Do Not Imitate, Reinforce: Iterative Classification via Belief Refinement

Anonymous authors
Paper under double-blind review

Abstract

Standard supervised classification trains models to imitate the labels in their training data. Since this typically happens in a single forward pass, models are locked into a fixed compute budget whether an input is simple or complex. Moreover, the imitation objective forces the model to express absolute certainty on its training data, which often carries over into evaluation, leading to overconfident predictions. Rather than forcing a rigid, single-step guess, early learning systems often framed learning as a trial-and-error process of gradual improvement. Reclaiming this perspective for classification, we propose Reinforced Iterative Classification (RIC), a reinforcement learning framework in which a recurrent actor iteratively refines a predictive distribution and a critic estimates the value of continued refinement. We study the properties of RIC and evaluate RIC-SPO, which trains RIC with Simple Policy Optimization. At inference time, the learned value function provides a principled stopping rule, halting when further refinement is no longer useful. Using image classification as a testbed, RIC-SPO matches or exceeds the accuracy of supervised baselines while reducing calibration error, with the largest gains under label noise and fine-grained classes.

1 Introduction

Modern classification models are trained to predict the class distribution in a single forward pass. From a sequential decision-making point of view, this represents a rigid form of imitation learning in which the model is trained to replicate the observed training labels in one shot. Although effective, this training scheme fixes the computation allocated to every input. Moreover, it couples label fitting with confidence growth, meaning easy inputs, ambiguous inputs, and corrupted labels all receive the same update pressure. Consequently, high training accuracy is usually accompanied by increasingly confident predictions. This raises a natural question: can classification instead be learned as a sequence of prediction refinements, with computation allocated according to how much an input still stands to gain?

Reinforcement Learning (RL) provides a natural framework for this question because it learns from evaluative feedback over sequences of decisions (Sutton & Barto, 2018). RL also has a long history of application to classification tasks (Wiering et al., 2011). Prior methods have used RL to learn where to look (Mnih et al., 2014), which layers to run (Wang et al., 2018), which features to acquire (Dulac-Arnold et al., 2011), and when to commit to a hard label (Lin et al., 2020). In these methods, however, RL controls how a classifier computes or when it stops; the full predictive distribution itself is not the action.

We address this gap with Reinforced Iterative Classification (RIC), an RL framework that treats classification as sequential prediction refinement. A recurrent policy maintains a class distribution and updates it over several steps. Each action is a probability vector, and each reward measures the improvement in log score relative to the previous prediction. Like cross entropy, the population RIC objective is optimized by the true conditional distribution. The two objectives share the same optimum but provide different learning signals. Cross entropy differentiates the loss of each prediction directly, whereas RIC updates a stochastic policy using the returns earned by sampled refinements. To support this refinement process, RIC uses a recurrent actor and critic architecture. We show that the state value equals the expected reduction in prediction error from continued refinement. The critic therefore estimates how much useful computation remains. At inference,

refinement stops once this estimate reaches zero, meaning adaptive computation requires neither a ponder penalty nor a separate halting network. This stopping rule allocates extra steps where refinement still reduces error and halts early otherwise. To optimize RIC, we use Simple Policy Optimization (SPO) (Xie et al., 2025), a PPO-style update (Schulman et al., 2017) that limits how far the policy moves between snapshots. We call the resulting method RIC-SPO.

Empirically, this alternative optimization route alters the dynamics of classification training. Accuracy rises first while confidence follows, and calibration emerges during training. RIC-SPO matches supervised accuracy on clean benchmarks, remains calibrated under label noise where cross entropy grows overconfident, and improves both accuracy and calibration on the fine-grained ImageWoof benchmark. Its critic concentrates computation on inputs that benefit from further refinement.

This paper develops RIC at three levels. **(i)** We formulate classification as a Markov decision process (MDP) in which full class probability distributions are actions and stepwise log score improvements are rewards. **(ii)** We show that discounting trains useful intermediate predictions while preserving the true conditional distribution as the optimum, and that state value equals the expected reduction in prediction error from further refinement, yielding a principled stopping criterion. **(iii)** We introduce RIC-SPO, a practical algorithm that trains the RIC framework using SPO. Across image classification benchmarks, RIC-SPO matches or exceeds supervised accuracy while reducing calibration error, especially under label noise.

2 Preliminaries

Supervised classification. We consider a K class classification problem with input space $\mathcal{X}$, label space $\mathcal{Y} = \{1, \ldots, K\}$, and an unknown data distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. We write $\Delta^{K-1} = \{p \in \mathbb{R}^K_{\geq 0} \mid \sum_{j=1}^K p_j = 1\}$ for the probability simplex and $q(\cdot \mid x) \in \Delta^{K-1}$ for the true conditional distribution over labels given input x. Cross entropy is the negative log score, $\mathrm{CE}(p, y) = -\log p_y$, where p_y is the probability that p assigns to class y. We measure calibration with expected calibration error (ECE) (Guo et al., 2017). A sample of n predictions is grouped into M equal width confidence bins $B_1, \ldots, B_M$, and $\mathrm{ECE} = \sum_{m=1}^M \frac{|B_m|}{n} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|$, the weighted average gap between bin accuracy and mean confidence.

Markov decision processes. A standard Markov decision process (MDP) is a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel P, reward function R, and discount factor $\gamma \in (0, 1)$. An agent's behavior is governed by a policy π that maps states to distributions over actions. The value function $V^\pi(s_t) = \mathbb{E}_\pi[\sum_{k=1}^\infty \gamma^{k-1} r_{t+k} \mid s_t]$ represents the expected cumulative discounted reward from state s_t onward. Actor critic methods learn a parameterized policy π_θ (the actor) together with a critic V_ϕ that approximates V^π and serves as a baseline for the policy update.

3 Reinforced Iterative Classification

RIC turns classification into an iterative decision process in which predictions are refined over time. The framework combines a classification MDP, a recurrent actor-critic, and a value-based halting rule. Starting from a uniform distribution, the actor proposes a new class distribution at each step and receives the corresponding change in the observed label's log score as reward. The critic estimates the expected improvement remaining and provides a principled signal for adaptive inference. These components define RIC independently of the learning algorithm used to optimize them. We train RIC with SPO (Xie et al., 2025) and call the resulting method RIC-SPO.

3.1 Classification MDP

RIC turns each labeled example into a small decision problem inside the classifier, where the agent acts on its own class distribution. A training episode is a single example $(x, y) \sim \mathcal{D}$. The reward measures how much the next prediction improves the log score of label y.

Action and state. Each episode begins from $s_0 = (x, a_0, \tau_0)$, where $a_0 = [1/K, \ldots, 1/K]$ is the uniform prior over classes and $\tau_0 = \mathbf{0}$ is the initial latent state. At refinement step $t \geq 0$, the state $s_t = (x, a_t, \tau_t)$

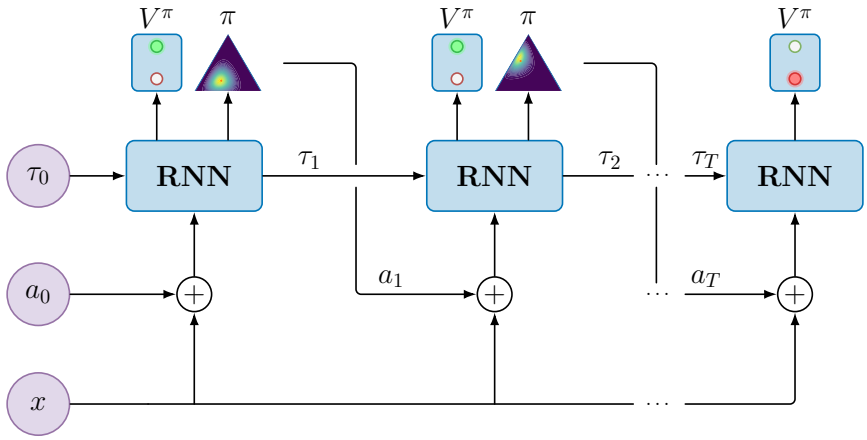

Figure 1: **Recurrent actor-critic architecture for RIC.** The recurrent module updates the latent state from the input and previous prediction. The policy head samples the next class distribution, and the value head predicts future improvement and supplies the stopping rule at inference.

contains the input, the current prediction, and the current latent state. The policy samples the next prediction $a_{t+1} \sim \pi_\theta(\cdot \mid s_t)$ with $a_{t+1} \in \Delta^{K-1}$. The label is used only to compute reward, so the policy conditions on s_t.

