Innovation-Gated Representations: Learning Physical State from Unreliable Sensing

Anonymous Author(s)
Affiliation
Address
email

Abstract

Learned quadrotor controllers are typically trained on reliable state, whereas deployed sensing can be delayed, stale or reordered, and a recurrent policy has no explicit mechanism for distinguishing such faults from genuine changes in the dynamics. We treat this as a representation-learning problem and propose INNOVATION-GATED REPRESENTATIONS (IGR). Rather than only encoding what it receives, the representation forms an internal expectation of each externally delivered channel from onboard signals and executed commands, and a normalised innovation conditions how strongly the delivered reading updates memory. Memory is supervised on state, inertial-forecast and identity targets inside a constrained teacher and distilled into a student of order 10^4 parameters. We compare supervising the channel expectation with letting the control task shape it. Across 23 held-out fault suites in simulation, task-trained IGR makes physical state readable from memory (external force R^2 0.64 vs. 0.29 on a 16-component projection of the recurrent state) while matching the strongest teacher-distilled GRU in success (63.1% vs. 62.8%, and 50.3% with expert labels). A Kalman-fed controller and learned memory fail in complementary places (89% vs. 18% under sustained drop-out, 59% vs. 87% under long freezes). Task-training the expectation improves on supervising it (63.1% vs. 58.8%) and keeps gradients bounded in the tested training configuration. Its physical heads respond strongly to real gusts and payloads and weakly to frozen, reordered or late data.

1 Introduction

Zero-shot adaptive policies control many vehicles by inferring their physics in context from sensing and action history [7, 18]. Real low-cost platforms deliver that sensing late, out of order or not at all.

In one such policy, 10–30 ms of velocity delay produces oscillations [7]. More generally, in-context adaptation treats every input as truthful, and nothing tells a policy which inputs to trust. A delivered reading that disagrees with the controller's expectation can have two causes that call for opposite responses. A stale or frozen reading is wrong and should not update memory; a gust or a payload change is a true change and should. The two can be distinguished only against a model of the vehicle's own dynamics, as when the accelerometer contradicts a frozen velocity.

Domain randomisation tolerates noise rather than reasoning about it, classical filters need accurate timing and a dynamics model, and auxiliary representation losses have produced mixed or null results, including no robustness gain beyond a bottleneck in a recent humanoid study [5].

We treat this as a question of what a representation should contain, and propose INNOVATION-GATED REPRESENTATIONS (IGR). Prior representations typically summarise the inputs they receive; IGR additionally forms, for every external channel, an expectation with a learned scale. The

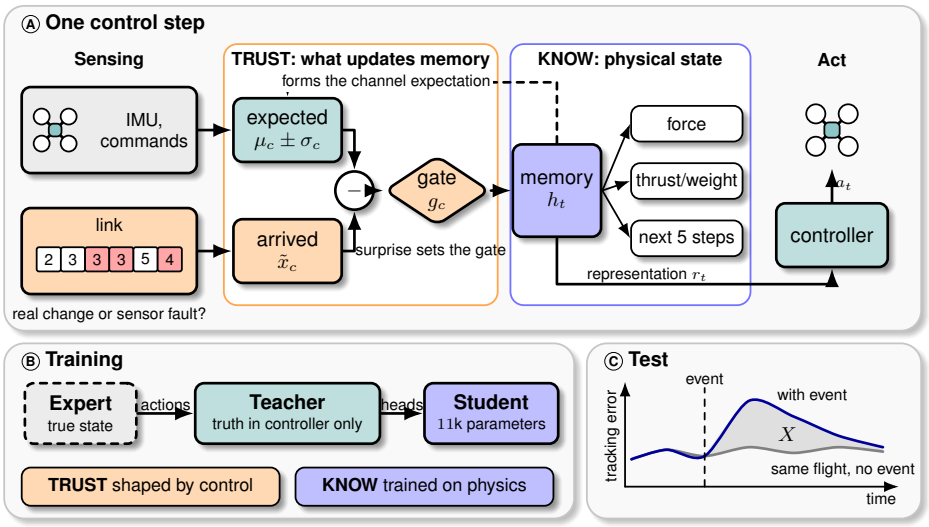


Figure 1: **How IGR works.** Ⓐ At each step, memory and onboard signals form an expectation of each channel; the innovation against the delivered value conditions how much that arrival updates memory. Memory exposes physical state and the controller acts on it. Ⓑ Physics targets train memory and control shapes the expectation. Ⓒ Paired flights measure the harm X of one event. Curves are schematic.

difference between the delivered and the expected value, scaled by it, is an *innovation*, and learned gates driven by the innovation condition how strongly a reading updates memory. The structure parallels a χ^2-gated Kalman filter [2] with learned models, and addresses learning physical state and characteristics from multimodal, unreliable sensing. Because a controller given privileged state can learn to ignore any estimate of it, privileged inputs reach only the teacher's controller. To see why a policy tolerates a fault, we evaluate with paired rollouts that differ in one event.

We find that supervising memory on physical targets while letting the control task shape the expectation behind the innovation gives a stronger combination of control and physical readability than supervising that expectation as well. We describe this as separating two jobs: *knowing* the physical state and *deciding what to trust*.

Our contributions are fourfold:

1. IGR, a predict–compare–write structure inside a recurrent controller, whose expectation is trained either by supervision (IGR-SUP) or through control (IGR-TASK) (Sec. 3.2).
2. A constrained-teacher procedure in which the representation is built from sensing alone, with privileged inputs reaching only the controller, and distilled into a student of 11,097 parameters, excluding training heads (Sec. 3.4).
3. A paired counterfactual evaluation that separates misleading data from real change (Sec. 4).
4. A comparison of supervised and task-trained expectations, showing control on par with the strongest GRU with physical readouts, distinct head responses to real change and sensor faults, complementary failure regimes with a filter, and a stability diagnosis (Secs. 4.2–4.6).

