GRADIENT-MOMENTUM COUPLING: A PARAMETER-SPACE PROXY FOR LEARNING PROGRESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Measuring learning progress is at the core of curiosity-driven exploration, which rewards an agent for going where its model is still learning. However, the abstract notion of learning progress is not directly measurable, and existing methods often derive it from the prediction error in the output space. This paper proposes Gradient-Momentum Coupling (GMC), which measures how strongly a sample drives change in the parameter space, given by the normalized absolute product of its gradient with the momentum of previous gradients. Directions of change that persist across samples accumulate in momentum, while noise cancels out. In controlled experiments GMC allocates near uniform priority across tasks with varying levels of noise, where prediction error chases the noisiest, and orders learnable tasks by improvement speed rather than difficulty. On four MiniGrid MultiRoom tasks, substituting GMC for prediction error inside the Intrinsic Curiosity Module (ICM) recovers exploration ICM loses to unpredictable observations.

1 INTRODUCTION AND BACKGROUND

A central challenge in reinforcement learning is efficient exploration, particularly in environments with sparse or deceptive rewards (Sutton & Barto, 1998). Intrinsic motivation signals address this by providing internal rewards that guide sample selection, curriculum design, and prioritization (Schmidhuber, 2010; Oudeyer et al., 2007). The goal is to identify where the agent's model is still making progress, a goal we share with the prior literature. What counts as progress, however, is not settled.

Schmidhuber (1991) proposed rewarding an agent for the reduction in prediction loss that an observation produces. In the approaches that follow, a learned dynamics model predicts some aspect of the environment (e.g., the next observation), and the intrinsic reward is derived from how well it does so. Intelligent Adaptive Curiosity (Oudeyer et al., 2007) estimates that reduction empirically within regions of the sensorimotor space, while the Intrinsic Curiosity Module (ICM) (Pathak et al., 2017) and Random Network Distillation (RND) (Burda et al., 2019b) use the prediction error itself, under the assumption that large errors mark situations worth prioritizing. These differ in what is predicted and in whether the loss or its change is used, but all of them read progress off the model's output.

Failure Modes of Loss-Based Curiosity. Two critical issues undermine standard loss-based curiosity approaches. First, the *noisy-TV problem* (Burda et al., 2019a): when prediction errors arise from aleatoric noise (irreducible randomness inherent in the environment, i.e. television static), they indicate uninformative surprises, not learning opportunities. An agent optimizing for prediction error will perpetually be drawn to noisy transitions, even after all learnable structure has been captured (Burda et al., 2019a). Even methods that modify the representation (Pathak et al., 2017) or prediction target (Burda et al., 2019b) still use instantaneous prediction error as the intrinsic signal, a proxy that remains susceptible to confounding noise with learnable structure.

Second, *the loss is an indirect view of what a model has learned*, and the two can come apart. Training can plateau followed by abrupt improvement (Power et al., 2022), and transient increases in loss can precede better performance. A signal defined as change in loss reports no progress across such a plateau and negative progress during such an increase, which would say that the model stopped learning and then unlearned, rather than that the loss is an incomplete view of its state. The decoupling may also run the other way, since a model can build structure of which only some part ever helps the

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.

task at hand. In reinforcement learning the distance between the two is larger still, because the agent is judged on return rather than on fit, and because the data the model learns from is produced by the agent's own exploration rather than given in advance.

To address these issues, we propose to measure change in the model's parameters directly, rather than change in loss. Momentum, the exponentially-weighted average of recent gradients, accumulates the directions that persist across samples and cancels those that do not, filtering random noise (which points in inconsistent directions across samples) and oscillatory dynamics (which cancel over time). A hard but learnable task may initially produce weakly aligned gradients, and as a useful representation forms its persistent component grows and accumulates in momentum. Pure noise, by contrast, has no sustained direction to accumulate once its predictable statistics have been fitted. GMC therefore prioritizes samples that drive the directions the model is already moving in, and defers weakly aligned ones, which can be confounded with noise, until the representation is more mature. The momentum vector over all model parameters reveals which parameters are currently undergoing the most change, and comparing a sample's gradient with this per-parameter activity signal enables prioritization by the proposed proxy.

In this paper, we denote *task performance* as the quantity an agent is ultimately judged on, the predictive model's loss or the policy's return, and we reserve *learning progress* for change in what the model has internalized, which may not always be directly reflected in the loss. Learning progress in this sense is an abstract target rather than a defined quantity, and prediction error, its change over time, and GMC are all proxies for it. Persistent parameter change is *necessary* for a model to train at all, but like a falling loss it is not *sufficient* for task performance to improve, since some of what a model internalizes may never turn out to be useful for the task at hand.

Relation to Prior Work. Our approach builds on learning-progress intrinsic motivation (Schmidhuber, 2010; Oudeyer et al., 2007), but shifts the measurement from external performance metrics to an internal quantity, parameter change as observed through gradient-momentum interaction. Prior formulations operationalize learning progress through observable changes in predictive performance, such as Schmidhuber's *prediction gain* (Schmidhuber, 1991; 2010) and Intelligent Adaptive Curiosity (Oudeyer et al., 2007), which takes the empirical derivative of prediction error within regions of the sensorimotor space. Learning Progress Monitoring (LPM) (Hou et al., 2026) also targets noise-robust exploration, and does so in output space by training a second model of how the prediction error evolves, which refines the estimator of a loss-difference quantity rather than the quantity that is read.

This is complementary to curiosity architectures that determine *what* to predict: ICM (Pathak et al., 2017) uses inverse dynamics to filter uncontrollable features, RND (Burda et al., 2019b) predicts fixed random targets for novelty, and Variational Information Maximizing Exploration (VIME) (Houthooft et al., 2016) maximizes information gain. Recent work on multi-step inverse dynamics (Efroni et al., 2021; Islam et al., 2023) extends single-step filtering to provably remove exogenous distractors, improving the *representation* on which curiosity is measured. GMC scores learning on whichever representation these methods produce, and could replace the prediction error used within them.

Curriculum learning (Bengio et al., 2009; Graves et al., 2017) designs explicit training schedules that progress from easy to hard, whereas GMC proposes a signal from which such prioritization can emerge directly, without manual scheduling.

Prior work has also used gradient information for sample prioritization. Gradient-norm methods (Katharopoulos & Fleuret, 2018; Zhao & Zhang, 2015) weight samples by $\|\nabla_\theta \mathcal{L}\|$ to reduce variance in SGD updates, and CurES (Zeng et al., 2025) selects samples whose gradients align with a held-out validation set, both targeting *optimization efficiency* in supervised settings with reliable labels and a fixed dataset. GMC instead targets online selection under aleatoric noise, since momentum-based filtering provides robustness to noise and the signal can be computed without reference to a validation set.

Closest within RL, Linke et al. (2020) study weight change $\|\Delta w\|$ as an intrinsic reward for tabular parallel general value functions. This rewards the magnitude of change but not whether successive changes share a direction, making it closest to the gradient-magnitude baseline we call NormAll (Section 3.1), which is degraded by aleatoric noise in our controlled experiments, whereas GMC couples the gradient to the per-parameter momentum and operates over deep dynamics models. See Appendix B for an extended discussion of related work.

2 METHOD

Section 1 motivated parameter dynamics as a proxy for learning progress, using momentum as a noise-filtered summary of ongoing change. The goal is to quantify how strongly a single sample's gradient contributes to ongoing parameter change. Consider the momentum $\mathbf{m} \in \mathbb{R}^d$ summarizing the current direction and magnitude of change across all d model parameters. A single sample's dynamics loss gradient $\mathbf{g} = \nabla_\theta \mathcal{L}_{dyn}(x) \in \mathbb{R}^d$ enters $\mathbf{m}$ through the exponential moving average, $\mathbf{m} \leftarrow \beta\mathbf{m} + (1-\beta)\mathbf{g}$, so the increment it induces, $\Delta_\mathbf{m} = (1-\beta)(\mathbf{g} - \mathbf{m})$, is small relative to $\|\mathbf{m}\|$. A first-order expansion of the induced change in momentum magnitude, $\|\mathbf{m} + \Delta_\mathbf{m}\| - \|\mathbf{m}\|$, splits into a term that is identical for every sample and a term that is not. Appendix A.1 derives the sample-dependent part, the *relative impact* of a sample on the momentum magnitude, up to a positive factor shared across samples, as:

$$\rho(x) \;=\; \frac{\mathbf{m}^\top \mathbf{g}}{\|\mathbf{m}\|} \tag{1}$$

This ranks samples by how strongly they accelerate or decelerate the ongoing parameter changes. Both reinforcing and opposing the current direction indicate the sample is *relevant* to actively-learned parameters, so the absolute value should be applied. Without this, the method would preferentially sample confirmatory examples.

Gradient-Momentum Coupling (GMC). Equation 1 measures the sample's influence on momentum as a single vector. We instead measure influence per parameter, applying the absolute value elementwise so that parameters changing in different directions contribute independently. For normalization across parameters, we replace $\|\mathbf{m}\|$ with a per-parameter scale

$$s_i = \max\!\big(\hat{v}_i,\, \alpha^2 \hat{q}_i\big)\,, \tag{2}$$

where $\hat{v}_i$ and $\hat{q}_i$ are the exponentially-weighted means of the squared gradient and squared parameter, respectively, and α is a constant. Different parameters can have gradient magnitudes that differ by orders of magnitude, so a global norm would let high-magnitude parameters dominate the signal. The maximum retains the gradient-based normalization when it is informative, while preventing a very small second moment from amplifying the score relative to the current parameter scale. Appendix A.4 compares design choices empirically. Formally, GMC is defined as:

$$r_{\text{gm}}(x) = \frac{1}{\sqrt{d}} \sum_i \left| \frac{\nabla_{\theta_i} \mathcal{L}_{dyn}(x) \cdot \hat{m}_i}{s_i} \right| \tag{3}$$

Here x is the current sample, i indexes all d model parameters, $\nabla_{\theta_i} \mathcal{L}_{dyn}(x)$ is the gradient of the dynamics loss with respect to parameter θ_i, and $\hat{m}_i$ is the gradient momentum. The score uses statistics accumulated through the previous batch and updates m, v, and q only after scoring the current samples. Thus a sample is compared against prior learning dynamics rather than statistics containing itself. $r_{\text{gm}}(x)$ adds $O(d)$ storage and elementwise work for d scored parameters, the same asymptotic order as the backward pass (see Appendix A.6).

We compute GMC on a reward-independent dynamics model rather than on the policy or value parameters. Scoring samples by the policy's own gradients would make the intrinsic reward depend on the updates it induces, and before any extrinsic reward is found those gradients carry little stable signal. A predictive dynamics objective is dense and grounded in observed transitions, and it lets GMC replace the prediction-error term used by architectures such as ICM.

3 EXPERIMENTS

We pose three questions: (1) Does GMC produce meaningfully different task prioritization than loss-based signals? (2) Can GMC distinguish learnable structure from aleatoric noise? (3) Does GMC improve exploration when it replaces prediction error inside a full RL pipeline?

To answer (1) and (2), we design controlled adaptive-sampling experiments where task difficulty and noise can be precisely manipulated, isolating the learning-progress signal from temporal credit assignment and policy optimization. To answer (3), we compute ICM's intrinsic reward with GMC on a family of sparse-reward MiniGrid navigation tasks.

