Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Modeling Delusions Through Hybrid Predictive Coding

Master's Thesis

2026
Mgr. Alica Bednaričová

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Modeling Delusions Through Hybrid Predictive Coding

Master's Thesis

Study Programme:	Cognitive Science
Field of Study:	Computer Science
Department:	Department of Applied Informatics
Supervisor:	doc. RNDr. Martin Takáč, PhD.

Bratislava, 2026
Mgr. Alica Bednaričová

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Mgr. Alica Bednaričová
Študijný program: kognitívna veda (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Modeling Delusions Through Hybrid Predictive Coding
Modelovanie bludov pomocou hybridného prediktívneho kódovania

Anotácia: V modeloch psychózy založených na prediktívnom kódovaní (PC) sa bludy (silné falošné presvedčenia odolné voči protichodným dôkazom) vysvetľujú ako účinok slabých apriórnych predpokladov. To však nedokáže vysvetliť živosť a percepčnú kvalitu bludov a prechod od počiatočných nestabilných presvedčení k neskorším pevným a odolným bludom. Tieto javy by podľa Hardinga a kol. (2024) bolo možné modelovať v hybridom PC (HPC – Tschantz et al., 2023), ktoré pridáva tzv. amortizovanú inferenciu. Cieľom diplomovej práce ju experimentálne overiť túto myšlienku.

Cieľ: 1. Reimplementovať model HPC (Tschantz et al., 2023).
2. Navrhnúť a otestovať scenáre skúmajúce podmienky, za ktorých môžu nastať a pretrvávať bludy.
3. Interpretovať výsledky s cieľom prispieť k hlbšiemu pochopeniu mechanizmov, ktoré sú základom psychotických symptómov.

Literatúra: Harding, J. N., Wolpe, N., Brugger, S. P., Navarro, V., Teufel, C., & Fletcher, P. C. (2024). A new predictive coding model for a more comprehensive account of delusions. The Lancet Psychiatry, 11(4), 295–302.
Tschantz, A., Millidge, B., Seth, A. K., & Buckley, C. L. (2023). Hybrid predictive coding: Inferring, fast and slow. PLOS Computational Biology, 19(8).

Vedúci: doc. RNDr. Martin Takáč, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.

Dátum zadania: 24.10.2025

Dátum schválenia: 24.10.2025

prof. Ing. Igor Farkaš, Dr.
garant študijného programu

.....
študent

.....
vedúci práce

Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Mgr. Alica Bednaričová
Study programme:	Cognitive Science (Single degree study, master II. deg., full time form)
Field of Study:	Computer Science
Type of Thesis:	Diploma Thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Modeling Delusions Through Hybrid Predictive Coding

Annotation: Positive symptoms of psychosis typically include delusions (strong false beliefs resistant to counterevidence) and hallucinations (experienced percepts without external stimuli). Standard predictive coding paradigm that models delusions as the effect of weak priors due to their low estimated precision cannot account for perception-like vividness of delusions and shift from initial unstable beliefs to later fixed and resistant delusions. Harding et al. (2024) suggest that a new modelling paradigm of hybrid predictive coding (HPC) that integrates standard predictive coding with amortized inference (Tschantz et al., 2023) can account for the missing phenomena. To our knowledge, their idea has not been tested yet in an implemented HPC model, and this is the goal of the proposed thesis.

Aim:
1. Reimplement the HPC model of Tschantz et al (2023).
2. Design and test scenarios examining the conditions under which delusions may occur and persist.
3. Interpret the results with the goal of contributing to a deeper understanding of the mechanisms underlying psychotic symptoms.

Literature:
Harding, J. N., Wolpe, N., Brugger, S. P., Navarro, V., Teufel, C., & Fletcher, P. C. (2024). A new predictive coding model for a more comprehensive account of delusions. The Lancet Psychiatry, 11(4), 295–302.
Tschantz, A., Millidge, B., Seth, A. K., & Buckley, C. L. (2023). Hybrid predictive coding: Inferring, fast and slow. PLOS Computational Biology, 19(8).

Supervisor:	doc. RNDr. Martin Takáč, PhD.
Department:	FMFI.KAI - Department of Applied Informatics
Head of department:	doc. RNDr. Tatiana Jajcayová, PhD.
Assigned:	24.10.2025
Approved:	24.10.2025

prof. Ing. Igor Farkaš, Dr.
Guarantor of Study Programme

Student

Supervisor

Acknowledgments: I wish to thank my supervisor doc. RNDr. Martin Takáč, PhD. for his guidance and ideas that shaped this work, and my family for their continuous support throughout my studies.

Abstract

The classic clinical definition of delusions describes them as fixed false beliefs held with strong conviction despite contradictory evidence. This clinical definition, however, captures only their surface. Phenomenologically, delusional beliefs often arrive suddenly and fully formed, are experienced with the immediacy of perception rather than as conclusions reasoned to, and resist correction in ways that ordinary mistaken beliefs do not. One of the computational accounts of delusions frames it within predictive coding, treating it as a disorder of how the brain weights sensory evidence against prior expectations during iterative inference. Because it is built entirely on iterative refinement, it fits poorly with the phenomenological features that needs to be explained. Harding and colleagues address this gap by proposing hybrid predictive coding with a fast amortized inference alongside the slow iterative one as better alternative. They propose that the amortized component encodes a library of context dependent mappings shaped by an individual's experiential history, and that delusions arises when a mapping appropriate to one context is applied where it does not belong and is left uncorrected by iterative inference. This thesis tests this preliminary proposal by adapting an existing hybrid predictive coding implementation to a context augmented classification task in which each image is paired with a context signal, allowing the model to commit both perceptual and context errors and manipulating the training distributions to simulate an individual's experiential history. Three experiments were conducted to examine whether imbalance in training distribution produces a measurable bias, where this bias resides in the architecture, and how the amortized and iterative components interact when they operate together. Across the three experiments, we show that fixedness in the model is not a property of its learned generative weights but a product of biased amortized initialization that iterative inference amplifies rather than corrects.