Transition. The dynamics are deterministic apart from the policy sample. Given s_t, the recurrent module computes $\tau_{t+1} = f_\theta(x, \tau_t, a_t)$. The sampled prediction then yields the next state $s_{t+1} = (x, a_{t+1}, \tau_{t+1})$. The latent state lets the classifier remember how its belief has been refined so far.

Reward. Standard classification gives the model a loss on a completed prediction. RIC pays each refinement step for the improvement it makes. The dense signal eases credit assignment across refinement steps. Let $a_{t,y}$ denote the probability that prediction a_t assigns to label y. For $t \geq 1$, we define

$$r_t = \log a_{t,y} - \log a_{t-1,y}. \tag{1}$$

The label is fixed within the episode, so each pair (x, y) induces its own MDP in which y is part of the environment. Let

$$J_{x,y}(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \,\middle|\, x, y\right] \tag{2}$$

be the discounted return on a single example, then the average $J(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[J_{x,y}(\theta)\right]$ is maximized.

3.2 Architecture and RIC-SPO

The architecture mirrors the MDP. A recurrent backbone carries the current latent state τ_t and produces τ_{t+1} with x and a_t as input (Figure 1). Then, a policy head samples the next class distribution, and a value head predicts how much refinement remains useful. The policy head parameterizes a Dirichlet distribution $\pi_\theta(\cdot \mid s_t)$ over Δ^{K-1} by producing a mean vector $\mu \in \Delta^{K-1}$ and a concentration parameter c. The action is then sampled as $a_{t+1} \sim \mathrm{Dir}\left(\mu \cdot c + \epsilon\right)$, where ϵ denotes a small safety offset, which counters numerical instability on the simplex border. The value head outputs $V_\phi(s_t) \in \mathbb{R}$. Both heads are reused at every refinement step and share the recurrent backbone.

Because reward is improvement in log score, the value function has a concrete meaning. It estimates the cumulative improvement still available from a state. At inference, RIC evaluates $V_\phi(s_t)$ before spending another refinement step. A positive value continues refinement. A nonpositive value stops and returns the current prediction. At evaluation, we take the Dirichlet mean as the action.

RIC can be trained using different policy optimization algorithms. In this paper, we use SPO as the policy update rule and call the resulting method RIC-SPO. During training, each episode is unrolled for a fixed

Algorithm 1: RIC-SPO

Input: dataset $\mathcal{D}$, horizon T, passes per snapshot P, discount γ, value coefficient c_v

Initialize parameters θ, ϕ ;
for $n \leftarrow 1, 2, \ldots$ **do**
  $\theta_{\text{old}} \leftarrow \theta, \quad \phi_{\text{old}} \leftarrow \phi$; `// snapshot`
  for $p \leftarrow 1$ **to** P **do**
    foreach *minibatch* $\mathcal{B} \subset \mathcal{D}$ **do**
      Roll out $\pi_{\theta_{\text{old}}}$ for T steps on each $x \in \mathcal{B}$;
      $\hat{A}, \hat{R} \leftarrow \text{GAE}(\mathcal{B}, V_{\phi_{\text{old}}}, \gamma)$; `// advantage and return calculation`
      Update θ, ϕ via $\nabla \hat{\mathbb{E}}_{\mathcal{B}}\big[L_{\text{SPO}}(\theta, \theta_{\text{old}}, \hat{A}) - c_v \|V_\phi - \hat{R}\|^2\big]$; `// update model parameters`

horizon of T steps, and value based stopping is reserved for inference. Training proceeds in rounds. We freeze a snapshot $(\theta_{\text{old}}, \phi_{\text{old}})$, collect rollouts with $\pi_{\theta_{\text{old}}}$ for P passes, and update the actor and critic with SPO (Xie et al., 2025) and generalized advantage estimation (Schulman et al., 2016). Algorithm 1 summarizes this loop.

4 Theoretical Analysis

RIC rewrites classification as an RL problem, so the return must have a statistical interpretation. The analysis gives that interpretation in three linked steps. Discounting evaluates predictions at a random refinement depth, the log score selects the true conditional distribution at those depths, and the critic compares current prediction error with expected future prediction error. The propositions characterize the RIC objective and value function independently of the learning algorithm used to train the policy.

4.1 Discounting Trains Predictions at Every Depth

Discounting determines which refinement depths are trained. Because reward is the change in log score, discounted return evaluates the prediction reached at a random stopping step. Earlier refinements receive more weight, but every depth has positive probability, so useful intermediate predictions are part of the objective.

Lemma 1 (Geometric horizon (Puterman, 1994))**.** *For any integrable reward sequence $\{r_t\}_{t\geq 1}$ and discount $\gamma \in (0, 1)$,*

$$\mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t\right] = \mathbb{E}_{N\sim\text{Geom}(1-\gamma)}\left[\sum_{t=1}^{N} r_t\right],$$

where the random horizon N is independent of the rewards.

For RIC, the reward is a log score difference, so the cumulative reward telescopes to the final log score minus the starting log score. Applying Lemma 1 to the single example return of Equation (2) gives

$$\begin{aligned} J_{x,y}(\theta) &= \mathbb{E}_{N\sim\text{Geom}(1-\gamma),\,\pi_\theta}\left[\sum_{t=1}^{N}\big(\log a_{t,y} - \log a_{t-1,y}\big) \,\middle|\, x, y\right] \\ &= \mathbb{E}_{N\sim\text{Geom}(1-\gamma),\,\pi_\theta}[\log a_{N,y} - \log a_{0,y} \mid x, y] \\ &= \mathbb{E}_{N\sim\text{Geom}(1-\gamma),\,\pi_\theta}[\log a_{N,y} \mid x, y] - \log a_{0,y}. \end{aligned} \tag{3}$$

Equation (3) evaluates the prediction reached at a random refinement depth. The return is high when that prediction assigns higher log probability to the label. The term $-\log a_{0,y}$ is constant. It equals $\log K$ under the uniform prior $a_0 = [1/K, \ldots, 1/K]$ and leaves the maximizing policy unchanged. Thus the discounted RL objective trains prediction quality at random refinement depths. This makes RIC an anytime classifier, with supervision supplied by reward in a form close to deep supervision (Lee et al., 2015).

4.2 The Optimal Prediction

Once discounting has placed supervision across refinement depths, the remaining issue is which prediction the log score favors at those depths. The answer is the same distribution selected by cross entropy, the true conditional distribution. Training samples actions on the simplex, so the expected log loss depends on both the average prediction and the spread around that average.

Lemma 2 (Log score decomposition for stochastic predictions (Gneiting & Raftery, 2007))**.** *Fix a state s with true conditional distribution $q = q(\cdot \mid x)$. Let $a \sim \pi(\cdot \mid s)$ have mean $\mu = \mathbb{E}_\pi[a \mid s]$ and assume $a_y > 0$ on the policy support. Then*

$$\mathbb{E}_{y\sim q,\, a\sim\pi}[-\log a_y \mid s] = H(q) + \mathrm{KL}(q \,\|\, \mu) + \mathbb{E}_{y\sim q,\, a\sim\pi}[D_F(a_y \,\|\, \mu_y) \mid s]\,, \tag{4}$$

where $D_F(x \,\|\, y) = x/y - \log(x/y) - 1 \geq 0$. The three terms are label ambiguity, mean prediction error, and spread from sampling.

Equation (4) separates the fixed label ambiguity from the errors the model can reduce. Under realizability, the mean prediction error and the sampling spread can both be removed. Because Equation (3) gives every refinement depth positive weight, an optimal policy cannot compensate for a bad early prediction with a good late one.

Proposition 1 (Optimal prediction target)**.** *If the true conditional distribution $q(\cdot \mid x)$ is realizable, an optimal RIC policy has mean prediction $q(\cdot \mid x)$ at each refinement step and zero sampling term in Lemma 2.*

Proof sketch. Equation (4) identifies the reducible terms, and Equation (3) gives every depth positive weight. Under realizability, any remaining mean error or sampling spread would lower the return, so an optimal policy removes both. The full proof is in Appendix A.2.

Proposition 1 shows that the main difference between RIC and common supervised approaches does not lie in the optimum, which maximizes their objective. Cross entropy pushes the predictions toward its label for as long as any gap remains. In contrast, RIC trains the same prediction through the policy gradient, which moves it only while sampled refinements earn distinguishable returns. Therefore, the main difference is the path how such an optimum can be reached.

4.3 The Critic Measures Remaining Prediction Improvement

At inference, the learned critic determines whether refinement should continue. Proposition 2 gives this decision a statistical meaning. The critic compares the prediction available now with the prediction expected after more refinement.

