FORWARD-ONLY LEARNING ACROSS BLACK-BOX INTERFACES: STATE PERSISTENCE AND PILOT RECOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

A changing black-box interface can present different coordinates to a downstream learner even when every realized transformation is invertible. We study how interface-state persistence and observability affect forward-only local learning without gradient propagation across the interface. Matched fixed and resampled channels share the same state distribution and paired upstream representations, separating per-state properties from consistency across calls. Experiments cover vanilla Forward-Forward, channel-wise competitive learning, supervised block-local learning, and an external DeeperForward ResNet. Matched Rademacher comparisons show that resampling invertible, norm-preserving states can cause substantial accuracy degradation. We introduce shared pilots that pass through the same hidden state as the data, making structured interface transformations observable from forward outputs. For invertible diagonal states and signed permutations, pilot constructions enable exact representation recovery under noiseless observations without upstream credit. On frozen DeeperForward representations, a single shared pilot restores downstream accuracy from 13.73% to 76.90%, matching the identity condition in all five paired seeds. These results distinguish per-state information preservation from downstream usability and identify state persistence or observability as key interface properties for forward-only learning.

1 INTRODUCTION

Forward-only local learning trains connected neural modules without propagating gradients across their interfaces. This setting is useful when a component is remote, proprietary, non-differentiable, or accessible only through forward queries. The Forward-Forward (FF) algorithm develops this approach through local goodness objectives (Hinton, 2022). Subsequent work extends local learning to convolutional architectures, layer coordination, and distributed implementations (Papachristodoulou et al., 2023; Lorberbom et al., 2023; Aktemur et al., 2024). These developments make the interface between modules part of the learning problem: avoiding backward communication does not determine whether the returned representation supports effective downstream training.

An unknown interface can alter both the information available to a learner and its representation. A fixed invertible transformation preserves information, although a restricted downstream model may struggle to use the transformed coordinates. If the hidden transformation changes between calls, the same upstream representation can produce different observations. Each realized map may remain invertible, while the marginal observations available without the state can lose label information. This distinction raises two related questions: how does hidden interface-state variation affect local learning, and when can forward observations support representation recovery?

We investigate these questions through matched fixed and resampled channels. The fixed condition retains one state throughout a run; the resampled condition draws a new state at each call. Both conditions use the same state distribution and paired upstream representations. Rademacher transformations provide a controlled comparison because every realized state preserves coordinatewise magnitudes, the Euclidean norm, and invertibility. Resampling nevertheless produces substantial downstream accuracy losses. Complementary analysis explains when normalization removes interface changes and how post-interface task information constrains prediction.

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.

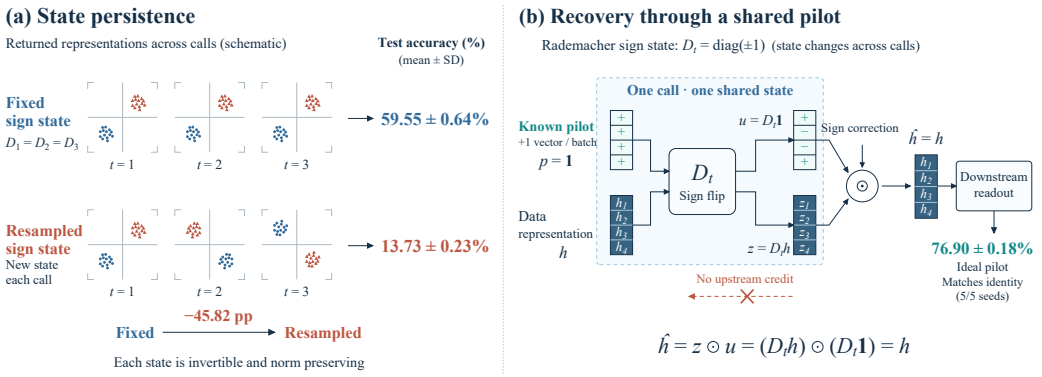

Figure 1: Forward-only learning across a changing interface. (a) Persistent and resampled Rademacher states produce different representation sequences, although every realized map is invertible and norm preserving. (b) A shared all-ones pilot reveals diagonal signs and restores the representation without upstream credit. Scatter plots and vectors are schematic. Accuracies use CIFAR-10 frozen DeeperForward features and a trained downstream readout, with mean ± sample SD over five paired seeds. The ideal pilot matches identity in every seed.

To recover a common representation when the interface state changes, we introduce *shared pilots*. Pilot-based channel estimation and representation canonicalization provide relevant precedents (Soltani et al., 2020; Kaba et al., 2023). Known reference vectors and data pass through the same unknown state within a logical call. A pilot with nonzero coordinates identifies an invertible diagonal state. For signed permutations, an all-ones pilot identifies the signs, and binary-coded pilots identify the coordinate permutation. Under noiseless shared-state observations, these constructions recover the original representation through forward outputs alone, without sending task or channel credit upstream. Figure 1 illustrates the state-persistence comparison and diagonal-pilot recovery.

Pilot recovery acts on the representation used for downstream training as well as prediction. Repeated-query averaging instead averages the scores of an already trained model. Our conditional concentration analysis characterizes how independent calls reduce sampling error when the mean-score margin is positive. This comparison separates reducing prediction variability from recovering the input on which the downstream module learns.

Evaluation covers vanilla FF, channel-wise competitive learning, supervised block-local learning, and an external DeeperForward ResNet (Sun et al., 2025). On frozen DeeperForward representations from CIFAR-10, downstream accuracy falls from 59.55% with a fixed Rademacher state to 13.73% with resampling. A single shared pilot restores accuracy to 76.90%, matching the identity-interface condition in all five paired seeds. Additional experiments examine pilot noise, pilot–data state mismatch, and communication and computational overhead. Together, these results connect interface-state properties, explicit recovery conditions, and downstream learning outcomes.

Our contributions are threefold:

- **Interface-state analysis.** Matched fixed and resampled channels show that per-state invertibility and norm preservation alone do not ensure effective downstream local learning.
- **Structured forward recovery.** We provide shared-pilot constructions and exact recovery conditions for invertible diagonal states and signed permutations, without upstream credit.
- **Empirical validation and recovery costs.** We evaluate state persistence and recovery across local learners and representations, including imperfect pilot observations and the cost of additional forward observations.

2 Related Work

Forward-Forward and local goodness learning. FF separates positives from negatives with a local goodness score (Hinton, 2022). Later work coordinates layers (Lorberbom et al., 2023), replaces

explicit negatives with cascaded class predictions (Zhao et al., 2023), adds recurrent predictive-coding dynamics (Ororbia & Mali, 2023), or adopts symmetric goodness objectives (Terres-Escudero et al., 2024). Self-contrastive FF constructs paired views for natural images and sequential data (Chen et al., 2025). We fix the learning rule and vary its input channel.

Dimensionality-compression FF uses correlated fluctuations for representation learning and mean-energy inference (Zhu et al., 2025), while Distance-Forward studies gradient, input, and sensor noise with a margin objective (Wu et al., 2024). We instead compare fixed and resampled post-representation channels to reveal how boundary variation affects downstream learning.

Noisy intermediate features and task-oriented communication. Communication between modules places task information under channel constraints. Split computing links intermediate-feature transmission to accuracy and communication cost (Matsubara et al., 2022). BottleNet++ learns an encoder and decoder around Gaussian and binary-erasure channels (Shao & Zhang, 2020). Task-oriented communication preserves label-relevant features (Shao et al., 2022), while robust information bottleneck learns protective redundancy (Xie et al., 2023). DFTS2 studies packet loss and concealment (Dhondea et al., 2021), and I-SPLIT selects split points using neuron importance (Cunico et al., 2022). With no downstream task or channel credit upstream, paired states isolate the downstream response to the interface.

Convolutional and scalable forward learning. CwC divides channels into class groups and trains each layer as a local classifier (Papachristodoulou et al., 2023). Recent systems use independent blocks, learning-rate schedules, modern backbones, hierarchical targets, and supervised contrastive features (Krutsylo, 2025; Torres et al., 2025; Yao et al., 2026). Binary stochastic FF studies device-oriented activations and hardware costs (Jaiswal et al., 2025), while distributed FF separates layers across devices (Aktemur et al., 2024).

Pilot observations can recover changing channel states. Communication systems use pilots for channel estimation, including learned placement (Soltani et al., 2020), while canonicalization maps transformed inputs to a common representation (Kaba et al., 2023). We use shared pilots to identify structured states at a forward-local interface, derive recovery conditions, and measure downstream accuracy.

Alternatives to end-to-end gradients. Supervised local-error networks use layerwise auxiliary objectives (Nøkland & Eidnes, 2019), while greedy and decoupled module training scale to ImageNet and relax update locking (Belilovsky et al., 2019; 2020). InfoPro regularizes task-relevant information passed between local modules (Wang et al., 2021). Approximate non-local credit can improve performance across local learning settings (Kunin et al., 2020), and direct feedback alignment sends output error through fixed random projections (Nøkland, 2016). Top-down feedback connects FF and PEPITA-like rules to adaptive feedback alignment (Srinivasan et al., 2023), whereas orthogonal constructions study how forward updates approximate backpropagation (Caillon et al., 2025). Our block-local and STE controls test how upstream credit changes the response to the same interface.

3 Problem Setting and Interface Protocol

The interface connects an upstream representation to a downstream learning rule. An upstream block produces $H = f_\theta(X)$, and the channel returns $Z = T_\xi(H)$. Its state ξ remains fixed in static conditions and is resampled in stochastic conditions. The downstream block g_ϕ learns from Z without differentiating through T_ξ. Vanilla FF separates positive and negative examples with local objectives. CwC minimizes cross-entropy over class-group squared activations. Supervised block-local BP gives each block a classifier and detaches the inter-block representation. In each case, the channel determines the representation distribution available for downstream training.