Keywords: predictive coding, hybrid predictive coding, amortized inference, delusions, computational psychiatry

Abstrakt

Klasická klinická definícia bludov ich opisuje ako pevné falošné presvedčenia, ktoré si človek zachováva s hlbokým presvedčením napriek protichodným dôkazom. Táto klinická definícia však zachytáva len ich povrch. Z fenomenologického hľadiska sa bludné presvedčenia často objavujú náhle a v plnej podobe, prežívajú sa skôr ako bezprostredný vnem než ako logicky odôvodnené závery a bránia sa korekcii spôsobom, akým to bežné mylné presvedčenia nerobia. Jedno z výpočtových vysvetlení bludov ich zasadzuje do rámca prediktívneho kódovania a považuje ich za poruchu spôsobu, akým mozog zvažuje zmyslové dôkazy voči predchádzajúcim očakávaniam počas iteratívneho usudzovania. Keďže je založené výlučne na iteratívnom zdokonaľovaní, zle zapadá do fenoménologických charakteristík, ktoré je potrebné vysvetliť. Harding a kolegovia riešia túto medzeru návrhom hybridného prediktívneho kódovania s rýchlym amortizovaným usudzovaním popri pomalom iteratívnom usudzovaní ako lepšou alternatívou. Navrhujú, aby amortizovaná zložka kódovala knižnicu kontextovo závislých priradení formovaných skúsenostnou históriou jednotlivca a aby bludy vznikali vtedy, keď sa priradenie vhodné pre jeden kontext aplikuje tam, kam nepatrí, a nie je opravené iteratívnou inferenciou. Táto práca overuje tento predbežný návrh prispôsobením existujúcej implementácie hybridného prediktívneho kódovania úlohe klasifikácie rozšírenej o kontext, v ktorej je každý obrázok spárovaný s kontextovým signálom, čo modelu umožňuje robiť chyby vnímania aj kontextové chyby, a manipuláciou trénovacích distribúcií s cieľom simulovať skúsenostnú históriu jednotlivca. Boli vykonané tri experimenty s cieľom preskúmať, či nerovnováha v trénovacej distribúcii vedie k merateľnej zaujatosti, kde sa táto zaujatosť nachádza v architektúre a ako spolu interagujú amortizované a iteratívne komponenty, keď fungujú spoločne. V rámci týchto troch experimentov ukazujeme, že fixnosť v modeli nie je vlastnosťou jeho naučených generatívnych váh, ale výsledkom zaujatého amortizovaného inicializovania, ktoré iteratívna inferencia skôr zosilňuje, ako koriguje.

Kľúčové slová: prediktívne kódovanie, hybridné prediktívne kódovanie, amortizovaná inferencia, bludy, výpočtová psychiatria

Contents

List of Figures

List of Tables

Introduction

Some beliefs arrive suddenly and with complete certainty, reorganizing how the world appears and resisting any evidence that would tell against them. What is outstanding about beliefs of this kind is not only that they are false but also how they are held. Rather than being "reasoned into place", they tend to "come fully formed" and the directness of the experience resembles something "seen" rather than something "concluded" with the persistence that cannot be shaken by contradictory evidence. Clinical definitions describe delusions as fixed false beliefs held with strong conviction despite contradictory evidence (American Psychiatric Association, 2013), but this captures only their surface. More phenomenological descriptions of delusions see them as disturbances in the way one experiences reality and not just false propositions about it (Mishara, 2010; Sass and Byrom, 2015). For instance, they can appear suddenly and sometimes as revelatory insight experiences that present themselves with the immediacy of perception rather as the product of inference (Jensen, 2022; Sips, 2019).

Recently, predictive coding has become the leading computational framework for perception and also a candidate for account of several psychiatric disorders. According to the dominant view regarding this account, delusions result from disorders of precision weighting in the inferential hierarchy (Adams et al., 2013; Fletcher and Frith, 2009; Corlett et al., 2019). While this account has had real success with some aspects of psychosis its difficulty is that it is built entirely on iterative inference and this commitment fits poorly with the features of delusions that needs explaining. Some may argue that a process that slowly accumulates evidence does not naturally produce a belief that appear all at once, nor a one that is experienced with the immediacy of an perceptual experience Harding et al. (2024); Corlett and Fletcher (2021).

Hybrid predictive coding suggested a way of addressing these difficulties. The architecture proposed by Tschantz et al. (2023) preserves the slow iterative inference of standard predictive coding but adds an alternative pathway alongside it, a learned feedforward mapping that produces an initial estimate of what the input means in a single pass without any iteration. Building on this, Harding et al. (2024) propose that this feedforward component encodes a library of context dependent mappings shaped by experience, and that delusion arises when a mapping appropriate to one context is applied where it does not belong. Because priors are weakened in the way the

standard account attributes to the prodromal phase, this initial wrong "guess" is then left uncorrected and reinforced over time. The proposal is, as Harding et al. (2024) themselves describes it, preliminary, and they ask for formal computational modeling of the two possibilities just sketched.

In this thesis we ask whether the computational properties Harding's account requires can in fact be observed in a trained hybrid predictive coding model. For our purposes,, we set an MNIST classification task where each handwritten digit is paired with one-hot encoded context signal and each pair combination is treated as a separate class. This setup allows the model to make two distinct kind of error: perceptual errors, in which the digit is wrongly predicted, and context errors, in which the digit is identified correctly but its context is not. The second mentioned is in our setting the analogue of misapplying a learned mapping to a context in which it does not belong. Additionally, training distibutions stand in as analogue to an individuals experiental history, in which a given digit can be paired more often with one context than the other. The experiments are build on the publicly available HPC implementation of Tschantz et al. (2023) and adapted to this context augmented task. Full details are given in Chapter 3.

Within this setup we ask three research questions presented in fuller detail in Section 2.2.5. The first one is focused on whether imbalanced experience reshapes the model's behavior at test, leaving a visible mark on what the model infers when it is later evaluated on a balanced test set. The second question asks where the bias lives, that is, whether the bias is encoded in the amortized network, in the generative weights, or in the hybrid process that combines them. The last question asks how the two inference modes interact when they operate together.

To further develop the theoretical background and address the research questions mentioned above, the structure of the thesis is organised as follows. Chapter 1 develops the theoretical framework, beginning with the Bayesian view of perception, deriving predictive coding from variational inference, identifying the limitations of the standard formulation, and introducing the hybrid extension of Tschantz et al. (2023). Chapter 2 turns to delusions, reviewing what standard predictive coding accounts explain well and explain poorly, presenting Harding's hybrid account, and stating the research questions in full. Chapter 3 describes the task, the model architecture, the three training configurations, and the inference modes used throughout. Chapters 4, 5, and 6 report the three experiments, examining respectively whether imbalanced experience produces a localisable bias, whether that bias can be reversed by corrective experience, and whether it transfers across stimuli.

Chapter 1

Predictive Coding

1.1 Bayesian inference and the brain

The starting point for the theoretical framework developed in this chapter is the idea that the brain is, in some functional sense, a Bayesian inference machine. The view originates with Helmholtz (Dayan et al., 1995), who proposed that perception involves unconscious inferences about the hidden causes of sensory data, and has been developed in modern computational neuroscience and cognitive science (Knill and Pouget, 2004; Friston, 2005; Clark, 2015; Hohwy, 2013). In this view, the brain does not passively receive what is in the world but it infers it, by combining incoming sensory evidence with prior expectations to arrive at probabilistic beliefs about what is most likely to have caused that evidence. However, how Bayesian inference is implemented in the brain remains an open question. While exact Bayesian inference is computationally intractable in any real-world setting, any biological implementation must be approximate. Predictive coding is one candidate hypothesis for how cortical circuits might implement such an approximation, and it is the framework on which the experiments in this thesis are built. The remainder of this chapter develops predictive coding from its foundations of variational-inference (Section 1.2), identifies the limitations of the standard formulation (Section 1.3), and the hybrid extension of Tschantz et al. (2023) is introduced in last section (Section 1.4), which is later tested in the experiments.