2 Related Work

Adaptive policies infer platform properties in context [7, 18] or condition on a learned latent of the dynamics [11], taking inputs at face value. In IGR the representation instead holds an expectation of each input and uses it to weigh that input, and the teacher's representation never receives privileged information.

Latent world models predict future observations or their embeddings [9, 1], so a stale or reordered stream becomes part of what is learned. We instead compare an expectation trained against the true channel value with one trained by control alone. Our use of prediction error is closest in spirit to predictive coding [14]; here the normalised error controls access to the memory of a controller.

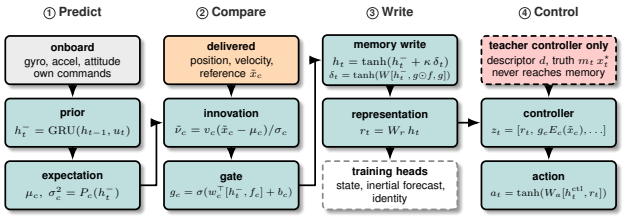


Figure 2: **The innovation-gated cell.** ① Prior and channel expectation. ② Normalised innovation and gate. ③ Gated write to memory. ④ Control; privileged inputs enter the teacher's controller only.

Learned filters fit gains, uncertainties or noise inside a predict–correct loop [15, 3, 8, 6], typically as estimators rather than as the memory of a controller. Our task-trained innovation extends the case for discriminative training [8] to closed-loop control under faults. The closest prior controller [13] uses a belief encoder that learns an attention gate over exteroception and reconstructs privileged information through the same gate. There the gate is computed from the belief state; ours acts on a normalised innovation and gates memory writes under packet-level faults.

We use an action buffer [4] and monitor copycat shortcuts [17]. Concurrent policy–estimator training [10] motivates heads on the representation, and our factorial ablation asks which objective helps on which part of the cell.

3 Method

3.1 Sensing Model

The vehicle carries *local* channels $\mathcal{C}_{\mathrm{loc}} = \{\mathrm{gyro}, \mathrm{accel}, \mathrm{attitude}\}$, available at control rate, and *external* channels $\mathcal{C}_{\mathrm{ext}} = \{\mathrm{ext_pos}, \mathrm{ext_vel}, \mathrm{reference}\}$ delivered over a link. External velocity comes from a motion-capture-like source or from optical flow. A packet k on channel c is sent at s_k and arrives at $a_k = s_k + \delta_k$ with random $\delta_k \geq 0$, not FIFO. With $k^*(t)$ the latest arrival, the controller observes $\tilde{x}_{c,t} = x_c(s_{k^*(t)}) + \eta_{c,t}$ and a validity bit $v_{c,t}$, but not the age $t - s_{k^*(t)}$: no timestamps are used. Real changes (gusts, payload, motor loss) alter the dynamics themselves. The policy π maps the history to four normalised motor commands $a_t \in [-1, 1]^4$ at 100 Hz.

3.2 Innovation-Gated Cell

Figure 2 details the loop of Fig. 1 in four steps.

Predict. Local sensing and the $K{=}6$ most recent commands, which the vehicle knows it sent, update a recurrent prior, and a predictor P_c outputs the mean and log-variance of each external channel:

$$u_t = [E_{\mathrm{loc}}(x_t^{\mathrm{loc}}),\ E_a(a_{t-K:t-1})], \quad h_t^- = \mathrm{GRU}_{\mathrm{rep}}(h_{t-1}, u_t), \quad (\mu_{c,t}, \log \sigma_{c,t}^2) = P_c(h_t^-). \quad (1)$$

Compare. The delivered value, in standardised units, is compared with the prediction, scaled by the predicted uncertainty:

$$\begin{aligned} \tilde{\nu}_{c,t} &= v_{c,t}\, \frac{\tilde{x}_{c,t} - \mu_{c,t}}{\sigma_{c,t}}, \qquad \mathrm{NIS}_{c,t} = \min\big(\|\tilde{\nu}_{c,t}\|^2,\ 50\big), \\ f_{c,t} &= E_\nu\big([\tilde{\nu}_{c,t},\ |\tilde{\nu}_{c,t}|,\ \log(1{+}\mathrm{NIS}_{c,t}),\ v_{c,t}]\big). \end{aligned} \quad (2)$$

Because a stale reading can be wrong in either direction, a gate linear in the signed $\tilde{\nu}$ could not depend on its magnitude, so we also supply $|\tilde{\nu}|$ and NIS.

Write. A scalar gate per external channel scales the innovation's access to memory:

$$\begin{aligned} g_{c,t} &= \mathrm{sigmoid}\big(w_c^\top [h_t^-, f_{c,t}] + b_c\big), \qquad r_t = W_r h_t, \\ h_t &= \tanh\Big(h_t^- + \kappa \tanh\big(W\big[h_t^-,\ (g_{c,t} f_{c,t})_c,\ (g_{c,t})_c\big]\big)\Big). \end{aligned} \quad (3)$$

The gate values are appended to the correction, so that "ignored" is distinguishable from "observed zero". The outer $\tanh$ keeps memory in the range of the GRU, and the scale $\kappa \in (0, 1]$ bounds how much it may change in one step; κ is a stability parameter of the recurrence (Sec. 4.4).

