImageNet; Tables 8 to 11).

Method	S0	S1	S2	S3	S4	S5	S6	S7	S8	AA
IPA	$81.7_{\pm 0.3}$	$77.2_{\pm 0.3}$	$73.2_{\pm 0.4}$	$69.8_{\pm 0.3}$	$66.4_{\pm 0.3}$	$63.6_{\pm 0.3}$	$61.3_{\pm 0.3}$	$58.8_{\pm 0.3}$	$56.7_{\pm 0.3}$	$67.6_{\pm 0.2}$
IPA-Approx	$81.7_{\pm 0.3}$	$77.1_{\pm 0.3}$	$73.5_{\pm 0.4}$	$\mathbf{70.1_{\pm 0.4}}$	$\mathbf{67.2_{\pm 0.3}}$	$\mathbf{64.4_{\pm 0.3}}$	$\mathbf{62.3_{\pm 0.3}}$	$\mathbf{60.0_{\pm 0.3}}$	$\mathbf{58.0_{\pm 0.3}}$	$\mathbf{68.3_{\pm 0.2}}$
IPA-Adaptive (o)	$81.7_{\pm 0.3}$	$78.5_{\pm 0.3}$	$75.6_{\pm 0.3}$	$73.2_{\pm 0.3}$	$70.9_{\pm 0.3}$	$68.9_{\pm 0.3}$	$67.4_{\pm 0.3}$	$65.6_{\pm 0.3}$	$64.1_{\pm 0.3}$	$71.8_{\pm 0.2}$
NCM-r	$81.0_{\pm 0.3}$	$72.0_{\pm 0.4}$	$68.2_{\pm 0.5}$	$65.0_{\pm 0.4}$	$62.0_{\pm 0.4}$	$59.2_{\pm 0.4}$	$57.3_{\pm 0.3}$	$54.9_{\pm 0.2}$	$53.1_{\pm 0.3}$	$63.7_{\pm 0.2}$
NCM	$81.6_{\pm 0.3}$	$76.2_{\pm 0.4}$	$71.9_{\pm 0.3}$	$68.5_{\pm 0.4}$	$65.5_{\pm 0.3}$	$62.6_{\pm 0.3}$	$60.6_{\pm 0.3}$	$58.1_{\pm 0.3}$	$56.1_{\pm 0.3}$	$66.8_{\pm 0.2}$
Finetune†	$83.2_{\pm 0.3}$	$14.2_{\pm 0.5}$	$6.6_{\pm 0.2}$	$5.6_{\pm 0.2}$	$5.0_{\pm 0.2}$	$4.7_{\pm 0.1}$	$4.5_{\pm 0.2}$	$4.2_{\pm 0.1}$	$3.9_{\pm 0.1}$	$14.7_{\pm 0.1}$
EWC†	$83.2_{\pm 0.3}$	$14.2_{\pm 0.5}$	$6.6_{\pm 0.2}$	$5.7_{\pm 0.2}$	$5.0_{\pm 0.2}$	$4.6_{\pm 0.1}$	$4.5_{\pm 0.2}$	$4.2_{\pm 0.1}$	$3.9_{\pm 0.1}$	$14.7_{\pm 0.1}$
ER†	$83.2_{\pm 0.3}$	$77.5_{\pm 0.3}$	$73.5_{\pm 0.3}$	$69.9_{\pm 0.3}$	$66.5_{\pm 0.2}$	$63.6_{\pm 0.2}$	$61.1_{\pm 0.2}$	$58.6_{\pm 0.2}$	$56.4_{\pm 0.2}$	$67.8_{\pm 0.2}$
Joint† (o)	$83.2_{\pm 0.3}$	$78.0_{\pm 0.4}$	$75.0_{\pm 0.3}$	$73.0_{\pm 0.3}$	$71.2_{\pm 0.3}$	$69.9_{\pm 0.2}$	$68.9_{\pm 0.2}$	$67.8_{\pm 0.2}$	$66.9_{\pm 0.1}$	$72.7_{\pm 0.2}$

Table 8: Per-session accuracy (%) ± 95 % CI for **CIFAR-100** with **ResNet-12** backbone (30 reps). S0 is the base session; S1–S8 are incremental sessions. **Bold**: IPA-Approx significantly outperforms any other non-oracle method ($p < 0.05$). (o) oracle upper bound; † are backpropagation algorithms.