1.2 Predictive coding

1.2.1 Perception as Bayesian inference

The Bayesian framing of perception treats the brain as inferring the posterior distribution over hidden causes given the data:

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})}{p(\mathbf{x})}, \tag{1.1}$$

where $p(\mathbf{x} \mid \mathbf{z})$ is the likelihood (how data are generated by causes), $p(\mathbf{z})$ is the prior (the brain's expectations about which causes are likely), and $p(\mathbf{x}) = \int p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})\, d\mathbf{z}$ is the marginal likelihood of the data. Equation 1.1 from a normative standpoint is appealing but the marginal likelihood $p(\mathbf{x})$ involves an integral over all possible causes, which is generally intractable for any realistic generative model (Fox and Roberts, 2012). Therefore if the brain implements anything like Bayesian inference it must do so approximately.

1.2.2 Variational inference

Variational inference has been studied in both machine learning (Beal, 2003; Wainwright and Jordan, 2008) and in theoretical neuroscience literature (Friston, 2005; Buckley et al., 2017) as one candidate approximation scheme. It postulates an approximate posterior $q_\lambda(\mathbf{z})$ parametrized by λ and treats inference as an optimization problem of choosing λ so that $q_\lambda(\mathbf{z})$ is as close as possible to the true posterior $p(\mathbf{z} \mid \mathbf{x})$, where closeness is measured by Kullback–Leibler divergence. This cannot be done because the true posterior is itself intractable and cannot be minimized directly. Variational inference instead minimizes an upper bound: the variational free energy

$$\mathcal{F}(\lambda, \mathbf{x}) = \mathbb{E}_{q_\lambda(\mathbf{z})}[\ln q_\lambda(\mathbf{z}) - \ln p(\mathbf{z}, \mathbf{x})] \geq D_{\mathrm{KL}}[q_\lambda(\mathbf{z}) \,\|\, p(\mathbf{z} \mid \mathbf{x})]. \tag{1.2}$$

Minimizing $\mathcal{F}$ over λ forces $q_\lambda(\mathbf{z})$ to approach the true posterior. Minimizing it with respect to the parameters of the generative model $p(\mathbf{z}, \mathbf{x})$ (denoted θ) maximizes a lower bound on the marginal likelihood and thus implements learning (Beal, 2003).

1.2.3 Predictive coding as iterative variational inference

Predictive coding can be derived as a specific implementation of variational inference under Gaussian assumptions (Friston, 2005; Buckley et al., 2017). For instance, one may assume that both the approximate posterior $q_\lambda(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mu, \sigma^2)$ and the factors of the generative model $p(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}(\mathbf{x}; f_\theta(\mathbf{z}), \sigma_l^2)$ and $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \bar{\mu}, \sigma_p^2)$ are Gaussian, where $f_\theta(\cdot)$ is a non-linear function parameterized by θ and $\bar{\mu}$ is the mean of the prior. Under these assumptions, the variational free energy reduces to a precision-weighted sum of squared prediction errors (Tschantz et al., 2023, Eq. 7):

$$\mathcal{F}(\mu, \mathbf{x}) = \frac{1}{2\sigma_l}\boldsymbol{\varepsilon}_l^2 + \frac{1}{2\sigma_p}\boldsymbol{\varepsilon}_p^2 + \frac{1}{2}\ln(\sigma_l \sigma_p), \tag{1.3}$$

where $\boldsymbol{\varepsilon}_l = \mathbf{x} - f_\theta(\mu)$ is the prediction error at the data level and $\boldsymbol{\varepsilon}_p = \mu - \bar{\mu}$ is the prediction error against the prior. Inference then proceeds by gradient descent on μ:

$$\dot{\mu} = -\kappa \frac{\partial \mathcal{F}}{\partial \mu} = -\kappa \left(\boldsymbol{\varepsilon}_p - \frac{\partial f_\theta(\mu)^\top}{\partial \mu} \boldsymbol{\varepsilon}_l \right), \tag{1.4}$$

with step size κ (Tschantz et al., 2023, Eq. 8). The same scheme applied to θ rather than μ implements learning of the generative model:

$$\dot{\theta} = -\kappa \frac{\partial \mathcal{F}}{\partial \theta} = -\kappa\, \boldsymbol{\varepsilon}_l f_\theta(\mu)^\top, \tag{1.5}$$

where μ is held at its converged value μ^* from iterative inference (Tschantz et al., 2023, Eq. 9).

1.2.4 Hierarchical predictive coding as a process theory

The single-level scheme above generalizes to hierarchies of L layers, where each layer's representation μ_i serves as an empirical prior for the layer below (Friston, 2005, 2008). The joint generative model factorizes as

$$p(\mathbf{z}, \mathbf{x}) = p(\mathbf{x} \mid \mathbf{z}_1)\, p(\mathbf{z}_1 \mid \mathbf{z}_2) \cdots p(\mathbf{z}_{L-1} \mid \mathbf{z}_L)\, p(\mathbf{z}_L), \tag{1.6}$$

with each conditional taking the Gaussian form of Section 1.2.3. The prediction errors at each layer are then propagated up to inform the update of higher-level representations. In turn, predictions at each layer propagate down to constrain representations at lower layers.

In this way, as a process theory of cortical computation, predictive coding posits two distinct populations of neurons within each level of the hierarchy: prediction units, which encode the top-down predictions, and prediction error units, which encode the mismatch between the current activity and the prediction (Rao and Ballard, 1999; Bastos et al., 2012). Top-down connections convey predictions while bottom-up connections convey prediction errors. The iterative refinement is, therefore, recurrent activity progressively minimizing prediction errors throughout the hierarchy.

1.3 The problem standard predictive coding has

1.3.1 The temporal cost of iterative inference

Given some input, the model initializes its representations at some value (typically random) and then updates them through gradient descent on the variational free energy, following the update rule of Equation 1.4. This update is done repeatedly until the representations converge on values that approximately minimize prediction error. So to extract the meaningful representations from sensory data, predictive coding must iteratively minimize prediction errors over multiple sequential steps (Bastos et al., 2012; Millidge et al., 2021). In neural terms this implies that perception requires multiple cycles of recurrent activity (Ahissar and Hochstein, 2004; Friston and Kiebel, 2009; van Bergen and Kriegeskorte, 2020). A single feedforward pass through this hierarchy

is insufficient and the brain must wait for the iterative process to settle before any meaningful representation is available. This is at odds with an empirical evidence where many aspects of human visual perception can occur remarkably rapidly, often within 150ms of stimulus onset (Carlson et al., 2013; Keysers et al., 2001; Thorpe et al., 1996; Thunell and Thorpe, 2019).

This evidence does not refute predictive coding, but it does constrain the interpretation. If iterative inference is the only route from data to beliefs, then either much of fast perception lies outside the scope of the framework, or the framework must be extended to accommodate inference processes that are not iterative. Tschantz et al. (2023) argue for the latter and propose HPC (Hybrid predictive coding) as the resulting extension.