The channel suite separates information loss, amplitude changes, and state persistence. It includes identity, orthogonal mixing, positive sample-dependent gain, saturation, support-matched quantization, and fixed low-rank projection. Fixed and random erasure, stochastic sign/flip, and drifting mixing add different forms of state variation. Static conditions retain one state throughout a run. Stochastic channels execute at every downstream update and inference call. Positive and negative

examples share the instantaneous state when they form paired measurements. State variation therefore remains part of the learning process.

Quantization levels match the support of the incoming activations. ReLU outputs use unsigned levels that retain zero. Signed post-normalization features use symmetric levels. Symmetric levels applied to sparse non-negative activations can shift zero and cause artificial collapse. The main aggregates use support-matched conditions to measure the channel effect.

Frozen representations let us measure class structure and state variation before downstream training. A cosine nearest-centroid probe fits centroids on training data and measures class structure available to that classifier. Repeated-forward cosine similarity measures directional stability across calls. Both diagnostics use training data, with test labels excluded from fitting. Their combination links task-relevant structure to the consistency of the outputs presented to the learner.

4 Information, State Persistence, and Recovery

Downstream FF layers normalize the interface output,

$$N_\varepsilon(z) = \frac{z}{\|z\|_2 + \varepsilon}. \tag{1}$$

This makes the direction of z more consequential than its raw amplitude.

Theorem 1 (Positive-radial equivalence)**.** *Suppose $T_2(h) = c(h)T_1(h)$, with $c(h) > 0$, wherever $T_1(h) \neq 0$. Then $N_0(T_2(h)) = N_0(T_1(h))$. Assume downstream computations access the interface through N_0 alone. Coupled minibatches, initialization, and optimizer randomness then give identical downstream parameter trajectories.*

For finite ε, let $r = \|T_1(h)\|_2$ and $T_2(h) = cT_1(h)$. Direct simplification gives

$$\|N_\varepsilon(T_2(h)) - N_\varepsilon(T_1(h))\|_2 = \frac{\varepsilon r|c-1|}{(cr+\varepsilon)(r+\varepsilon)}. \tag{2}$$

An L-Lipschitz downstream score changes by at most L times this quantity. Ideal normalization removes positive gain, while finite stabilization introduces the stated difference. This explains why an amplitude change can leave downstream learning unchanged.

Differentiability cannot overcome lost task information. For the Markov chain $Y \to H \to Z = T(H)$, every downstream classifier has risk at least

$$R^*(Z) = 1 - \mathbb{E}_Z \max_y p(y \mid Z), \tag{3}$$

the post-interface Bayes risk. A channel that merges class-conditional distributions raises this floor, regardless of the downstream objective.

Proposition 1 (Pilot identifiability for diagonal state)**.** *Let a call return $D_s z$, where D_s is an unknown invertible diagonal state. A known pilot p has no zero coordinate and shares that state. Its output identifies $D_s = \mathrm{diag}((D_s p) \oslash p)$. The downstream block then recovers $z = D_s^{-1}(D_s z)$ through forward outputs alone.*

For Rademacher states and $p = \mathbf{1}$, recovery reduces to $(D_s z) \odot (D_s \mathbf{1}) = z$. Sharing the diagonal state makes the pilot informative about the data transformation. The downstream module can then recover the input from the returned pair.

Proposition 2 (Coded-pilot identification of signed permutations)**.** *Let a call return $D_s P_s z$, with unknown permutation P_s and diagonal sign matrix D_s. One all-ones pilot identifies D_s. Another $\lceil \log_2 d \rceil$ pilots assign distinct binary codes to the source coordinates. Their sign-corrected outputs identify the source coordinate at each output position. Scattering the corrected data to those coordinates recovers z exactly.*

The construction uses $1 + \lceil \log_2 d \rceil$ pilots, giving 16 pilots at our $d = 20{,}480$ interface. Distinct binary codes identify coordinate permutations, while dense linear maps require richer observations. Figure A1 illustrates how the codes reveal the source coordinates.

Shared-state interface and local computation. The interface accepts known feature-space pilots alongside data representations and applies one shared hidden state within each logical call. The receiver knows the pilots and observes only the returned data and pilot responses, not the state itself. Paired FF positive and negative examples share this call. Pilot rows are removed before downstream batch normalization, classification heads, or local-loss evaluation. In Algorithm 1, g_ϕ denotes the original downstream computation, including any frozen subsequent blocks; only its designated parameters ϕ_{train} are updated.

Algorithm 1 Shared-pilot recovery for forward-local training and inference

Input: Frozen upstream f_θ; downstream computation g_ϕ and original local loss $\mathcal{L}_{\text{local}}$; trainable subset ϕ_{train}; shared-state interface of family $\mathcal{F}$; feature dimension d.

1 $P \leftarrow \text{ConstructPilots}(\mathcal{F}, d)$; $N_p \leftarrow \text{rows}(P)$
2 **for each** local-training minibatch B **do**
3 $(H, \tau) \leftarrow \text{ConstructLocalExamples}(f_\theta, B)$
4 $H \leftarrow \text{stop_gradient}(H)$
5 $V \leftarrow \text{QueryOnce}(\text{concat_rows}(P, H))$ *one shared state*
6 $(U, Z) \leftarrow \text{SplitPilotData}(V, N_p)$ *Z contains data rows only*
7 **if** $\mathcal{F}$ is diagonal **then**
8 $\widehat{a} \leftarrow U_{0,:} \oslash P_{0,:}$
9 $\widehat{H} \leftarrow Z \oslash \widehat{a}$ (Rademacher: $\widehat{H} \leftarrow Z \odot \widehat{a}$)
10 **else if** $\mathcal{F}$ is signed permutation **then**
11 $\widehat{s} \leftarrow U_{0,:}$
12 $b \leftarrow \text{BinaryDecode}(U_{1:m,:} \odot \widehat{s})$, $m = \lceil \log_2 d \rceil$
13 $\widehat{H}_{:,b_j} \leftarrow \widehat{s}_j Z_{:,j}$ **for all** $j = 0, \ldots, d-1$
14 **end if**
15 Restore $\widehat{H}$ to the native feature shape expected by g_ϕ
16 $\ell \leftarrow \mathcal{L}_{\text{local}}(g_\phi(\widehat{H}), \tau)$
17 Update ϕ_{train} using ℓ and the original local training rule
18 **end for**

Pilots: Diagonal states use $P_{0,:} = \mathbf{1}$; signed permutations add m binary-code rows (Appendix B.4). Products and divisions broadcast across rows. The Rademacher implementation retains multiplicative correction under pilot perturbations.

Inference: Initialize P, N_p with line 1, construct H using frozen f_θ, and apply lines 4–15. Use the original downstream prediction rule with no parameter updates.

Repeated queries average the variation from a changing channel. For a fixed example with class y, let $S^{(q)}$ denote the score vector from query q. Write the average as $\bar{S}_Q = Q^{-1} \sum_q S^{(q)}$. The mean-score margin is $\Delta = \mathbb{E}S_y - \max_{j \neq y} \mathbb{E}S_j$.

Theorem 2 (Repeated-query concentration)**.** *Condition on a fixed example and trained model, and assume independent calls. Each centered pairwise difference $(S_j - S_y) - \mathbb{E}(S_j - S_y)$ is assumed to be $2\sigma^2$-sub-Gaussian. For $\Delta > 0$,*

$$\Pr[\arg\max_j \bar{S}_{Q,j} \neq y \mid h, y] \leq (K-1) \exp\left(-\frac{Q\Delta^2}{4\sigma^2}\right). \tag{4}$$

A sub-Gaussian tail bound for each competing score, followed by a union bound, gives the result. Additional calls reduce sampling error around the stochastic model's mean-score classifier. Training on changing representations can also change that classifier. Query averaging reduces prediction variation, while pilot recovery restores the representation used for learning.

5 Experiments

5.1 Setup

We evaluate MNIST, Fashion-MNIST, and CIFAR-10. CIFAR-10 has $50{,}000$ training and $10{,}000$ test images across ten classes (Krizhevsky & Hinton, 2009). Vanilla FF follows the sequential full-batch

PyTorch reference, using label overlay, two 500-unit layers, and threshold two. Adam trains each layer for 1,000 epochs (Kingma & Ba, 2014). The CIFAR model uses the CwC class-group design (Papachristodoulou et al., 2023). Its four convolutional layers have 20, 80, 240, and 480 channels, alternating dense and ten-group convolutions. A linear head follows the frozen backbone. Batch normalization operates at local block boundaries (Ioffe & Szegedy, 2015). Each layer and the head train for 20 epochs.

The supervised controls share the grouped-convolution design with 20, 80, 240, and 480 channels. End-to-end BP and STE train for 20 epochs. Block-local BP trains each of two blocks for 20 epochs with a detached interface. Primary stochastic comparisons use five paired seeds. Each nonidentity run loads its seed's clean upstream checkpoint and initializes a new downstream component. Pairing holds the upstream representation fixed across channel conditions. Resource counts include complete training from scratch. Figure 2 summarizes the state-persistence and pilot-recovery comparisons across the three learners.

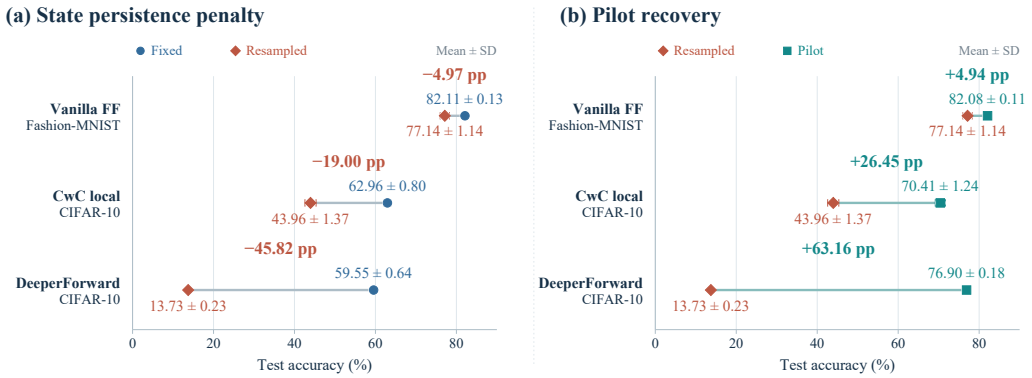

Figure 2: State persistence and pilot recovery across three learners. Fixed and resampled Rademacher states share the same distribution. Points show mean accuracy with sample SD over five paired seeds. The pilot recovers the original representation from a shared channel state. Difference annotations are means of the five within-seed paired changes, in percentage points.