Method	S0	S1	S2	S3	S4	S5	S6	S7	S8	AA
IPA	$92.1_{\pm 0.6}$	$88.3_{\pm 0.7}$	$84.7_{\pm 0.6}$	$81.5_{\pm 0.5}$	$78.9_{\pm 0.4}$	$76.1_{\pm 0.4}$	$73.5_{\pm 0.5}$	$71.1_{\pm 0.4}$	$69.0_{\pm 0.4}$	$79.5_{\pm 0.4}$
IPA-Approx	$91.9_{\pm 0.6}$	$88.2_{\pm 0.7}$	$85.0_{\pm 0.6}$	$82.2_{\pm 0.6}$	$\mathbf{79.9_{\pm 0.5}}$	$\mathbf{77.2_{\pm 0.5}}$	$\mathbf{74.7_{\pm 0.5}}$	$\mathbf{72.5_{\pm 0.4}}$	$\mathbf{70.4_{\pm 0.4}}$	$\mathbf{80.2_{\pm 0.5}}$
IPA-Adaptive (o)	$92.0_{\pm 0.6}$	$89.4_{\pm 0.7}$	$86.8_{\pm 0.6}$	$84.5_{\pm 0.5}$	$82.7_{\pm 0.5}$	$80.6_{\pm 0.5}$	$78.5_{\pm 0.5}$	$76.7_{\pm 0.4}$	$75.0_{\pm 0.4}$	$82.9_{\pm 0.5}$
NCM-r	$92.0_{\pm 0.6}$	$88.7_{\pm 0.7}$	$85.3_{\pm 0.6}$	$82.0_{\pm 0.5}$	$79.2_{\pm 0.5}$	$76.3_{\pm 0.6}$	$73.3_{\pm 0.5}$	$71.1_{\pm 0.4}$	$68.7_{\pm 0.5}$	$79.6_{\pm 0.4}$
NCM	$92.2_{\pm 0.6}$	$88.9_{\pm 0.7}$	$85.4_{\pm 0.6}$	$82.3_{\pm 0.5}$	$79.6_{\pm 0.4}$	$76.8_{\pm 0.4}$	$74.2_{\pm 0.4}$	$71.8_{\pm 0.4}$	$69.7_{\pm 0.4}$	$80.1_{\pm 0.4}$
Finetune†	$83.8_{\pm 1.1}$	$13.3_{\pm 1.7}$	$6.6_{\pm 0.3}$	$5.8_{\pm 0.2}$	$5.5_{\pm 0.1}$	$5.1_{\pm 0.2}$	$4.6_{\pm 0.1}$	$4.5_{\pm 0.1}$	$4.3_{\pm 0.1}$	$14.8_{\pm 0.3}$
EWC†	$83.8_{\pm 1.1}$	$13.4_{\pm 1.7}$	$6.7_{\pm 0.3}$	$5.8_{\pm 0.2}$	$5.5_{\pm 0.1}$	$5.0_{\pm 0.2}$	$4.6_{\pm 0.1}$	$4.5_{\pm 0.1}$	$4.3_{\pm 0.1}$	$14.8_{\pm 0.3}$
ER†	$83.8_{\pm 1.1}$	$87.3_{\pm 0.7}$	$83.7_{\pm 0.7}$	$80.3_{\pm 0.6}$	$77.5_{\pm 0.5}$	$74.7_{\pm 0.5}$	$71.9_{\pm 0.5}$	$69.6_{\pm 0.4}$	$67.3_{\pm 0.4}$	$77.3_{\pm 0.6}$
Joint† (o)	$83.8_{\pm 1.1}$	$96.2_{\pm 0.3}$	$95.8_{\pm 0.2}$	$95.3_{\pm 0.3}$	$94.9_{\pm 0.3}$	$94.3_{\pm 0.2}$	$93.5_{\pm 0.3}$	$93.0_{\pm 0.2}$	$92.3_{\pm 0.2}$	$93.2_{\pm 0.3}$

Table 9: Per-session accuracy (%) $\pm$ 95 % CI for **MiniImageNet** with **ResNet-12** backbone (30 reps). S0 is the base session; S1–S8 are incremental sessions. **Bold**: IPA-Approx significantly outperforms any other non-oracle method ($p < 0.05$). (o) oracle upper bound; † are backpropagation algorithms.