Control. The controller reads the representation with the gated raw inputs, local sensing and the action history:

$$z_t = \big[r_t,\ (g_{c,t}E_c(\tilde{x}_{c,t}),\ g_{c,t})_c,\ u_t\big], \quad a_t = \tanh\big(W_a[\mathrm{GRU}_{\mathrm{ctl}}(h^{\mathrm{ctl}}_{t-1}, z_t),\ r_t]\big). \tag{4}$$

Eqs. (1)–(3) keep the skeleton of the χ^2-gated update $\hat{x} = \hat{x}^- + g\,P^- H^\top S^{-1}\nu$ [2] and make each piece learnable: the gain becomes W, S becomes σ_c^2, and the test becomes the gate. The update is nonlinear and learned, so it does not inherit the optimality or calibration of a Kalman filter, and the response need not be a closing gate: the innovation features and the correction also carry it.

3.3 Representation Objectives

The objectives fall into two groups. Memory supervision teaches h_t about the vehicle through an inertial forecast, state and identity, and applies in every configuration; channel supervision instead trains the expectation P_c directly and is the term we vary. Targets are standardised with fixed statistics from expert rollouts and squared norms are averaged over dimensions:

$$\begin{aligned}
\mathcal{L}_{\mathrm{pred}} &= \tfrac{1}{5}\textstyle\sum_{j=1}^{5} \big\|\hat{y}_{t+j} - y_{t+j}\big\|^2, \quad \mathcal{L}_P = \sum_c \Big[\tfrac{(x_c(t)-\mu_c)^2}{2\sigma_c^2} + \tfrac{1}{2}\log\sigma_c^2\Big], \\
\mathcal{L}_{\mathrm{state}} &= \big\|\hat{s}_t - s^*_t\big\|^2, \quad s^* = [p_{\mathrm{err}}, v_{\mathrm{err}}, R_{6\mathrm{D}}, \omega, \mathrm{rpm}, f_{\mathrm{ext}}, \tau_{\mathrm{ext}}], \\
\mathcal{L}_{\mathrm{id}} &= \big\|\hat{\imath}_t - \imath^{\mathrm{eff}}_t\big\|^2, \quad \imath^{\mathrm{eff}} = [r_{t2w}, \log r_{t2i}, T_{\mathrm{up}}, T_{\mathrm{down}}].
\end{aligned} \tag{5}$$

$\mathcal{L}_{\mathrm{pred}}$ forecasts the next five inertial steps from h_t and the executed commands, $\mathcal{L}_{\mathrm{state}}$ reconstructs state and disturbance under corruption, and $\mathcal{L}_{\mathrm{id}}$ tracks physical characteristics at time t; $\mathcal{L}_P$ is the Gaussian likelihood of the true current channel value. The identity target is the effective one, $r_{t2w}(t) = k_{\mathrm{batt}}(t)\,\bar{k}(t)\,T_{\mathrm{tot}}/(m(t)g)$, so that a payload or a lost motor moves it. Discrete events are up-weighted with $w_t \propto 1 + \beta e_t$, normalised to mean one, giving $\mathcal{L}_{\mathrm{rep}} = \sum_k \lambda_k \overline{w_t \mathcal{L}_k}$, where e_t marks the span of a discrete fault or change.

Training the two jobs. Memory is always supervised on physics. For the expectation we compare two designs. IGR-SUP also trains P_c with $\mathcal{L}_P$ against the true current value $x_c(t)$, not the delayed value that will be delivered, so that the innovation stays tied to the physical state. The innovation then reads μ_c and σ_c through a stop-gradient, and P_c is shaped by the likelihood alone. IGR-TASK sets $\lambda_P = 0$, removes the channel term from distillation and the stop-gradient, so control gradients reach μ_c and σ_c. The expectation then becomes a task-shaped feature, tuned to what helps flight, while the physical objectives still ground memory. This is our proposed configuration.

3.4 Teacher–Student Training

An expert E is trained by PPO [16] on clean state and a drone descriptor d. The teacher T uses the same cell, wider and channel-supervised, trained by PPO with imitation of E decaying to 0.05. Its information flow is $r_t = \varphi(x^{\mathrm{loc}}_{\le t}, a_{<t}, \tilde{x}_{\le t})$ and $a_t = \pi(r_t, \dots, d, m_t x^\star_t)$, with the mask m_t annealed to zero over the first 60% of training. The descriptor and truth reach the controller only, so the representation has to infer identity and state from sensing through the objectives in Eq. (5).

The student has the cell at about one third of the width, no descriptor and no truth. After a short teacher-driven warm-up [7] it runs on-policy against the teacher's actions and capabilities:

$$\mathcal{L}_S = \|a_S - a_T\|^2 + \lambda_s(\mathcal{L}_{\mathrm{rep}} + \mathcal{L}_{\mathrm{cap}}), \quad \mathcal{L}_{\mathrm{cap}} = \lambda_T\|\mathrm{heads}_S - \mathrm{heads}_T\|^2 + \lambda_g \textstyle\sum_c (g_{S,c} - g_{T,c})^2, \tag{6}$$

where the heads are the state, identity and forecast outputs and, for IGR-SUP, the channel means, and $\lambda_s = 0.01$, the lowest value of a validation sweep at $\kappa = 1$, is shared by all variants. New students use $\kappa = 0.5$ and a 100-step backward window with recurrent state carried across windows (Sec. 4.4).

4 Experiments

4.1 Experimental Setup

Platforms and Faults. Drones are sampled over thrust-to-weight, mass, inertia and motor lag following prior work [7], in a JAX simulator with 2 ms physics and 10 ms control (App. B). The 23

Table 1: **Success (%) by suite family**, mean over suites and students (per suite in App. A).