3.1 Signal Evaluation under Controlled Difficulty and Noise

3.1.1 Environment

This reduced environment provides no extrinsic reward. The agent's sole objective is to prioritize tasks according to the learning-progress proxy supplied as its intrinsic signal. This mimics the situation in sparse-reward RL where no reward has yet been discovered and exploration must be guided entirely by intrinsic motivation. By observing which samples each signal prioritizes, we can infer its implicit answer to what counts as progress, and how it tackles different levels of noise.

The agent consists of an actor and an image classifier, which we call the dynamics model by analogy with RL. Every episode is a single step, as in a multi-armed bandit setting (Sutton & Barto, 1998). During the episode: the actor selects an action, without receiving any environment or task state; the environment returns an image together with a *target* drawn from the 10 classes; the dynamics model predicts that target and the episode ends. This is analogous to an RL agent choosing which part of an environment to visit, each of which presents different learning opportunities, but without state or temporal credit assignment. The dynamics loss $\mathcal{L}_{dyn}$ is the cross-entropy between the dynamics model's 10-way prediction and the target supplied by the environment. To emulate observations, we sample images from MNIST (Lecun et al., 1998) and CIFAR-10 (Krizhevsky et al., 2009). Targets are reassigned within groups, as described below, so a target is in general not the image's original dataset label. Each of the 10 dataset classes is assigned to one of four groups: $G_0 = \{0\}$, $G_1 = \{1, 2\}$, $G_2 = \{3, 4, 5\}$, and $G_3 = \{6, 7, 8, 9\}$. We test two conditions that differ in what an action selects and in how the target is drawn, giving controlled variation in task difficulty and in irreducible noise, targeting questions (1) and (2) respectively:

Condition 1 - Curriculum: Before training, each image is permanently assigned a target drawn uniformly from the classes of its own group (e.g., a cat image becomes "dog") and keeps it for the whole run. The target is therefore a deterministic and fully learnable, but arbitrary, function of the image. The 10 reassigned classes are the actions: selecting class c returns an image whose fixed target is c, so here the target coincides with the chosen action. The groups become four tasks of differing intrinsic difficulty: the single class of G_0 is untouched, whereas each class of G_3 pools images from four visually distinct source classes and must essentially be memorized. We report results per group by aggregating the classes belonging to each group.

Condition 2 - Noise: The four groups are the actions, and the target is redrawn uniformly from the selected group's classes *each time* an image is drawn, so it is unpredictable from the image and the action fixes only the distribution the target comes from. Groups G_0–G_3 therefore carry 0%, 50%, 67%, and 75% irreducible label noise respectively (larger groups offer more equally valid targets).

3.1.2 Compared Signals

We compare six methods for computing intrinsic rewards from the dynamics model:

- **Uniform**: No intrinsic reward $r_{\text{int}} = 0$; baseline with random sampling across groups.
- **Curiosity**: Prediction loss of the classifier $r_{\text{int}} = \mathcal{L}_{dyn}$; canonical curiosity baseline using loss as proxy for learning.
- **Gradient-Momentum Coupling (GMC)**: $r_{\text{int}} = d^{-1/2} \sum_i |\nabla_{\theta_i} \mathcal{L}_{dyn} \cdot \hat{m}_i / \max(\hat{v}_i, \alpha^2 \hat{q}_i)|$; our proposed method.
- **NormOut**: Squared L_2 norm of the gradient of the dynamics loss with respect to the model's output, $r_{\text{int}} = \|\nabla_z \mathcal{L}_{dyn}\|_2^2$, where z denotes the 10 output logits; like Curiosity it is measured in output space, but it looks at gradient magnitude rather than loss itself.
- **NormAll**: Squared L_2 norm of the gradient of the same loss with respect to all d model parameters, $r_{\text{int}} = \|\nabla_\theta \mathcal{L}_{dyn}\|_2^2$; takes the same gradient-magnitude idea into parameter space, and is similar to GMC but without the momentum weighting and per-parameter normalization.
- **DeltaLoss**: $r_{\text{int}} = |\bar{\mathcal{L}}_{g,t-1} - \bar{\mathcal{L}}_{g,t-2}| / (N_{g,t-1} + N_{g,t-2})$, a numerical approximation of the derivative of group g's loss with respect to the number of samples that group has received; tests direct learning-progress estimation from observed loss changes.

The agent, or selection policy, is trained using policy gradient with entropy regularization (Sutton & Barto, 1998). Because the random policy input is independent of the environment, the actor learns a selection distribution over groups or classes. The policy loss is therefore:

$$\mathcal{L}_{\text{policy}} = -\mathbb{E}_{\pi}[\log \pi(a) \cdot r_{\text{int}}] - \lambda_{\text{ent}} H(\pi) \tag{4}$$

Here r_{int} is the intrinsic signal from the corresponding method, $\pi(a)$ is the probability assigned to the selected action a by the actor, and λ_{ent} weights the entropy term $H(\pi) = -\sum_{i \in \text{all actions}} \pi(a_i) \log(\pi(a_i))$ to prevent collapse to a single group. λ_{ent} is shared across all methods. Because the policy-gradient term scales with the magnitude of r_{int}, every learned signal is passed through the same automatic scale calibration, so that no method is favored or penalized against the entropy term by its raw scale alone (Appendix A.6). Increasing λ_{ent} moves every method toward uniform allocation, while decreasing it sharpens allocations toward collapse.

At the end of each epoch, we compute a *Balanced Evaluation Loss*: the average loss over an equal number of samples from each class or group. In the Curriculum condition, evaluation uses the same fixed image-label assignments as training and therefore measures a memorization fit. In the Noise condition, labels are re-randomized per sample, leaving an irreducible floor. To visualize task prioritization, we report "Training Allocation / Uniform": the mean per-class visitation within each group during a training epoch, divided by the visitation each class would receive under uniform sampling. All runs in this section use 20 random seeds with different dataset shuffles and network initializations. Shaded regions show 95% confidence intervals (CI) following Patterson et al. (2024). These are computed as $\bar{x} \pm t_{0.975,\, n-1} \cdot \sigma / \sqrt{n}$ where σ is the sample standard deviation and n the number of seeds. Two variants of each experiment were run, one for the MNIST dataset and one for CIFAR-10. All reports are computed separately for each dataset. Further implementation details are available in Appendix A.6.

3.1.3 Results: Curriculum Condition

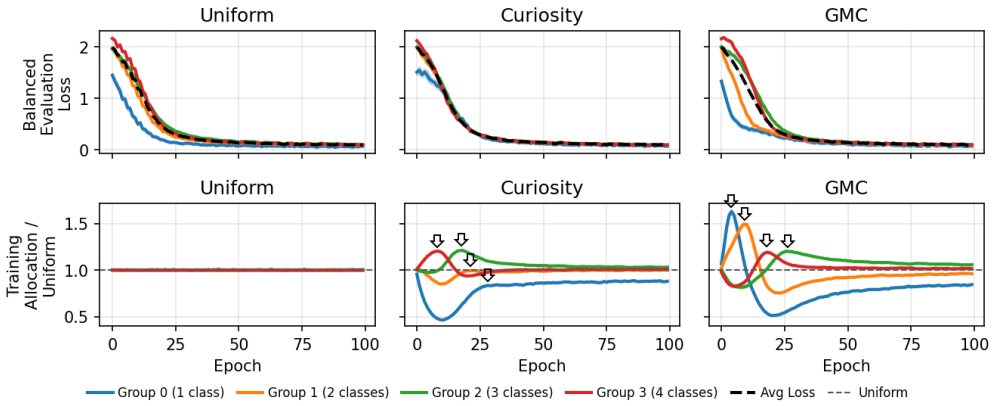

Figure 1: CIFAR-10 Curriculum Condition. **Top:** Per-group Balanced Evaluation Loss. **Bottom:** Mean per-class training allocation within each group. Arrows mark each group's visitation peak.

This experimental condition creates tasks of varied difficulty by scrambling the labels once, over varying group sizes. A zero loss is theoretically achievable through memorization, but within the model's structural limitations and training time, this is not achieved. This section compares how Uniform, Curiosity and GMC vary in the prioritization of tasks with respect to difficulty.

Figure 1 reveals each method's implicit learning-progress measure through its sampling allocation. Uniform has no prioritization, and it reveals some differences in difficulty between the groups. Group size controls how many labels can be assigned, but it does not determine the empirical difficulty by itself, because the particular classes placed in each group also matter. Notably, the four-class G_3 starts with the highest loss of all groups under Uniform sampling, yet its curve crosses below the three-class G_2 partway through training. Curiosity shifts focus based on absolute loss magnitude, which can be observed in how the losses become similar for all groups, the loss curves contract. GMC produces an almost opposite pattern, and its loss curves spread out instead.

The difference in prioritization is clearest in the training allocation, where each group's visitation peaks at a different time. Curiosity peaks in the order $G_3 \to G_2 \to G_1 \to G_0$, which in this case coincides with the ordering from highest initial loss to lowest. GMC peaks in the order $G_0 \to G_1 \to G_3 \to G_2$, which appears to favor the groups whose loss is falling fastest at the time. We see that GMC visits G_2 last even though it sits below G_3 for the first part of training under Uniform sampling. For G_3 to catch up and cross below G_2, its loss must have been descending more steeply over that interval, and that is where GMC directs its budget. The pattern is one of taking the quick wins first and deferring the groups whose loss is moving slowly, even when that means postponing progress on a group that is still far from converged. GMC's prioritization of G_0 is visible in the early loss curves, where G_0 falls faster than under any other method, indicating that the early allocation is spent productively on this group. In contrast, for Curiosity G_0 visitation drops rapidly at the start since it focuses on groups with higher absolute loss. The MNIST (Appendix A) results show more pronounced effects due to larger difficulty differences between groups.

To test whether this ordering reflects the particular classes placed in each group, Appendix A.2 repeats the experiment on two additional partitions with randomly drawn class-to-group assignments and the same group sizes, which produce visibly different difficulty orders among the groups, and GMC's visitation order changes with them. GMC does not converge faster than Uniform or Curiosity in this condition (Figure 12), so in this simple setting the signal produces a difference in prioritization rather than a convergence advantage.

3.1.4 Results: Noise Condition

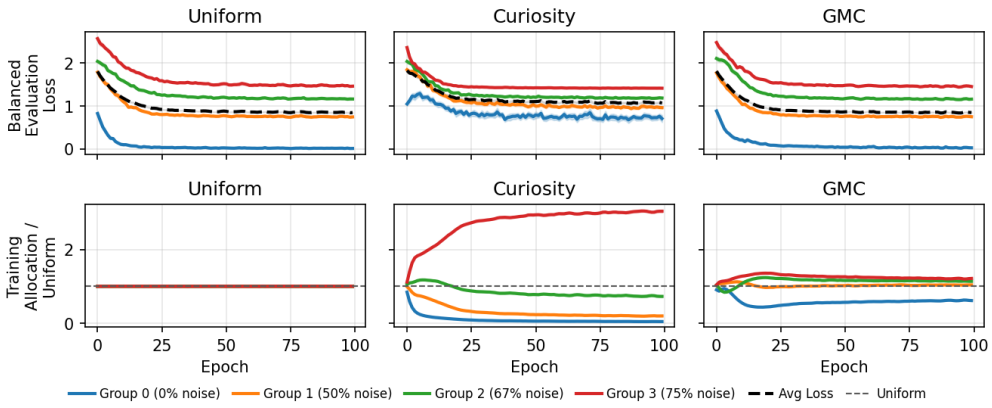

Figure 2: CIFAR-10 Noise Condition. **Top:** Per-group Balanced Evaluation Loss under uniform sampling (0%–75% label noise). **Bottom:** Mean per-class training allocation within each group.