Method	S0	S1	S2	S3	S4	S5	S6	S7	S8	AA
IPA	$75.3_{\pm 0.3}$	$73.3_{\pm 0.4}$	$71.3_{\pm 0.4}$	$69.5_{\pm 0.3}$	$67.4_{\pm 0.3}$	$66.0_{\pm 0.4}$	$65.0_{\pm 0.4}$	$63.7_{\pm 0.3}$	$62.6_{\pm 0.3}$	$68.2_{\pm 0.2}$
IPA-Approx	$76.4_{\pm 0.3}$	$74.1_{\pm 0.4}$	$72.3_{\pm 0.4}$	$\mathbf{70.6_{\pm 0.4}}$	$\mathbf{68.7_{\pm 0.4}}$	$\mathbf{67.6_{\pm 0.3}}$	$\mathbf{66.8_{\pm 0.3}}$	$\mathbf{65.6_{\pm 0.3}}$	$\mathbf{64.7_{\pm 0.2}}$	$\mathbf{69.6_{\pm 0.2}}$
IPA-Adaptive (o)	$75.4_{\pm 0.3}$	$75.1_{\pm 0.3}$	$74.5_{\pm 0.4}$	$74.2_{\pm 0.3}$	$73.4_{\pm 0.3}$	$73.1_{\pm 0.3}$	$73.0_{\pm 0.3}$	$72.2_{\pm 0.2}$	$71.9_{\pm 0.1}$	$73.6_{\pm 0.2}$
NCM	$74.4_{\pm 0.3}$	$72.0_{\pm 0.4}$	$69.5_{\pm 0.4}$	$67.4_{\pm 0.3}$	$65.2_{\pm 0.4}$	$63.5_{\pm 0.4}$	$62.3_{\pm 0.4}$	$60.8_{\pm 0.3}$	$59.6_{\pm 0.3}$	$66.1_{\pm 0.3}$
NCM-r	$73.5_{\pm 0.4}$	$71.9_{\pm 0.4}$	$70.0_{\pm 0.4}$	$68.2_{\pm 0.5}$	$66.2_{\pm 0.4}$	$64.8_{\pm 0.4}$	$63.5_{\pm 0.4}$	$62.3_{\pm 0.3}$	$60.9_{\pm 0.3}$	$66.8_{\pm 0.2}$
Finetune†	$82.9_{\pm 0.2}$	$6.2_{\pm 0.2}$	$5.9_{\pm 0.2}$	$5.8_{\pm 0.1}$	$5.3_{\pm 0.1}$	$5.1_{\pm 0.1}$	$5.0_{\pm 0.1}$	$4.6_{\pm 0.1}$	$4.4_{\pm 0.1}$	$13.9_{\pm 0.0}$
EWC†	$82.9_{\pm 0.2}$	$6.2_{\pm 0.2}$	$6.1_{\pm 0.1}$	$5.8_{\pm 0.1}$	$5.3_{\pm 0.1}$	$5.1_{\pm 0.1}$	$4.9_{\pm 0.1}$	$4.6_{\pm 0.1}$	$4.4_{\pm 0.1}$	$13.9_{\pm 0.0}$
ER†	$82.9_{\pm 0.2}$	$77.2_{\pm 0.4}$	$73.4_{\pm 0.3}$	$69.9_{\pm 0.3}$	$66.7_{\pm 0.3}$	$64.1_{\pm 0.3}$	$62.1_{\pm 0.3}$	$60.0_{\pm 0.3}$	$58.2_{\pm 0.3}$	$68.3_{\pm 0.3}$
Joint† (o)	$82.9_{\pm 0.2}$	$80.3_{\pm 0.3}$	$79.3_{\pm 0.3}$	$78.3_{\pm 0.3}$	$77.2_{\pm 0.3}$	$76.6_{\pm 0.2}$	$76.0_{\pm 0.2}$	$75.4_{\pm 0.1}$	$74.8_{\pm 0.1}$	$77.9_{\pm 0.2}$

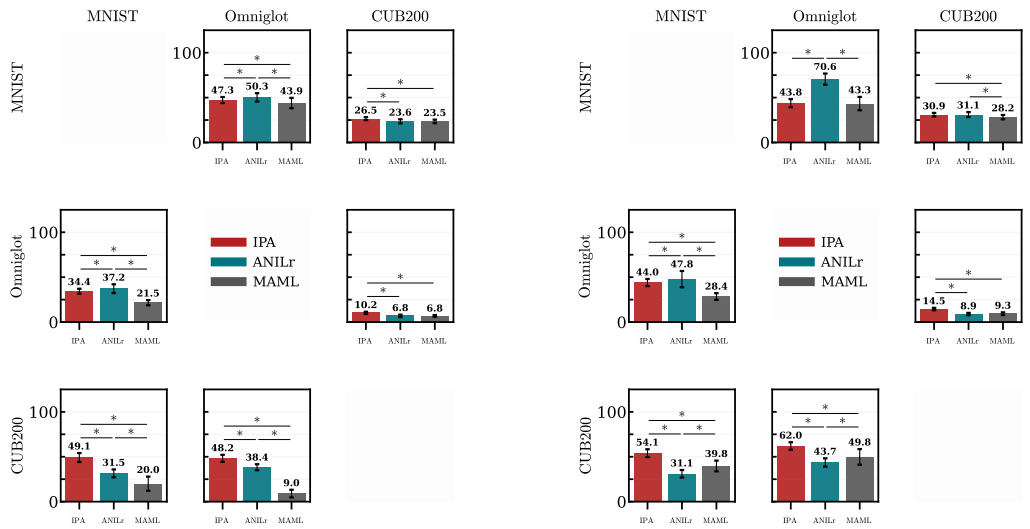
Table 10: Per-session accuracy (%) $\pm$ 95 % CI for **CIFAR-100** with **DINO** backbone (30 reps). S0 is the base session; S1–S8 are incremental sessions. **Bold**: IPA-Approx significantly outperforms any other non-oracle method ($p < 0.05$) without finetuning. (o) oracle upper bound; † are backpropagation algorithms that require additional training.