1.3.2 Memoryless inference

A second limitation of standard predictive coding follows directly from its iterative structure. Because inference is performed afresh from each input, with the representations μ initialized at random or at some fixed default, the model retains no information about the outcomes of previous inferences (Gershman and Goodman, 2014). Each new input is treated as if it were the first input the model had ever seen, with the entire iterative process running from scratch.

This has both positive and negative aspects: Tschantz et al. (2023) note that it can be useful in so far as it ignores any biases that may have been introduced by previous data points but it is also computationally wasteful: if the same input is encountered repeatedly, predictive coding will rediscover the same representation each time, paying the full iterative cost on every encounter.

One might attempt to address this limitation within standard predictive coding by arguing for changes in the generative parameters θ: as the model learns from experience, its priors are updated, and this in turn shapes future inference. However, learning generative parameters is slow and depends on the average structure in the dataset. It cannot easily account for rapid, stimulus-specific effects of recent experience.

In contrast, amortized inference provides a fast mechanism by learning a function from data to beliefs and applying it directly to new inputs without any need for iteration (Kingma and Welling, 2013).

1.3.3 Generative without discriminative

A third limitation arises from the type of inference predictive coding implements. Predictive coding models the world in a top-down, generative manner. It learns to predict sensory data from latent causes, $p(\mathbf{x} \mid \mathbf{z})$, rather than to predict latent causes from sensory data, $p(\mathbf{z} \mid \mathbf{x})$ (Bishop, 2006).

Generative models tend to be more sample-efficient than discriminative models, they generalize more robustly to novel inputs (Rezende et al., 2014) and can be used for downstream tasks beyond classification (Kingma and Welling, 2013). On the other hand, discriminative models often outperform generative ones at a given sample size (LeCun et al., 2015). Discriminative methods are only sensitive to the features relevant for discrimination, while generative methods model the entire data distribution itself. Standard predictive coding misses out on the potential performance gains offered by discriminative methods.

1.3.4 A common structure

These three limitations described above are not independent but follow from the same underlying feature of standard predictive coding. Which is that iterative inference is the only mechanism by which the model converts data into beliefs.

The speed limitation arises because iteration is intrinsically slow. Memorylessness limitation arises because iteration starts each inference from scratch, with no learned shortcut from data to typical outcomes. The lack of a discriminative pathway follows from the fact that iteration is a way of inverting a generative model, not a way of mapping data directly to latent causes. Each limitation reflects a different consequence of the same structural commitment.

Rather than treating the limitations as separate problems to be addressed by separate modifications, one might add a single new component: a feedforward, learned, data-to-beliefs mapping, that addresses all three at once. Such a component would be fast (no iteration required), would carry information across inferences (the mapping is shared across the dataset), and would be discriminative (it predicts latent causes from data rather than the reverse). Tschantz et al. (2023) preserve what predictive coding does well while addressing what it does poorly. The next section describes the resulting architecture.

1.4 Hybrid predictive coding

1.4.1 Architecture

The HPC model proposed by Tschantz et al. (2023) preserves the hierarchical structure of standard predictive coding. A network of L layers represents the world at progressively more abstract levels: the lowest layer represents sensory data $\mathbf{x}$, while higher layers represent latent causes of that data. Each layer i contains a population of representation units, whose activity is denoted μ_i and a population of error units $\boldsymbol{\varepsilon}_i$ that signal mismatches between predictions and current activity at that layer.

The difference from standard predictive coding is that HPC introduces a second set of connections running in the opposite direction. Where standard predictive coding has top-down predictions only, HPC adds bottom-up connections that predict the activity of layer $i + 1$ from the activity of layer i, through a separate set of amortisation parameters ϕ. Top-down predictions take the form

$$\mu_{i-1} = f_{\theta_i}(\mu_i), \tag{1.7}$$

with the lowest layer predicting the sensory data, $\mathbf{x} = f_{\theta_0}(\mu_0)$. Bottom-up predictions take the form

$$\mu_i = f_{\phi_i}(\mu_{i-1}), \tag{1.8}$$

with the lowest layer operating directly on the input, $\mu_0 = f_{\phi_0}(\mathbf{x})$. The functions $f_\theta(\cdot)$ and $f_\phi(\cdot)$ are non-linear mappings parameterized by θ and ϕ respectively; in the implementations used both by Tschantz et al. (2023) and in subsequent chapters, these are realized as fully-connected neural networks with tanh activations.

The two sets of connections do not duplicate each other but rather they serve a different functions. Top-down connections implement a generative model where given a high-level representation, they predict what the lower-level activity should be. These are the same standard predictive coding connections that support the same kind of iterative inference. On the other hand, bottom-up connections, given the sensory input, they produce a direct estimate of what each layer's representation should be. Tschantz et al. (2023) call this an amortized inference because the mapping is computed once during learning and applied uniformly across the dataset (in contrast to iterative inference, where the optimization is repeated afresh for each input). Both sets of connections target the same representation units μ, what makes the model genuinely a single model rather than two networks running in parallel (the two pathways converge on a shared set of latent variables, and inference involves reconciling their predictions).

1.4.2 Two types of inference

Inference in HPC proceeds in two phases:

In the amortisation phase, sensory data $\mathbf{x}$ is propagated up the hierarchy through the bottom-up connections in a single feedforward pass. Each layer's representation is initialized to whatever the amortisation network predicts based on the layer below: $\mu_0 = f_{\phi_0}(\mathbf{x})$, $\mu_1 = f_{\phi_1}(\mu_0)$, and so on up the hierarchy. After this pass, every layer has an initial estimate of its representation. This entire process is a single forward sweep where no iterative computation occured yet.

In the iterative phase, these initial estimates are refined by the same prediction error minimization process that defines standard predictive coding. At each layer, the

prediction error is the mismatch between that layer's current activity and the top-down prediction generated from the layer above:

$$\boldsymbol{\varepsilon}_i = \mu_{i-1} - f_{\theta_i}(\mu_i). \tag{1.9}$$

Representations are then updated by gradient descent on the variational free energy $\mathcal{F}$, which under the Gaussian assumptions used in predictive coding can be written as a precision-weighted sum of squared prediction errors (Tschantz et al., 2023, Eq. 7). The update for each μ_i takes the form introduced in Equation 1.4, where in the hierarchical setting $\boldsymbol{\varepsilon}_p$ is the prediction error from the layer above, $\boldsymbol{\varepsilon}_l$ is the prediction error from the layer below. This update is applied iteratively for N steps; Tschantz et al. (2023) use $N = 100$ in their MNIST experiments, and the same value is used throughout subsequent chapters of this thesis unless stated otherwise. After N iterations, inference is taken to be complete and the representations at the top of the hierarchy are read out as the model's belief about the input.