This experimental condition creates tasks with varied amount of aleatoric noise by scrambling the labels each time, over varying group sizes. A zero loss is now impossible, and methods distracted by noise are expected to fail. This section compares the noise robustness of Uniform, Curiosity and GMC.

As Figure 2 shows, Curiosity dramatically over-allocates to the noisiest group, whereas GMC maintains near-uniform allocation across noise levels. Furthermore, GMC shows a slight bias against the no-noise group after saturation. This could be expected from the numerical properties of its expression. When losses plateau the momentum becomes small $\mathbf{m} \approx \mathbf{0}$, however the gradients are only small for the no-noise group $|\nabla_{\theta_i} \mathcal{L}_0| \approx 0$. In the no-noise case, GMC multiplies two near zero values, whereas in the other groups only one value is near zero. The MNIST variant (Appendix A) shows similar patterns.

3.1.5 Overall Convergence Across Methods

The per-group figures above cover Uniform, Curiosity and GMC. Table 1 summarizes all six methods on CIFAR-10 by the area under the overall Balanced Evaluation Loss curve. In the noise condition Curiosity and NormOut are the two methods clearly above the Uniform level, both are driven by

Condition	Uniform	Curiosity	GMC	NormOut	NormAll	DeltaLoss
Curriculum	1.00 ± 0.01	0.99 ± 0.00	1.01 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	$\mathit{1.14 \pm 0.01}$
Noise	1.00 ± 0.00	$\mathit{1.23 \pm 0.00}$	0.99 ± 0.00	$\mathit{1.34 \pm 0.00}$	$\mathit{1.07 \pm 0.00}$	1.02 ± 0.00

Table 1: Normalized area under the overall Balanced Evaluation Loss curve on CIFAR-10, scaled so that Uniform is 1.00 in each condition. This table reports failures in italics.

the magnitude of an output-space error and spend their budget on the noisiest groups. NormAll sits between them and Uniform, so moving the same gradient-magnitude idea from output space into parameter space reduces the damage. GMC and DeltaLoss stay at the Uniform level, and the loss curves in Appendix A show that they and Uniform reach the same final loss, with DeltaLoss slower to get there. In the curriculum condition all six methods end at the same loss, and only DeltaLoss converges visibly more slowly.

This disqualifies NormOut and NormAll as noise-robust signals, and leaves DeltaLoss as the one baseline that filters noise as GMC does. However, DeltaLoss estimates progress from outside the model by differencing per-task loss averages over completed epochs, which requires those averages to be tracked per task and makes the estimate visibly unstable, as the oscillations in its allocation show. LPM (Hou et al., 2026) takes that approach further and addresses the instability by fitting a separate model of how the prediction error evolves, which makes it a trained component with its own dynamics rather than a quantity that can be substituted for r_{int} in place, as every signal compared here is. DeltaLoss reaches the same prioritization order as GMC here, but GMC arrives at that order by measuring parameter dynamics directly, without externally tracking an estimate of rate of change. Per-group losses and allocations for NormOut, NormAll and DeltaLoss in all four conditions are given in Appendix A.

3.2 MultiRoom Benchmark: GMC as a Filter for ICM

Having analyzed GMC's properties in isolation, we next integrate it into a full RL pipeline on a standard sparse-reward benchmark, addressing question (3). GMC and ICM (Pathak et al., 2017) address different parts of the problem, as discussed in Section 1 and Appendix B. ICM determines which features the dynamics model represents, while GMC determines how a sample is scored on those features. ICM's inverse-dynamics objective does not remove unpredictability from the signal, it relocates it into the learned feature space. MultiRoom lets us test a scenario where prediction error degrades as an intrinsic signal wherever the feature space retains variation the forward model cannot resolve. We keep the ICM architecture and its training objective unchanged and replace only the quantity used as the intrinsic reward, which isolates whether scoring the same features with GMC avoids that degradation.

3.2.1 Setup

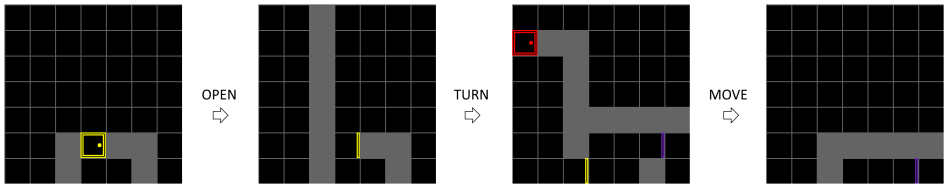

Figure 3: Four consecutive observations from a MultiRoom episode, with the action taken between them shown above the arrows. Opening a door reveals part of the next room, turning right and moving forward hides it again. The view is egocentric and partial, so what a rotation brings into view is not contained in the observation that precedes it.

We evaluate on the MiniGrid MultiRoom family (Chevalier-Boisvert et al., 2023), in which the agent must pass through a chain of rooms connected by closed doors and reach a goal tile in the final room. A single sparse reward is given on reaching the goal, and no other reward is provided. Observations are partial and egocentric, so the agent sees a small view cone in front of itself and receives no

information about what lies behind it. The relevant actions are turn left, turn right, move forward, and toggle, for opening doors. Figure 3 shows a short sequence of observations and the actions that produced them. We run four instances of increasing sizes (2, 3, 4, and 6), with the step budget scaled to the difficulty of each. Each configuration is run for 50 random seeds, and shaded regions show 95% confidence intervals computed as in Section 3.1.

Compared Methods. We compare three methods, all built on Proximal Policy Optimization (PPO) (Schulman et al., 2017). **PPO** is vanilla PPO with no intrinsic reward. **ICM** is PPO augmented with ICM (Pathak et al., 2017), where a learned feature encoder, inverse dynamics model, and forward dynamics model in the learned feature space provide intrinsic reward as the forward model's prediction error. **ICM+GMC** retains the full ICM architecture and trains it with the standard ICM loss, but computes the intrinsic reward as GMC (Equation 3) on the ICM forward model rather than as its prediction error.

3.2.2 Results

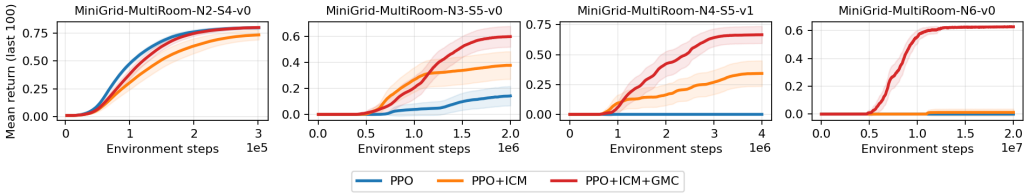

Figure 4: MiniGrid MultiRoom mean episodic return. Each panel is one environment, ordered by increasing size and step budget. The advantage of ICM+GMC over ICM grows as the task gets harder.

The advantage of ICM+GMC over ICM grows with environment size (Figure 4). In the smallest environment exploration is not the bottleneck, and the fact that ICM ends just below PPO there indicates that its intrinsic reward already biases the policy away from the extrinsic optimum when no exploration help is needed, which computing the same reward with GMC avoids. From three rooms onward the two intrinsic methods separate in the order ICM+GMC, ICM, and the gap between them persists once PPO stops reaching the goal altogether, so the sparse reward is findable without intrinsic motivation only in the smaller instances. By six rooms ICM's advantage over PPO has essentially vanished although the same intrinsic reward was clearly useful at three and four rooms, whereas ICM+GMC still solves the task, so what degrades with scale is the quality of the intrinsic signal rather than the usefulness of having one.

The MultiRoom observation is egocentric and partial, and the turning actions rotate the view cone. What a turn reveals is not contained in the observation preceding it (Figure 3), so the forward model cannot predict it from the current state and action however long it trains, and from the model's point of view the residual is aleatoric. ICM's inverse-dynamics objective does not suppress this and to some extent encourages it, since rotations are among the actions most easily identified from a pair of consecutive observations and keeping the visible layout in the encoding is what makes them identifiable. The features that make the inverse model accurate are therefore the same ones whose forward prediction stays unreliable. Its consequence is that ICM's prediction error keeps rewarding rotation, so the larger the environment the more of the budget is spent turning rather than crossing rooms. GMC scores the same features but is robust to unpredictable variation, for the reasons described in Section 3.1, so unlike ICM it is not drawn to the rotation residual.

Appendix A.3 reports the behavioral quantities that this account predicts. In all four environments ICM assigns a persistently larger share of its intrinsic reward to rotation than ICM+GMC does, and correspondingly turns more and moves forward less, with the difference widening as the environments grow.

4 Discussion and Conclusion

The controlled experiments reveal noise robustness and distinct prioritization behavior, while the MultiRoom study shows that substituting GMC for prediction error inside ICM restores exploration that ICM loses on the MultiRoom family of tasks.

The Value of Depth. NormOut and NormAll apply the same gradient-magnitude idea in output space and in parameter space respectively, so comparing them isolates what measuring in parameter space adds. NormOut fails under noise while NormAll is less affected. A possible explanation is that a noisy target produces a large output gradient, but that gradient attenuates through backpropagation and its contribution to the full-network sum is proportionally smaller.

The Value of Momentum Weighting. The comparison of NormAll to GMC serves as an ablation study on the importance of momentum weighting for detecting learning saturation. In the MNIST curriculum condition (Appendix A), once the three easier groups saturate, NormAll increases sampling of the hardest group only to match the other groups, resulting in slow convergence. GMC, in contrast, dramatically boosts sampling of the unsaturated group, producing a clear curriculum signal. This seems to illustrate the benefit of momentum weighting, GMC leverages it to better detect learning saturation, while gradient magnitude alone does not.

Limitations and Scope. While momentum filters oscillatory dynamics, environments with different oscillation characteristics may require different momentum decay rates. Appendix A.5 reports a sensitivity analysis over $\beta \in \{0.99, 0.999, 0.9999\}$ on both CIFAR-10 conditions. GMC's qualitative behavior is stable across this range, and the main effect of a longer averaging window is to move allocations closer to uniform sampling, most visibly among groups that differ only in noise level. This supports the reading that GMC assumes no particular noise distribution and only requires that unlearnable components lack a persistent gradient direction over the averaging window. As with learning rate or batch size, optimal settings likely depend on the problem, and the values used here were selected through preliminary experiments. In RL, noise can be time-dependent because a changing policy shifts the visited distribution, which suggests choosing the decay rates with the policy's rate of change in mind, favoring slower momentum where a wider range of fluctuations must cancel. Like any intrinsic reward added to the objective, GMC can bias the policy away from the extrinsic optimum, which also holds for prediction-error curiosity and can be mitigated by annealing the intrinsic-reward weight. As noted in Section 1, GMC filters incoherent aleatoric variation but does not separate learning that helps the task from learning that does not, such as memorization, which we regard as a property of the target rather than a defect of the proxy and a limit shared by reward-free prediction-error curiosity. In the controlled conditions GMC's prioritization coincides with the ordering a loss-difference signal produces, so these experiments do not exercise the decoupling between loss and learning discussed in Section 1, and evaluating a scenario where that alignment breaks down is left for future work. The LPM signal could be expected to help on MultiRoom as well, but it introduces a second network with its own training dynamics, so substituting it changes more than the scored quantity and makes for a less direct comparison than the ones reported here. Integration into other architectures such as RND, and evaluation on domains with continuous control or rich visual observations, are natural extensions. GMC scores *whether learning is occurring*, methods like ICM determine *what features* to measure it on, and the MultiRoom experiments show that these roles compose.