Method	S0	S1	S2	S3	S4	S5	S6	S7	S8	AA
IPA	$89.6_{\pm 0.2}$	$88.4_{\pm 0.3}$	$87.4_{\pm 0.2}$	$86.6_{\pm 0.2}$	$85.8_{\pm 0.3}$	$85.1_{\pm 0.3}$	$84.2_{\pm 0.3}$	$83.4_{\pm 0.2}$	$82.9_{\pm 0.2}$	$85.9_{\pm 0.2}$
IPA-Approx	$89.8_{\pm 0.2}$	$88.5_{\pm 0.3}$	$87.7_{\pm 0.2}$	$87.1_{\pm 0.2}$	$86.4_{\pm 0.3}$	$\mathbf{85.9_{\pm 0.2}}$	$\mathbf{85.1_{\pm 0.2}}$	$\mathbf{84.5_{\pm 0.2}}$	$\mathbf{84.0_{\pm 0.1}}$	$86.5_{\pm 0.1}$
IPA-Adaptive (o)	$89.6_{\pm 0.2}$	$89.0_{\pm 0.2}$	$88.6_{\pm 0.2}$	$88.3_{\pm 0.2}$	$88.0_{\pm 0.2}$	$87.7_{\pm 0.2}$	$87.2_{\pm 0.2}$	$86.9_{\pm 0.1}$	$86.6_{\pm 0.1}$	$88.0_{\pm 0.1}$
NCM-r	$85.1_{\pm 0.3}$	$83.0_{\pm 0.4}$	$81.0_{\pm 0.5}$	$79.3_{\pm 0.5}$	$77.5_{\pm 0.5}$	$76.5_{\pm 0.5}$	$74.8_{\pm 0.5}$	$73.8_{\pm 0.4}$	$72.7_{\pm 0.3}$	$78.2_{\pm 0.3}$
NCM	$89.1_{\pm 0.2}$	$87.5_{\pm 0.3}$	$86.1_{\pm 0.2}$	$85.1_{\pm 0.2}$	$83.9_{\pm 0.3}$	$83.0_{\pm 0.3}$	$82.0_{\pm 0.3}$	$81.1_{\pm 0.2}$	$80.4_{\pm 0.2}$	$84.2_{\pm 0.2}$
Finetune†	$99.9_{\pm 0.0}$	$14.9_{\pm 1.8}$	$8.7_{\pm 0.4}$	$9.7_{\pm 0.3}$	$10.9_{\pm 0.5}$	$12.6_{\pm 0.5}$	$14.1_{\pm 0.4}$	$15.4_{\pm 0.5}$	$16.1_{\pm 0.6}$	$22.5_{\pm 0.3}$
EWC†	$99.9_{\pm 0.0}$	$14.1_{\pm 1.0}$	$8.7_{\pm 0.3}$	$9.6_{\pm 0.3}$	$11.0_{\pm 0.5}$	$12.8_{\pm 0.6}$	$14.2_{\pm 0.5}$	$15.4_{\pm 0.6}$	$16.2_{\pm 0.7}$	$22.4_{\pm 0.3}$
ER†	$99.9_{\pm 0.0}$	$95.9_{\pm 0.2}$	$92.3_{\pm 0.2}$	$89.5_{\pm 0.3}$	$86.7_{\pm 0.4}$	$84.3_{\pm 0.4}$	$82.1_{\pm 0.4}$	$80.0_{\pm 0.3}$	$78.2_{\pm 0.4}$	$87.7_{\pm 0.3}$
Joint† (o)	$99.9_{\pm 0.0}$	$99.9_{\pm 0.1}$	$99.9_{\pm 0.1}$	$99.9_{\pm 0.0}$	$99.9_{\pm 0.0}$	$99.9_{\pm 0.0}$	$99.9_{\pm 0.0}$	$99.8_{\pm 0.0}$	$99.9_{\pm 0.0}$	$99.9_{\pm 0.0}$

Table 11: Per-session accuracy (%) $\pm$ 95 % CI for **MiniImageNet** with **DINO** backbone (30 reps). S0 is the base session; S1–S8 are incremental sessions. **Bold**: IPA-Approx significantly outperforms most methods ($p < 0.05$) without finetuning (exception for ER: AA). † are backpropagation algorithms that require additional training.