5.2 Protocol validation and deterministic controls

Deterministic channels show why differentiability alone does not determine downstream accuracy. Vanilla FF reaches 92.75% on MNIST, close to the public implementation's reported regime. Positive radial gain gives the same 92.75%, consistent with normalization invariance. Orthogonal mixing reaches 93.22%. Unsigned quantization ranges from 92.00% at one bit to 92.77% at eight bits. These single-seed results connect preserved direction and activation support to effective local learning.

The CIFAR-10 diagnostics distinguish changes in class structure from changes across calls. Probe accuracy is 33.90% for identity, 30.33% for unsigned one-bit quantization, and 31.83% for stochastic sign/flip. Repeated-direction similarities are 1.000, 1.000, and 0.214, respectively. Quantization reduces measured class structure while retaining a fixed direction. Sign/flip preserves similar one-shot probe accuracy but presents changing directions to the downstream learner.

5.3 Stochastic interfaces affect distinct local objectives

The stochastic interface reduces accuracy relative to the clean condition under both local objectives in every paired seed (Table 1). Their mean drops differ by 2.23 points, with the same direction of change across objectives. BP+STE reduces the loss to 5.19 points relative to clean BP. The result links the interface penalty to how training uses the changing representation. Figure 3 connects the representation diagnostics to query averaging. Fixed and resampled channels occupy different regions of the diagnostic plane. Additional queries improve prediction, with smaller gains as the query count increases.

Training with upstream credit changes the response to the stochastic channel. STE averages 70.984%, exceeding CwC local by 11.662 points and block-local BP by 7.782 points. These comparisons share

Table 1: Paired CIFAR-10 accuracy under stochastic sign/flip states (mean ± sample SD; five paired seeds). Bold marks the highest resampled mean and smallest loss.

	Test accuracy (%)		
Training regime	**Clean**	**Resampled ↑**	**Loss (pp) ↓**
CwC local	70.41 ± 0.58	59.32 ± 1.51	11.09
Block-local BP	76.52 ± 1.71	63.20 ± 1.06	13.32
End-to-end BP+STE	76.17 ± 1.40	**70.98** ± 0.93	**5.19**

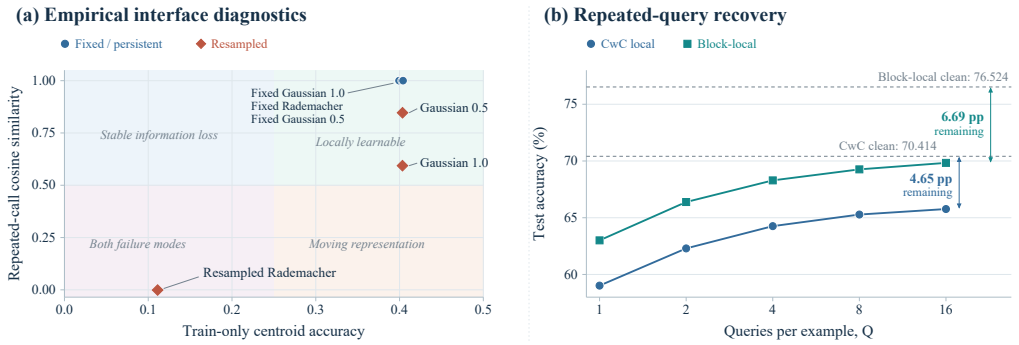

Figure 3: Interface diagnostics and repeated-query recovery on CIFAR-10. (a) Frozen centroid accuracy and repeated-call cosine similarity separate class structure from state variation; coordinates are five-seed means. Pale regions are qualitative guides, not calibrated decision boundaries or sufficient conditions for learning. (b) Logit averaging improves accuracy over five training seeds and five evaluation streams per checkpoint. At $Q = 16$, clean gaps remain 4.65 and 6.69 points for CwC local and block-local BP, respectively, with 12.21 GiB of logical inference traffic.

the architecture family, seeds, data, interface, and training duration. Because the objective and access to upstream credit change together, the comparison reflects both training differences.

5.4 Held-out predictions and pilot recovery

Held-out channels test whether the diagnostics predict downstream ordering before training. We split CIFAR-10 training data into 45,000 fit and 5,000 validation examples. Each seed's first two clean CwC blocks remained frozen. The diagnostic score multiplies validation cosine-centroid accuracy by repeated-call directional stability. We fixed predictions for equal-retention erasure and additive Gaussian noise using these training-data measurements. For Gaussian noise, the prespecified ordering from highest to lowest was fixed ($\sigma = 0.5$), fixed ($\sigma = 1$), resampled ($\sigma = 0.5$), and resampled ($\sigma = 1$). Downstream blocks and heads then trained on all 50,000 images with the same 20/20/20-epoch protocol. Five paired seeds were evaluated once on the test set, linking the frozen predictions to downstream outcomes.

The downstream results follow every frozen directional prediction for erasure and Gaussian noise (Table 2). Mild settings produce small differences between fixed and resampled conditions. Rademacher resampling causes a larger loss of 19.00 points on CIFAR-10, with the same direction in every seed. Each realized Rademacher map is invertible and norm preserving. Fashion-FF loses 4.97 points under the same state comparison. These results identify state persistence as a factor beyond per-state invertibility (Figure 2).

Making the hidden state observable restores the representation and downstream accuracy. The CIFAR diagonal pilot recovers 26.45 points, placing its mean within 0.008 points of identity. For signed permutations, 16 coded pilots raise accuracy from 12.03% to 70.41%. Diagonal-pilot noise with standard deviation 0.10 gives 69.25%, and 10% pilot-state mismatch gives 66.90%. Fashion-FF recovers 4.94 points and matches every clean-seed result. Appendix Figure A5 links the held-out predictions to these state-persistence comparisons.

Table 2: Prespecified interface tests (mean ± sample SD; five seeds). Differences use within-seed left-minus-right values before rounding.

Comparison	Left (%)	Right (%)	Diff. (pp)
Erasure: fixed / resampled, 10%	69.69 ± 0.66	69.27 ± 0.88	0.43
Erasure: fixed / resampled, 30%	69.88 ± 0.83	67.87 ± 1.80	2.02
Erasure: resampled, 10% / 30%	69.27 ± 0.88	67.87 ± 1.80	1.40
Gaussian: fixed / resampled, $\sigma = 0.5$	70.31 ± 1.55	69.63 ± 1.36	0.69
Gaussian: fixed / resampled, $\sigma = 1$	70.21 ± 0.97	66.61 ± 2.78	3.60
Gaussian: resampled, $\sigma = 0.5$ / 1	69.63 ± 1.36	66.61 ± 2.78	3.02
Rademacher: fixed / resampled	62.96 ± 0.80	43.96 ± 1.37	19.00
Rademacher: pilot / resampled	70.41 ± 1.24	43.96 ± 1.37	26.45
Signed permutation: pilot / resampled	70.41 ± 1.24	12.03 ± 0.84	58.37
Fashion-FF: fixed / resampled	82.11 ± 0.13	77.14 ± 1.14	4.97
Fashion-FF: pilot / resampled	82.08 ± 0.11	77.14 ± 1.14	4.94

5.5 Accuracy versus forward-query cost

Query averaging recovers accuracy most rapidly over the first two to four calls (Table 3). All ten stream-averaged checkpoint curves are monotone. At $Q = 16$, mean within-checkpoint SD is 0.150 points for CwC and 0.187 for block-local BP. Every seed retains a clean gap larger than the variation across evaluation streams. Over this measured query range, the pattern is consistent with changes to the trained model or its mean-score margin.

Table 3: Repeated-query accuracy averaged over five training seeds and five evaluation streams per checkpoint. Gaps use the paired clean model. Bold marks the best value in each metric column.

	CwC local		Block-local BP	
Q	**Acc. (%) ↑**	**Gap (pp) ↓**	**Acc. (%) ↑**	**Gap (pp) ↓**
1	59.017	11.397	63.006	13.518
2	62.296	8.118	66.384	10.140
4	64.259	6.155	68.291	8.233
8	65.280	5.134	69.262	7.262
16	**65.766**	**4.648**	**69.834**	**6.690**

5.6 Resource accounting

Training structure determines both the number of active parameters and the volume of interface traffic (Table 4). CwC uses fewer simultaneously trainable parameters. Its sequential layers and frozen-backbone head cross the interface three times. This produces more optimizer updates and three times the logical traffic of block-local BP. At inference, sixteen queries transfer 12.21 GiB across the CIFAR test-set interface, compared with 0.763 GiB for one. Query-based recovery therefore trades additional traffic for improved prediction.

Table 4: Training cost in the current CIFAR-10 implementation. Traffic is logical FP32 tensor volume.