Stopping at state s_t keeps the current prediction a_t. Its expected log loss is $H(q(\cdot \mid x)) + L_t$, where $L_t = \mathrm{KL}(q(\cdot \mid x) \| a_t)$ is the current prediction error. Continuing under π replaces that current error by the expected error at the random future depth. The label ambiguity $H(q(\cdot \mid x))$ is tied to the input, so it is identical in both losses and cancels. The critic therefore estimates whether refinement lowers the KL error.

Proposition 2 (Value of refinement)**.** *For any policy π and state $s_t = (x, a_t, \tau_t)$, let $L_j = \mathrm{KL}(q(\cdot \mid x) \| a_j)$ denote the prediction error at refinement depth j. The value function under the reward of Equation* (1) *satisfies*

$$V^\pi(s_t) \;=\; L_t \;-\; \mathbb{E}_{N\sim\mathrm{Geom}(1-\gamma),\,\pi}[\,L_{t+N} \mid s_t\,]\,. \tag{5}$$

Proof sketch. Apply Lemma 1 from s_t and average the unknown label under $q(\cdot \mid x)$. The log loss split above turns the current and future log scores into entropy plus KL error. The common entropy cancels and leaves Equation (5). The full proof is in Appendix A.2.

Equation (5) is the stopping rule used in Section 3.2. If L_t is larger than the expected future error, then $V^\pi(s_t) > 0$ and refinement is expected to help. If L_t is not larger, then $V^\pi(s_t) \leq 0$ and the current prediction is at least as good as continuing. The cancellation of the input-dependent label ambiguity $H(q(\cdot \mid x))$ explains why state value is distinct from predictive confidence. Because the critic tracks the remaining reducible prediction error , it separates label ambiguity from model error. For highly ambiguous inputs, the policy

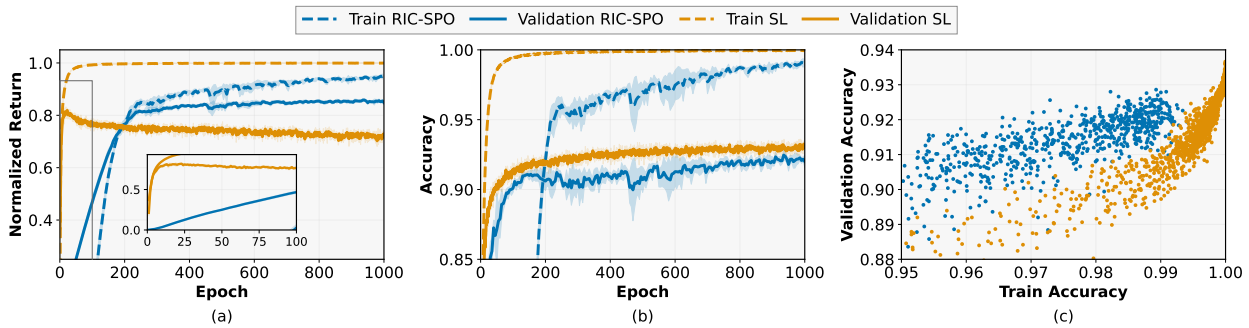

Figure 2: **CIFAR-10 training dynamics of RIC-SPO and SL.** Trained for 1000 epochs with 5 random seeds. (a) Return normalized by the $\log K$ prior offset over training. RIC-SPO improves more slowly because policy gradient updates have higher variance, but attains higher validation return. (b) Classification accuracy on train and validation sets. Both objectives optimize predictive performance, and RIC-SPO reaches comparable accuracy. (c) Evaluation accuracy as a function of training accuracy across epochs. At matched training accuracy, RIC-SPO achieves higher evaluation accuracy, indicating better generalization.

can halt early with a high-entropy prediction as soon as all reducible KL error is eliminated. Conversely, a confident but incorrect prediction will maintain a positive value as long as subsequent refinement steps can steer the distribution closer to q. This mechanism instantiates the classical monitoring rule for anytime algorithms (Russell & Wefald, 1991; Hansen & Zilberstein, 2001), and Proposition 2 shows that the critic trained alongside the policy already estimates it.

5 Results

The preceding sections motivate the RIC formulation as an RL alternative to single-pass supervised learning (SL) and yield concrete predictions about optimization behavior and adaptive computation. The target preservation result predicts that metrics such as accuracy and return remain comparable between the two paradigms. In addition, the value identity predicts that the critic should allocate computation while refinement can still reduce prediction error.

Therefore, we empirically evaluate RIC-SPO against SL, ACT (Graves, 2016), PonderNet (Banino et al., 2021), and Evidential Deep Learning (EDL) (Sensoy et al., 2018) and assess it as a practical image classifier. Our results are organized to **(i)** compare training dynamics and confidence timing against these paradigms, **(ii)** benchmark test set accuracy and calibration across datasets, and **(iii)** analyze value based halting.

5.1 Setup

All methods use the same ResNet-34 backbone (He et al., 2016) to encode an image x into an embedding and are trained end to end. The baselines are chosen so that each shares one ingredient with RIC-SPO. SL predicts from the shared embedding in a single forward pass and uses the standard cross entropy objective. ACT and PonderNet use the same GRU refinement module under supervised adaptive computation objectives. PonderNet's halting prior is matched to the RIC discount ($\lambda_p = 1 - \gamma$). EDL shares the Dirichlet output family. Full hyperparameter settings and sweeps are given in the appendix. RIC-SPO uses a Dirichlet policy over the simplex with discount γ=0.8 and truncated rollouts of T=20 refinement steps with value bootstrapping. At γ=0.8 the residual weight $\gamma^{20} \approx 0.01$, so truncation negligibly affects the infinite horizon objective. We evaluate on CIFAR-10 (Krizhevsky, 2009), CIFAR-10N (Wei et al., 2022b), SVHN (Netzer et al., 2011), and ImageWoof (Howard, 2019), a fine grained subset of ImageNet (Deng et al., 2009). In all plots, shaded bands show 95% confidence intervals across random seeds.

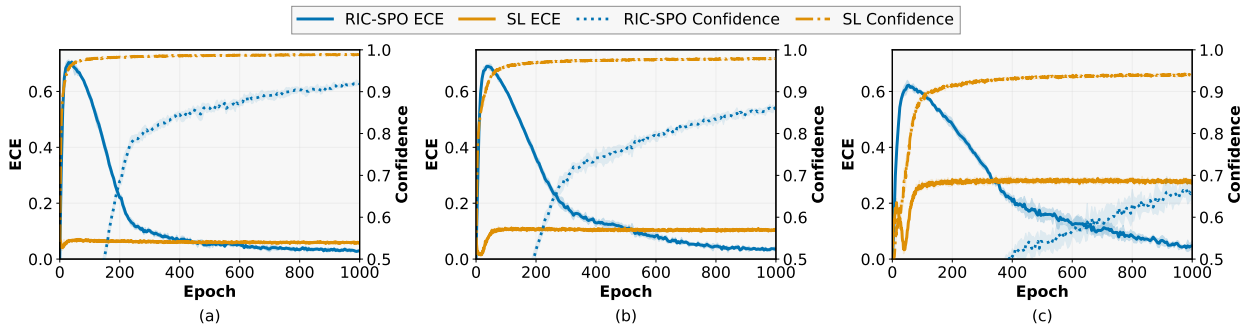

Figure 3: **Calibration dynamics under varying label noise.** Validation ECE and confidence over training time on CIFAR-10 (a), CIFAR-10N with aggregated human labels (9.03% noise) (b), and CIFAR-10N with worst case labels (40.21% noise) (c). Confidence lags accuracy under the SPO update, so RIC-SPO is underconfident early in training. As a consequence ECE peaks above 0.6 independent of label noise. Even so, RIC-SPO keeps lower ECE than SL at a later training stage and remains stable as label noise increases.

5.2 Training dynamics and confidence timing

Figure 2 tests whether the shared target preserves metrics on CIFAR-10. At convergence, RIC-SPO achieves higher validation sequential return and maintains it through the end of training, whereas SL overfits and its validation return declines (Figure 2(a)). However, RIC-SPO improves more slowly in the early phase because policy gradient estimates are noisier. Although RIC-SPO accuracy trails SL for most of training, it ends at a comparable level (Figure 2(b)). In addition, Figure 2(c) reveals a systematic generalization advantage. At a fixed level of training accuracy, RIC-SPO attains higher evaluation accuracy, which suggests that the remaining accuracy gap is a training speed effect. We also notice that RIC-SPO maintains a low validation loss through the later stages of training and, as shown later in this section, exhibits the same persistence in calibration. For supervised learning, both validation performance and calibration peak only ephemerally before continued fitting degrades them.