Table 12: Published FSCIL results on the same protocol: NC-FSCIL[1] (Yang et al., 2023), TOPIC[2] (Tao et al., 2020) and GKEAL[3] (Zhuang et al., 2023) (ResNet-18 and ResNet-20 backbones as reported). S0 and S8 are accuracy (%) at the base and final incremental sessions; AA is the mean across all sessions; PD=S0−S8.

	CIFAR-100				MiniImageNet			
	S0	S8	AA	PD	S0	S8	AA	PD
NC-FSCIL[1]	82.52	56.11	67.50	26.41	84.02	58.31	67.82	25.71
TOPIC[2]RS-18	64.10	29.37	42.62	34.61	61.31	24.42	39.64	36.89
GKEAL[3]	74.01	51.40	61.35	22.61	73.59	51.31	60.48	22.28



(a) 1-shot (b) 5-shot

Figure 12: Domain adaptation between different datasets. Models trained on a source dataset (row-specified), are cross-validated on a new domain dataset (column-specified). We adapt new domain tasks to correspond with the original meta-task size (e.g., MNIST-5 is tested on Omniglot-5).

C.11 FSCIL WITH PRE-TRAINED DINO BACKBONE

Comparison of the DINO ViT-S/8 (Caron et al., 2021) backbone for FSCIL (Table 13). IPA enables off-the-shelf FSCIL without any backbone fine-tuning; labeled shots are still used to initialize class centroids. DINO-ViT-S/8 is self-supervised on ImageNet-1k, of which MiniImageNet is a subset, so absolute DINO numbers are not comparable to FSCIL methods trained on base-session data only; all methods below share the same frozen backbone, so comparisons between methods are unaffected. Besides oracle upper bounds, IPA-Approx outperforms all gradient-based methods for CIFAR-100 and is only outperformed by ER(20) on MiniImageNet. Consistent with the ResNet-12 results, IPA-Approx exhibits the lowest performance drop across sessions for both datasets. The same positive backward transfer is visible for CIFAR-100 (Figure 13), while less apparent for MiniImageNet where most classes already show high accuracy at introduction.

Table 13: FSCIL results with frozen **DINO-ViT-S/8** backbone. S0 and S8 are accuracy (%) after the base and final incremental session; AA is the mean over all sessions; PD is performance drop (S0 − S8). (o) denotes oracle upper bounds, † denotes methods that require additional backbone or head training for FSCIL; unmarked IPA variants use only K labeled shots to initialize centroids and apply no gradient updates at deployment. All results averaged over 30 repetitions.

CIFAR-100	S0	S8	AA	PD
IPA	75.29	62.55	68.24	12.74
IPA-Approx	76.43	64.74	**69.65**	11.69
IPA-Adaptive (o)	75.38	71.85	73.64	3.53
NCM	74.36	59.56	66.05	14.80
NCM-r	73.53	60.92	66.80	12.61
Finetune†	82.95	4.39	13.91	78.56
EWC†	82.95	4.37	13.93	78.58
ER†	82.95	58.24	68.29	24.70
Joint† (o)	82.95	74.76	77.86	8.19

MiniImageNet	S0	S8	AA	PD
IPA	89.58	82.86	85.93	6.72
IPA-Approx	89.78	84.04	86.55	5.74
IPA-Adaptive (o)	89.56	86.60	87.98	2.96
NCM	89.07	80.43	84.24	8.65
NCM-r	85.10	72.72	78.19	12.38
Finetune†	99.95	16.09	22.48	83.86
EWC†	99.95	16.23	22.44	83.72
ER†	99.95	78.17	**87.67**	21.78
Joint† (o)	99.95	99.86	99.89	0.09

C.12 FSCIL ADAPTATION DATA SPLIT

We test whether IPA's centroid refinement requires access to the test distribution at deployment. Table 14 compares four conditions on FSCIL with the ResNet-12 backbone (30 repetitions): (A) standard 50/50 test-set split, half for adaptation and half for evaluation; (B) 25/75 split (less adaptation

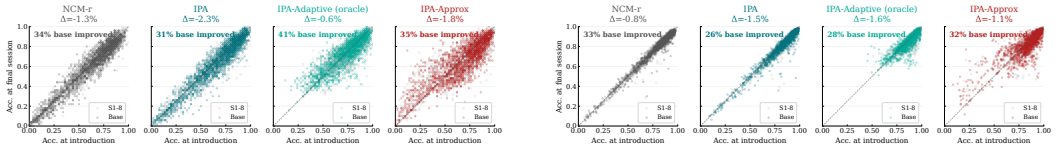


Figure 13: Per-class accuracy at introduction vs. final session under the frozen DINO-ViT-S/8 backbone. Points above the diagonal indicate classes that improved during continual deployment. With DINO features the IPA-Approx variant retains nearly all base classes within $\pm 5\%$ and improves a substantial fraction (CIFAR-100, left; MiniImageNet, right).