Training regime	Params.	Peak trainable	Updates	Traffic (GiB)
CwC local	586,490	307,210	19,600	228.88
Block-local BP	792,940	585,850	7,840	76.29
End-to-end BP+STE	588,130	588,130	3,920	76.29

Pilot recovery uses extra vectors and computation to expose the channel state (Figure A4). An A6000 microbenchmark measures the 256 × 20,480 FP32 interface. The ideal diagonal pilot takes 0.344

ms/call, compared with 0.261 ms for resampled Rademacher (1.315×). Incremental peak allocation rises from 20.08 to 40.31 MiB. Noisy and mismatch pilots take 0.637 and 0.578 ms/call, respectively. The diagonal pilot adds 0.391% logical vector traffic per batch. Signed-permutation recovery uses a larger pilot code and a coordinate scatter operation. A separate frozen benchmark measures its 16-vector code. Logical traffic increases by 6.25%, and latency reaches 1.261 versus 0.383 ms/call (3.296×). Peak incremental allocation rises from 40.23 to 67.56 MiB. These measurements connect the recovery construction to the cost of its concatenation and scatter operations.

5.7 State persistence and recovery on DeeperForward

We further evaluate state persistence and recovery using the external DeeperForward ResNet (Sun et al., 2025). Our CIFAR-10 reproduction reaches 86.06%, within 0.16 percentage points of the reported 86.22% result. We place the interface after residual block 2, freeze the upstream learner, and train only a downstream readout. Five paired seeds cover identity, fixed and resampled Rademacher states, ideal and noisy pilots, and pilot-state mismatch.

The external results reproduce the loss caused by hidden state variation and the gain from making that state observable. The fixed state exceeds resampling by 45.82 points, while the ideal pilot recovers 63.16 points and matches identity in every seed. Pilot noise and 10% pilot–data state mismatch reduce ideal-pilot accuracy by only 2.42 and 2.20 points, respectively. Figure A2 shows the paired seed-wise changes. Appendix D.2 isolates test-time perturbations, while Appendices F.4–F.6 examine representation alignment, selected recovery cases, and class-wise outcomes.

Table 5: External DeeperForward validation.

Interface condition	Test accuracy (%)
Identity	**76.90** ± 0.18
Fixed Rademacher	59.55 ± 0.64
Resampled Rademacher	13.73 ± 0.23
Ideal pilot	**76.90** ± 0.18
Noisy pilot	74.47 ± 0.06
10% state mismatch	74.70 ± 0.37

6 Discussion

Results indicate that interface design should distinguish per-state information preservation from cross-call consistency because invertibility is insufficient under cross-call hidden-state variation. Pilot recovery applies when shared forward observations identify the state, subject to channel structure and pilot–data alignment.

The state-persistence effect appears across local objectives and representations. Five-seed comparisons cover CIFAR-10 and Fashion-MNIST with vanilla FF, CwC, block-local BP, and a frozen DeeperForward representation. The internal CwC model follows the published class-group design; its seed-1 clean accuracy is 71.00%, versus 78.11% implied by the published error. The external reproduction adds a second backbone to the paired interface test. Newer FF systems offer further architectural settings (Chen et al., 2025; Yao et al., 2026). Matching upstream representations within each comparison keeps the state-persistence question consistent across learners.

Recovery depends on pilot observability. Diagonal states require a nonzero pilot, while signed permutations require distinct noiseless coordinate codes. Information loss and independently changing pilot/data states alter these conditions. Class-structure and repeated-direction diagnostics characterize interface behavior. Their product predicts the tested orderings but is an ordering diagnostic, not a calibrated accuracy estimate. The appendix connects these observations to pilot perturbations, the exploratory training intervention, and measured resource costs.

7 Conclusion

Forward-only learning across black-box interfaces depends on task information and the representation presented to the downstream module. Matched channels show that per-state invertibility and norm preservation do not ensure effective learning when hidden states change. Shared pilots make structured states observable and recover the original representation without upstream credit. On DeeperForward, one pilot restores identity accuracy across five paired seeds. Together with the local-learning comparisons, this result links successful learning to state persistence or observability at the interface.

AI USE STATEMENT

Generative AI tools were used to assist with programming, figure preparation, and language editing. All research questions, methodological choices, experimental analyses, reported results, and conclusions were independently determined and verified by the authors. The authors take full responsibility for the manuscript.

REPRODUCIBILITY STATEMENT

The appendix specifies data splits, architectures, optimization budgets, interface semantics, pairing rules, and theorem assumptions. Supplementary code generates aggregate tables and figures from per-seed JSON records. Paired summaries use every requested seed.

REFERENCES

Ege Aktemur, Ege Zorlutuna, Kaan Bilgili, Tacettin Emre Bok, Berrin Yanikoglu, and Suha Orhun Mutluergil. Going forward-forward in distributed deep learning. *arXiv preprint arXiv:2404.08573*, 2024. URL https://arxiv.org/abs/2404.08573.

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to ImageNet. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 583–593. PMLR, 2019. URL https://proceedings.mlr.press/v97/belilovsky19a.html.

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Decoupled greedy learning of CNNs. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 736–745. PMLR, 2020. URL https://proceedings.mlr.press/v119/belilovsky20a.html.

Paul Caillon, Alex Colagrande, Erwan Fagnou, Blaise Delattre, and Alexandre Allauzen. Forward only learning for orthogonal neural networks of any depth. *arXiv preprint arXiv:2512.20668*, 2025. URL https://arxiv.org/abs/2512.20668.

Xing Chen, Dongshu Liu, Jérémie Laydevant, and Julie Grollier. Self-contrastive forward-forward algorithm. *Nature Communications*, 16:5978, 2025. doi: 10.1038/s41467-025-61037-0. URL https://www.nature.com/articles/s41467-025-61037-0.

Federico Cunico, Luigi Capogrosso, Francesco Setti, Damiano Carra, Franco Fummi, and Marco Cristani. I-SPLIT: Deep network interpretability for split computing. In *2022 26th International Conference on Pattern Recognition*, 2022. URL https://arxiv.org/abs/2209.11607.

Ashiv Dhondea, Robert A. Cohen, and Ivan V. Bajić. DFTS2: Deep feature transmission simulation for collaborative intelligence. In *2021 IEEE International Conference on Visual Communications and Image Processing*, pp. 1–5, 2021. URL https://arxiv.org/abs/2112.00794.

Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022. URL https://arxiv.org/abs/2212.13345.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456. PMLR, 2015. URL https://proceedings.mlr.press/v37/ioffe15.html.

Risi Jaiswal, Supriyo Datta, and Joseph G. Makin. Energy-efficient supervised learning with a binary stochastic forward-forward algorithm. *arXiv preprint arXiv:2507.06461*, 2025. URL https://arxiv.org/abs/2507.06461.

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 15546–15566, 2023. URL https://proceedings.mlr.press/v202/kaba23a.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. URL https://arxiv.org/abs/1412.6980.

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

Andrii Krutsylo. Scalable forward-forward algorithm. *arXiv preprint arXiv:2501.03176*, 2025. URL https://arxiv.org/abs/2501.03176.

Daniel Kunin, Aran Nayebi, Javier Sagastuy-Brena, Surya Ganguli, Jonathan Bloom, and Daniel Yamins. Two routes to scalable credit assignment without weight symmetry. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5511–5521. PMLR, 2020. URL https://proceedings.mlr.press/v119/kunin20a.html.

Guy Lorberbom, Itai Gat, Yossi Adi, Alex Schwing, and Tamir Hazan. Layer collaboration in the forward-forward algorithm. *arXiv preprint arXiv:2305.12393*, 2023. URL https://arxiv.org/abs/2305.12393.

Yoshitomo Matsubara, Marco Levorato, and Francesco Restuccia. Split computing and early exiting for deep learning applications: Survey and research challenges. *ACM Computing Surveys*, 2022. URL https://arxiv.org/abs/2103.04505.

Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/d490d7b4576290fa60eb31b5fc917ad1-Abstract.html.

Arild Nøkland and Lars Hiller Eidnes. Training neural networks with local error signals. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4839–4850. PMLR, 2019. URL https://proceedings.mlr.press/v97/nokland19a.html.

Alexander Ororbia and Ankur Mali. The predictive forward-forward algorithm. *arXiv preprint arXiv:2301.01452*, 2023. URL https://arxiv.org/abs/2301.01452.

Andreas Papachristodoulou, Christos Kyrkou, Stelios Timotheou, and Theocharis Theocharides. Convolutional channel-wise competitive learning for the forward-forward algorithm. *arXiv preprint arXiv:2312.12668*, 2023. doi: 10.48550/arXiv.2312.12668. URL https://arxiv.org/abs/2312.12668.

Jiawei Shao and Jun Zhang. BottleNet++: An end-to-end approach for feature compression in device-edge co-inference systems. In *2020 IEEE International Conference on Communications Workshops*, 2020. doi: 10.1109/ICCWorkshops49005.2020.9145068. URL https://arxiv.org/abs/1910.14315.

Jiawei Shao, Yuyi Mao, and Jun Zhang. Learning task-oriented communication for edge inference: An information bottleneck approach. *IEEE Journal on Selected Areas in Communications*, 40: 197–211, 2022. doi: 10.1109/JSAC.2021.3126087. URL https://arxiv.org/abs/2102.04170.

Mehran Soltani, Vahid Pourahmadi, and Hamid Sheikhzadeh. Pilot pattern design for deep learning-based channel estimation in OFDM systems. *arXiv preprint arXiv:2003.08980*, 2020. URL https://arxiv.org/abs/2003.08980.

Ravi Srinivasan, Francesca Mignacco, Martino Sorbaro, Maria Refinetti, Avi Cooper, Gabriel Kreiman, and Giorgia Dellaferrera. Forward learning with top-down feedback: Empirical and analytical characterization. *arXiv preprint arXiv:2302.05440*, 2023. URL https://arxiv.org/abs/2302.05440.

Liang Sun, Yang Zhang, Weizhao He, Jiajun Wen, Linlin Shen, and Weicheng Xie. Deeperforward: Enhanced forward-forward training for deeper and better performance. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kOYnXVQCtA.

Erik B. Terres-Escudero, Javier Del Ser, and Pablo Garcia Bringas. A contrastive symmetric forward-forward algorithm for continual learning tasks. *arXiv preprint arXiv:2409.07387*, 2024. URL https://arxiv.org/abs/2409.07387.