Figure 3 shows how confidence evolves as label noise increases. The SPO policy update caps how far the class distribution can move between snapshots. Consistent with this, confidence evolves more slowly than accuracy. RIC-SPO is therefore underconfident early, while supervised and evidential objectives let confidence rise as fast as fitting allows. ECE peaks above 0.6 early in training, before confidence catches up to the accuracy already reached. SL models maintain low ECE only during a short early phase before calibration deteriorates as confidence continues to increase. On standard CIFAR-10, RIC-SPO reaches a lower minimum ECE than SL, and this minimum occurs at a later stage when validation accuracy is already high (Figure 2(b)), so calibration improves alongside predictive performance throughout training. The gap widens as label noise increases. Under aggregated human annotations (9.03% noise) and worst case labels (40.21% noise), SL calibration degrades sharply. RIC-SPO keeps predictions on noisy examples away from the simplex boundary and stays well calibrated. Confidence also rises more slowly as label noise increases.

This timing difference is the central training signature of RIC-SPO. Predictive accuracy improves faster than model confidence. Cross entropy couples the two and reaches peak accuracy when confidence is already extreme. The separation is what leaves room for calibration. On clean data, RIC-SPO reaches high accuracy while confidence is still rising, so longer training continues to increase confidence. Under label noise, the target itself is less concentrated, so confidence rises toward a lower level and mislabeled examples keep moderate probabilities.

5.3 Test set calibration and adaptive computation

Figure 4 compares the final predictive distributions on the test set. SL models produce a sharply peaked confidence distribution, concentrating predictions near extreme confidence values. Furthermore, the accuracy of the SL model is substantially lower than its mean confidence. As a consequence, the corresponding reliability diagram reveals systematic overconfidence across all bins. In contrast, RIC-SPO produces a broader confidence distribution and is mildly underconfident across most bins. Accuracy and confidence are

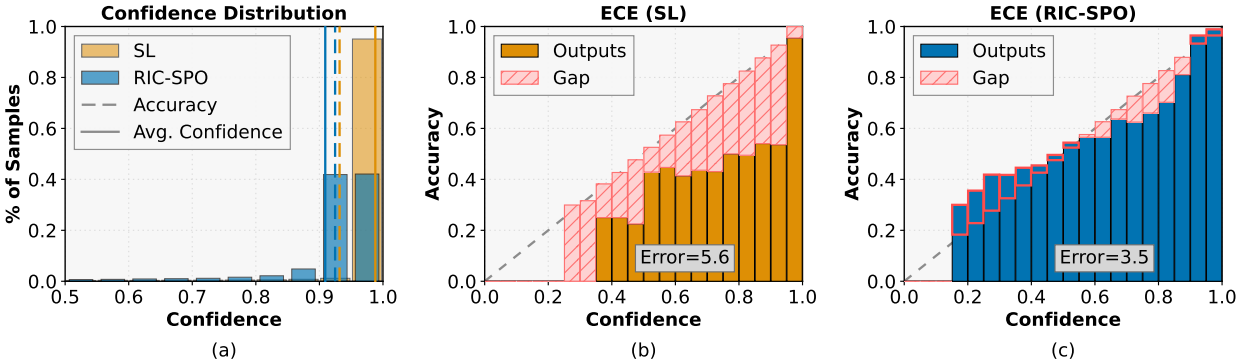

Figure 4: **Confidence distributions and reliability diagrams on CIFAR-10 (test).** (a) Confidence histogram averaged across models trained with five random seeds. (b) Reliability diagram for SL. (c) Reliability diagram for RIC-SPO. SL produces a sharply peaked confidence distribution and exhibits systematic overconfidence. RIC-SPO predictions are more dispersed and mildly underconfident across most bins, which improves calibration compared to SL.

much better aligned. The reliability diagrams are the endpoint of the training dynamics in Figure 3. Cross entropy sharpens early and stays sharp, driving probabilities toward the boundary. RIC-SPO sharpens while refinement still improves prediction error, and its test distribution stays broader and better calibrated.

Table 1 summarizes accuracy and calibration across all datasets. RIC-SPO matches or exceeds SL test accuracy while substantially reducing ECE, and it does so despite lower training accuracy, consistent with the generalization advantage in Figure 2(c). ACT and PonderNet track SL's ECE across datasets, so adaptive recurrence alone does not calibrate.

PonderNet is the closest recurrent supervised baseline in our suite. It trains per-step cross entropy under learned halting weights, with its geometric prior matched to RIC's discount. Its ECE nevertheless remains similar to SL. EDL shares the Dirichlet output family and improves on SL only modestly. The refinement architecture, the depth-weighted objective, and the output family each fail to close the gap on their own. The remaining difference is how the objective is optimized. RIC-SPO learns from sampled predictions under capped policy updates and exhibits the confidence lag in Figure 3. On SVHN, nearly saturated accuracy leaves little room, yet RIC-SPO still nearly halves ECE. On fine grained ImageWoof, RIC-SPO improves both test accuracy (85.4 vs 82.6) and calibration (ECE 0.049 vs 0.102) over SL. These are the largest gains among our benchmarks. Gradual confidence helps most with fine grained classes and limited data.

Figure 5 evaluates the critic interpretation in Proposition 2. The identity predicts that refinement should continue while prediction error can decrease and halt once expected improvement is small. Figure 5(a) shows the stopping behavior on examples grouped by supervised model consensus on the ImageWoof test set. We evaluate five independently trained SL ResNet-34 models on each test image and count how many predict the correct class. Images classified correctly by all five models are labeled Easy, images classified correctly by

	CIFAR-10			SVHN			ImageWoof		
	$\mathrm{Acc}_{\mathrm{train}}$ [%]	$\mathrm{Acc}_{\mathrm{test}}$ [%]	$\mathrm{ECE}_{\mathrm{test}}$	$\mathrm{Acc}_{\mathrm{train}}$ [%]	$\mathrm{Acc}_{\mathrm{test}}$ [%]	$\mathrm{ECE}_{\mathrm{test}}$	$\mathrm{Acc}_{\mathrm{train}}$ [%]	$\mathrm{Acc}_{\mathrm{test}}$ [%]	$\mathrm{ECE}_{\mathrm{test}}$
SL	99.83	92.57	0.059	99.88	95.80	0.033	90.91	82.61	0.102
EDL	99.57	92.27	0.049	99.39	95.84	0.029	87.66	83.78	0.078
ACT	99.78	92.51	0.059	99.80	95.68	0.033	91.90	83.62	0.097
PonderNet	99.78	92.43	0.059	99.83	95.79	0.033	91.52	83.68	0.097
RIC-SPO (ours)	99.33	**93.05**	**0.035**	99.40	**95.86**	**0.018**	86.37	**85.39**	**0.049**

Table 1: **Test accuracy and ECE across CIFAR-10, SVHN, and ImageWoof.** Results are averaged over five seeds. Models are selected by best validation accuracy. RIC-SPO matches or exceeds supervised test accuracy across datasets while consistently reducing calibration error. Appendix Table 3 gives full confidence intervals.

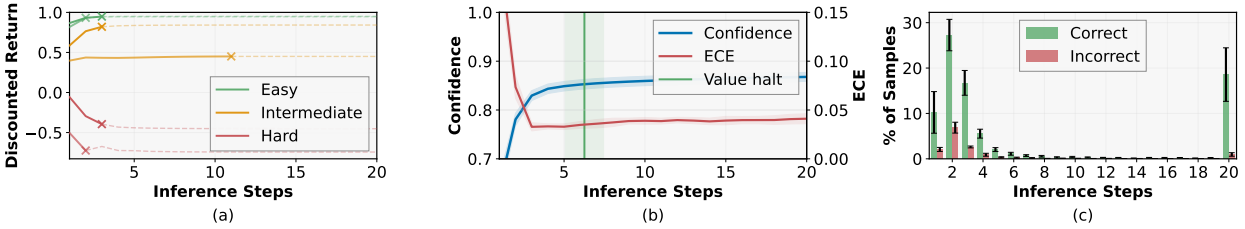

Figure 5: **Adaptive computation analysis on ImageWoof (test).** (a) Normalized return by supervised-consensus difficulty (easy, intermediate, hard). On easy inputs the policy takes confident predictions early and the value function imposes early stopping. On intermediate examples the policy takes more compute and gradually increases return. On hard images the value function stops early when return is about to decrease. (b) Average confidence and ECE as a function of inference steps. The vertical line marks the mean value based stopping step. (c) Distribution of stopping steps for correctly and incorrectly classified inputs.

more than half but not all models are labeled Intermediate, and images classified correctly by at most half of the models are labeled Hard. Easy inputs saturate quickly, intermediate inputs benefit from additional refinement steps, and on hard images the policy stops before additional negative reward accrues.