Model	val_id	test_id	Held-out	Sensing	Actuation	Dynamics	Task	Combined
B4 (GRU, expert labels)	76.8	76.8	50.3	57.4	1.0	67.0	50.4	24.0
B5m (Kalman-fed GRU)	86.7	86.1	61.7	**76.3**	5.0	73.2	42.7	36.1
A7 (GRU, teacher labels)	**88.7**	**88.1**	62.8	71.5	**5.2**	73.3	**67.7**	**40.3**
A9 (GRU with physical heads)	84.8	84.1	61.1	69.4	3.1	72.1	67.1	38.6
IGR-SUP	82.9	82.2	58.8	69.5	3.1	69.8	53.4	36.3
IGR-TASK	85.7	85.1	**63.1**	72.8	3.2	**73.7**	66.1	38.9
gain over B4 (points)	+8.9	+8.3	+12.7	+15.4	+2.3	+6.6	+15.7	+14.8
T (wide teacher, with truth in training)	96.2	95.6	69.0	80.5	8.8	78.4	67.2	44.6

held-out suites are grouped by design into sensing (12), actuation, dynamics, task and combined faults (App. A). Each model is evaluated on 1024 episodes per suite on held-out drones. There are three teacher seeds; students train for 1000 epochs with one selection rule.

Baselines. B0, a released prior foundation policy [7], is evaluated through our corrupted sensing layer and shown for reference only (App. A). B4 is a size-matched GRU on the same native inputs with expert labels. B5m is a size-matched GRU with teacher labels fed by a delay-aware Kalman filter with χ^2 gating and no timestamps, the same-information classical pipeline. A7 is a plain GRU with the same teacher labels, and A9 adds the physical heads to it. A1 is IGR-SUP with expert labels and A3 the cell with no auxiliary objectives. All students share one training budget. IGR cells and A9 use a 100-step backpropagation window for stability (Sec. 4.4) while B4, A7 and B5m use 500; comparisons among A3, A9 and IGR share a horizon, and we report the comparison with A7 as parity. IGR-TASK and the leave-one-out ablations have three students, the others six.

Metrics and Protocol. Success is survival for the whole 20 s episode. For the event evaluation each episode is run twice with identical random numbers, with and without one event, and the harm of the event is the tracking error it adds,

$$X = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \int_{t_e}^{t_e + T_e + 2\,\mathrm{s}} \max\big(0,\, e_i(t) - e_i^{\mathrm{cf}}(t)\big)\, dt, \tag{7}$$

with e the position error, $\mathcal{S}$ the drones whose counterfactual survives the integration window, and a crash counted as maximal error from the crash onward. X needs no access to hidden states, so it compares architectures on equal terms. Misleading events are E1 (0.5 s velocity freeze), E4 (1 s reorder) and E5 (1 s late reference); real changes are E2 (gust of $0.25\,mg$) and E7 (payload $+30\%$), with detection read from the physical heads (App. A). A probe regresses a physical quantity on the recurrent state, reduced to 16 principal components to match capacity, on held-out episodes.

Analysis. Each subsection below answers one question. The teacher seed is the unit of replication, so intervals are 95% t-intervals over the three teacher means. Selection and evaluation use separate seeds.

4.2 How Much Comes from Teacher Distillation?

Training labels come first. On the same architecture, teacher labels add 12.5 points over expert labels (A7 vs. B4, 62.8% vs. 50.3%; interval 1.8 to 23.2; +9.6, +17.4, +10.4 per teacher). The delay at 50% success rises from 6 ms for the released policy and 98 ms for B4 to 145 ms for A7 and 140 ms for IGR-TASK, against at most 40 ms in training (Table 1).

4.3 Can Memory Become Readable without Losing Control?

Table 2 and Fig. 3a show that it can. **Physical objectives make recurrent memory readable**: at matched capacity the external-force probe rises from 0.29 for A7 to 0.62 for a plain GRU with heads (A9), so readability alone does not need the cell. IGR-TASK combines the highest force probe, 0.64, with held-out success of 63.1%, against 61.1% for A9. On the full state the gap to A7 is smaller (0.66 vs. 0.50). It matches A7 in aggregate (difference +0.3, interval −1.9 to +2.4) with the same crash-inclusive error (1.124 vs. 1.119 m) and stays 12.7 points above B4 (interval 2.9 to 22.6). Within the cell the objectives also add modest control: against the same cell with none (A3),

Table 2: **Factorial ablation** with teacher labels (bold: best in column): success (%), PCA-16 probe R^2 for velocity error and force at 100 ms delay, and mean $\bar{X}$ over E1, E4, E5 (not run for A9).

Variant	Structure	Objectives	Held-out	Sensing	R^2_v	R^2_f	$\bar{X}$
A7	plain GRU	none	62.8	71.5	0.67	0.29	0.239
A9	plain GRU	state, forecast, identity	61.1	69.4	**0.88**	0.62	—
A3	IGR cell	none	62.4	71.9	0.71	0.42	**0.231**
IGR-SUP	IGR cell	all, channel supervised	58.8	69.5	0.79	0.45	0.283
IGR-TASK	IGR cell	all but channel; task-trained	**63.1**	**72.8**	0.85	**0.64**	0.253