Mauricio Ortiz Torres, Markus Lange, and Arne P. Raulf. On advancements of the forward-forward algorithm. *arXiv preprint arXiv:2504.21662*, 2025. URL https://arxiv.org/abs/2504.21662.

Yulin Wang, Zanlin Ni, Shiji Song, Le Yang, and Gao Huang. Revisiting locally supervised learning: An alternative to end-to-end training. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=fAbkE6ant2.

Yujie Wu, Siyuan Xu, Jibin Wu, Lei Deng, Mingkun Xu, Qinghao Wen, and Guoqi Li. Distance-forward learning: Enhancing the forward-forward algorithm towards high-performance on-chip learning. *arXiv preprint arXiv:2408.14925*, 2024. URL https://arxiv.org/abs/2408.14925.

Songjie Xie, Shuai Ma, Ming Ding, Yuanming Shi, Mingjian Tang, and Youlong Wu. Robust information bottleneck for task-oriented communication with digital modulation. *IEEE Journal on Selected Areas in Communications*, 41(8):2577–2591, 2023. doi: 10.1109/JSAC.2023.3288252. URL https://arxiv.org/abs/2209.10382.

Jie-En Yao, Hong-En Chen, and C.-C. Jay Kuo. HCL-FF: Hierarchical and contrastive learning for forward-forward algorithm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Yao_HCL-FF_Hierarchical_and_Contrastive_Learning_for_Forward-Forward_Algorithm_CVPR_2026_paper.html.

Gongpei Zhao, Tao Wang, Yidong Li, Yi Jin, Congyan Lang, and Haibin Ling. The cascaded forward algorithm for neural network training. *arXiv preprint arXiv:2303.09728*, 2023. URL https://arxiv.org/abs/2303.09728.

Zhichao Zhu, Yang Qi, Hengyuan Ma, Wenlian Lu, and Jianfeng Feng. Stochastic forward-forward learning through representational dimensionality compression. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/00358de35a101a372ea0412bed913c86-Abstract-Conference.html.

A Reproducibility Records

Per-seed JSON records provide the inputs for all primary aggregates. Paired summaries include every requested seed for CwC, block-local, repeated-query, and resource studies. Experiments use PyTorch 2.11.0 and two NVIDIA RTX A6000 GPUs. CIFAR-10 files come from the public fast.ai ImageFolder mirror with the standard ten-class 50k/10k split.

Pairing and frozen predictions. Each nonidentity CIFAR condition loads its seed's first two clean blocks and initializes new downstream blocks and a head. Held-out diagnostics use training-directory split seed 7301, with 45,000 centroid-fit and 5,000 validation examples. The protocol and JSON records were fixed before downstream training. Erasure and Gaussian studies each use five seeds across four conditions. Rademacher comparisons use five seeds across two state-persistence conditions. Query evaluation crosses five training seeds with RNG seeds 9001 through 9005 and validates all 50 files before aggregation.

The Fashion-MNIST comparison tests the same state-persistence mechanism in vanilla FF. Its protocol was fixed before training and uses the two-layer, 500-unit reference with five paired seeds. Fixed and resampled states follow the CIFAR-10 call semantics. The summary includes all ten requested records.

Exact state-matched control. Matched Rademacher states preserve the same per-state properties under both conditions. Each state is diagonal with independent entries in $\{-1, +1\}$. The fixed condition retains one seed-17 state throughout the run. Resampling draws a state per interface call and shares it across that call's examples. Every realized map preserves each $|z_i|$ and the norm $\|z\|_2$, and is invertible. The comparison isolates state persistence. Differences between clean and fixed conditions reflect accessibility within the downstream convolutional hypothesis class.

Pilot intervention. Pilots and data share one state per logical call, making recovery possible from returned outputs. Ideal diagonal and signed-permutation studies evaluated seed 1 before seeds 2 through 5 under a fixed protocol. Diagonal studies transmit $B + 1$ vectors, and signed-permutation studies transmit $B + 16$, with $d = 20{,}480$ for the latter. The summarizers check records, protocol fields, and upstream pairing before aggregation.

Executable audits. The audit wrapper checks interface invariants, held-out and query protocols, regenerated summaries, and manuscript numbers against JSON. It also scans source and PDF metadata for identities and absolute paths. Hardware records retain 30 timing blocks per condition. Earlier simultaneous-training, once-sampled stochastic, and support-mismatched quantization runs remain outside the scientific aggregates.

B Additional Proof Details

An error occurs only if some competing class has $\bar{S}_{Q,j} - \bar{S}_{Q,y} \geq 0$. This difference has expectation at most $-\Delta$. Its assumed $2\sigma^2$-sub-Gaussian variation gives tail probability at most $\exp[-Q\Delta^2/(4\sigma^2)]$. Summing over $K - 1$ competitors yields the theorem. A target conditional error δ is reached when

$$Q \geq \frac{4\sigma^2}{\Delta^2} \log \frac{K-1}{\delta}. \tag{5}$$

B.1 Positive-radial equivalence with finite stabilization

Let $u = T_1(h)$, $r = \|u\|_2$, and $T_2(h) = cu$ for $c > 0$. Then

$$N_\varepsilon(cu) - N_\varepsilon(u) = u\left(\frac{c}{cr+\varepsilon} - \frac{1}{r+\varepsilon}\right) = u\frac{\varepsilon(c-1)}{(cr+\varepsilon)(r+\varepsilon)}.$$

Taking norms gives the expression in Section 4. At $\varepsilon = 0$, the equality requires $r > 0$. With the convention $N_0(0) = 0$, zero outputs also agree when $T_2(h) = c(h)T_1(h)$. Exact trajectory coupling requires downstream access through the normalized value and shared algorithmic randomness. These assumptions specify the normalization control.

B.2 BAYES-RISK FLOOR AFTER AN INTERFACE

For a classifier $a : \mathcal{Z} \to \mathcal{Y}$, the conditional error at $Z = z$ is $1 - p(a(z) \mid z)$. Choosing a maximizer of $p(y \mid z)$ gives minimum error $1 - \max_y p(y \mid z)$. Averaging over Z yields $R^*(Z)$. A fixed invertible transformation preserves Bayes risk: $R^*(Z) = R^*(H)$. A restricted downstream hypothesis class may still fail to represent or learn the inverse. This separates information preservation from model accessibility. Resampled hidden states instead define a stochastic observation channel, whose marginal output can lose label information.

B.3 EXACT RECOVERY FROM A DIAGONAL PILOT

Let every coordinate of p be nonzero. The returned pilot $\tilde{p} = D_s p$ identifies diagonal entries $\tilde{p}_i/p_i$. For returned data $\tilde{z} = D_s z$, recovery gives $z_i = (p_i/\tilde{p}_i)\tilde{z}_i$. Rademacher states are their own inverses, so $p = \mathbf{1}$ gives elementwise sign correction. Exact recovery requires a diagonal invertible channel, nonzero pilot coordinates, and shared pilot/data state. All operations use forward outputs.

B.4 IDENTIFICATION OF A SIGNED PERMUTATION

Assign coordinate $j \in \{0, \dots, d-1\}$ a distinct code $c(j) \in \{0,1\}^m$, with $m = \lceil \log_2 d \rceil$. Permutation leaves an all-ones pilot unchanged, so its output reveals the signs. Sign-corrected coded-pilot outputs then reveal each output position's source coordinate. Scattering corrected data to those coordinates recovers the input. The construction uses $m+1$ pilots with noiseless code observations. Nearest-code decoding suggests an error-correcting extension for noisy codes; the current experiments use exact codes.

For Algorithm 1, use zero-based indices and pilot rows $P_{0,i} = 1$ and $P_{\ell+1,i} = \mathrm{bit}_{m-1-\ell}(i)$ for $\ell = 0, \dots, m-1$, with the most significant bit first. Here bit_0 is the least significant bit. If output coordinate j comes from input coordinate b_j, the returned pilot batch U gives

$$\widehat{s}_j = U_{0,j}, \qquad C_{\ell,j} = \widehat{s}_j U_{\ell+1,j}, \qquad b_j = \mathrm{BinaryDecode}(C_{:,j}) = \sum_{\ell=0}^{m-1} 2^{m-1-\ell} C_{\ell,j}.$$

In the noiseless setting, $C_{\ell,j} \in \{0,1\}$ and b is a permutation of $\{0, \dots, d-1\}$.The receiver scatters $\widehat{s}_j Z_{:,j}$ to column b_j. Equivalently, for data row r and original coordinate i,

$$\widehat{H}_{r,i} = \widehat{s}_{b^{-1}(i)} Z_{r,b^{-1}(i)}.$$

Thus b maps returned positions to source positions; its inverse specifies the column order for gathering the sign-corrected data.

Figure A1 illustrates the same decoding with column-stacked pilots and one-based coordinate labels.

C COMPLETE EXPERIMENTAL PROTOCOL

C.1 INTERFACE SEMANTICS

A logical call specifies which examples share the same channel state. Fixed conditions draw one state for the run. Resampled conditions draw a new state for each call. Algorithm 1 concatenates pilots and data, queries their shared state once, separates the returned rows, and recovers the data. Paired FF positive and negative examples use the same state. Pilot rows never enter downstream batch normalization, classification heads, or local losses. This sequence gives the downstream module the shared observations used in the recovery propositions.

The two invertible channel families hide different coordinate transformations. Rademacher states multiply coordinates by independent signs. Signed permutations also rearrange their positions. Both preserve the Euclidean norm in every realization. Rademacher states preserve coordinatewise magnitudes, while signed permutations preserve their multiset. Fixed and resampled conditions within each family differ in state persistence. Erasure and Gaussian channels add comparisons with information loss and other forms of variation.