Figure 5(b) shows confidence rising across refinement steps while ECE falls. At early steps the agent stays cautious and sharpens its belief as more computation is used. As confidence stabilizes, predicted future return approaches zero, and the value rule stops inference once further refinement yields marginal gains. Figure 5(c) confirms that the critic allocates more steps to inputs that are ultimately classified correctly, and stops early on hard images that are misclassified. However, at the same time some images still take the maximum amount of steps. This is mainly an effect of an overestimation of the state value, showing that the performance of value-based halting for adaptive computation is linked to the performance of the value function.

6 Related Work

RIC sits at the intersection of several strands of work that have studied learning from evaluative feedback, adaptive computation, classification, deliberation, and calibration. We organize the related work around the differences to RIC as a formulation in which the class distribution itself is the action, reward measures its stepwise improvement, and the critic decides when refinement should stop.

Reward driven learning. RIC draws on the reward driven view of learning as improvement from evaluative feedback (Sutton & Barto, 2018). Early machine learning already tied learning to improvement through experience, as in Samuel's checkers program (Samuel, 1959). Temporal difference learning later trained predictions from differences between successive predictions (Sutton, 1988). The supervised error correction line, including Widrow and Hoff's adaptive switching circuits, trains from desired outputs by reducing error (Widrow & Hoff, 1960). RIC connects these histories inside a classifier. The class distribution is the action, and reward measures prediction improvement.

Adaptive computation. Adaptive computation methods share RIC's goal of spending compute according to input difficulty, usually through supervised halting machinery. Graves (2016) introduces Adaptive Computation Time, which trades task loss against an explicit ponder cost, and Banino et al. (2021) learn a distribution over stopping times regularized toward a geometric prior. Recurrent refinement with shared weights also appears in the Universal Transformer (Dehghani et al., 2019), and spatially adaptive depth extends this idea to convolutional residual networks (Figurnov et al., 2017). RIC uses the critic as the stopping signal. Proposition 2 gives this rule the classical value of computation interpretation for monitoring anytime algorithms (Russell & Wefald, 1991; Hansen & Zilberstein, 2001).

RL control for classification. RL control methods are closer in training signal, but their actions select classifier machinery rather than predictions. Mnih et al. (2014) learn a sequential attention policy while training the final class prediction with a supervised objective. SkipNet and BlockDrop learn policies that choose which layers or residual blocks to execute for each input, with rewards reflecting both accuracy and

computation (Wang et al., 2018; Wu et al., 2018). Wiering et al. (2011) frame classification as an MDP whose actions edit feature buffers and whose classes are read from state values, Mousavi et al. (2019) train moving agents whose consensus readout receives a single terminal loss, and Lin et al. (2020) emit one hard label per sample with class-weighted accuracy rewards. Feature acquisition methods make stopping itself a discrete classify-now action (Dulac-Arnold et al., 2011; Janisch et al., 2019). Across this line, the action is an attention location, a subnetwork, a feature, or a hard label, and the reward is accuracy or compute cost. To our knowledge, RIC is the first formulation in which the action is the class distribution itself, the reward is the stepwise improvement of a strictly proper score of that action, and the learned value of remaining improvement decides when refinement stops.

Deliberation and recurrent decision making. Recent work treats deliberation and computation control as part of the action space. Hanna & Corrado (2025) formalize deliberation as mental actions in a thought MDP, and Orenstein et al. (2025) give agents explicit compute actions and feedback about compute cost. Latent reasoning in language models studies both the medium and duration of computation before answering (Hao et al., 2025; Ning et al., 2025). Together, these works show that reasoning can move into a latent space and that reward can choose how long it continues. Related findings also support the recurrent design. Recurrent agents trained by RL can develop speed accuracy tradeoffs and changes of mind (Wispinski et al., 2026), and recurrent value networks can learn hidden states correlated with Bayesian beliefs (Lambrechts et al., 2022). RIC places the action directly on the class prediction.

Calibration and uncertainty. Calibration provides another reason to pursue iterative belief refinement. Cross-entropy-trained modern networks are often overconfident (Guo et al., 2017), a behavior Wei et al. (2022a) connect to continued logit-norm growth during training. In language models, PPO-based RLHF can amplify verbalized overconfidence (Leng et al., 2025), and overconfidence can undermine pass@N scaling under repeated sampling (Chen et al., 2025). A common approach is post-hoc adjustment such as temperature scaling (Guo et al., 2017). Training rewards shape predicted uncertainty during learning itself (Damani et al., 2026; Bani-Harouni et al., 2026). RIC follows this route. RIC shares its Dirichlet output family with evidential deep learning, which predicts Dirichlet parameters in a single supervised forward pass to represent uncertainty (Sensoy et al., 2018). Evidential methods emit one input-conditioned Dirichlet in a single forward pass, while RIC samples and refines actions through an RL objective. Juergens et al. (2024) identify limitations in the faithfulness of epistemic uncertainty estimates from evidential methods, including problems of identifiability and convergence.

7 Conclusion

In this work, we introduced Reinforced Iterative Classification, a formulation of classification as reward driven belief refinement. RIC preserves the usual training target, but changes how that target is reached. Our RIC-SPO instantiation matches or exceeds supervised accuracy while improving calibration, particularly under label noise and fine grained classes. More importantly, RIC and supervised learning follow fundamentally different training dynamics. Supervised learning couples accuracy with rapid and often extreme certainty. RIC instead learns from the returns earned by sampled refinements, causing confidence to lag behind accuracy. Consequently, while low calibration error is often only a brief phase of supervised training, RIC-SPO maintains it as a stable regime once high accuracy is reached, even under heavy label noise. Its contribution is therefore not simply better calibration, but a learning process in which accuracy and confidence evolve more favorably. Why these dynamics differences arise remains open, and we intend to study them in future work.

Limitations and Prospects. Relative to supervised learning, RIC-SPO is slow to train because of policy optimization over the simplex requiring many more passes over the data. Its slow convergence may leave much of RIC's potential unrealized due to potential exposition to overfitting during prolonged training. We believe better reinforcement learning algorithms could improve both efficiency and final performance while preserving the training dynamics observed herein. Another noteworthy limitation is The Dirichlet distribution becoming unstable near the simplex boundary; especially as the number of classes grows. This limits our current validation to relatively small label spaces. Scaling RIC will require more stable ways of navigating the simplex. Overall, we believe RIC is a first step towards learning systems that improve their predictions through reinforcement rather than imitation.

References

David Bani-Harouni, Chantal Pellegrini, Paul Stangel, Ege Özsoy, Kamilia Zaripova, Nassir Navab, and Matthias Keicher. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yResLmrVO1.

Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.

Feng Chen, Allan Raventós, Nan Cheng, Surya Ganguli, and Shaul Druckmann. Rethinking fine-tuning when scaling test-time compute: Limiting confidence improves mathematical reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/e8f4eae0a41cab67fdead3aa6b77f083-Abstract-Conference.html.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training LMs to reason about their uncertainty. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ASQ649zdHm.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyzdRiR9Y7.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Gabriel Dulac-Arnold, Ludovic Denoyer, Philippe Preux, and Patrick Gallinari. Datum-wise classification: a sequential approach to sparsity. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, 2011.

Michael Figurnov, Maxwell D Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. Spatially adaptive computation time for residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1039–1048, 2017.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Josiah P. Hanna and Nicholas E. Corrado. When can model-free reinforcement learning be enough for thinking? In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/2a4179ef39846557e99f6bfac580ea2e-Abstract-Conference.html.

Eric A. Hansen and Shlomo Zilberstein. Monitoring and control of anytime algorithms: A dynamic programming approach. *Artificial Intelligence*, 126(1–2):139–157, 2001.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *International Conference on Learning Representations (ICLR)*, 2025.

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*, pp. 770–778, 2016.

Jeremy Howard. Imagenette and imagewoof datasets. https://github.com/fastai/imagenette, 2019. Accessed: 2026-03-03.

Jaromír Janisch, Tomáš Pevný, and Viliam Lisý. Classification with costly features using deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Mira Juergens, Nis Meinert, Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. Is epistemic uncertainty faithfully represented by evidential deep learning methods? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22624–22642. PMLR, 2024. URL `https://proceedings.mlr.press/v235/juergens24a.html`.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Gaspard Lambrechts, Adrien Bolland, and Damien Ernst. Recurrent networks, hidden states and beliefs in partially observable environments. *Transactions on Machine Learning Research*, 2022. URL `https://openreview.net/forum?id=dkHfV3wB2l`.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pp. 562–570. PMLR, 2015.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in LLMs: Reward calibration in RLHF. In *International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=l0tg0jzsdL`.