data); (C) 50% of the training set for adaptation, evaluated on a disjoint val split (no test data ever seen during adaptation); (D) 50% of val for adaptation, while evaluating accuracy during stream on the full val set (i.e., a realistic setting where we adapt while performing the unlabeled classification task). condition (C) — the strictly inductive regime — changes AA by at most $+0.4$ here; in the DINO-backbone variant of the same ablation it costs 2.0 (CIFAR-100) and 0.75 (MiniImageNet) AA, while condition (D) changes AA by at most 0.03 in all configurations. The transductive advantage of our 50/50 split is therefore at most 1–2% AA, and does not affect the relative ordering. We conclude that IPA requires *any* in-distribution unlabeled stream and did not influence the FSCIL results from using test samples.

Table 14: IPA-Approx under four sources of the unlabeled adaptation stream (ResNet-12 backbone, matching Table 1; 30 repetitions). (A) 50% test / 50% test; (B) 25% test / 75% test; (C) 50% train / 100% val (strictly inductive: no test data during adaptation); (D) 50% val / 100% val. S0 and S_{last} are accuracy (%) at the base and final incremental sessions; AA is the average across all sessions; PD=S0$-S_{last}$.

CIFAR-100 (ResNet-12)

Cond.	Method	S0	S_{last}	AA	PD
A	IPA-Approx	81.61	57.84	68.25	23.77
B	IPA-Approx	81.67	57.58	68.05	24.09
C	IPA-Approx	81.68	58.51	68.61	23.18
D	IPA-Approx	81.68	57.98	68.23	23.70

MiniImageNet (ResNet-12)

Cond.	Method	S0	S_{last}	AA	PD
A	IPA-Approx	91.93	70.36	80.21	21.57
B	IPA-Approx	91.92	70.37	80.21	21.55
C	IPA-Approx	91.92	70.39	80.21	21.54
D	IPA-Approx	91.92	70.40	80.21	21.52

CIFAR-100 (DINO)

Cond.	Method	S0	S_{last}	AA	PD
A	IPA-Approx	73.96	60.67	66.08	13.29
B	IPA-Approx	74.03	60.07	65.71	13.96
C	IPA-Approx	74.12	57.49	64.08	16.64
D	IPA-Approx	74.01	60.66	66.05	13.34

MiniImageNet (DINO)

Cond.	Method	S0	S_{last}	AA	PD
A	IPA-Approx	89.04	83.29	85.75	5.75
B	IPA-Approx	89.02	83.30	85.75	5.72
C	IPA-Approx	89.05	81.80	85.00	7.25
D	IPA-Approx	89.05	83.28	85.75	5.76

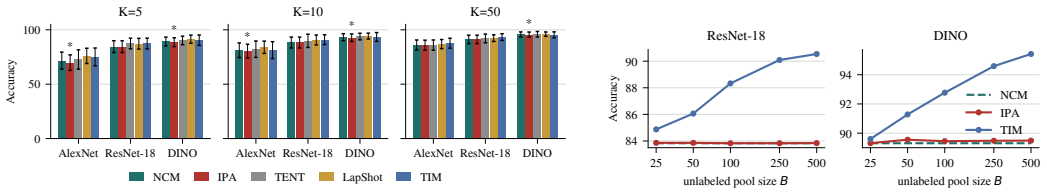
C.13 Data-access-matched baselines

All baselines below receive exactly the same unlabeled data, frozen features, and adaptation budget as IPA (produced during the author response in a single evaluation harness whose feature pipeline differs from that of Table 1, most visibly for the DINO backbone on MiniImageNet, where NCM reaches 80.88 here against 84.24 in Table 13; comparisons are therefore made within this table). FSCIL with the identical 50% unlabeled stream, AA over 30 repetitions; TIM (Boudiaf et al., 2020), TEEN (Wang et al., 2023), TENT (Wang et al., 2021):

Table 15: FSCIL under matched data access (AA %, 30 repetitions). Bold: significantly best over every baseline (paired two-sided t-tests over shared repetitions, $p < 0.001$); on MiniImageNet/ResNet-12 IPA-Approx is statistically tied with TEEN ($p = 0.57$) while keeping the final-session margin over NCM (70.35 vs 69.75).