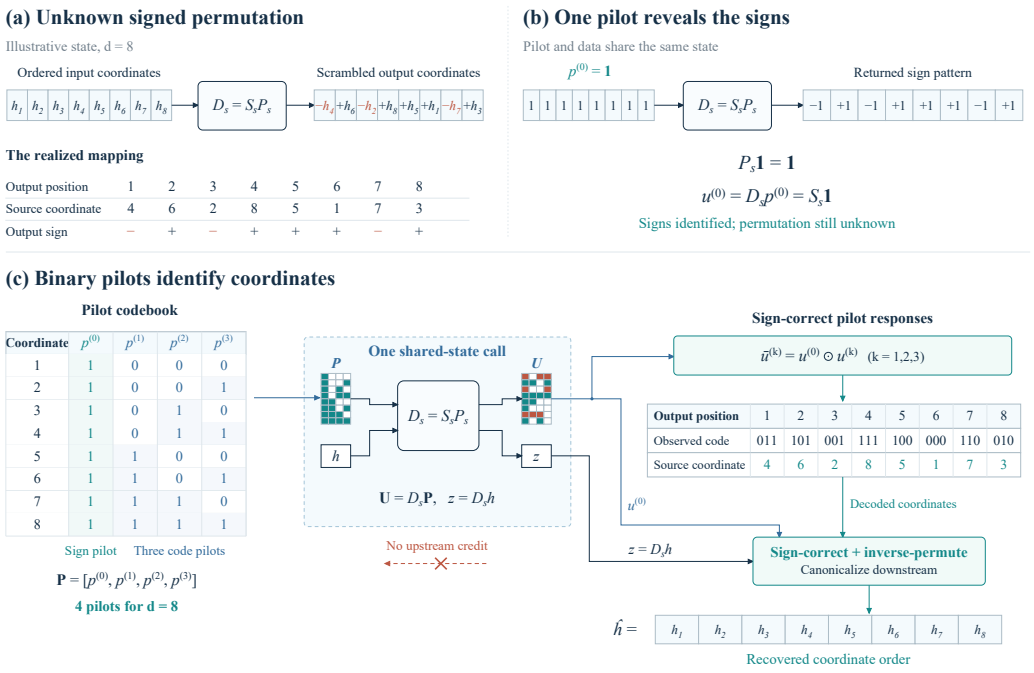

Figure A1: Coded pilots recover a signed permutation. The diagram denotes the full state by $D_s = S_s P_s$, with S_s holding the signs. One all-ones pilot identifies output signs; binary coordinate codes reveal the source position of each output. Bold $\mathbf{P}$ stacks the pilot columns and $\mathbf{U}$ their returned responses; P_s denotes the unknown permutation. Pilots and data share one state in a logical call. With noiseless observations, this construction uses $1 + \lceil \log_2 d \rceil$ pilots: four for the $d = 8$ example and 16 for the experimental $d = 20{,}480$ interface.

C.2 Training, pairing, and evaluation

CIFAR-10 uses the standard 50k/10k train/test split. CwC and block-local comparisons pair each non-identity condition with its seed's clean upstream checkpoint. Training then updates the downstream component. Five-seed summaries report means, sample SDs, paired differences, and seed-level consistency. Query evaluation uses five independent streams per checkpoint and validates all 50 records before aggregation.

The external protocol starts from the DeeperForward clean configuration, which reached 86.06%. Its 0.16-point difference from the reference met the preregistered 1.5-point reproduction threshold. The transformation acts after residual block 2 on a $400 \times 16 \times 16 = 102{,}400$-dimensional boundary representation. The remaining frozen blocks produce the $25{,}600$-dimensional features presented to the readout. A downstream readout trains for 50 epochs with batch size 128 and learning rate 10^{-3}. All conditions share the external checkpoint, whose parameters remain fixed during readout training. The supplementary pilot and representation analyses use documented new evaluation streams; historical checkpoints did not retain their original interface RNG states. Each analysis specifies its evaluation unit and seed set.

C.3 Preregistration and exclusion rules

The supplementary package includes the protocols for held-out channels, Fashion transfer, pilots, signed permutations, and hardware studies. Exploratory seed-1 runs are separated from the remaining seeds. Aggregation requires all requested seeds. Excluded runs include once-sampled stochastic states, support-mismatched quantization, and simultaneous local training.

D Complete External Validation Results

D.1 Paired readout results

Table 6 gives each seed; Table 7 summarizes their paired differences.

Table 6: Per-seed external DeeperForward accuracy. Identity and ideal pilot coincide because pilot canonicalization reconstructs the returned representation. Bold marks the tied highest means.

	Test accuracy (%)					
Seed	**Identity**	**Fixed**	**Resampled**	**Pilot**	**Noisy**	**Mismatch**
1	76.64	60.00	13.79	76.64	74.39	74.79
2	76.99	59.83	13.74	76.99	74.50	74.57
3	77.06	58.44	14.09	77.06	74.44	75.08
4	77.02	59.89	13.49	77.02	74.49	74.91
5	76.77	59.59	13.56	76.77	74.55	74.13
Mean	**76.896**	59.550	13.734	**76.896**	74.474	74.696
SD	0.182	0.638	0.234	0.182	0.061	0.367

Table 7: Five-seed paired differences. Consistency counts seeds with positive differences.

Comparison	**Mean diff. (pp)**	**Consistency**
Fixed − resampled	45.816	5/5
Pilot − resampled	63.162	5/5
Identity − pilot	0.000	0/5 (ties)
Pilot − noisy pilot	2.422	5/5
Pilot − mismatch pilot	2.200	5/5

Every non-tied external comparison has the same direction in all five seeds. Identity and ideal pilot tie because recovery is algebraically exact in the tested implementation. Noise ($\sigma = 0.10$) and state mismatch (10% of coordinates) each use one preregistered severity. Figure A2 places the seed-level changes beside their mean effects.

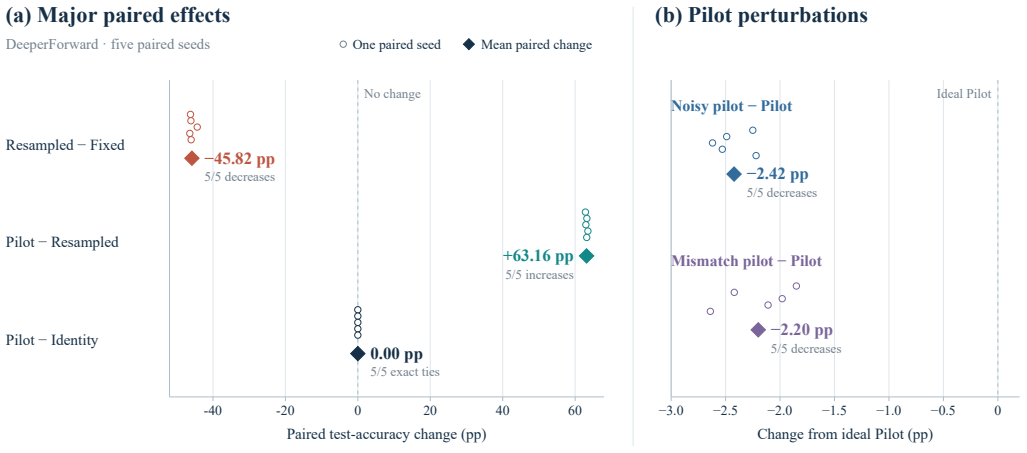

Figure A2: Paired external DeeperForward effects over five seeds. Small points show individual seeds; diamonds show mean paired changes (percentage points). Vertical offsets separate seeds and carry no quantitative meaning. Signs follow the comparison labels.

D.2 TEST-TIME PILOT RELIABILITY

The main noisy and mismatch conditions perturb the interface during both training and testing. Figure A3 instead isolates inference-time sensitivity: all model blocks and the five ideal-pilot-trained readouts remain fixed. Accuracy uses all 10,000 CIFAR-10 test images under the supplementary evaluation streams. This protocol also fixes the interpretation of the no-pilot control, which feeds unrecovered inputs to the same ideal-trained readout rather than using the separately trained Resampled readout.