Enlu Lin, Qiong Chen, and Xiaoming Qi. Deep reinforcement learning for imbalanced classification. *Applied Intelligence*, 50:2488–2502, 2020.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. *Advances in neural information processing systems*, 27, 2014.

Hossein K Mousavi, Mohammadreza Nazari, Martin Takáč, and Nader Motee. Multi-agent image classification via reinforcement learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5020–5027. IEEE, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, 2011. URL `https://research.google/pubs/reading-digits-in-natural-images-with-unsupervised-feature-learning/`.

Alex Ning, Yen-Ling Kuo, and Gabe Gomes. Learning when to stop: Adaptive latent reasoning via reinforcement learning. *arXiv preprint arXiv:2511.21581*, 2025.

Adrian Orenstein, Jessica Chen, Gwyneth Anne Delos Santos, Bayley Sapara, and Michael Bowling. Toward agents that reason about their computation. *arXiv preprint arXiv:2510.22833*, 2025.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994. ISBN 0471619779.

Stuart Russell and Eric Wefald. Principles of metareasoning. *Artificial Intelligence*, 49(1–3):361–395, 1991.

Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2 edition, 2018.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 409–424, 2018.

Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning (ICML)*, 2022a.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022b. URL `https://openreview.net/forum?id=TBWA6PLJZQm`.

Bernard Widrow and Marcian E. Hoff, Jr. Adaptive switching circuits. In *1960 IRE WESCON Convention Record*, volume 4, pp. 96–104, 1960.

Marco A Wiering, Hado Van Hasselt, Auke-Dirk Pietersma, and Lambert Schomaker. Reinforcement learning algorithms for solving classification problems. In *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pp. 91–96. IEEE, 2011.

Nathan J Wispinski, Scott A Stone, Anthony Singhal, Patrick M Pilarski, and Craig S Chapman. Primate-like perceptual decision making emerges through deep recurrent reinforcement learning. *arXiv preprint arXiv:2601.12577*, 2026.

Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8817–8826, 2018.

Zhengpeng Xie, Qiang Zhang, Fan Yang, Marco Hutter, and Renjing Xu. Simple policy optimization. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68813–68824. PMLR, 13–19 Jul 2025. URL `https://proceedings.mlr.press/v267/xie25m.html`.

A Theoretical Supplement

A.1 Regularity Conditions

The results in this paper hold for any continuous simplex-valued policy satisfying the conditions below. The Dirichlet policy of Section 3.2 is one example.

1. **Interior support.** The policy is supported on the interior of the simplex, so $a_k > 0$ on the policy support and the log score $\log a_y$ is finite.
2. **Integrability.** $\mathbb{E}_\pi[\sum_{t=1}^{\infty} \gamma^{t-1} |\log a_{t,y}|] < \infty$, ensuring that the geometric mixture objective and all sum interchange steps are valid.

Verification for the Dirichlet policy. For a $\mathrm{Dir}(c\mu)$ policy with c clipped to $[c_{\min}, c_{\max}]$ and softmax outputs ensuring μ_k bounded away from zero, $\mathbb{E}_\pi[|\log a_y|^2]$ is finite at each step (via digamma/trigamma bounds on the Dirichlet log-moments). Since $N \sim \mathrm{Geom}(1-\gamma)$ has finite moments, the integrability condition follows. Other continuous simplex-valued families satisfy the same conditions under analogous parameter constraints.

Boundary behavior. When $q(\cdot \mid x)$ assigns zero probability to some classes, the optimal action lies on the simplex boundary. The Dirichlet policy, which has interior support, approaches such boundary points as $c \to \infty$. The optimality statements in the main text refer to the supremum of the expected objective.

A.2 Proofs

A.2.1 Derivation of Lemma 2 (Log Score Decomposition for Stochastic Predictions)

Let $q = q(\cdot \mid x)$ and $\mu = \mathbb{E}_\pi[a \mid s]$. For the convex function $-\log$, the associated Bregman divergence is

$$D_F(x\|y) = -\log x + \log y + \frac{x-y}{y}.$$

Setting $x = a_y$ and $y = \mu_y$:

$$D_F(a_y\|\mu_y) = -\log a_y + \log \mu_y + \frac{a_y - \mu_y}{\mu_y}. \tag{6}$$

Rearranging for $-\log a_y$:

$$-\log a_y = -\log \mu_y - \frac{a_y - \mu_y}{\mu_y} + D_F(a_y\|\mu_y).$$

Take expectation under $a \sim \pi$ with $\mathbb{E}_\pi[a_y \mid s] = \mu_y$:

$$\begin{aligned}\mathbb{E}_\pi[-\log a_y \mid s] &= -\log \mu_y - \frac{\mathbb{E}_\pi[a_y \mid s] - \mu_y}{\mu_y} + \mathbb{E}_\pi[D_F(a_y\|\mu_y) \mid s] \\ &= -\log \mu_y + \mathbb{E}_\pi[D_F(a_y\|\mu_y) \mid s],\end{aligned} \tag{7}$$

where the linear correction cancels. Averaging over $y \sim q$ gives

$$\mathbb{E}_{y\sim q,\, a\sim\pi}[-\log a_y \mid s] = \mathbb{E}_{y\sim q}[-\log \mu_y] + \mathbb{E}_{y\sim q,\, a\sim\pi}[D_F(a_y\|\mu_y) \mid s].$$

The first term is the cross entropy between q and μ:

$$\mathbb{E}_{y\sim q}[-\log \mu_y] = H(q) + \mathrm{KL}(q\|\mu).$$

Combining the two displays proves Eq. (4).

Since $D_F(x\|y) = x/y - \log(x/y) - 1 \geq 0$ with equality only when $x = y$, the sampling term is zero exactly when the sampled coordinate equals its mean for the labels weighted by q.

Optional local approximation for intuition. The same point has a local Taylor view. Start from $-\log a_y$ and expand around $a_y = \mu_y$.

$$-\log a_y = -\log \mu_y + \frac{1}{\mu_y}(\mu_y - a_y) + \frac{1}{2\mu_y^2}(a_y - \mu_y)^2 + O\left(\frac{(a_y - \mu_y)^3}{\mu_y^3}\right).$$

Substitute into the Bregman definition and cancel linear terms.

$$D_F(a_y \| \mu_y) = \frac{(a_y - \mu_y)^2}{2\mu_y^2} + O\left(\frac{(a_y - \mu_y)^3}{\mu_y^3}\right).$$

Take expectation.

$$\mathbb{E}_\pi[D_F(a_y \| \mu_y)] \approx \frac{\mathrm{Var}_\pi(a_y)}{2\mu_y^2}. \tag{8}$$

Combining with the identity above gives

$$\mathbb{E}_\pi[\log a_y] \;\approx\; \log \mu_y - \frac{\mathrm{Var}_\pi(a_y)}{2\mu_y^2}.$$

The penalty is proportional to the squared coefficient of variation of the action. For a $\mathrm{Dir}(c\mu)$ policy, $\mathrm{Var}_\pi(a_y) = \mu_y(1-\mu_y)/(c+1)$, so the penalty scales as $O(1/c)$.

A.2.2 Proof of Proposition 1 (Optimal Prediction Target)

We give a direct proof that avoids any independence assumption between refinement steps. The argument has four parts.

Step 1. Geometric stopping reduces the objective to terminal log score. By Lemma 1 and telescoping of the log score improvements in Eq. (1), the example level objective equals $J_{x,y}(\pi) = \mathbb{E}_{N\sim\mathrm{Geom}(1-\gamma),\, a_{1:N}\sim\pi}[\log a_{N,y}] - \log a_{0,y}$. Since a_0 is constant, maximizing $J_{x,y}$ is equivalent to maximizing $\mathbb{E}_{N\sim\mathrm{Geom}(1-\gamma),\,\pi}[\log a_{N,y}]$.

Step 2. Expand by terminal time. Since $\Pr(N = t) = (1-\gamma)\gamma^{t-1} > 0$ for all $t \geq 1$, the law of total expectation gives

$$\mathbb{E}_{N\sim\mathrm{Geom}(1-\gamma),\,\pi}[\log a_{N,y}] = \sum_{t=1}^{\infty}(1-\gamma)\gamma^{t-1}\,\mathbb{E}_\pi[\log a_{t,y}].$$

The decomposition uses only the law of total expectation over N and requires no independence between steps.