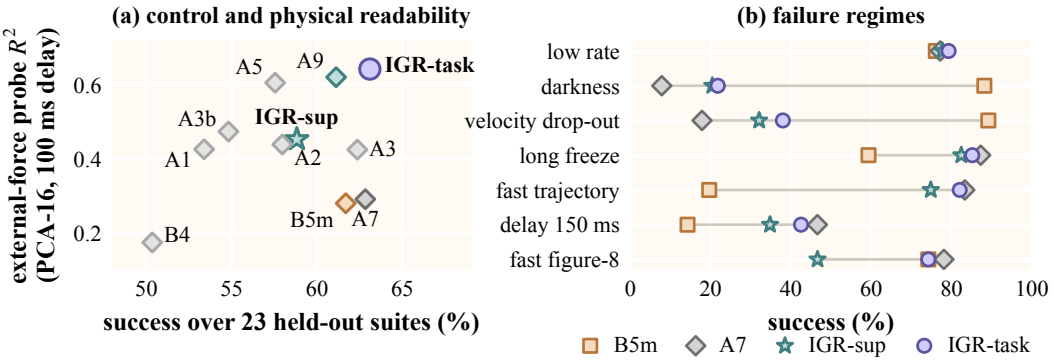


Figure 3: **Control, readability and failure regimes.** (a) Held-out success against an external-force probe; the upper right combines both, where IGR-TASK and A9 sit closest. (b) Success on selected suites: memory and the Kalman-fed controller fail in different places.

matched on student seed, IGR-TASK gains 0.95 points held-out (interval 0.27 to 1.64) and 1.40 on sensing (interval 0.85 to 1.95), every teacher positive, while its velocity and force probes rise from 0.70 to 0.85 and from 0.43 to 0.64 in every seed.

4.4 What Changes when Control Trains the Expectation?

This is the central comparison. **IGR-TASK improves on IGR-SUP by 4.3 points over the held-out suites** (interval 3.3 to 5.3; +4.6, +3.8, +4.4 per teacher), with the same memory objectives. The cost of supervision concentrates out of distribution: on a fast figure-eight never seen in training, every cell with a supervised channel lies at or below 61% (46.6% for IGR-SUP), and the two without it reach 72% and 74%.

Task training also kept the recurrence stable in our tested configurations. IGR-TASK and A3, with no channel supervision, peaked at 0.42 and 0.48 with no rejected update, as plain GRUs do (Fig. 4c). With a supervised channel, the correction path can instead amplify gradients by about 1.05 per step, which compounds over 500 steps. At $\kappa = 1$ with full-length backpropagation, peaks exceeded 10^{20} after epoch 400 on all three students (Fig. 4a), yet clipping kept training running and the first 400 epochs looked healthy.

Damping helps but is not a general cure. Halving κ alone left peaks of 4×10^{21}. Adding a 100-step window bounded IGR-SUP (peak 39) and raised validation success at epoch 1000 from 76% to 82% (Fig. 4b), but other supervised variants still reached 10^8–10^{11}. Supervised variants without gates (A2) or innovation features (A5) stayed stable, so the instability appears to need the supervised channel and gated innovation together.

4.5 Where Does the Structure Help or Fail?

IGR-TASK leads A7 on 6 of 23 suites and trails by 3.0 points in distribution. **Its clearest advantage is sparse sensing.** At 5–20 Hz it improves success from 77.2% to 79.3% (interval 1.2 to 2.9, positive for every teacher) and crash-inclusive error from 0.850 to 0.759 m (interval −0.16 to −0.02). A9, on the same horizon, reaches 75.2% and 0.908 m.

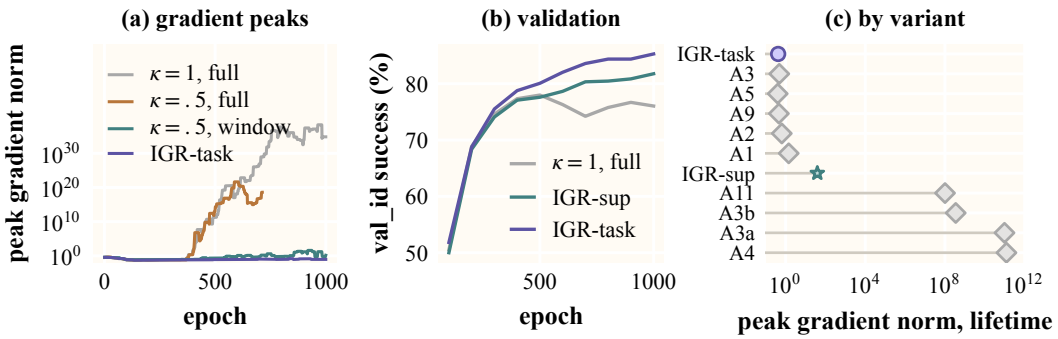


Figure 4: **Numerical stability.** (a) Running peak gradient norm, largest over students. (b) Validation success. (c) Lifetime peak by variant at $\kappa = 0.5$ with the window.

Table 3: **Event evaluation.** Added error X in m·s (bold: lowest) and head detection among survivors.

	misleading data			real change	
Model	E1	E4	E5	E2	E7
B5m (Kalman-fed GRU)	2.074	0.416	**0.233**	0.035	0.274
A7 (GRU, teacher labels)	**0.339**	**0.117**	0.260	0.039	**0.265**
A2 (cell without gates)	0.412	0.133	0.235	0.048	0.281
IGR-SUP	0.402	0.149	0.298	0.040	0.276
IGR-TASK	0.342	0.138	0.280	**0.033**	0.269
Harm reduction, IGR-TASK vs. B5m	+84%	+67%	−20%	+4%	+2%
Head detection, IGR-SUP	—	—	—	74%	96%
Head detection, IGR-TASK	—	—	—	**82%**	**96%**