AI use statement

Generative AI tools were used for language editing and LaTeX formatting. The authors reviewed and verified all AI-assisted changes and take responsibility for the final content of this work.

Reproducibility statement

The method and experimental setup are described in Section 2 and Section 3. Appendix A.6 provides the architectures, optimizer settings, hyperparameters, compute resources, and other implementation details needed to reproduce the experiments. Code for reproducing all experiments is available at `https://github.com/[to-be-filled-upon-publication]`.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in Neural Information Processing Systems*, 30, 2017.

Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andrew Bolt, and Charles Blundell. Never give up: Learning directed exploration strategies. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, volume 29, pp. 1471–1479, 2016.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos J. Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019a.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *Seventh International Conference on Learning Representations*, pp. 1–17, 2019b.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and J K Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *Advances in Neural Information Processing Systems*, 36:73383–73394, 2023.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Yonathan Efroni, Dipendra Misra, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Provably filtering exogenous distractors using multistep inverse dynamics. In *International Conference on Learning Representations*, 2021.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *International conference on machine learning*, pp. 1050–1059, 2016.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for reinforcement learning. In *International Conference on Machine Learning*, pp. 1311–1320. PMLR, 2017.

Zhibo Hou, Zhiyu An, and Wan Du. Beyond noisy-tvs: Noise-robust exploration via learning progress monitoring. In *International Conference on Learning Representations*, volume 2026, pp. 79369–79386, 2026.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Riashat Islam, Manan Tomar, Alex Lamb, Yonathan Efroni, Hongyu Zang, Aniket Rajiv Didolkar, Dipendra Misra, Xin Li, Harm van Seijen, Remi Tachet des Combes, et al. Principled offline RL in the presence of rich exogenous information. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 14670–14691. PMLR, 2023.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pp. 2525–2534. PMLR, 2018.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

DP Kingma and J Ba. Adam: A method for stochastic optimization. In *ICLR 2015-Conference Track Proc. 1*, 2015.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images., 2009.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, 2010.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems*, 31, 2018.

Cam Linke, Nadia M Ady, Martha White, Thomas Degris, and Adam White. Adapting behavior via intrinsic reward: A survey and empirical study. *Journal of artificial intelligence research*, 69: 1287–1332, 2020.

Manuel Lopes, Tobias Lang, Marc Toussaint, and Pierre-Yves Oudeyer. Exploration in model-based reinforcement learning by empirically estimating learning progress. *Advances in neural information processing systems*, 25, 2012.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *arXiv preprint arXiv:1707.00183*, 2017.

Yurii E Nesterov. A method of solving a convex programming problem. *Doklady Akad. Nauk SSSR*, 269:543–547, 1983.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. *Advances in neural information processing systems*, 29, 2016.

Georg Ostrovski, Marc G. Bellemare, Aaron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2721–2730. PMLR, 2017.

Pierre-Yves Oudeyer, Frederic Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research, pp. 5062–5071. PMLR, 2019.

Andrew Patterson, Samuel Neumann, Martha White, and Adam White. Empirical design in reinforcement learning. *Journal of Machine Learning Research*, 25(318):1–63, 2024.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Vitchyr Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, Proceedings of Machine Learning Research, pp. 7783–7792. PMLR, 2020.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. URL https://arxiv.org/abs/2201.02177.

Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pp. 222–227, 1991.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

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

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.

Yongcheng Zeng, Zexu Sun, Bokai Ji, Erxue Min, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Haifeng Zhang, Xu Chen, and Jun Wang. Cures: From gradient analysis to efficient curriculum learning for reasoning llms. *arXiv preprint arXiv:2510.01037*, 2025.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *International conference on machine learning*, pp. 1–9. PMLR, 2015.

A Supplementary Experimental Results

This section contains detailed experimental results and architectural details supporting the main paper findings.

A.1 Derivation of the Relative Impact on Momentum Magnitude

Equation 1 follows from a first-order Taylor approximation of the change in momentum magnitude induced by a single sample's gradient. Let $\mathbf{m} \in \mathbb{R}^d$ be the current momentum vector and $\mathbf{g} = \nabla_\theta \mathcal{L}_{dyn}(x) \in \mathbb{R}^d$ the gradient for sample x.

Momentum increment. Following Adam's definition of momentum with decay rate $\beta \in (0, 1)$, the updated momentum is $\beta\mathbf{m} + (1-\beta)\mathbf{g}$, so the increment induced by $\mathbf{g}$ is:

$$\Delta_{\mathbf{m}} = \big(\beta\mathbf{m} + (1-\beta)\mathbf{g}\big) - \mathbf{m} = (\beta - 1)\mathbf{m} + (1-\beta)\mathbf{g} \tag{5}$$

First-order expansion. Define $f(\mathbf{x}) = \|\mathbf{m} + \mathbf{x}\|$ for $\mathbf{x} \in \mathbb{R}^d$, so that $f(\mathbf{0}) = \|\mathbf{m}\|$. The quantity of interest is the change in momentum magnitude, which is exactly:

$$\Delta = f(\Delta_{\mathbf{m}}) - f(\mathbf{0}) = \|\mathbf{m} + \Delta_{\mathbf{m}}\| - \|\mathbf{m}\| \tag{6}$$

Expanding f around $\mathbf{x} = \mathbf{0}$ to first order:

$$f(\mathbf{x}) \approx f(\mathbf{0}) + \nabla f\big|_{\mathbf{x}=\mathbf{0}}^{\top}\mathbf{x}, \qquad \nabla f(\mathbf{x}) = \frac{\mathbf{m}+\mathbf{x}}{\|\mathbf{m}+\mathbf{x}\|}, \qquad \nabla f\big|_{\mathbf{x}=\mathbf{0}} = \frac{\mathbf{m}}{\|\mathbf{m}\|} \tag{7}$$

Evaluating the expansion at $\mathbf{x} = \Delta_{\mathbf{m}}$:

$$\Delta \approx \frac{\mathbf{m}^\top \Delta_{\mathbf{m}}}{\|\mathbf{m}\|} \tag{8}$$

Substituting the increment. Inserting Equation 5 and using $\mathbf{m}^\top\mathbf{m} = \|\mathbf{m}\|^2$:

$$\Delta \approx \frac{(\beta-1)\,\mathbf{m}^\top\mathbf{m} + (1-\beta)\,\mathbf{m}^\top\mathbf{g}}{\|\mathbf{m}\|} = (\beta-1)\|\mathbf{m}\| + (1-\beta)\,\frac{\mathbf{m}^\top\mathbf{g}}{\|\mathbf{m}\|} \tag{9}$$

Relative impact. The momentum $\mathbf{m}$ is held fixed while the current batch is scored, so the first term of Equation 9 is a decay term identical for every sample, and the remaining factor $(1-\beta) > 0$ is likewise common to all samples. Neither affects how samples compare to one another: discarding them is a strictly increasing affine map of Δ, which leaves the ordering of samples unchanged. What remains is the relative impact of a gradient on the momentum magnitude:

$$\rho(x) = \frac{\mathbf{m}^\top\mathbf{g}}{\|\mathbf{m}\|} \tag{10}$$

This is the scalar projection of $\mathbf{g}$ onto $\mathbf{m}$, measuring the component of the sample gradient in the direction of ongoing learning. The absolute value in Equation 3 is applied to ρ rather than to Δ, since it is the sample-dependent part whose sign distinguishes reinforcing from opposing gradients. The expansion is valid when $\|\Delta_{\mathbf{m}}\| = (1-\beta)\|\mathbf{g} - \mathbf{m}\| \ll \|\mathbf{m}\|$, which the factor $(1-\beta)$ enforces at the decay rates we use, except where an individual gradient is orders of magnitude larger than the accumulated momentum.

A.2 Robustness to Alternative Class-to-Group Assignments

The Curriculum condition in Section 3.1 uses a single fixed assignment of the ten CIFAR-10 classes to the four groups. The prioritization order it produces could therefore reflect the visual properties of the particular classes placed in each group rather than a property of the compared signals. To test this, we repeated the CIFAR-10 Curriculum experiment on two additional partitions whose class-to-group assignments were drawn at random (Table 2). The group sizes $(1, 2, 3, 4)$ are held fixed, so the number of admissible labels per group is unchanged and only the identity of the classes varies. Each alternative partition was run for 20 random seeds with all remaining hyperparameters as in Table 4.

The Uniform panel of each figure shows how the class assignment changes the empirical difficulty of the groups, and the singleton G_0 is the easiest in all three partitions. Beyond that the groups reorder, and both randomized partitions are more strongly differentiated than the original one, where the three larger groups end up close together.

Partition	G_0	G_1	G_2	G_3
Original	$\{0\}$	$\{1, 2\}$	$\{3, 4, 5\}$	$\{6, 7, 8, 9\}$
Randomized 1	$\{7\}$	$\{2, 5\}$	$\{1, 6, 9\}$	$\{0, 3, 4, 8\}$
Randomized 2	$\{4\}$	$\{0, 8\}$	$\{2, 3, 7\}$	$\{1, 5, 6, 9\}$

Table 2: CIFAR-10 class-to-group assignments. The two randomized partitions preserve the group sizes $(1, 2, 3, 4)$ of the original partition and reassign the class identities at random.

Randomized partition 1. Here G_3 is visibly harder than the other groups under Uniform sampling: its loss is the highest from the start and stays clearly above the rest for the whole run, while G_1 and G_2 track each other closely (Figure 5). GMC visits G_0 first and G_3 last, with G_1 and G_2 in between and too close to separate reliably. Curiosity does the opposite, allocating most to G_3 from the first epochs and least to G_0.

Randomized partition 2. This partition produces a more interesting case (Figure 6). Under Uniform sampling G_1 has the lowest loss of the multi-class groups at the start, yet it is overtaken by all of them later and finishes highest, so it is easy at first and hard to master. Curiosity, following absolute loss, allocates least to G_1 early and only turns to it once its loss has become the largest, peaking around epoch 20 and staying elevated thereafter. GMC's allocation for G_1 has two distinct phases: a small bump in the first epochs while its loss is still falling quickly, then a decline, and finally a sustained commitment from roughly epoch 35 onward as the other groups saturate and G_1 becomes the only group where learning is still under way. This is the behavior the signal is intended to produce, and it is not available to a rule based on loss magnitude alone, which would have committed to G_1 either at the start or at the end but not at both.

Interpretation. Across all three partitions, GMC's allocation order changes together with the empirical learning dynamics of the groups rather than with their class identities, so the prioritization pattern reported in Section 3.1 is not an artifact of the original assignment. Curiosity, by contrast, remains primarily responsive to absolute prediction loss, so its prioritization begins with the groups that have the highest losses at the time, regardless of whether those groups are the ones currently improving fastest. These are observations about which groups each signal visits and when, read off the allocation curves, and we do not claim a general correspondence between these orderings and any single scalar measure of difficulty.