Step 3. The log-score decomposition bounds each per-step term. Fix any step t and condition on x. Let $\mu_t = \mathbb{E}_\pi[a_t \mid x]$. Lemma 2 gives

$$\mathbb{E}_{y\sim q(\cdot|x),\, a_t\sim\pi}[-\log a_{t,y} \mid x] = H(q(\cdot \mid x)) + \mathrm{KL}(q(\cdot \mid x)\|\mu_t) + \mathbb{E}_{y\sim q(\cdot|x),\, a_t\sim\pi}[D_F(a_{t,y}\|\mu_{t,y}) \mid x].$$

Equivalently,

$$\mathbb{E}_{y\sim q(\cdot|x),\, a_t\sim\pi}[\log a_{t,y} \mid x] \leq -H(q(\cdot \mid x)),$$

with equality only when $\mu_t = q(\cdot \mid x)$ and the sampling penalty is zero.

Step 4. Positive weights force optimality at every step. The full objective is upper bounded by replacing each per-step term with its best value $-H(q(\cdot \mid x))$. Under realizability, a policy whose mean prediction equals $q(\cdot \mid x)$ and whose sampling penalty is zero at every step achieves this bound. Since every weight $(1-\gamma)\gamma^{t-1}$ is strictly positive, any optimal policy π^* must attain this value at every t. The optimal value conditioned on input x is

$$\mathbb{E}_{y\sim q(\cdot|x)}[J_{x,y}(\pi^*)] = -H(q(\cdot \mid x)) - \mathbb{E}_{y\sim q(\cdot|x)}[\log a_{0,y}].$$

Since a_0 is uniform, this becomes $-H(q(\cdot \mid x)) + \log K$.

A.2.3 Proof of Proposition 2 (Value of Refinement)

Step 1. Geometric stopping rewrites the value. Fix a state $s_t = (x, a_t, \tau_t)$. The state contains x and the current prediction, but not the label. Hence conditioning on s_t averages labels under $q(\cdot \mid x)$. Applying Lemma 1 from step t gives

$$V^\pi(s_t) = \mathbb{E}_{y\sim q(\cdot|x)}\, \mathbb{E}_{N\sim \mathrm{Geom}(1-\gamma),\, \pi}\left[\sum_{k=1}^{N} r_{t+k} \mid s_t, y\right].$$

Step 2. The reward increments telescope. For a fixed label and rollout, the log-score increments telescope:

$$\sum_{k=1}^{N} r_{t+k} = \sum_{k=1}^{N} (\log a_{t+k,y} - \log a_{t+k-1,y}) = \log a_{t+N,y} - \log a_{t,y}.$$

Therefore

$$V^\pi(s_t) = \mathbb{E}_{N\sim \mathrm{Geom}(1-\gamma),\, \pi}\left[\mathbb{E}_{y\sim q(\cdot|x)}\left[\log a_{t+N,y} - \log a_{t,y}\right] \mid s_t\right]. \tag{9}$$

The exchange of the label and rollout expectations is valid because the policy and transition do not observe y.

Step 3. Averaging the label converts log scores into prediction errors. For any fixed prediction a in the simplex interior, Lemma 2 with zero sampling penalty reduces to the usual log-score identity

$$\mathbb{E}_{y\sim q(\cdot|x)}[\log a_y] = -H(q(\cdot \mid x)) - \mathrm{KL}(q(\cdot \mid x)\|a). \tag{10}$$

Apply Eq. (10) to the current prediction a_t and to each sampled future prediction a_{t+N}. Define

$$L_j = \mathrm{KL}(q(\cdot \mid x)\|a_j).$$

Then

$$\begin{aligned}\mathbb{E}_{y\sim q(\cdot|x)}\left[\log a_{t+N,y} - \log a_{t,y}\right] &= \left[-H(q(\cdot \mid x)) - L_{t+N}\right] - \left[-H(q(\cdot \mid x)) - L_t\right] \\ &= L_t - L_{t+N}.\end{aligned} \tag{11}$$

Step 4. Substitution gives the claim. Substituting Eq. (11) into Eq. (9) yields

$$\begin{aligned}V^\pi(s_t) &= \mathbb{E}_{N\sim \mathrm{Geom}(1-\gamma),\, \pi}\left[L_t - L_{t+N} \mid s_t\right] \\ &= L_t - \mathbb{E}_{N\sim \mathrm{Geom}(1-\gamma),\, \pi}\left[L_{t+N} \mid s_t\right],\end{aligned} \tag{12}$$

which is the claim. The regularity conditions in Appendix A.1 justify the expectations and the geometric sum.

B Additional Experimental Results

B.1 Hyperparameters

A short overview of the hyperparameter sweep during training is shown in Table 2. We tuned optimizer choice, activation function, and batch size over the options listed there, with optimizer $\in \{\mathrm{SGD}, \mathrm{Adam}\}$, activation $\in \{\mathrm{ReLU}, \mathrm{SiLU}\}$, and batch size $\in \{128, 256\}$. For Adam, we used a learning rate of 3×10^{-4}, weight decay of 0.001, and gradient clipping of 0.5, while for SGD we used standard settings including momentum 0.9. No learning rate scheduler was used. For the table, supervised baselines were trained for 300 epochs and RIC for 2000 epochs, independent of the dataset. The number of passes per snapshot was fixed to 5. To make compute budgets comparable to our RIC method with discount $\gamma = 0.8$, we matched both adaptive methods to the same geometric horizon. We set PonderNet's geometric prior to $\lambda_p = 1 - \gamma = 0.2$ (implying $\mathbb{E}[N] = 1/\lambda_p = 5$) and used the standard KL weight $\beta = 0.01$, while keeping

Table 2: Hyperparameter sweep considered in our experiments.

Group	Hyperparameter	Value / Options
Optimizer	Choice	{SGD, Adam}
Optimizer	Learning rate	$3 \times 10^{-4}, 0.1$
Training	Batch size	{128, 256}
Model	Activation	{ReLU, SiLU}

`max_steps` = 20. For ACT, we fixed the numerical/stability constants to paper defaults ($\epsilon = 0.01$, positive halting-bias initialization) and set the ponder penalty to $\tau = -\ln(\gamma) = 0.223$, which corresponds to a geometric continuation probability γ under a linear compute cost. For EDL, the evidential head used the standard evidence parametrization, converting pre-activations to evidence via clipped exponentiation and Dirichlet parameters as $\alpha = \exp(\text{clip}(z, -10, 10)) + 1$. Models were optimized with the Dirichlet Bayes risk cross entropy loss plus a KL regularizer to the uniform Dirichlet prior with weight 1.0, linearly annealed over the first 10 epochs. These settings give an analytically matched starting point for compute comparisons, while each method still learns its own step count.

The parameters of the RIC Dirichlet policy are given by $\alpha = \mu \cdot c + \epsilon$. The concentration scale is clipped to $c \in [1, 10]$ with $\epsilon = 0.01$ added for numerical stability. A clipping ratio $\epsilon = 0.25$ is used for the SPO objective and the critic is trained with GAE$-\lambda = 0.5$

For low resolution datasets such as CIFAR-10, we modify the encoder stem to better preserve spatial information, using a 3×3 stride 1 convolution and removing the initial max pooling layer to avoid early spatial downsampling. The original ResNet architecture is otherwise preserved, with the activation function selected from the sweep in Table 2. Encoding and latent state dimensions are set equal. All experiments were conducted on NVIDIA L40 GPUs.

B.2 Results

The appendix gives validation accuracy, reliability diagrams, full confidence intervals, and the ablation. Figure 6 shows validation accuracy as a function of training accuracy for CIFAR-10 (a), SVHN (b), and ImageWoof (c). RIC-SPO achieves better generalization, with the gap increasing on harder datasets. Figure 7 and Figure 8 show confidence distributions and reliability diagrams on SVHN and ImageWoof (test). Both models produce sharply peaked confidence distributions due to the high accuracy on SVHN. At high accuracies, the calibration advantage is reduced. On ImageWoof, SL exhibits systematic overconfidence, whereas RIC-SPO predictions are more dispersed and underconfident on all bins, improving calibration. Table 3 gives the same trend with test accuracy, train accuracy, and ECE confidence intervals across CIFAR-10, SVHN, and ImageWoof. RIC-SPO achieves competitive test accuracy across datasets while reducing calibration error. Figure 9 shows a small ablation study. The Dirichlet head improves stability and training speed over a logistic Gaussian head. Replacing SPO with PPO gives RIC-PPO, a second algorithm for the same MDP. In this ablation, its faster training comes with higher calibration error. SPO is designed to constrain probability ratios within the trust region more effectively than PPO (Xie et al., 2025).