Functionally, the amortisation phase provides a fast, learned guess at what the input means. The iterative phase refines that guess by checking it against the model's generative predictions and adjusting. If the amortisation network has learned good mapping, the initial guess will already be close to the final answer and few iterations will be needed to confirm it. On the other hand, if the amortisation guess is poor, more iterations will be needed to correct it.

1.4.3 Learning

HPC is trained by minimizing prediction errors at all layers, with both sets of weights (generative parameters θ and amortisation parameters ϕ updated after each inference cycle.

The generative parameters θ are updated in the same way as in standard predictive coding. After iterative inference has converged on a set of representations μ^*, each layer's top-down prediction is compared with the actual activity at the layer below, and θ is adjusted to reduce the prediction error:

$$\dot{\theta}_i = -\alpha \frac{\partial \mathcal{F}}{\partial \theta_i} = -\alpha \left(\boldsymbol{\varepsilon}_l f_{\theta_i}(\mu_i)^\top \right), \tag{1.10}$$

where α is a learning rate (Tschantz et al., 2023, Eq. 9).

After iterative inference has produced its converged representations μ^*, the bottom-up predictions made during the amortisation phase are compared with these converged values:

$$\boldsymbol{\varepsilon}_i^{\phi} = \mu_{i+1}^* - f_{\phi_i}(\mu_i), \tag{1.11}$$

This tells us how far the the initial guess of the amortisation network was from what iterative inference eventually settled on. Amortisation parameters are then updated to

reduce this error:

$$\dot{\phi}_i = -\alpha \left(\boldsymbol{\varepsilon}_i^{\phi} f_{\phi_i}(\mu_i)^{\top} \right). \tag{1.12}$$

Put simply, the amortization network is learning to predict what iterative inference will produce. The outputs of the iterative procedure, therefore, serve as the targets for the amortized learning. Tschantz et al. (2023) describe this as "learning to infer through inference" where the iterative process produces the targets that the amortisation network learns to reproduce.

Chapter 2

Modeling of Delusions

2.1 Predictive coding and delusions

The previous sections developed predictive coding and its hybrid extension as general computational framework for perceptual inference. This section moves toward a more specific question: how predictive coding accounts can be used to understand psychotic phenomena, and in particular delusions? Our aim is not to provide a comprehensive review but to identify what standard predictive coding accounts of delusions have explained well, what they have struggled to explain, and where the gaps lie. The gaps identified here will motivate the development of a hybrid predictive coding account of delusions in the next section.

Delusions are conventionally defined as fixed false beliefs that are held with strong conviction despite contradictory evidence (American Psychiatric Association, 2013). This definition captures only the surface of the phenomenon. Phenomenological descriptions of delusions (Mishara, 2010; Sass and Byrom, 2015) emphasize that delusions are not simply incorrect beliefs but transformations how reality is experienced. They often emerge from a prodromal phase, the so-called Wahnstimmung or delusional mood (Mishara, 2010), characterized by uncertainty, altered salience, and a sense that ordinary events carry hidden meaning. They can also appear suddenly, sometimes as 'insight experiences" that present themselves with the immediacy of perception rather than the deliberation of inference (Jensen, 2022; Sips, 2019). And once established, they are typically resistant to counter-evidence in ways that ordinary mistaken beliefs are not.

According to (Teufel and Fletcher, 2016; Sass and Byrom, 2015) any useful computational account of delusions must engage with this phenomenological aspect, not just with the surface features of incorrect belief content. Predictive coding has been advanced as one of those an account, and while it can explain some aspects, it is less successful in explaining others.

2.1.1 Precision and the prodromal phase

Within this dominant account, delusions have been modeled as disorders of precision weighting in the inferential hierarchy (Adams et al., 2013; Fletcher and Frith, 2009; Corlett et al., 2019). Specifically, the account holds that sensory prediction errors are estimated as more precise than they should be, while priors are estimated as less precise. The result is that ordinary sensory data, which under typical precision weighting would be discounted as noise or accommodated within existing priors, is instead taken to be highly informative and drives updates to higher-level beliefs.

2.1.2 What the standard account explains poorly

The precision-weighting account of delusions struggles with several features that are central to the clinical phenomenology and that have been highlighted in recent critiques (Harding et al., 2024; Corlett and Fletcher, 2021).

Sudden onset Since the standard account views the process of delusion formation as gradual and iterative, prediction errors accumulating to gradually update prior beliefs and over time the system settles on a new, delusional interpretation of the world (Corlett et al., 2019). This does not fit well with common pattern in which a delusional belief appears suddenly and fully formed, often experienced as a revelatory insight that transforms the person's understanding of their situation in a single moment (Jensen, 2022; Sips, 2019). Jensen (2022), writing from personal experience of schizophrenia, describes the appearance of delusional ideas as having "the experiential quality" of self-evident truths rather than reasoned conclusions. An iterative inference process that gradually accumulates evidence does not naturally produce this phenomenology.

Fixedness A second difficulty concerns the resistance of established delusional beliefs to counter evidence. The precision-weighting account, predicts that beliefs are updated when prediction errors are large. But established delusions are typically held with tenacity even when the person is presented with evidence that should refute them (American Psychiatric Association, 2013). The same computational principle that was hypersensitive to evidence during the prodromal phase appears to become unusually insensitive to it once a delusional belief is established.

Perception like quality A third difficulty is more phenomenological than computational. Delusional beliefs are often described as having a perception-like immediacy. They are not experienced as conclusions one has reasoned to, but as fact about the world that present themselves directly, in the same way as visual experience (Jensen, 2022; Sass and Byrom, 2015).

The precision-weighting account, framed entirely in terms of iterative inference, does not naturally reflect this matter. Iterative inference is, by its nature, a process of gradual conclusion-drawing. It does not produce outputs that have the immediacy of percepts.

2.2 Hybrid predictive coding and delusions

2.2.1 The library of amortized mappings

Harding et al. (2024) introduced a useful informal extension of the HPC architecture. The idea is that the amortized network does not implement a single mapping from data to beliefs but rather a library of context-dependent mappings. To borrow their illustration, the same image of a knife can evoke markedly different inferences in the context of a horror film, a cooking programme, or a friend's home, and these differences are the result of the rapid, automatic application of a context-dependent mapping. The mapping selected determines not just the belief that arises but the experiental character of that belief.

This idea is not part of the formal HPC model as developed by Tschantz et al. (2023), which trains a single set of amortized parameters ϕ across the whole dataset. But it is natural extension: in any sufficiently rich training regime, different stimuli or stimulus-context combinations will be associated with different inferential outcomes, and the amortized network will encode this structure in its weights. Whether one calls the result a "library of mappings" or simply a context-sensitive amortized function, the point is that what the amortized network has learned is shaped by the distribution of experience the system has been exposed to.