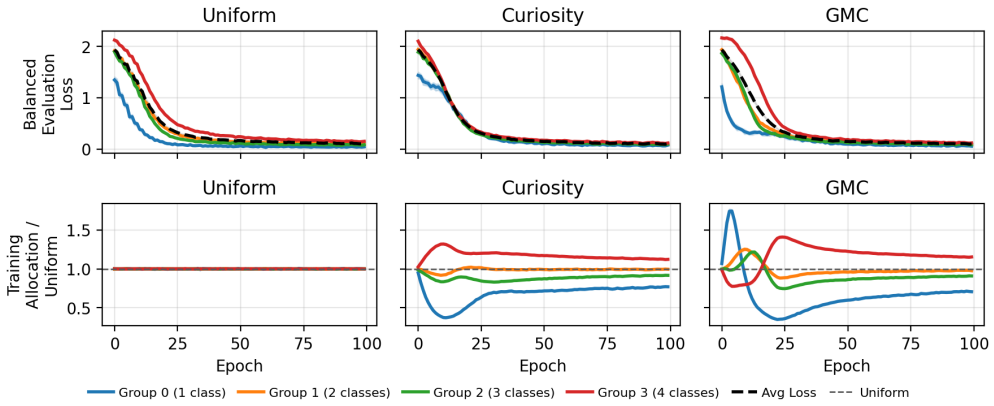

Figure 5: CIFAR-10 Curriculum Condition under randomized partition 1. **Top:** Per-group Balanced Evaluation Loss. **Bottom:** Mean per-class training allocation within each group. G_3 is the clearly hardest group in this partition, and GMC is the only method that visits it last.

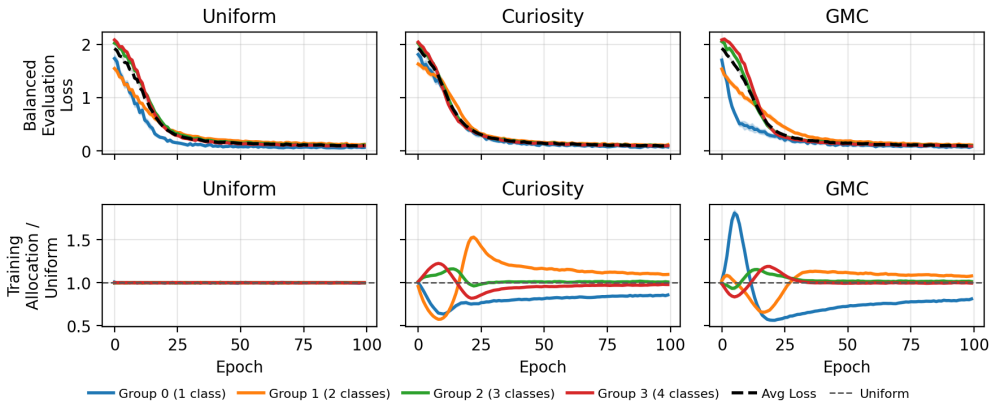

Figure 6: CIFAR-10 Curriculum Condition under randomized partition 2. **Top:** Per-group Balanced Evaluation Loss. **Bottom:** Mean per-class training allocation within each group. Under Uniform sampling G_1 starts with the lowest loss of the multi-class groups and finishes with the highest. Curiosity turns to it only once its loss is largest, whereas GMC visits it briefly at the start and then commits to it from the middle of training onward.

A.3 MultiRoom Action Composition and Intrinsic Reward Allocation

Section 3.2 attributes ICM's declining advantage in the larger MultiRoom environments to the prediction error that rotation actions generate under partial observability. Figure 7 reports the behavioral quantities that this account predicts, measured over training for all three methods and all four environments. The columns are the fraction of actions that are rotations, the fraction that are forward moves, and the share of intrinsic reward received by rotation actions relative to forward moves. The rows are the four environments, in order of increasing size. The reward-share column is undefined for PPO, which has no intrinsic reward.

ICM assigns a persistently larger share of its intrinsic reward to rotation than ICM+GMC does, and its action distribution remains more heavily weighted toward turning and correspondingly less toward forward movement. The difference is present in all four environments and does not diminish with training, which is what distinguishes an irreducible residual from a transient one. It is also largest in the six-room environment, where ICM turns for the great majority of its steps while ICM+GMC turns less than in any smaller environment, matching the return curves in Figure 4.

A.4 Ablation: GMC Variants

We compare three variants of the GMC signal:

1. **GMC**: $d^{-1/2} \sum_i |g_i \hat{m}_i / s_i|$
2. **Dot Product**: $d^{-1/2} \sum_T |\langle g_T, \hat{m}_T / s_T \rangle|$
3. **Cosine Similarity**: $d^{-1/2} \sum_T |\cos(g_T, \hat{m}_T / s_T)|$

Here $s_i = \max(\hat{v}_i, \alpha^2 \hat{q}_i)$ as in Equation 2, and T ranges separately over every weight and bias tensor.

Theoretical Differences. The Dot Product variant aggregates products within each parameter tensor before taking magnitude, permitting cancellation within a tensor while keeping weight and bias contributions from each layer separate. In contrast, GMC applies the absolute value elementwise, allowing parameters that change in different directions to contribute independently. As a result, GMC captures simultaneous multi-dimensional changes rather than collapsing them into a single axis. The Cosine Similarity variant computes an absolute cosine separately for every weight and bias tensor, using the same normalized momentum $\hat{m}/s$ as the other variants. It is scale-invariant within each tensor and measures only directional alignment, discarding magnitude information.

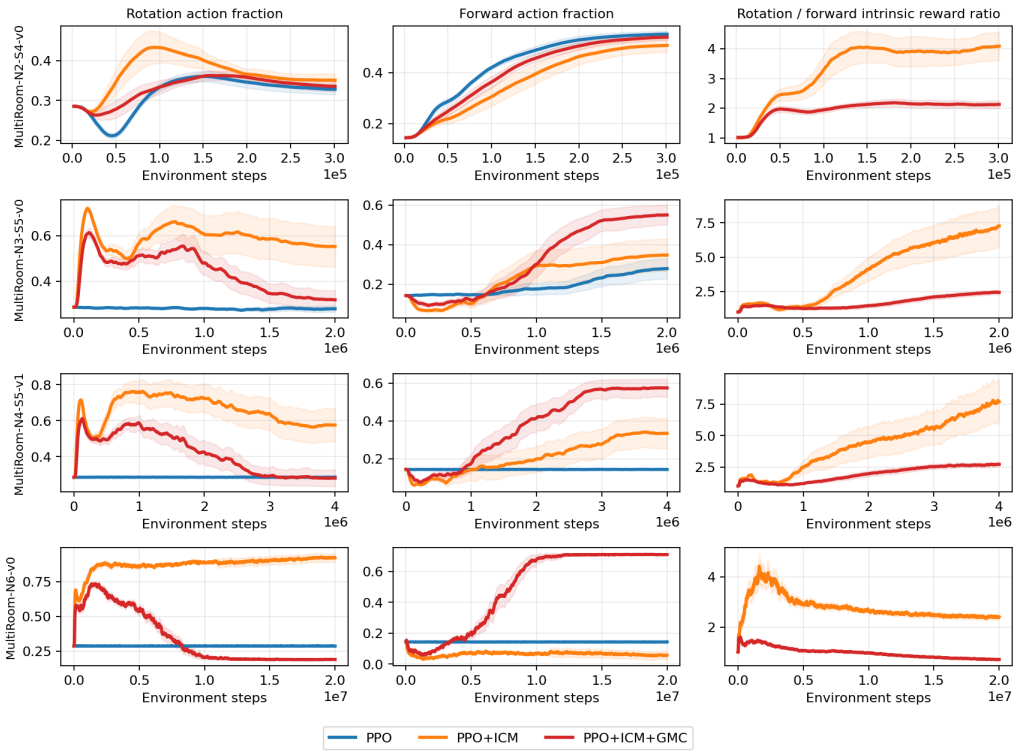

Figure 7: MultiRoom behavioral diagnostics. **Columns:** fraction of actions that are rotations, fraction that are forward moves, and share of intrinsic reward received by rotation actions relative to forward moves. **Rows:** the four environments in order of increasing size. The reward-share column omits PPO, which receives no intrinsic reward.

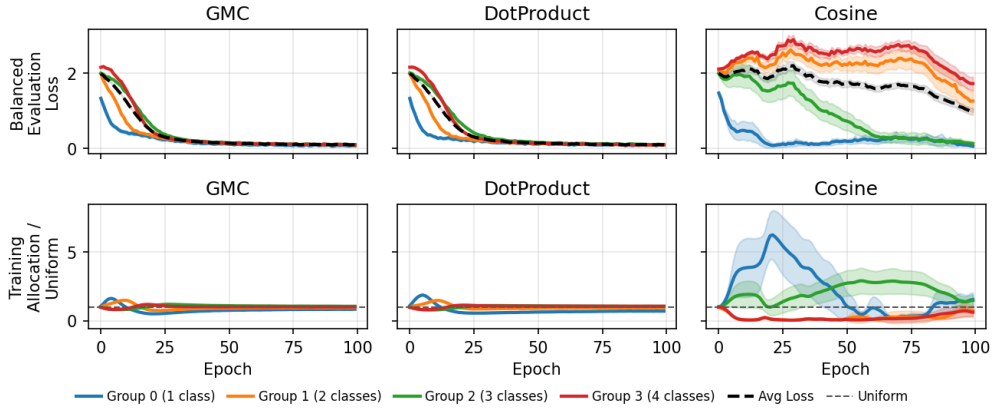

Figure 8: CIFAR-10 Curriculum Condition for the three GMC variants, one per column: GMC, Dot Product, and Cosine Similarity. **Top row:** Per-group Balanced Evaluation Loss. **Bottom row:** Mean per-class training allocation within each group. GMC and Dot Product produce broadly similar allocations, while Cosine fluctuates throughout.

Empirical Observations. Cosine Similarity is clearly unsuitable: it tends to oversample the easiest group early in training and exhibits fluctuating sampling weights throughout (Figure 8). By construction, cosine normalization amplifies even residual gradient alignment once magnitudes shrink near convergence, injecting noise into the signal precisely when it should vanish.

Dot Product and GMC yield broadly similar sampling behavior. Given these results, both GMC and Dot Product are viable. We default to GMC because its elementwise absolute value preserves independent per-parameter contributions across multiple simultaneous directions of change, whereas Dot Product permits cancellation among parameters within each tensor. This ablation does not establish that either viable variant is superior in more complex domains.

A.5 Sensitivity to the EMA Decay Rate

GMC maintains three exponentially-weighted moving averages: the momentum m_i, second moment v_i, and squared-parameter scale q_i. In this sweep their shared decay rate β sets the length of the averaging window. To quantify how strongly the signal depends on this choice, we repeated the CIFAR-10 Curriculum and Noise conditions with $\beta_m = \beta_v \in \{0.99, 0.999, 0.9999\}$, using 10 random seeds per configuration. All remaining hyperparameters are as in Table 4.

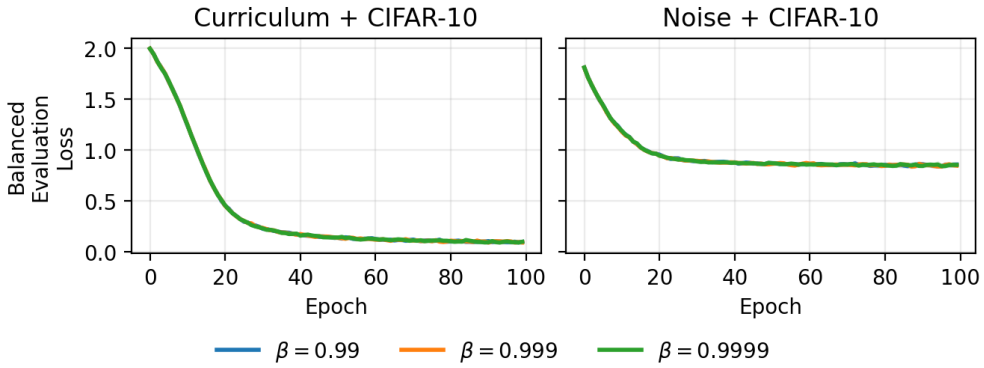

Figure 9: Balanced Evaluation Loss averaged over all classes for the three decay rates. The three curves are indistinguishable in both conditions.