Learned memory and a filter fail in opposite places (Fig. 3b). B5m is strongest where a signal is absent: 88% and 89% in darkness and sustained drop-out, against 8% and 18% for A7. It is weakest where a signal is present but stale, or the motion leaves its model: 59% on long freezes and 20% on fast trajectories, against 87% and 83% for A7. Without the two outage suites the sensing means are 83.2% for A7, 81.4% for IGR-TASK and 73.8% for B5m. An oracle choosing the better of A7 and B5m per episode would reach 70.9%, which suggests the two are complementary. The gated cell narrows the gap on long outages absent from training: under expert labels it raises their mean success from 5.2% (B4) to 28.1% (A1; interval 7.2 to 38.7 points, every teacher positive), and under teacher labels from 12.8% (A7) to 29.9% (IGR-TASK), a gain of 17.0 points (+40.3, +11.4, −0.6). Within IGR-SUP, removing the gates costs 9.9 points on these suites and removing innovation features 9.6, positive for every teacher.

The gains are concentrated, not uniform. IGR-TASK shows no general robustness advantage over a plain recurrent controller: it improves under sparse sensing and outages while aggregate success stays comparable to A7. With the readability of Sec. 4.3, it suits settings that need physically informative memory, not general-purpose flight.

4.6 Does the Representation Distinguish Physical Change from Sensing Faults?

The physical heads respond differently to real change and sensor faults (Fig. 5c,d). During a gust the IGR-TASK force head rises by $0.053\,mg$ on average, against at most $0.004\,mg$ during a freeze, reorder or late reference. After a payload its thrust-to-weight head falls by 0.56, against at most 0.04 during those faults, and settles close to the teacher's reading. Heads detect 82% of gusts and 96% of payloads, against 21% and 49% for untrained heads. This holds on a figure-eight outside the training tasks, where IGR-TASK detects 72% and 94%, against 26% and 72% for IGR-SUP.

Control responses are a separate question. X on real changes is the lowest of any model on gusts and level with A7 on payloads. Against B5m, IGR-TASK cuts the harm of a freeze by 84% and of a reorder by 67%; a freeze at peak acceleration crashes B5m in 37% of valid episodes, against 3% for

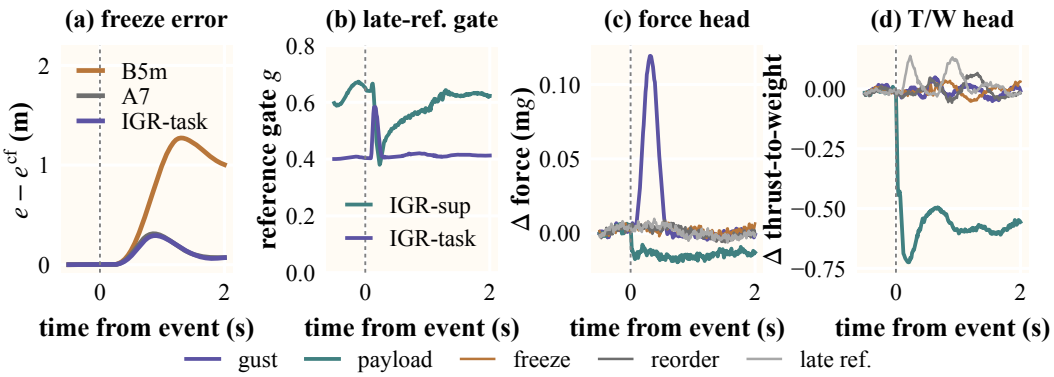


Figure 5: **Paired-rollout traces**, means over drones and students. (a) Error added by a freeze. (b) Reference gate during a late reference. (c, d) Change in the IGR-TASK heads for two real changes (heavier) and three sensor faults.

IGR-TASK and 1% for A7 (Fig. 5a). On the figure-eight IGR-TASK lowers X against IGR-SUP on all five events and against A7 on reorder, late reference and gust. In the in-distribution protocol A7 has slightly lower X on the three misleading events (Table 3), in line with the concentrated gains of Sec. 4.5.

The response is broader than a closing gate. For IGR-SUP, velocity NIS doubles under reorder (1.46 to 3.11), and the reference gate closes under a late reference (0.59 to 0.51, Fig. 5b). IGR-TASK has lower X on that event with a gate that stays open, so the response is also carried by the features and memory. Clamping the representation to its episode mean removes 82 points of IGR-SUP success and 87 for A3, so control depends on it.

5 Conclusion

We proposed IGR, a predict–compare–write structure that conditions memory updates on learned, normalised discrepancy features inside a recurrent controller. Physical objectives on memory with a task-trained expectation gave a stronger combination of control and readability than a supervised expectation. The resulting controller matches the strongest distilled GRU in aggregate, adds physical readouts and gains at low sensing rates. A distilled GRU suits flight alone, IGR-TASK adds readouts and sparse sensing, and a filter suits long outages.

Limitations. All experiments are in simulation on our own sensing model, without hardware or sim-to-real validation. Results rest on three teacher seeds, with three IGR-TASK students, and outage gains vary by teacher; one expert left 15% of training drones unsolved (masked from labels). IGR-TASK changes the likelihood, its distillation term and the stop-gradient together, so our finding concerns that package, not one cause. Its μ_c, σ_c are control features, not calibrated estimates. Detection is measured among survivors, and without false-alarm rates or calibrated magnitudes the heads show distinct responses, not a validated detector. The controller has 11,097 parameters excluding heads; with the state and identity heads kept as readouts it has 15,254. Strong actuation delay and severe fault combinations remain unsolved.