Harding et al. (2024) use this framing to make two specific claims about delusion formation. The first concerns the consequences of selecting an inappropriate mapping. If, for some reason, the amortized network applies a mapping that is appropriate to one context (say, the horror film) in a context where it is not (the friend's home), the resulting inference will be inappropriate in a way that resembles delusional display: "the person might, automatically and unquestioningly, infer ominous consequences seeing a knife in the home of a friend." Importantly, this inappropriate inference will not be corrected by iterative inference if the iterative system is itself disturbed and the standard predictive coding account of the prodromal phase, with its weakened priors and elevated sensory precision, supplies precisely such a disturbance. Harding et al. (2024) note that under these conditions, the amortized mapping "persists uncorrected": the iterative system, lacking strong priors against which to evaluate the amortized guess, accepts it.

The second claim concerns what shapes the library of amortized mappings in the

first place. The mappings the amortized network learns reflect the experiences the system has had, and individuals differ in their experiential histories. Harding et al. (2024) suggest that this provides a computational handle on the well-established association between childhood trauma and the development of psychotic disorders (Bailey et al., 2018; Croft et al., 2019, 2022). According to this account, traumatic experiences alter input-belief mappings that the amortized network learns and make certain inferential patterns more readily available and harder to correct. A history of threatening experiences may, on this view, shape the amortized network such that the mapping from ambiguous social input to threatening interpretation is the one most readily applied. Iterative correction of this mapping requires not just intact iterative inference but a generative model that can supply alternatives, and where the entire experiential history has been weighted toward threat, the alternatives may be limited.

The second claim is particularly relevant to the experiments in subsequent sections. The trauma framing mentioned above, although not literally clinical, captures a feature of how amortized networks should acquire biases. Amortized networks encode the statistics of the training distribution (individual experiential history) including imbalances in how stimuli have been paired with outcomes.

2.2.2 Perception-like inference

A separate but related point in Harding et al. (2024) concerns the phenomenological quality of delusional beliefs. As discussed in Section 2.1.2, delusions are often described as having an immediacy more characteristic to perception than to belief. They be presenting themselves as facts about the world rather than as conclusions one has reasoned to. As stated before, the standard predictive coding account lacks this register by its commitment to gradual iterative refinement.

Harding et al. (2024) observe and propose that the amortized inference component of HPC could supplies what is missing here. The amortisation network map directly from data to beliefs at every level of the hierarchy and it is doing so in a single feedforward pass with no iterative duty. Harding et al. (2024) note further that the timescale of amortized processing is plausibly independent of the level of the hierarchy at which it operates whereas iterative inference operates more slowly at higher levels that at lower ones (Hasson et al., 2008). Amortisation produces its output in a single feedforward pass regardless of where in the hierarchy that output sits suggesting that high-level inferences (of the kind that delusional beliefs typically constitute) could in principle be furnished by amortisation with the same immediacy that characterizes low-level visual inference.

2.2.3 Local minima

Harding et al. (2024) propose a mechanism that exploits the interaction between the two HPC inference components and his argument runs as follows. Iterative inference, formulated as gradient descent on variational free energy (Equation 1.4), is in general not guaranteed to find the global minimum of the loss landscape. Both, standard predictive coding and HPC are vulnerable to settling on local minima during inference.

Where they differ is in their starting points: standard predictive coding initializes iterative inference at random or at some fixed default value and HPC initializes iterative inference at the output of the amortized network (which has been shaped by previous experience to predict where iterative inference has tended to settle). If the amortized mapping has previously led to a local minimum (because of the system's experiental history, or because of an inappropriate mapping selection of the kind discussed in Section 2.2.1) then iterative inference initialized from that mapping will tend to settle in the same local minimum again. The amortized network is then updated with this new converged value as its target, reinforcing the mapping that led there. Over repeated cycles, the amortized network and the iterative process can become mutually reinforcing in a way that locks the system into a particular interpretation. Harding et al. (2024) suggest that this self-reinforcing dynamic could underpin the phenomenological transition from the delusional mood to the entrenched delusional state.

2.2.4 What the experiments in this thesis can and cannot test

The proposal sketched above is computationally underexplored, and Harding et al. (2024) are explicit about it. They describe their account as "preliminary" and call for formal computational modeling focusing on (i) learning and optimal selection of the amortized mapping, and (ii) how the amortized and iterative parts of the system interact. The experiments reported in subsequent chapters take a step toward this formalisation, but it is important to be clear about what they do and do not establish.

What they test The experiments use the Tschantz et al. (2023) HPC architecture, trained on MNIST to investigate three specific computational properties:

1. Firstly, whether imbalanced experience produces stimulus-specific biases in the trained model, and where in the architecture those biases are localized (Chapter 4). This is relevant to the "library of mappings" framing in Section 2.2.1: it asks whether biased experience is encoded primarily in the amortized network, and whether iterative inference corrects, amplifies, or leaves it unchanged.

2. Secondly, whether such biases can be reversed by subsequent corrective experience and what the form of the reversal reveals about where the bias lives (Chapter 5).

3. Finally, whether environmental biases acquired through experience with some stimuli transfer to other stimuli whose own training contradicts that environmental bias (Chapter 6).

These three questions are each motivated by Harding et al. (2024) proposal, but they do not test the proposal in its entirety.

What they do not test In the experiments we are using a toy domain, specifically MNIST handwritten digits paired with context signals, and analogy to biased experience through imbalanced training distributions. These are not models of clinical delusions, and their findings should not be read as it. The use of terminology like "trauma," "healing," "threatening" and "benign" contexts is a deliberate analogy to the framing in Harding et al. (2024) we presented in previous sections. There are also two specific aspects of Harding et al. (2024) proposal that the present design does not engage with. First, context is given to the model as an observed input clamped at the sensory layer rather than as a latent variable the model must infer, so the experiments speak to how the library of mappings is formed by experience but not to how a mapping is selected from the library at inference. Second, the prodromal disturbance that Harding et al. (2024) place at the centre of the first proposal, namely weakened priors that let a misapplied amortized guess persist uncorrected, is not manipulated here.

Why test these properties anyway Despite these limitations, the experiments are worth conducting because they test the computational properties that are important for Harding et al. (2024) proposal. Firstly, if imbalanced experience does not produce stimulus-specific biases in the amortized network, the library of amortized mappings framing loses its accountability for how the library is formed. Secondly, if such biases are easily corrected by iterative inference initialised from them, the account of the fixedness of delusional beliefs loses one of the two mechanisms Harding et al. (2024) propose for it. And finally, if amortized biases transfer across stimuli it loses the context sensitivity.

2.2.5 Research questions

Does imbalanced experience reshape the model's behavior at test? This question aims to explore whether the HPC model trained on imbalanced distribution where a digit paired with one context appear more often than the same digit with another context leaves a measurable mark on what the model infers when later tested. In the chapter's analogy, it is the question of whether a biased experiential history produces the stimulus-specific inferential tendencies that the library-of-mappings framing requires. If imbalanced experience left no trace at test, the premise that amortized

networks encode the statistics of their training distribution would therefore not hold for this architecture.