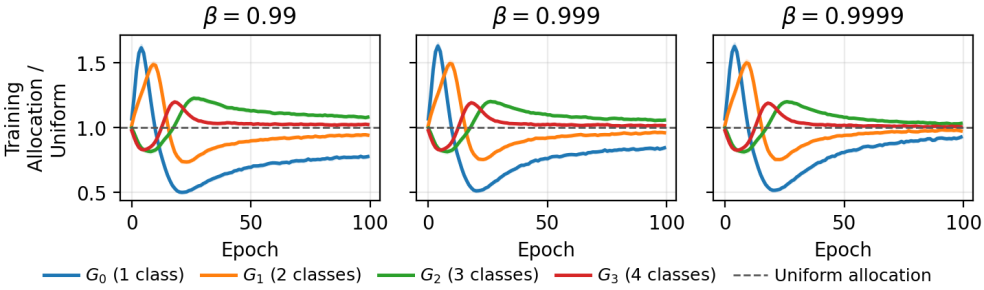

Figure 10: GMC decay-rate sensitivity on the CIFAR-10 Curriculum Condition, mean per-class training allocation within each group. The visitation peaks occur in the same order for all three decay rates, and the allocations contract toward uniform sampling as β increases.

Convergence. The overall loss is unaffected by the decay rate (Figure 9). In both conditions the three curves are indistinguishable over the whole of training, so the choice of β within this range has no measurable effect on convergence speed or final loss. What it does affect is the allocation.

Curriculum Condition. The ordering of the visitation peaks is the same for all three values of β, namely $G_0 \to G_1 \to G_3 \to G_2$ (Figure 10), so the prioritization reported in Section 3.1 is not an artifact of the particular decay rate used there. The peak heights are also largely unchanged. The visible effect of a longer window is a contraction of the later allocations toward uniform sampling, resembling the effect of raising the entropy-regularization weight λ_{ent}.

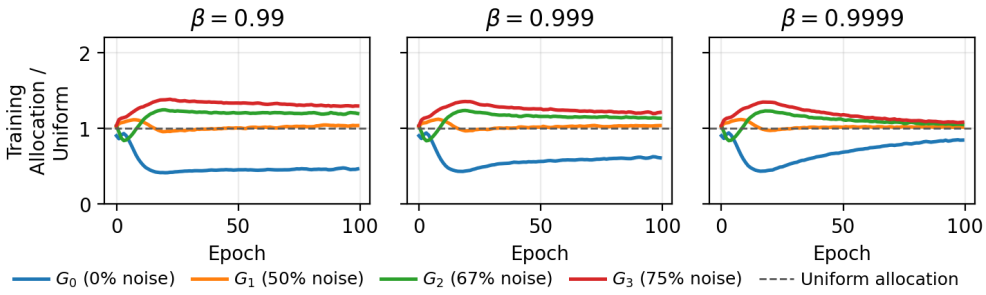

Figure 11: GMC decay-rate sensitivity on the CIFAR-10 Noise Condition, mean per-class training allocation within each group. The three noisy groups draw closer together as β increases, while the gap between them and the noise-free G_0 remains visible at every decay rate.

Noise Condition. The same contraction appears here where the four groups differ only in noise level (Figure 11). As β increases, the separation between the three noisy groups shrinks, and by $\beta = 0.9999$ their final allocations nearly coincide. The gap between those groups and the noise-free G_0 persists at every decay rate, although it too narrows. This is the behavior expected from the mechanism described in Section 1. A larger β corresponds to a longer effective averaging window, over which inconsistent gradient components have more opportunity to cancel, which makes GMC less sensitive to transient differences between groups that differ only in noise level. It is the empirical counterpart of the assumption stated in the Limitations, that GMC requires only that unlearnable components lack a persistent gradient direction over the averaging window.

Summary. Sensitivity over the tested range is limited: convergence is unaffected and the prioritization order is unchanged, while a longer window moves allocations toward uniform sampling and slightly improves noise robustness. We use $\beta = 0.999$ in both the controlled experiments and the MultiRoom experiments.

A.6 Implementation Details

Component	Type	Configuration
Dynamics Model (Classifier)	MLP	Input: $32 \times 32 \times 3$ (CIFAR-10) / $28 \times 28 \times 1$ (MNIST) Flattened to 3072 / 784 neurons Hidden Layer 1: 512 neurons, ReLU Hidden Layer 2: 512 neurons, ReLU Output Layer: 10 neurons (logits) Loss: Cross-Entropy
Policy Network (Actor)	MLP	Input: 4 (groups) or 10 (classes) neurons of independent uniform random noise Hidden Layer 1: 512 neurons, ReLU Hidden Layer 2: 512 neurons, ReLU Output Layer: 4 (groups) or 10 (classes) neurons (logits) Action: softmax(output) for group selection

Table 3: Network architectures for dynamics model and policy. Both use identical hidden dimension (512) and ReLU activations. The dynamics model predicts class labels, the policy provides sampling probabilities at each step.

Hyperparameter Selection. PPO hyperparameters follow widely-used defaults (Schulman et al., 2017; 2016). Preliminary experiments confirmed stable training with these values across all methods. GMC-specific parameters (the moment decay rates and parameter-floor scale) were chosen based on the theoretical motivation that longer averaging windows smooth noise more effectively and that the denominator should remain meaningful when gradients become small, and were validated through preliminary experiments to confirm stable signal behavior. ICM hyperparameters were set to values from the original work (Pathak et al., 2017). Controlled experiment hyperparameters (Table 4) were similarly determined through initial experimentation, selecting values that produced stable

Hyperparameter	Value
Optimization	
Dynamics Model	Adam, learning rate 1×10^{-3}
Policy Network	Adam, learning rate 5×10^{-6}, $\beta_1 = 0.99$
Training	
Batch Size	256
Evaluation Batches per Epoch	100
Num Epochs (MNIST Curriculum)	400
Num Epochs (other conditions)	100
Entropy Regularization Weight (λ_{ent})	0.1
Intrinsic Signal Weight	1.0
Signal-scale EMA Decay	0.9
Signal-scale Calibration Window	First 10% of training
GMC	
Momentum Decay (β_m)	0.999
Second-Moment Decay (β_v)	0.999
Squared-Parameter Decay ($\beta_q = \beta_v$)	0.999
Parameter-Floor Scale (α)	1×10^{-2}
Data	
Random Seeds	0-19 (20 total)

Table 4: Training hyperparameters for Signal Evaluation under Controlled Difficulty and Noise experiments (CIFAR-10, MNIST). The policy is updated after every batch. Settings not listed are library defaults. Policy network receives lower learning rate than dynamics model to ensure stable learning. Momentum decay parameters for GMC use longer averaging windows than Adam's default β_1, capturing more extended learning history.

learning dynamics across all methods. A sensitivity analysis of GMC's decay rates is reported in Appendix A.5.

Computational Complexity. For d scored parameters, GMC stores a momentum, a gradient second moment, and a squared-parameter moving average per parameter and computes an $O(d)$ elementwise product and sum. This is the same asymptotic order as the backward pass, though the current implementation has practical time and memory overhead from obtaining per-sample gradients. The momentum m_i, second moment v_i, and squared-parameter average q_i are exponentially-weighted moving averages of the batch-mean gradient $\bar{g}_i$ and the parameter value θ_i:

$$m_i \leftarrow \beta_m m_i + (1-\beta_m)\bar{g}_i, \quad v_i \leftarrow \beta_v v_i + (1-\beta_v)\bar{g}_i^2, \quad q_i \leftarrow \beta_v q_i + (1-\beta_v)\theta_i^2. \tag{11}$$

The bias-corrected values $\hat{m}_i$, $\hat{v}_i$, and $\hat{q}_i$ from prior batches are used to score the current batch with denominator $\max(\hat{v}_i, \alpha^2 \hat{q}_i)$, after which the current batch statistics are inserted. We use $\beta_m = \beta_v = 0.999$ and $\alpha = 10^{-2}$ in the controlled experiments. Equal momentum and second-moment decay rates imply $\hat{m}_i^2 \leq \hat{v}_i$ by Jensen's inequality, while the parameter-relative floor further prevents amplification when $\hat{v}_i$ is very small.

GMC maintains its own momentum, second-moment, and parameter-scale estimates for scoring. These read-only statistics do not interact with the optimizer's update rule or apply momentum twice. We keep them separate to allow decay rates different from Adam's defaults ($\beta_1 = 0.9$, $\beta_2 = 0.999$). With matched rates, the optimizer's own estimates could be reused.

We normalize GMC by $1/\sqrt{d}$, where d is the number of scored parameters. In the controlled experiments all non-uniform methods, including NormOut, NormAll, and DeltaLoss, then use the shared early-signal calibration described below, with no method-specific scaling constants.

Signal Scale Calibration. In the controlled experiments the policy-gradient term scales with the magnitude of r_{int}, so a signal whose raw values are large relative to λ_{ent} would sharpen its allocation for reasons unrelated to what it measures. The six methods produce values on very different scales, and to compare them under a single shared λ_{ent} we track an EMA (decay 0.9) of each signal's batch-mean magnitude during the first 10% of training, divide the signal by that running value, and then freeze the divisor for the rest of the run. Freezing matters, since a divisor that kept updating would normalize away the decay of the signal as learning saturates, which is precisely the behavior under study. This is a device for fair comparison rather than a component of the method.

Hyperparameter	Value
Environments	
Environment 1	MiniGrid-MultiRoom-N2-S4-v0, 300,000 steps
Environment 2	MiniGrid-MultiRoom-N3-S5-v0 (locally registered), 2,000,000 steps
Environment 3	MiniGrid-MultiRoom-N4-S5-v1, 4,000,000 steps
Environment 4	MiniGrid-MultiRoom-N6-v0, 20,000,000 steps
Observation	Partial, egocentric (as distributed)
Environment Modifications	None
PPO	
Learning Rate	2.5×10^{-4}, linearly annealed
Discount Factor (γ)	0.99
GAE λ	0.95
Entropy Coefficient	0.01
PPO Epochs per Rollout	4
Mini-Batch Size	512
Parallel Environments	16
Rollout Length per Environment	128
Hidden Dimension	256
ICM	
Feature Dimension	128
Learning Rate	3×10^{-4}
Epochs per Rollout	2
Mini-Batch Size	256
Forward / Inverse Loss Weight	0.2 / 0.8
Intrinsic Reward (ICM and ICM+GMC)	
Intrinsic Reward Coefficient	5×10^{-4}
Data	
Random Seeds	0-49 (50 total)

Table 5: Hyperparameters for the MiniGrid MultiRoom reinforcement learning experiments. The environments are used exactly as distributed, with no modified observations or rewards, and each is run for the step budget listed above. The distributed set contains no three-room instance, so we register one locally with the same entry point as the other variants and three rooms of maximum size 5. Only settings that were chosen for these experiments are listed: the GMC decay rates and parameter-floor scale are unchanged from Table 4, and the remaining PPO and ICM settings are the commonly used values of Schulman et al. (2017) and Pathak et al. (2017). ICM and ICM+GMC use the same architecture, the same dynamics model hyperparameters, and the same intrinsic reward coefficient. PPO hyperparameters are identical across all three methods.