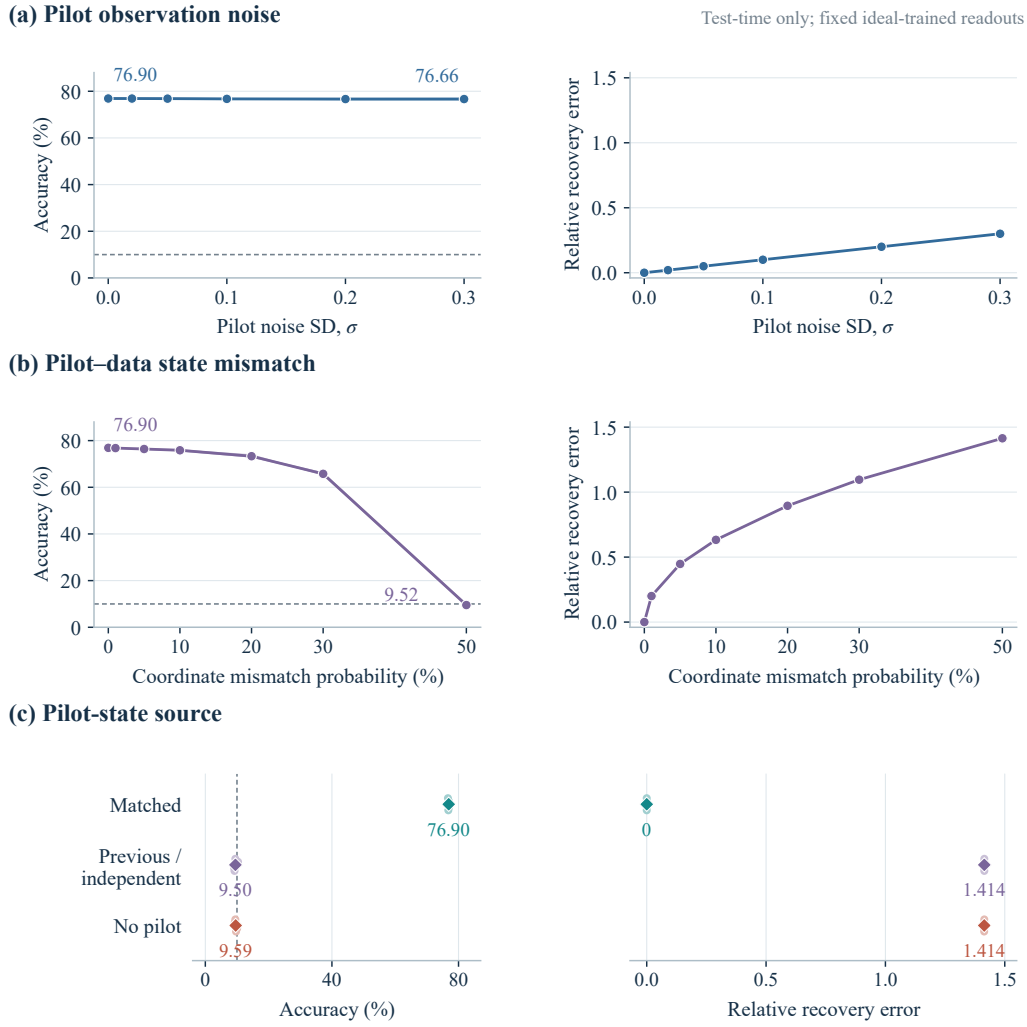

Figure A3: Pilot reliability with fixed ideal-pilot-trained readouts and test-time perturbations. Rows vary (a) Gaussian pilot observation noise, (b) the probability of a pilot-coordinate sign mismatch, and (c) the pilot-state source. Columns show test accuracy and boundary recovery error. Lines or diamonds are means over five readout seeds; pale traces or points retain individual seeds, and bands or bars show sample SD. Dashed accuracy lines mark 10% uniform guessing. Previous and independent controls are merged because their paired measurements coincide.

Noise and coordinate mismatch. Gaussian noise is added to the returned pilot before multiplicative recovery. State mismatch independently flips each pilot coordinate with probability r. The recovery error at the transformed boundary is

$$E_{\mathrm{rec}} = \sqrt{\frac{\sum_i \|\hat{h}_i - h_i\|_2^2}{\sum_i \|h_i\|_2^2 + \epsilon}}. \tag{6}$$

At noise standard deviation $\sigma = 0.30$, mean accuracy is 76.66%, a decrease of 0.24 percentage points from ideal recovery, while relative recovery error reaches 0.300. Coordinate mismatch produces a larger degradation: at $r = 0.30$, accuracy is 65.76%, and at $r = 0.50$ it is 9.52%. The joint accuracy and recovery-error curves distinguish tolerance to small pilot perturbations from failure when pilot signs cease to identify the data state.

Which state does the pilot observe? A matched current-state pilot retains 76.90% accuracy. A previous-call or independent-state pilot gives 9.50%, and removing the pilot gives 9.59%. The previous and independent controls have identical paired measurements under the exported construction, which swaps the same two independent sign draws in an elementwise product. They therefore appear as one row. These controls locate the recovery benefit in the correspondence between pilot and data states.

E Reproducibility, Compute, and Audit Details

The resource measurements link recovery to the operations performed at the interface. Training scripts generate the traffic and update counts in Table 4. Figure A4 reports A6000 latency and temporary allocation for the tested implementation.

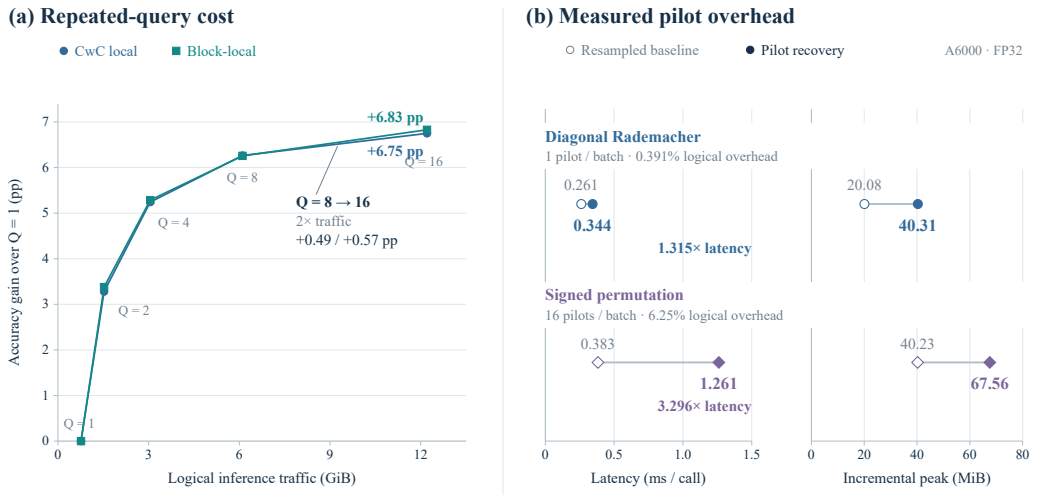

Figure A4: Accuracy and interface cost. (a) Repeated-query accuracy gain relative to $Q = 1$ versus logical FP32 inference traffic over the 10,000-example test set. Doubling from eight to sixteen queries adds 0.49 and 0.57 points for CwC and block-local, respectively. (b) Median latency over 30 timing blocks and incremental peak allocation; open markers denote resampled baselines and filled markers denote pilot recovery. Pilot counts and logical overhead appear beside each row. Hardware measurements use an A6000 at batch size 256 and dimension $20{,}480$, with FP32 tensors and separate baselines for the two channel families; measurements are implementation-specific.

The supplementary wrapper checks invariants, seed sets, pairing, manuscript numbers, and anonymization. A clean-package smoke test reruns representative paths in isolation. Primary claims trace to canonical JSON records. Licensing records are maintained separately.

F Additional Interface Analyses

F.1 Held-out interface predictions

Frozen diagnostics predict the preregistered directional orderings for erasure, Gaussian noise, and Rademacher states. Mild erasure and Gaussian settings yield small differences between fixed and resampled means. Larger perturbations and Rademacher resampling separate the conditions more

strongly. Directional stability alone outperforms the diagnostic product in some comparisons. The scores therefore describe interface mechanisms and ordering, with numerical accuracy measured by downstream training.

Figure A5 places these held-out predictions beside the CIFAR-10 and Fashion-MNIST state-persistence effects.

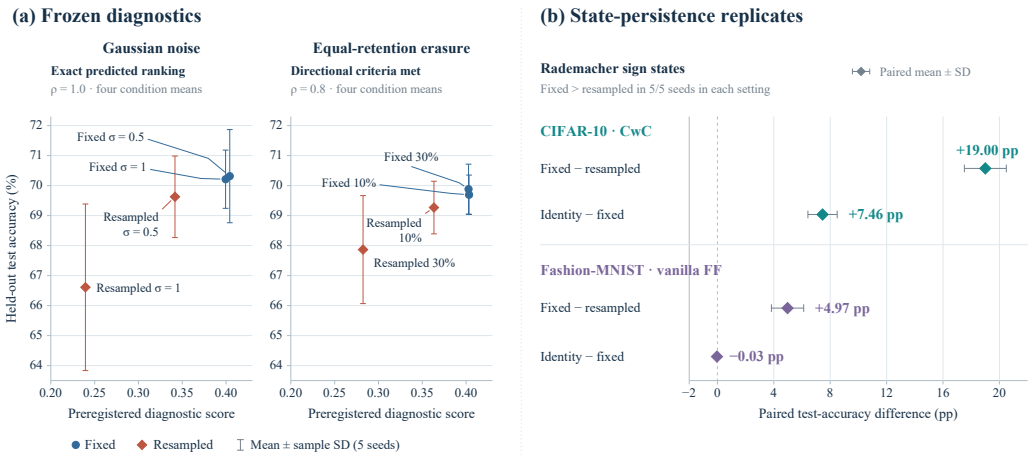

Figure A5: Frozen diagnostics and state-persistence replicates. (a) Preregistered scores and held-out accuracies for Gaussian noise and equal-retention erasure. Vertical bars show accuracy sample SD over five seeds; Spearman ρ is descriptive over four condition means per family. (b) Paired state effects on CIFAR-10 CwC and Fashion-MNIST FF. Horizontal bars show sample SD of the five within-seed paired differences.

F.2 Repeated queries after training with changing states

Repeated inference queries improve prediction after training on a changing interface. Averaging 2, 4, 8, and 16 independent calls increases mean accuracy for both local learners. At sixteen calls, clean gaps remain 4.648 points for CwC and 6.690 for block-local BP. Mean within-checkpoint SD across evaluation streams is 0.150 and 0.187 points, respectively. The gaps exceed stream variation, consistent with changes associated with downstream training on resampled representations.

F.3 Training averaging and state mismatch

Training-time averaging and pilot-state mismatch probe different parts of the recovery process. An exploratory four-state training intervention fell 0.73 points below its frozen seed-1 continuation threshold and stopped. Fixed Rademacher states also fall below identity in some settings. Their signs can make preserved information less accessible to a restricted downstream model. With 10% coordinate mismatch, the CwC pilot remains 22.940 points above resampling and 3.508 points below the ideal pilot. These observations connect task information, model accessibility, and shared-state observations to downstream accuracy.

F.4 Cross-call representation alignment

Figure A6 tests correspondence on the ten lowest CIFAR-10 test indices per class using the 25,600-dimensional DeeperForward readout features. For call-specific matrices X_A, X_B and corresponding Identity features H_{id}, normalized drift is

$$E_{\mathrm{drift}} = \frac{\|X_A - X_B\|_F}{\|H_{\mathrm{id}}\|_F + \epsilon}. \tag{7}$$

The true-match margin subtracts the largest nonmatching gallery cosine from the same-image cosine. Its descriptive CDF pools 100 queries across five interface-state seeds, rather than independent backbone training runs.