Where does the bias live? Afterwards, given that a bias exists, second question localizes it within the HPC architecture. Is the bias encoded primarily in the amortized network, the feedforward data-to-belief mapping, or does it also reside in, or get modified by, the iterative inference process? This matters because the mentioned account hinges on the amortized component carrying biases shaped by experience while iterative inference should serve as the corrective mechanism.

How do the two inference modes interact? Finally, the last question examines the relationship between the amortized and iterative components when they operate together. It asks a question whether iterative inference reverses an amortized bias, reinforces it, or settles into the same biased solution the amortized network initialized it toward. This connects to the local-minima mechanism of Section 2.2.3, where iterative inference initialized from a biased mapping tends to re-converge on the same solution, which then becomes the amortized network's new target and therefore the mechanism produces a self-reinforcing loop.

Chapter 3

Methods

3.1 Task setup

The Harding et al. (2024) account of delusions fundamentally involves the misapplication of learned inference to inappropriate contexts. According to this, amortized inference operates through learned feedforward mappings from sensory inputs to higher layers, where the selection of the appropriate mapping depends on contextual information and past experiences Harding et al. (2024).

When we look at real-world scenarios, the same perceptual input can have radically different meanings depending on the context in which it is perceived. In standard MNIST classification, there is no ambiguity in interpretation, and there is no opportunity for the model to select between multiple valid interpretations of the same input. Therefore, the inference problem becomes trivial. Simple mapping of visual features into digit identity eliminates the core mechanism to underlies delusions, which is the inappropriate selection of one mapping over another based on context. Furthermore, in delusions, the problematic belief is not that the person cannot perceive accurately but rather they apply inappropriate interpretation framework. Without context, we cannot distinguish between perceptual errors and inferential errors.

To create a task in which the model can make perceptual errors (wrong digit) but also interpretive errors (right digit, wrong context), we augmented each MNIST image input vector with context signals that determines whether the digit should be classified as benign (Context 1, "C1") or threatening (Context 2, "C2"). This setup makes the context signal a part of the model's observation and not a latent variable.

Each combination of digit identity and context is treated as a separate class, giving a 20-way output where classes 0–9 correspond to digit d in C1 (class index d), and classes 10–19 to digit d in C2 (class index $d + 10$). Labels are one-hot vectors over these 20 classes, soft-labeled with 0.97 on the correct class and 0.03 elsewhere. The two contexts signals are represented as vectors of dimension two: C1 $= [10, 0]$ and C2

$= [0, 10]$.

The dataset is created from standard MNIST split (60,000 training, 10,000 test) where pixel values of each sample are scaled to $[0, 1]$ and flattened to 784-dimensional vectors. For further reference, how the input is preprocessed, see Figure 3.1.

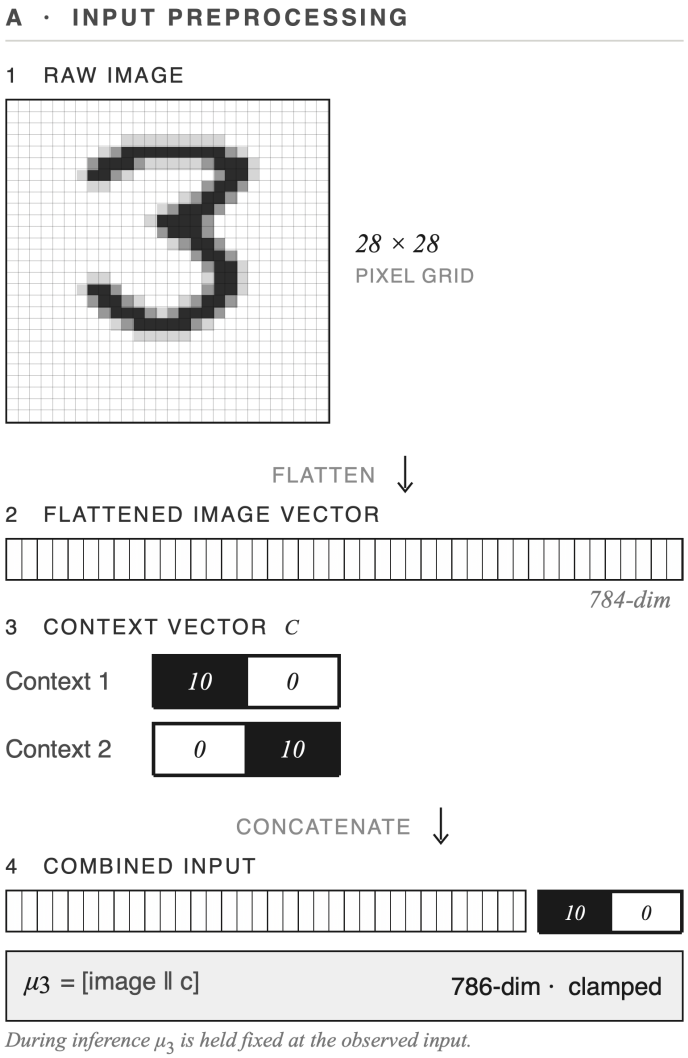

Figure 3.1: A 28x28 MNIST image is flattened into a 784 dimensional vector and concatenated with a 2 dimensional context vector c. The resulting 786 dimensional vector is assigned to the latent state μ_3, which is held clamped at the observed input throughout inference.

3.2 Model architecture

The model has two parameter sets that are trained jointly in the default configuration. One is a generative network with parameters θ that supports iterative inference, and

second is an amortised network with parameters ϕ.

The generative network is a three-layer hierarchy with latent states μ_l at four node groups indexed $l = 0, 1, 2, 3$ and dimensions $[20, 128, 256, 786]$, where μ_0 is the class layer and μ_3 is the sensory (image + context) layer.[1] The three weight layers connect $\mu_0 \to \mu_1$, $\mu_1 \to \mu_2$, and $\mu_2 \to \mu_3$, with tanh activations on the two hidden mappings and a linear activation on the sensory layer. The top-layer state μ_0 corresponds directly to the 20 classes, and classification is performed by argmax over μ_0 without an additional readout.

Second, amortised network is a separate feedforward network with the inverse layer structure $[786, 256, 128, 20]$ and the same activation rule (tanh on hidden, linear on output). It is initialised identically to the generative network and the single forward pass of the amortised network produces initial values for all hidden latent states μ_0, μ_1, μ_2 simultaneously, for more details see Figure 3.2

3.3 Training configurations

Throughout experiments in this thesis, we are evaluating three architectures which are not three different models but three different training configurations of the same HPC model:

- **Hybrid model.** Both amortization and generative networks are trained jointly.
- **Feedforward model.** Only the amortization network is trained while generative weights remain frozen at their random initialization. At the test time the classification is read directly from the amortised output with no iterative inference.
- **Generative model.** Only the generative network is trained. The amortised network remains at its random initialisation throughout training and is bypassed at test time.