All experiments were run using PyTorch 2.0.1 on Ubuntu with an NVIDIA GeForce RTX 3080 GPU. The MNIST and CIFAR experiments took approximately 30 minutes per seed. Curriculum+MNIST took approximately four times longer, as it was run for 400 rather than 100 analyzed epochs, requiring about 120 minutes per seed. NormAll increased runtime, taking about 408 minutes on Curriculum+MNIST and about 90 minutes on the remaining MNIST/CIFAR experiments. For MultiRoom each different configuration ran a different number of steps. For all of them PPO ran roughly at 3000 steps per second, ICM at 2600, and ICM+GMC at 2500. However, with a PyTorch native reimplementation of GMC these differences would most probably be negligible.

A.7 Complete Results Across All Conditions

The main text reports the per-group results for Uniform, Curiosity and GMC on CIFAR-10. Here we provide the overall loss of all six methods on both datasets, and the per-group losses and allocations for the remaining combinations of dataset, condition and method. Table 6 gives the summary metric of Table 1 for both datasets and additionally for the two GMC variants of Appendix A.4.

	Curriculum		Noise		
Method	MNIST	CIFAR-10	MNIST	CIFAR-10	Total
Uniform	1.00 ± 0.01	1.00 ± 0.01	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.00	4.00 ± 0.01
Curiosity	$\mathbf{0.92 \pm 0.01}$	$\mathbf{0.99 \pm 0.00}$	1.13 ± 0.00	1.23 ± 0.00	4.27 ± 0.01
GMC	0.96 ± 0.01	1.01 ± 0.00	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	$\mathbf{3.96 \pm 0.01}$
NormOut	0.94 ± 0.01	1.00 ± 0.00	1.13 ± 0.00	1.34 ± 0.00	4.41 ± 0.01
NormAll	1.02 ± 0.01	1.00 ± 0.00	1.01 ± 0.00	1.07 ± 0.00	4.10 ± 0.01
DeltaLoss	1.00 ± 0.01	1.14 ± 0.01	$\mathbf{1.00 \pm 0.00}$	1.02 ± 0.00	4.16 ± 0.01
DotProduct	0.97 ± 0.01	1.02 ± 0.00	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.00	3.99 ± 0.01
Cosine	5.30 ± 1.59	4.64 ± 0.19	1.04 ± 0.00	1.78 ± 0.02	12.76 ± 1.60

Table 6: Normalized area under the overall Balanced Evaluation Loss curve, lower is better. Values are scaled so that Uniform is 1.00 in each column. The best mean in each column is shown in bold. The Total column sums the four conditions and is reported only as a compact overall ordering. DotProduct and Cosine are the GMC variants of Appendix A.4.

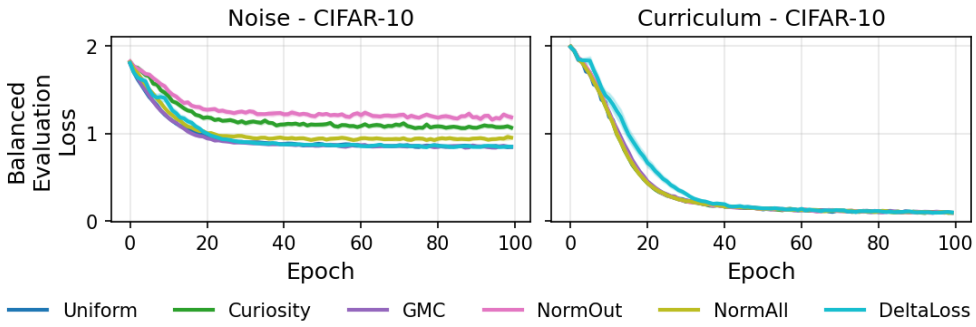

Figure 12: CIFAR-10 Balanced Evaluation Loss averaged over all classes, for all six methods in the Noise (left) and Curriculum (right) conditions.

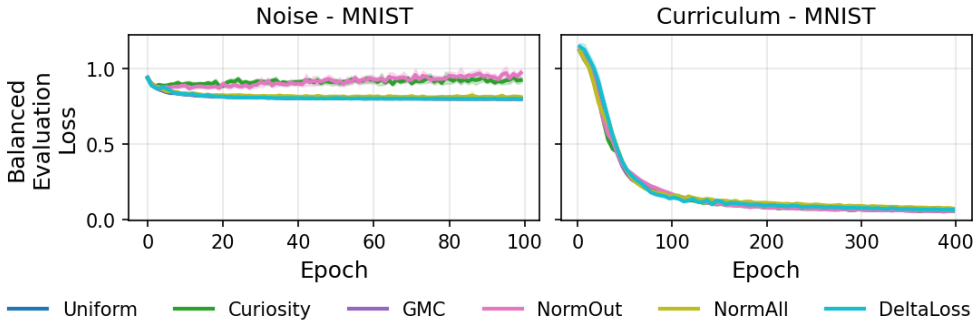

Figure 13: MNIST Balanced Evaluation Loss averaged over all classes, for all six methods in the Noise (left) and Curriculum (right) conditions. The split is the same as on CIFAR-10 (Figure 12).

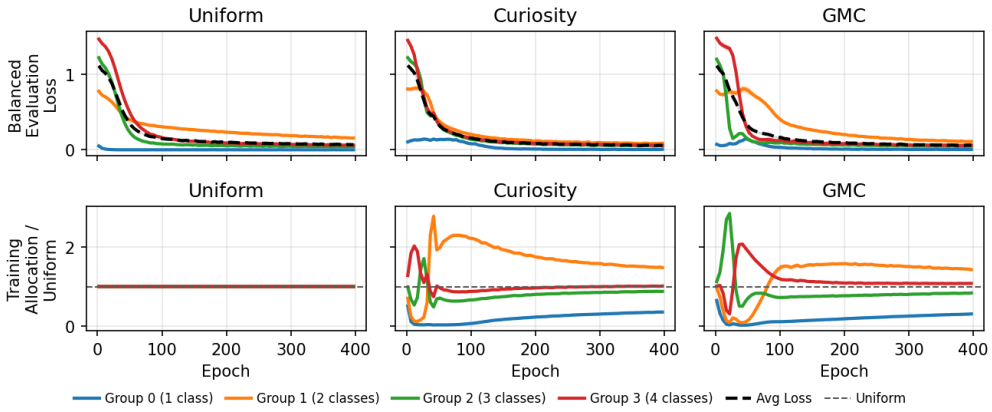

Figure 14: MNIST Curriculum Condition. **Top:** Per-group Balanced Evaluation Loss. **Bottom:** Mean per-class training allocation within each group. MNIST shows larger difficulty differences between groups than CIFAR-10, producing more pronounced prioritization differences. GMC peaks in the order $G_2 \to G_3 \to G_1$, while Curiosity prioritizes by initial loss magnitude ($G_3 \to G_2 \to G_1$). G_0 is left out of both orderings: it is the easiest group and its allocation stays low for both methods without forming a distinct peak, so it has no well-defined position in the sequence.

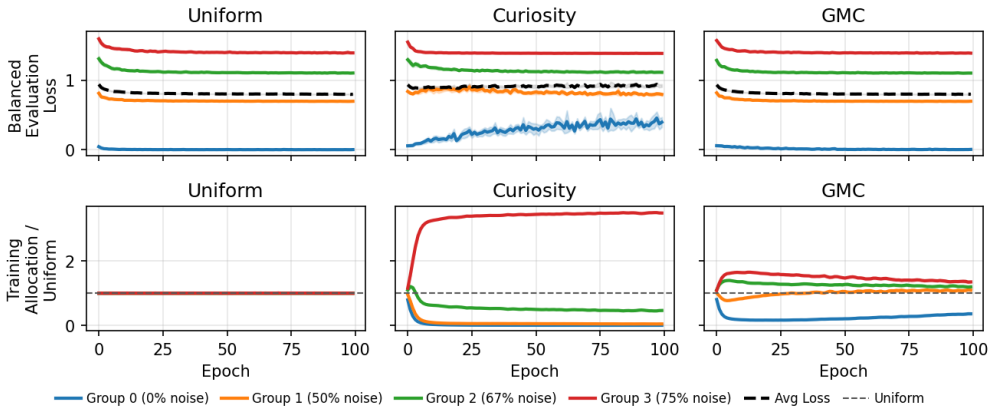

Figure 15: MNIST Noise Condition. **Top:** Per-group Balanced Evaluation Loss. **Bottom:** Mean per-class training allocation within each group. Like CIFAR-10, Curiosity concentrates on the noisiest group while neglecting others. GMC maintains stable allocation.

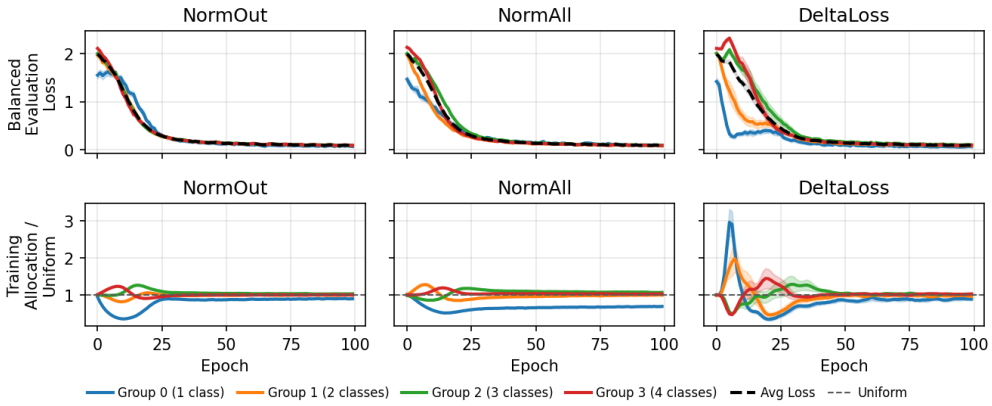

Figure 16: CIFAR-10 Curriculum Condition for NormOut, NormAll, and DeltaLoss. **Top row:** Per-group Balanced Evaluation Loss. **Bottom row:** Mean per-class training allocation within each group. NormOut shows prioritization similar to Curiosity and NormAll to GMC, but both exhibit slower convergence. DeltaLoss peaks in the order $G_0 \to G_1 \to G_3 \to G_2$, the same order as GMC in Figure 1, but its allocation is oscillatory and has much wider confidence intervals, reflecting the variance of consecutive-epoch loss estimates.

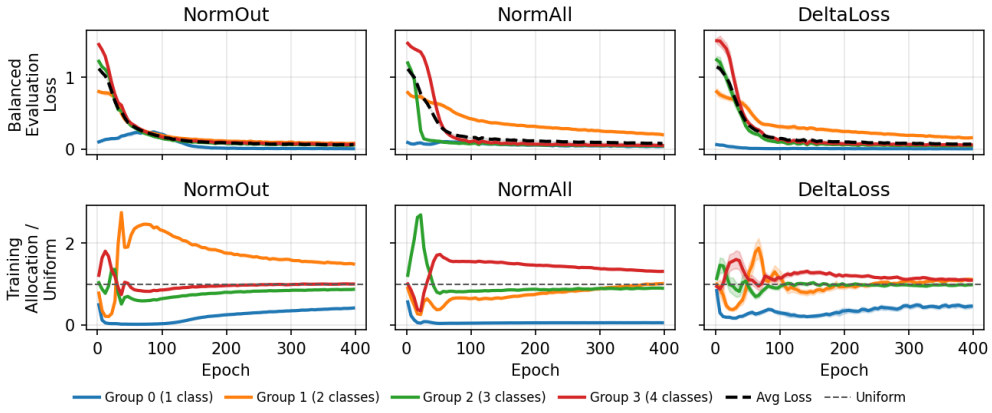

Figure 17: MNIST Curriculum Condition for NormOut, NormAll, and DeltaLoss. **Top row:** Per-group Balanced Evaluation Loss. **Bottom row:** Mean per-class training allocation within each group. NormAll fails to prioritize the hardest group after easier groups saturate, leading to slower learning. These results emphasize the role of momentum weighting. DeltaLoss exhibits pronounced oscillations due to high variance in consecutive-epoch loss differences, which harm the convergence speed.