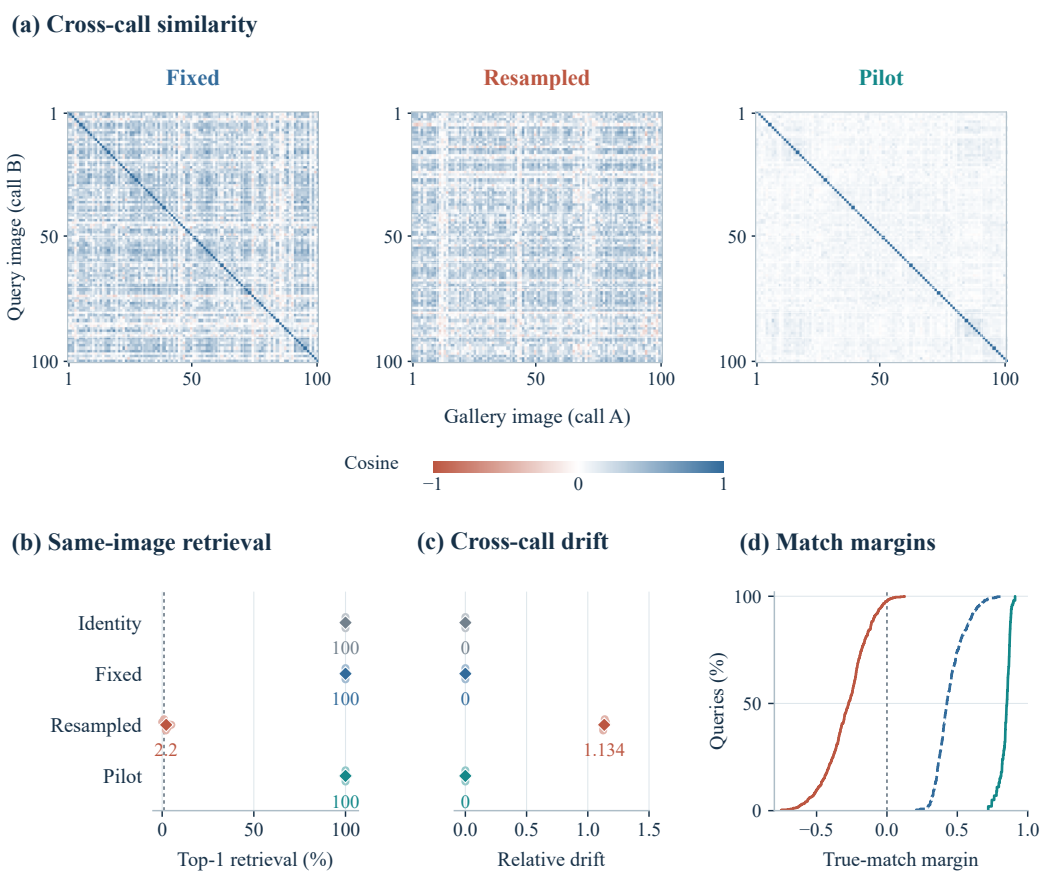

Figure A6: Cross-call alignment on 100 class-balanced CIFAR-10 images. (a) State-seed-1 cosine matrices on a shared scale; separators delimit classes. (b) Same-image retrieval and (c) normalized drift across five paired interface-state seeds: pale points show individual seeds; diamonds and bars show mean and sample SD. Chance retrieval is 1%. (d) Margin CDFs: red denotes Resampled, dashed blue Fixed, and teal both Pilot and Identity.

Resampled states give $2.20 \pm 1.92\%$ same-image retrieval and 1.134 ± 0.009 drift; Fixed, Identity, and Pilot give 100% retrieval and zero drift. Fixed has a different margin distribution from Pilot, so stable coordinates need not equal Identity. Pilot and Identity have coincident exported similarity matrices and margins.

F.5 Selected repeated-call and feature-recovery examples

Figure A7 retains the first four seed-1 examples selected for Resampled failure and Identity/Pilot success. Selection prioritizes class diversity, then ground-truth probability gain, with test-index tie breaking. Each method uses its corresponding trained readout. On 16 fresh prediction calls, Resampled succeeds on 1, 0, 2, and 1 calls for indices 357, 3217, 496, and 3543, respectively; Pilot succeeds on all 16 for each input.

Three separately exported feature calls expose the sign correction at the transformed boundary. All 12 full 102,400-dimensional recovered vectors agree elementwise with the canonical representations. Figure A8 extends the selected-case analysis to the complete test set.

F.6 Class-wise prediction recovery

Figure A8 evaluates the seed-1 readouts on all 1,000 test images per class using the supplementary stream. Corrections change Resampled errors to Pilot successes; regressions reverse this change. Their difference gives the recall gain: 44.6–78.5 percentage points across classes. Overall, 64.88%

are corrected and 2.36% regressed, giving a 62.52-point gain from 14.13% to 76.65% accuracy. These single-seed results complement the main five-seed aggregate; they provide no across-seed class-wise uncertainty estimate.

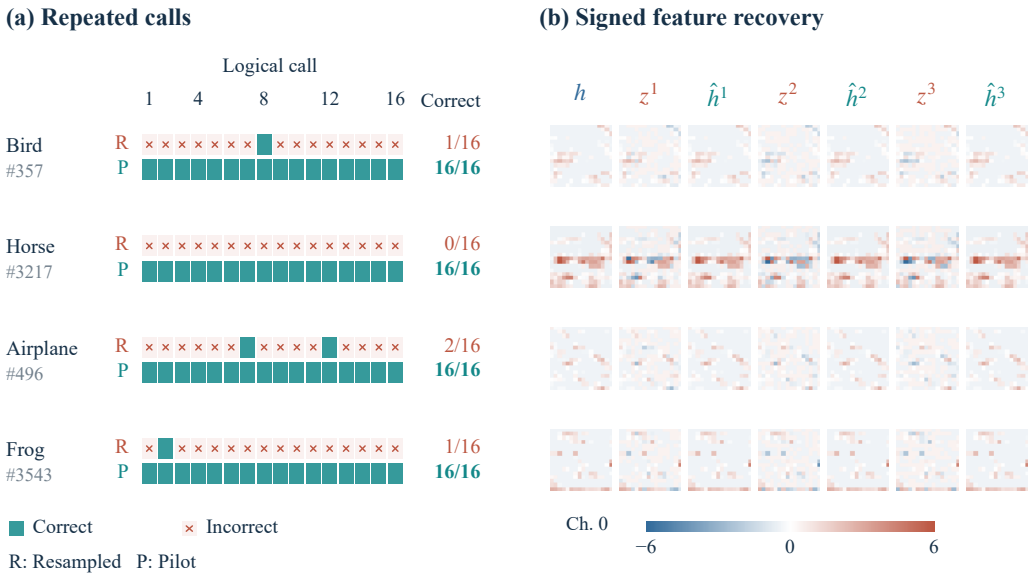

Figure A7: Four selected seed-1 correction cases. (a) Correctness over 16 fresh calls; R/P denote Resampled/Pilot. (b) Channel 0 (16×16): canonical h, returned $z^{(q)}$, and recovered $\hat{h}^{(q)}$ from three separate feature calls, all on $[-6, 6]$. Cases require Resampled failure and Identity/Pilot success.

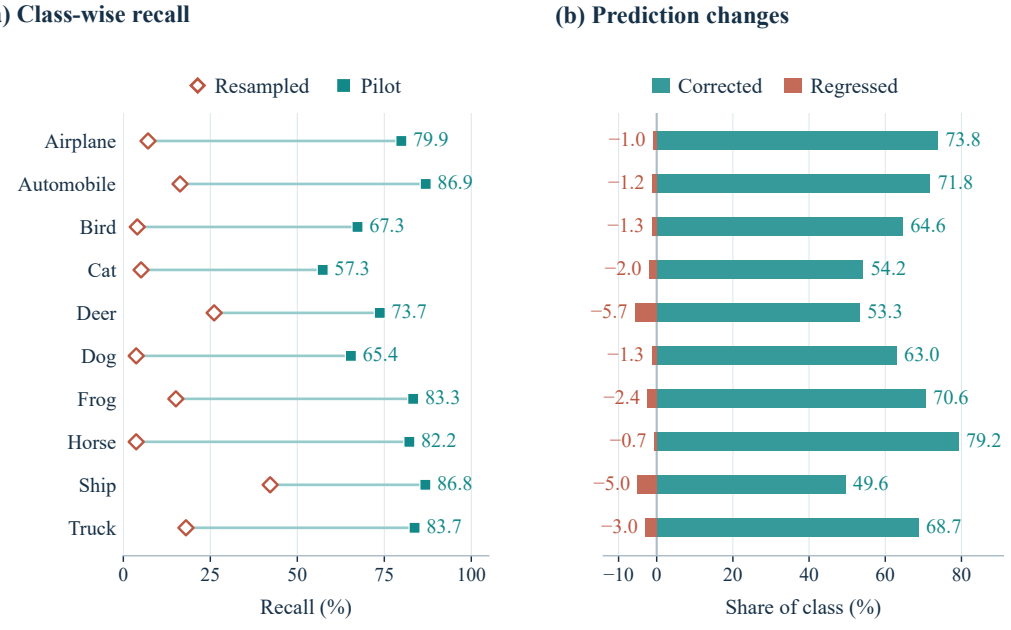

Figure A8: Class-wise recovery on the complete seed-1 CIFAR-10 test set. (a) Recall; labels give Pilot values. (b) Corrected shares extend right and regressed shares left. Leftward bars encode positive regression rates. One readout seed; supplementary evaluation stream.