	CIFAR-10			SVHN			ImageWoof		
	$\text{Acc}_{\text{train}}$ [%]	Acc_{test} [%]	ECE_{test}	$\text{Acc}_{\text{train}}$ [%]	Acc_{test} [%]	ECE_{test}	$\text{Acc}_{\text{train}}$ [%]	Acc_{test} [%]	ECE_{test}
SL	99.83 ± 0.02	92.57 ± 0.17	0.059 ± 0.002	99.88 ± 0.03	95.80 ± 0.04	0.033 ± 0.0001	90.91 ± 0.81	82.61 ± 0.50	0.102 ± 0.005
EDL	99.57 ± 0.09	92.27 ± 0.18	0.049 ± 0.002	99.39 ± 0.15	95.84 ± 0.12	0.029 ± 0.002	87.66 ± 1.11	83.78 ± 0.62	0.078 ± 0.008
ACT	99.78 ± 0.10	92.51 ± 0.14	0.059 ± 0.002	99.80 ± 0.02	95.68 ± 0.09	0.033 ± 0.0009	91.90 ± 0.57	83.62 ± 0.29	0.097 ± 0.003
PonderNet	99.78 ± 0.04	92.43 ± 0.23	0.059 ± 0.002	99.83 ± 0.07	95.79 ± 0.15	0.033 ± 0.0005	91.52 ± 0.48	83.68 ± 0.22	0.097 ± 0.004
RIC-SPO (ours)	99.33 ± 0.33	$\mathbf{93.05 \pm 0.07}$	$\mathbf{0.035 \pm 0.004}$	99.40 ± 0.20	$\mathbf{95.86 \pm 0.20}$	$\mathbf{0.018 \pm 0.006}$	86.37 ± 0.55	$\mathbf{85.39 \pm 0.28}$	$\mathbf{0.049 \pm 0.010}$

Table 3: **Test accuracy, train accuracy, and ECE with confidence intervals across CIFAR-10, SVHN, and ImageWoof.** RIC-SPO matches or exceeds supervised test accuracy across datasets while improving calibration.

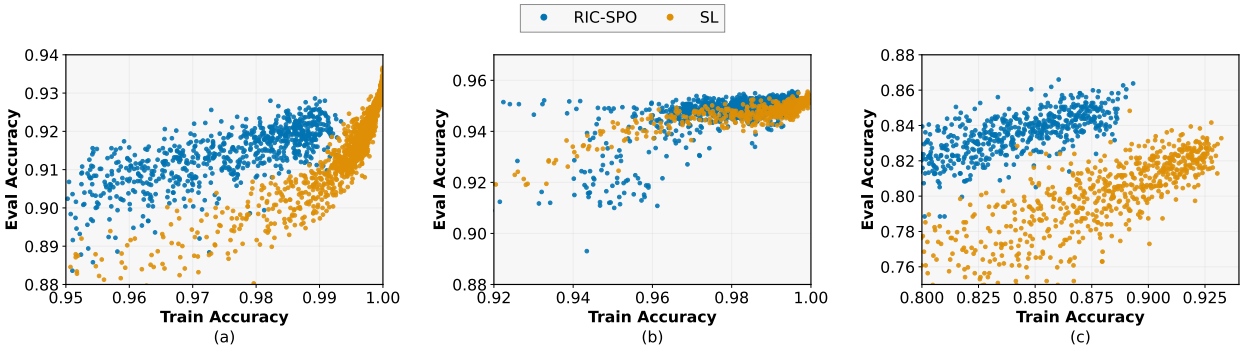

Figure 6: **Validation accuracy as a function of training accuracy.** (a) On CIFAR-10, (b) on SVHN, (c) on ImageWoof. RIC-SPO achieves better generalization.

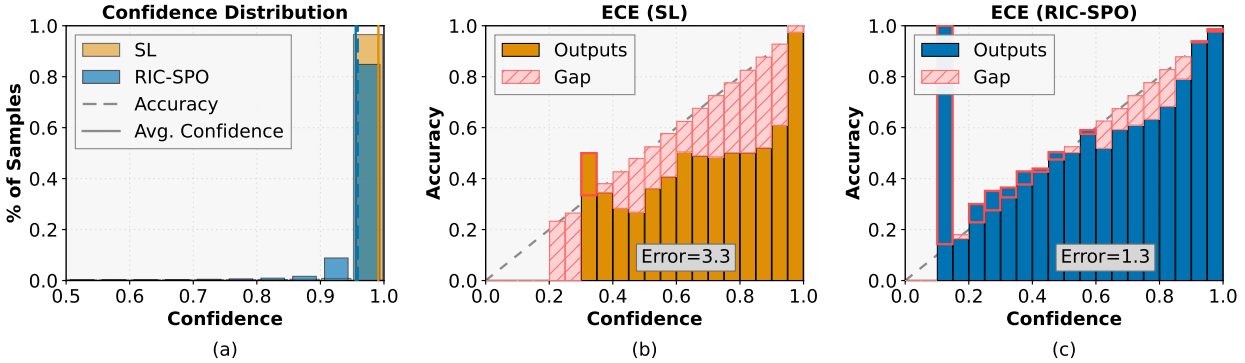

Figure 7: **Confidence distributions and reliability diagrams on SVHN (test).** (a) Confidence histogram averaged across models trained with five random seeds. (b) Reliability diagram for SL. (c) Reliability diagram for RIC-SPO.

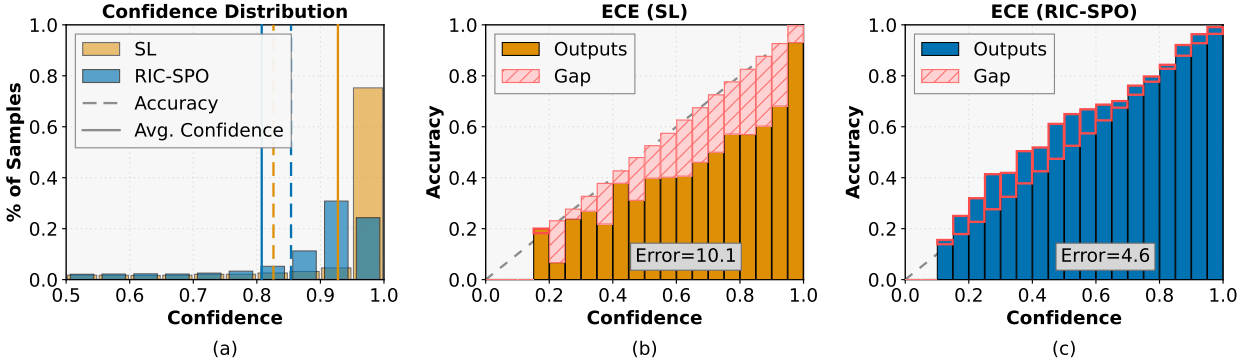

Figure 8: **Confidence distributions and reliability diagrams on ImageWoof (test).** (a) Confidence histogram averaged across models trained with five random seeds. (b) Reliability diagram for SL. (c) Reliability diagram for RIC-SPO.

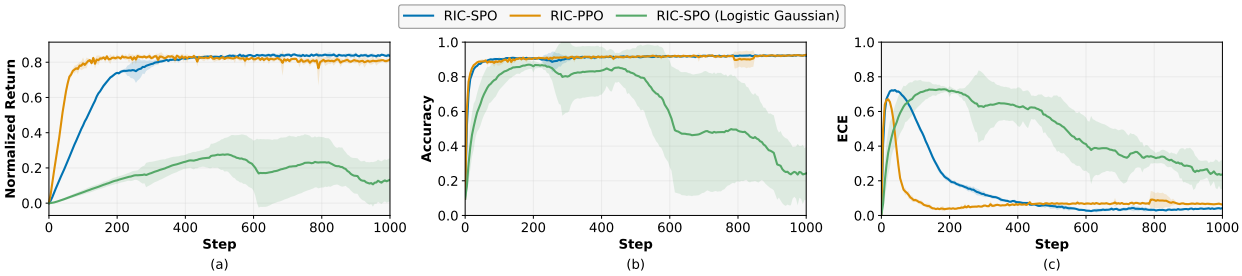

Figure 9: **Ablation study.** (a) Normalized return, (b) validation accuracy, and (c) ECE over training time for ablations. Standard RIC-SPO (blue) with Dirichlet head outperforms the logistic Gaussian head in stability and training speed. RIC-PPO, the same MDP trained with a PPO update, trains faster and calibrates worse.