Config	NCM	TENT	TIM	TEEN	IPA-Approx
CIFAR-100/ResNet-12	66.84	55.52	66.29	66.66	**68.26**
CIFAR-100/DINO	66.05	65.32	67.74	67.60	**69.52**
MiniImageNet/ResNet-12	80.16	66.42	80.03	80.19	80.22
MiniImageNet/DINO	80.88	81.54	82.65	82.65	**83.76**

Every other adaptation method falls below plain NCM in at least one configuration (TENT by 5 to 14 points; TIM and TEEN on CIFAR-100/ResNet-12); IPA-Approx never does. On few-shot episodes (CIFAR-FS features, 5-shot, disjoint evaluation, 30 episodes), batch-transductive methods win balanced query pools — TIM 90.6 vs IPA/NCM 83.8 on ResNet-18 (paired $p < 10^{-7}$), LaplacianShot (Ziko et al., 2020) $+3.2$ — but require the entire query pool jointly; under a 5:1 pool imbalance TIM's margin over NCM shrinks to $+3.6$ (ResNet-18) and $+1.7$ (DINO, no longer significant), while IPA stays within 0.7 of NCM. A training-free Tip-Adapter cache (Zhang et al., 2022) (support-only) adds $+0.1$ to $+0.2$ over NCM. The full grid over backbones and shots is shown in Figure 8, and Figure 14 shows the imbalanced-pool results and the dependence on the unlabeled-pool budget B: restricting both methods to the same B samples, TIM's margin over IPA decays monotonically from $+6.7$ ($B = 500$) to $+1.0$ ($B = 25$, $p = 0.047$) on ResNet-18 and from $+5.9$ to $+0.3$ ($p = 0.59$) on DINO, while IPA is flat across budgets (83.83–83.86 and 89.31–89.55).



(a) imbalanced unlabeled pool (up to 5:1)

(b) unlabeled-pool budget ($K = 5$)

Figure 14: Batch-transductive methods depend on the unlabeled pool. (a) CIFAR-FS 5-way accuracy under a class-imbalanced unlabeled pool (same harness and episodes as Figure 8). (b) Accuracy as a function of the number B of unlabeled samples available to TIM and IPA on the same 30 episodes ($K = 5$, disjoint evaluation); NCM uses no unlabeled data.

C.14 Initialization versus adaptation

Crossing the head initialization with the adaptation mechanism (Table 16) separates the contributions of labeled construction and unlabeled adaptation (CIFAR-100/ResNet-12 FSCIL, AA %, 30 repetitions; random init $\mathcal{N}(0, 1/\sqrt{D})$).

Table 16: Initialization $\times$ adaptation grid, CIFAR-100/ResNet-12 FSCIL (AA %, 30 repetitions; NCM: 66.84).

$\mathbf{W}_0$	none	IPA	entropy-min (TENT)
zero	1.28 (chance)	66.52	1.28 ± 0.00
prototype rows (NCM-style)	67.63	**68.25**	55.52
random	1.36	10.41	1.33 ± 0.13

Prototype initialization plus IPA is best overall, so IPA acts as a refinement operator on an NCM-style initialization. From $\mathbf{W}_0 = 0$ IPA rebuilds near-NCM accuracy from chance: the zero start lies inside the attractor basin. Entropy minimization cannot: at $\mathbf{W} = 0$ the uniform softmax is the entropy maximum, so its gradient is identically zero, and from a random start it reinforces its own noise. The random row does not recover under IPA either, consistent with the contraction result being local rather than global (Section 3.2).

C.15 DEPLOYMENT COST

One IPA update costs $\mathcal{O}(mn)$: measured 619 µs per update on CPU and 152 µs on GPU (inference 13.4 µs/sample); a full FSCIL session takes 6.19 s; persistent state is 250 KiB versus 23.4 MiB for an ER replay buffer, with no optimizer state or backward graph. On the same head, a TENT entropy step measures 292 µs and a backpropagation step 144 µs per batch of 64 on GPU (batched figures; the IPA figures are per-sample latencies). Accuracy versus adaptation steps per class (CIFAR-100/ResNet-12 FSCIL, AA %): $1{:}\,67.76$, $5{:}\,67.96$, $10{:}\,68.05$, $25{:}\,68.17$, $50{:}\,68.26$, $100{:}\,68.25$, $250{:}\,68.13$, $500{:}\,67.85$ (NCM 66.84): one update per class already exceeds NCM and the curve saturates at 25–50 updates per class. A single pass over the stream (predict, one update, never revisit) matches the multi-pass protocol exactly (AA 68.26 ± 0.21 in both).

C.16 SEMI-SUPERVISED META-LEARNING COMPARATOR

A soft-k-means ProtoNet (Ren et al., 2018) meta-trained end-to-end under our protocol (unlabeled refinement inside the meta-training loop; same backbone and episode protocol as Section 4.2) on MiniImageNet 5-way, 30 repetitions $\times$ 600 episodes: at $K{=}1$, ProtoNet 39.32 ± 0.45, soft-k-means 41.32 ± 0.53 (5 unlabeled/class) and 42.08 ± 0.61 (100/class); at $K{=}5$, 64.06 ± 0.25, 65.00 ± 0.26, and 66.55 ± 0.28 (mean $\pm$ 95% CI over repetitions). The refinement gains ($+0.9$ to $+2.8$) match the magnitudes reported by Ren et al. (2018); Meta-IPA reaches 57.60 and 80.36 (Table 7) while adapting on support embeddings only at evaluation, i.e., with strictly less unlabeled data than either variant.