Future Work. We plan to extend IGR-TASK to more seeds on a common horizon, separate supervision from gradient routing, measure head false alarms, learn a filter–memory switch, and fly on hardware.

6 Broader Impacts

Fault-tolerant learned control may improve the safety of autonomous vehicles; the techniques could also serve dual-use systems, and over-generalising from simulation to real flight is a risk we limit by stating scope.

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[2] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley, 2001.

[3] Philipp Becker, Harit Pandya, Gregor Gebhardt, Cheng Zhao, C. James Taylor, and Gerhard Neumann. Recurrent Kalman networks: Factorized inference in high-dimensional deep feature spaces. In *International Conference on Machine Learning (ICML)*, 2019.

[4] Yann Bouteiller, Simon Ramstedt, Giovanni Beltrame, Christopher Pal, and Jonathan Binas. Reinforcement learning with random delays. In *International Conference on Learning Representations (ICLR)*, 2021.

[5] Chayanin Chamachot. Evaluating factor-wise auxiliary dynamics supervision for latent structure and robustness in simulated humanoid locomotion. *arXiv preprint arXiv:2603.21268*, 2026.

[6] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[7] Jonas Eschmann, Dario Albani, and Giuseppe Loianno. RAPTOR: A foundation policy for quadrotor control. *Science Robotics*, 2026. arXiv:2509.11481.

[8] Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel. Backprop KF: Learning discriminative deterministic state estimators. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[9] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, 2019.

[10] Gwanghyeon Ji, Juhyeok Mun, Hyeongjun Kim, and Jemin Hwangbo. Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robotics and Automation Letters*, 7(2), 2022.

[11] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems (RSS)*, 2021.

[12] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5), 2008.

[13] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62), 2022.

[14] Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1), 1999.

[15] Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar. KalmanNet: Neural network aided Kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70, 2022.

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

[17] Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[18] Dingqi Zhang, Antonio Loquercio, Xiangyu Wu, Ashish Kumar, Jitendra Malik, and Mark W. Mueller. Learning a single near-hover position controller for vastly different quadcopters. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. arXiv:2209.09232.

A Sensing Layer and Fault Suites

Table 4: **Channels seen by every native-sensing model**, each through the sensing layer.

Channel	Dim.	Source and faults
gyro, accel	$3 + 3$	onboard IMU; noise, bias random walk, vibration
attitude	6 (6D)	Mahony filter [12] with a heading reference; gain varied across drones
ext_pos, ext_vel	$3 + 3$ + valid	link: delay, jitter, drops, freezes, reorder bursts, outages; velocity from a motion-capture-like source or optical flow with low-light noise
reference	9 + valid	lookahead of the reference over the link, same faults
action history	24	last $K{=}6$ commands (actuation delay applies)

Table 5: **Held-out suites.** Each changes the nominal evaluation condition as stated. Training covers delay $\leq$ 40 ms, actuation delay $\leq$ 20 ms, freezes $\leq$ 0.3 s, reorder window $\leq$ 2, wind $\leq$ 3 m/s, payload $\leq 20\%$, and no full darkness, velocity outage, fast trajectory or figure-eight.

Family	Suites
Sensing (12)	external delay 60, 100, 150 ms; delay burst ($+200$ ms for 0.3–1 s); severe reorder (window 4–6); late reference (100 ms, window 4, 20% drop); long freeze (0.5–1 s); low rate (5–20 Hz); sustained velocity drop-out (2–5 s); darkness (flow, zero light, 3–6 s); inertial bias $\times 3$; vibration $\times 3$
Actuation (2)	actuation delay 40, 60 ms
Dynamics (4)	strong wind (5–8 m/s with gusts); payload up to $+40\%$; motor loss; high thrust-to-weight drones
Task (3)	fast trajectory; fast figure-eight; mid-air recovery (100 ms delay)
Combined (2)	delay with gusts; all faults together

Table 6: **Event library** for paired rollouts. Events start at peak reference acceleration.

Kind	Event	What happens	Detection readout
misleading data	E1	external velocity frozen for 0.5 s	—
	E4	reorder window of 6 packets for 1 s	—
	E5	reference delayed 150 ms, reordered for 1 s	—
real change	E2	lateral gust, $0.25\,mg$ for 0.6 s	force head rises within 0.5 s
	E7	payload $+30\%$	thrust-to-weight head deviates within 1 s

Table 7: **Success (%) by suite**, mean over students.