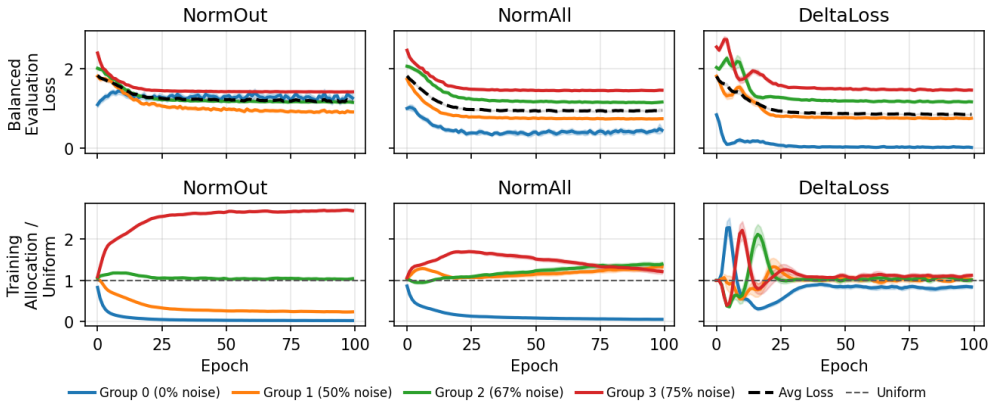

Figure 18: CIFAR-10 Noise Condition for NormOut, NormAll, and DeltaLoss. **Top row:** Per-group Balanced Evaluation Loss. **Bottom row:** Mean per-class training allocation within each group. NormOut is strongly affected by noise. NormAll's allocation is also affected, but less severely, highlighting the contribution of measuring the gradient over all parameters rather than at the output alone. DeltaLoss is not attracted to high prediction error alone.

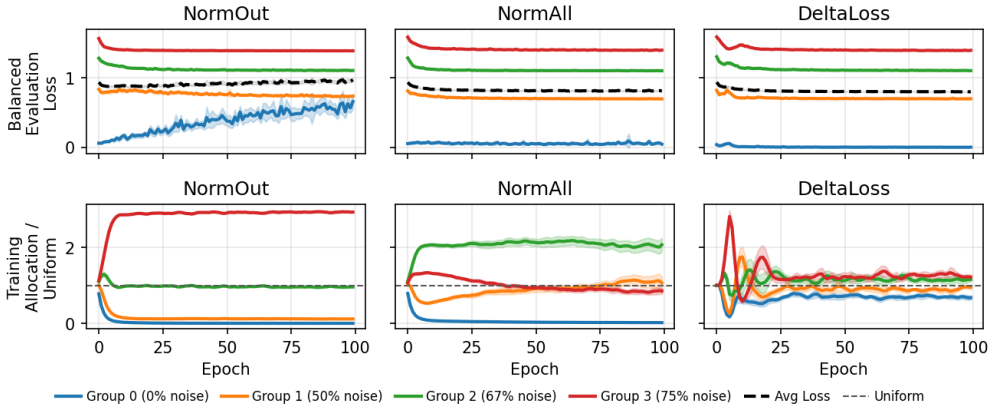

Figure 19: MNIST Noise Condition for NormOut, NormAll, and DeltaLoss. **Top row:** Per-group Balanced Evaluation Loss. **Bottom row:** Mean per-class training allocation within each group. The behavior is similar to CIFAR-10.

B Related Work: Extended Discussion

B.1 Learning Progress and Intrinsic Motivation

Schmidhuber's formal theory of creativity (Schmidhuber, 2010) proposes that agents should maximize "improvements of the world model", the rate of learning. Oudeyer and Kaplan Oudeyer et al. (2007) develop this in developmental robotics. However, "learning progress" has never been given an agreed definition, and what the literature provides are operationalizations rather than definitions. Prior work measures progress through auxiliary task performance or prediction accuracy (Lopes et al., 2012; Bellemare et al., 2016; Pathak et al., 2019), or through prediction gain (Schmidhuber, 1991) and compression progress (Schmidhuber, 2010), all of which read progress off the model's output and inherit both high variance and the ambiguities described in Section 1.

GMC operationalizes the same idea at the level of model parameters rather than through external performance metrics.

B.2 Curiosity-Based Exploration

Curiosity-driven exploration methods typically address sparse rewards by using model prediction error as a proxy for learning progress. Rather than explicitly estimating improvement over time, which is statistically challenging and often high variance, these approaches reward states or transitions where the agent's predictive models incur high loss, under the assumption that such regions offer greater potential for learning. ICM (Pathak et al., 2017) operationalizes this idea by defining intrinsic reward as the prediction error of a learned forward dynamics model, while constraining the underlying representation through an auxiliary inverse-dynamics objective that encourages features to be informative about the agent's actions. RND (Burda et al., 2019b) uses prediction error against a fixed random target network as a novelty signal, while VIME (Houthooft et al., 2016) employs Bayesian information gain in the model's belief state. Despite their differences, these methods share a common foundation: instantaneous model loss is used as a surrogate for learning progress.

This surrogate introduces a fundamental limitation. High prediction error may arise either from learnable structure or from irreducible stochasticity, yet scalar loss provides no mechanism to distinguish between the two.

ICM's Controllability Filter: Overfiltering and Underfiltering. In ICM, the inverse-dynamics objective is intended to mitigate this issue by filtering out uncontrollable sources of variation. However, this mechanism introduces two complementary failure modes.

First, overfiltering: by biasing representations toward information that is locally predictive of action, ICM disfavors state information whose relevance is temporally delayed or only indirectly connected to control. In environments where meaningful cues are revealed during phases of limited or no controllability, such as observations encountered during free fall after an irreversible action, these cues provide little or no inverse-dynamics learning signal. As a result, they are weakly represented or discarded, despite being critical for downstream task performance. This reflects a structural bias toward short-horizon controllability rather than long-term task relevance.

Second, underfiltering: the inverse-dynamics objective does not explicitly penalize retaining information unrelated to action prediction. In regions of the state space where actions have little influence on outcomes, the inverse loss becomes uninformative, providing no incentive to suppress irrelevant or stochastic features. Consequently, representations may continue to encode uncontrollable noise, and the forward-model prediction error, and thus intrinsic reward, remains persistently high. The agent is therefore still attracted to stochastic transitions, reproducing the noisy-TV failure mode the inverse model was intended to prevent.

Together, these issues show that while ICM reframes curiosity through controllability-aware representation learning, it neither guarantees the preservation of task-relevant but uncontrollable information nor reliably excludes irreducible randomness. The fundamental ambiguity of prediction error as a proxy for learning progress remains unresolved.

Recent work on multi-step inverse dynamics (Efroni et al., 2021; Islam et al., 2023) addresses the underfiltering problem specifically: by conditioning on actions over multiple timesteps, these methods

can provably separate endogenous (controllable) from exogenous (uncontrollable) state components, filtering distractors that single-step inverse dynamics misses. However, multi-step conditioning requires a sufficiently long action horizon to capture the relevant dynamics, introducing practical challenges around tractability and variance that grow with horizon length. Moreover, these methods improve the representation, not the measurement: the forward model's prediction error remains the intrinsic signal.

RND's One-Shot Geometry. RND defines intrinsic reward as prediction error to a fixed random target. While effective as a novelty heuristic, this design introduces two distinct limitations.

First, the induced representation geometry is arbitrary: because the target network is randomly initialized and never trained, the distances and novelty measures it induces bear no relationship to task-relevant structure. Random features may amplify or compress certain regions of the state space for reasons entirely unrelated to controllability, learnability, or reward. This can cause the agent to over-explore regions that merely happen to have high random-target prediction error, while under-exploring regions with more meaningful learning potential.

Second, RND's novelty signal is one-shot: because the reward decays monotonically as the predictor distills the fixed target, RND provides no mechanism to revisit regions that were encountered early in training but were not yet learnable with the agent's then-current representation capacity. If a region requires foundational skills or abstractions that have not yet been developed, the agent may visit it once, fail to learn, and then never return, even after acquiring the necessary prerequisites. This stands in contrast to learning-progress signals that can detect when previously-difficult regions become newly learnable.

Complementarity. ICM determines *what features to learn* and RND *what to treat as novel*, while GMC scores *whether learning is occurring* on whichever representation is used, so one could replace prediction error with GMC in either architecture.

Recent work addresses other aspects of exploration. Never Give Up (Badia et al., 2020) combines episodic and life-long novelty, Go-Explore (Ecoffet et al., 2019) uses archive-based exploration, and counting-based methods (Bellemare et al., 2016; Ostrovski et al., 2017) reward rarely-seen states. Goal-conditioned approaches (Andrychowicz et al., 2017; Pong et al., 2020) learn to reach diverse goals. These focus on state-space coverage (discovering unknown unknowns), while GMC focuses on efficient learning within known unknowns. Combining coverage-based methods with learning-progress signals is a promising direction.

B.3 Curriculum Learning

Curriculum learning (Bengio et al., 2009) posits that learning is more efficient when examples progress from easy to hard. Automated approaches include teacher networks (Graves et al., 2017), teacher-student frameworks (Matiisen et al., 2017), and self-paced learning (Kumar et al., 2010). Most require explicit difficulty rankings or auxiliary teacher models.

In our controlled partitions, GMC induces a prioritization that begins with the groups whose loss is falling fastest and defers the rest, reproduced across three different class-to-group assignments (Appendix A.2). Tasks where learning is rapid and persistent receive high scores. Tasks that are difficult (weakly aligned gradients), too noisy (incoherent gradients), or already learned (small gradients from saturation) receive low scores. Curiosity, by contrast, prioritizes tasks by highest loss first, and both signals shift focus over time. The observed ordering arises without manual tuning, but it is not a general guarantee of a convergence benefit, which remains task-dependent.

B.4 Momentum in Optimization

Classical momentum (Polyak, 1964) accumulates gradients to accelerate learning and reduce oscillation. Modern variants include Nesterov acceleration (Nesterov, 1983) and adaptive methods like Adam (Kingma & Ba, 2015).

We repurpose momentum from an optimization tool to a learning signal. Rather than using momentum to smooth updates, we use it as a record of *which parameters are currently moving*. High coupling (positive or negative) between a sample's gradient and momentum indicates the sample affects

actively-changing parameters. Low coupling indicates it affects stagnant parameters. This perspective aligns with work on optimization dynamics (Li et al., 2018) showing that successful optimization navigates with consistent patterns despite high dimensionality.

B.5 Noise Robustness and Uncertainty

Bayesian approaches (Gal & Ghahramani, 2016; Kendall & Gal, 2017) and ensemble methods (Osband et al., 2016; Pathak et al., 2019) provide mechanisms for estimating uncertainty, often useful for capturing epistemic uncertainty and, in some formulations, separating it from aleatoric noise. However, they introduce additional computational overhead and require an additional scoring or decision rule to translate uncertainty into sample prioritization.

Our approach is more direct: random noise produces gradients pointing in inconsistent directions, which cancel in momentum over a sufficiently long averaging window. Only consistent structure builds up. In our controlled setting, gradient-momentum coupling thereby filters noise while requiring no additional forward passes or ensemble maintenance.