C.17 EVOLVED ABCD AND END-TO-END IB REFERENCES

CMA-ES-evolved ABCD. Evolving the $\mathbf{ABCD}$ parameters with CMA-ES on the same CIFAR features (2.4M sample evaluations versus zero for the closed form): from the same 25 labeled shots and $\mathbf{ABCD} = 0$ the evolved rule reaches 54.1 ± 3.5; seeded with IPA's centroid construction, 84.3 ± 3.3; with $20\times$ the labels from scratch, 80.0 ± 3.6; the IPA closed form reaches 84.4 ± 3.3 (10 evaluation episodes each). The information is carried by the construction, not found by the search. On a matched toy task both rules reach 100% accuracy, but IPA operates at $\mathcal{I}(X;O) = 0.235$ versus the evolved rule's 0.483: the same performance at less than half the information cost, as compression never enters the CMA-ES fitness.

End-to-end IB head. A VIB head (Alemi et al., 2017) trained by gradient on the same K shots and episodes (CIFAR-FS, ResNet-18, 5-shot) reaches 81.19 versus IPA's 83.84; adding entropy minimization on the unlabeled pool at weight 1.0 collapses it to 61.1 ± 6.5, and it requires the weight tuned to 0.1 to reach 85.9. The end-to-end head can win with per-task tuning but is tuning-fragile precisely at deployment, where tuning data does not exist.

Large-label probe. With many-shot centroids on a 100-way task, NCM 62.71, IPA 62.68, IPA one-pass 62.75: adaptation is neutral when prototypes are strong; its gains concentrate in the weak-prototype regime.

C.18 IPA-APPROX ROBUSTNESS UNDER PROTOTYPE NOISE, IMBALANCE, AND NON-STATIONARITY

We stress-test IPA-Approx at benchmark scale on CIFAR-100 FSCIL (ResNet-12 backbone, AA over 30 repetitions per condition). *Imbalanced* subsamples the unlabeled adaptation stream with rotating per-class keep fractions (10:1 skew at most), leaving the evaluation half untouched; *noise* adds Gaussian noise of standard deviation σ times the per-dimension feature standard deviation to every k-shot prototype; *non-stationary* orders the stream blockwise by class; *K=1* uses a single support shot per incremental class. TENT (Wang et al., 2021) is head-only entropy minimization on the identical stream.

IPA-Approx leads NCM in every condition. Under imbalance it loses 0.5 points where TENT loses 5.3; under non-stationarity it is essentially unaffected; and at $\sigma{=}0.5$ its margin over NCM grows to $+1.9$, i.e., the adaptation partially repairs corrupted prototypes. Gains concentrate where prototypes are weak (incremental classes, 1-shot, corruption) and are neutral, never negative, where they are strong. Under the same conditions the closed-form calibration of TEEN exceeds IPA-Approx only

Table 17: Robustness at benchmark scale: CIFAR-100 FSCIL (ResNet-12), AA (%) over 30 repetitions.

Condition	NCM	IPA-Approx	TENT
balanced	66.84	**68.26**	55.52
imbalanced stream	66.84	**67.73**	50.19
noise σ=0.1	66.98	**68.19**	55.60
noise σ=0.5	64.51	**66.44**	55.53
noise σ=1.0	62.90	**63.44**	55.25
non-stationary stream	66.84	**68.23**	56.09
K=1 support	63.92	**65.22**	55.49

at $\sigma \geq 0.5$ (67.12 vs. 66.44 at $\sigma = 0.5$; 64.42 vs. 63.44 at $\sigma = 1.0$), and TIM stays at or below NCM throughout. The toy-problem dynamics of Figure 5 illustrate the mechanism: under an 8:1:1 imbalance the input de-meaning partially neutralizes the shift toward the heavy class, and moderate class-mean drift is tracked through IPA-Approx's own predictions: under a linear drift of about one class separation (3 Gaussians, 20 seeds) IPA-Approx ends at $80.4 \pm 0.4\%$ against $68.4 \pm 0.8\%$ for a static prototype classifier, and an exponential moving average of the prototypes ($\alpha = 0.01$) reaches $95.1 \pm 0.4\%$; under a 60° rotation of the class means the numbers are 86.1, 49.3, and 98.7%. At 90° the unsupervised tracker partially loses lock and at 120° fully, once classes swap basins and their identity is no longer identifiable without labels.