	Suite	B0	B4	B5m	A7	A9	A3	IGR-SUP	IGR-TASK
S	delay 60 ms	6.6	80.3	88.3	**90.2**	87.7	88.8	86.0	88.6
S	delay 100 ms	0.5	48.1	73.6	**82.4**	79.4	76.7	74.0	79.6
S	delay 150 ms	0.0	12.0	14.2	**46.6**	42.5	36.7	34.8	42.5
S	delay burst	37.2	81.6	83.2	**90.6**	87.7	89.6	86.6	88.4
S	severe reorder	9.4	76.9	76.1	**88.4**	86.6	87.4	84.7	87.4
S	late reference	50.2	79.1	87.5	**88.1**	84.6	86.8	83.1	86.1
S	long freeze	31.6	77.0	59.3	**87.3**	84.0	86.0	82.4	85.2
S	low rate	8.2	59.6	76.1	77.2	75.2	78.4	77.7	**79.3**
S	sustained drop-out	4.7	8.5	**89.2**	17.9	21.4	35.2	32.1	38.0
S	darkness	1.2	1.8	**88.2**	7.8	8.3	18.4	20.4	21.8
S	inertial bias $\times 3$	49.6	83.9	91.0	**92.0**	89.2	90.7	88.2	90.1
S	vibration $\times 3$	50.9	79.7	88.6	**89.4**	86.0	88.0	84.5	87.1
A	actuation delay 40 ms	0.4	1.9	9.1	**9.6**	5.8	7.2	5.7	6.0
A	actuation delay 60 ms	0.0	0.0	**0.8**	0.7	0.4	0.5	0.4	0.5
D	strong wind	51.6	83.1	90.5	**91.0**	87.9	89.9	87.3	89.6
D	payload $+40\%$	43.7	75.4	81.5	**81.8**	79.9	81.5	77.4	80.8
D	motor loss	42.8	68.0	68.1	68.0	69.2	**70.9**	66.0	70.2
D	high thrust-to-weight	11.0	41.6	53.0	52.3	51.5	52.6	48.6	**54.0**
K	fast trajectory	0.1	60.7	19.6	**83.3**	81.6	83.0	74.8	82.0
K	fast figure-8	24.4	71.1	74.2	**78.1**	77.0	71.8	46.6	74.2
K	mid-air recovery	0.0	19.4	34.4	41.7	**42.6**	38.4	38.8	42.0
C	delay and gust	0.2	48.1	72.2	**80.6**	77.2	75.7	72.6	77.7
C	all faults	0.0	0.0	0.0	0.0	0.0	**0.1**	0.0	0.0

B Implementation Details

Table 8: **Training configuration.**

Stage	Settings
Expert E	PPO, 3×10^8 steps per seed, 4096 environments, learning rate 3×10^{-4} decayed linearly, $\gamma = 0.99$, GAE 0.95
Teacher T	recurrent PPO, 5×10^8 steps, 2048 environments, 64-step backpropagation, learning rate 2×10^{-4}, $\kappa = 1$, objectives weighted 1, 1, 1, 0.5 (forecast, state, channel, identity); widths $72/48/32$ (memory, controller, r); 79,869 deployed parameters
Student S	1000 epochs, 1024 environments, 500-step sequences, 64 sequences per mini-batch (16 Adam steps per epoch), constant learning rate 10^{-4}, gradient clipping at norm 1, 10 warm-up epochs, $\beta = 4$, checkpoints every 100 epochs; widths $24/16/16$; $\kappa = 0.5$, 100-step window (B4, A7, B5m: 500-step); $\lambda_{\text{pred}} = \lambda_{\text{state}} = \lambda_P = 1$, $\lambda_{\text{id}} = \lambda_T = 0.5$, $\lambda_g = 0.1$, $\lambda_s = 0.01$
Filter (B5m)	per-axis position, velocity and bias; χ^2 gate at 99%; 1 s lock-out; process-noise scale 1000 and nominal delay 20 ms selected on validation
Parameters	IGR-SUP and IGR-TASK 11,097 without heads, 15,254 with state and identity heads; B4 and A7 11,138; B0 2,084; all heads 8,987 (state 3,225, forecast 4,830, identity 932)

Lessons from earlier runs. A cosine learning rate from 10^{-3} diverged, so students use a constant 10^{-4} with overflow-safe clipping. An unbounded write let memory grow without limit, which motivated the outer $\tanh$. The stop-gradient through the supervised innovation cuts a feedback loop whose gain reaches $1/\sigma$. A filter noise sweep that was too narrow rejected valid packets, so the sweep was widened, with a guard against estimators worse than raw data. Sec. 4.4 gives the κ and window results.

C Leave-One-Out Ablations

Table 9: **Difference from IGR-SUP** in success points with teacher-level 95% intervals, and lifetime peak gradient norm.

Variant	Students	Held-out	Sensing	Peak gradient
IGR-TASK (no channel supervision, task-trained)	3	+4.3 [3.3, 5.3]	+3.3 [0.1, 6.5]	0.42
A7 (plain GRU, same teacher)	6	+4.0 [2.3, 5.7]	+1.9 [−2.2, 6.1]	0.6
A3 (no auxiliary objectives)	6	+3.5 [2.3, 4.8]	+2.4 [−0.3, 5.0]	0.48
A9 (plain GRU with heads)	6	+2.3 [−1.1, 5.7]	−0.2 [−5.8, 5.5]	0.46
A2 (no gates)	3	−0.9 [−2.5, 0.8]	−2.1 [−5.4, 1.2]	0.64
A5 (raw embeddings, no innovation)	3	−1.3 [−1.5, −1.1]	−2.6 [−4.7, −0.5]	0.4
A11 (no capability or gate distillation)	3	−2.2 [−5.5, 1.1]	−3.9 [−8.2, 0.4]	1.0×10^8
A3a (no inertial forecast)	3	−2.7 [−5.3, −0.1]	−3.2 [−7.3, 0.9]	9.7×10^{10}
A3b (no state or disturbance target)	3	−4.0 [−6.6, −1.5]	−4.0 [−5.9, −2.1]	3.5×10^8
A4 (no control gradient into the representation)	3	−4.8 [−12.6, 3.0]	−4.5 [−13.1, 4.1]	1.2×10^{11}
A1 (expert labels)	6	−5.4 [−17.1, 6.3]	−6.7 [−20.6, 7.3]	1.4

D Reproducibility and Compute

All runs used one NVIDIA L40S GPU with JAX 0.12. Expert training took about 13 minutes, teachers 56 minutes, and the students, evaluation and analysis 3.9 hours, of which the 36-student program took 89 minutes. Reported numbers use evaluation seed 54321 and selection seed 777. Code and configuration will be released in anonymised form; every number, table and data figure here is generated from stored evaluation output by scripts in that release.