3.4 Inference modes

A trained model can be evaluated under several inference modes depending on which component drives the prediction. **Hybrid inference.** The amortised network is run forward to initialise μ_0, μ_1, μ_2 and the input is clamped at μ_3. After that 100 iterative steps are run on the free latent states. Classification happens as the argmax of μ_0 after the last PC iteration. The hybrid inference mode requires both networks to have been trained and is therefore applied only to the hybrid model. **Amortization-only**

[1] This indexing reverses the convention of Section 1.4, in which we denote μ_0 as the sensory layer. The reversal now matches the code implementation and the rest of this chapter.

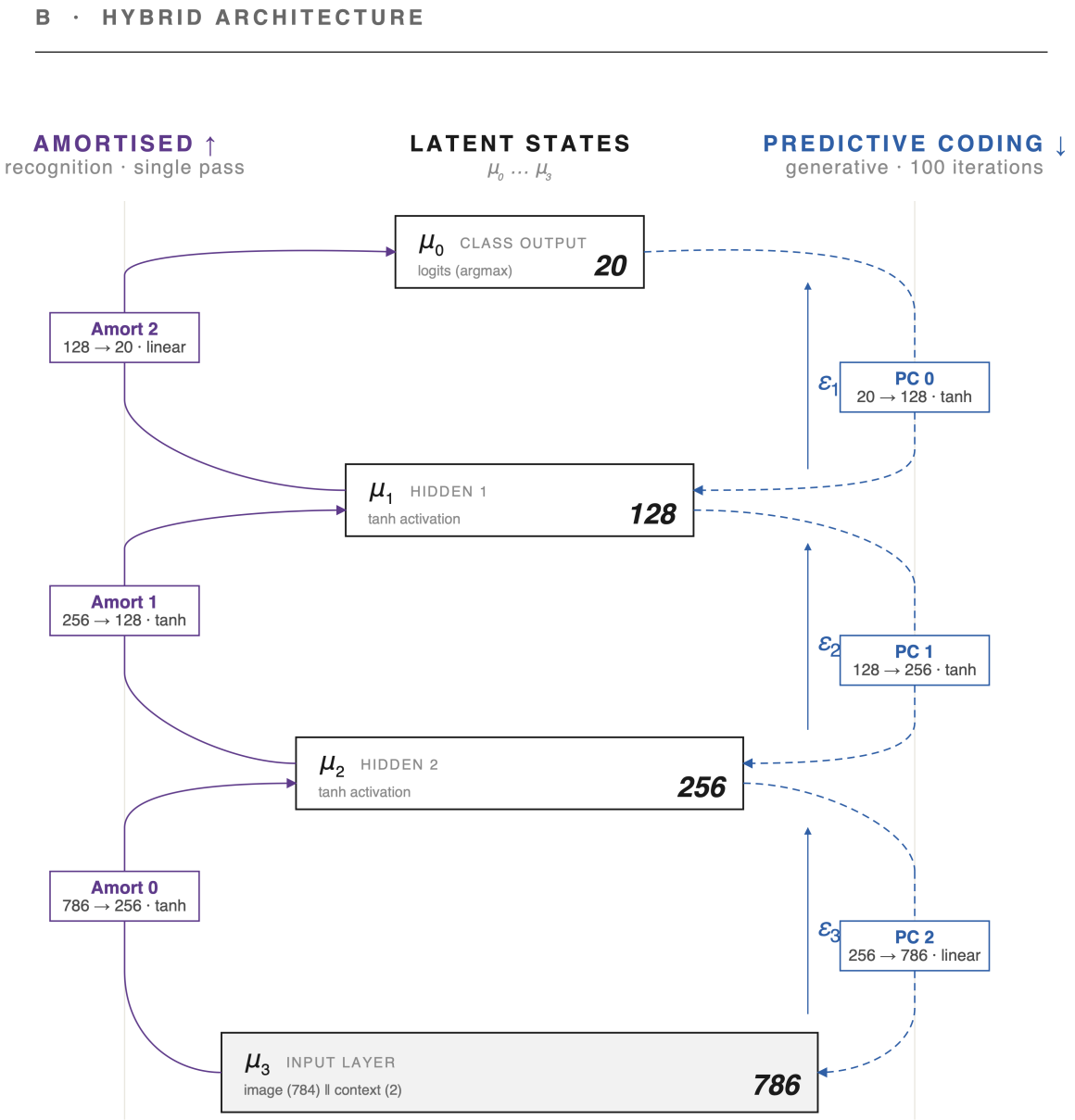

Figure 3.2: Ilustration of hybrid predictive coding architecture focusing on two networks. The model has two parameter sets (amortised network ϕ; generative network θ) that work together over a shared hierarchy of latent states $\mu_0, \dots, \mu_3$ with dimensions $[20, 128, 256, 786]$. The amortised network, shown on the left (purple) maps the input upward through three weight layers in a single feedforward pass. This produces initial values for all hidden states μ_0, μ_1, μ_2 simultaneously. The generative network on the right side of the figure maps top-down through three weight layers producing predictions and the differences, between these and latent states define the prediction errors $\varepsilon_1, \varepsilon_2, \varepsilon_3$. These errors then drive the updates of the free latent states over 100 steps at test time. The input layer μ_3 shown in grey combines the 784-dimensional image vector with a 2-dimensional context signal and is clamped throughout the whole inference process. The class output μ_0 is then read out by argmax of its 20-dimensional activation.

inference. In this mode, we run forward the amortised network, and its top-layer output is used directly as the classification with no iterative refinement. This mode is applied to the feedforward model and to the hybrid model when we want to isolate the amortised network's contribution. **PC-only inference.** Here the amortised network is bypassed and the hidden latent states are initialised from a random distribution. The input is clamped at μ_3, and 100 iterative steps are run. Throughout the experiments we use four initialisation schemes: random Gaussian noise at $\sigma \in \{0.01, 0.1, 1.0\}$, and a **general prior** in which hidden states are initialised to the mean latent state the model settles on over the training distribution. Concretely, before evaluation, the model is run over every batch of the training data with both the input and the true label clamped (a training-style forward pass with 20 iterative steps and no weight update), and the resulting μ_l values are averaged across all samples and batches at each hidden layer. The three averaged vectors are then used as the initial latent states. PC inference is applied to the generative model and to the hybrid model.

3.5 Evaluation

Test set. All experiments are evaluated on a balanced test set constructed from the MNIST test split. For each test image, the context label is assigned to C1 or C2 with $p = 0.5$ meaning that digit 3, which has 1,010 samples in the MNIST test split, this produces approximately 505 samples per context. The balanced test set deliberately differs from the training distributions used in the experiments in order to expose any context bias the model has acquired.

Outcome categories. Each test sample is classified into one of four outcome categories based on the model's final prediction relative to ground truth of the test sample: **perfect** (digit and context both correct), **context error** (digit correct, context wrong), **digit error** (digit wrong, context correct), and **both wrong**.

Metrics. Our primary metric is the C1→C2 error rate, computed per digit:

$$\text{C1}\rightarrow\text{C2 rate}_d = \frac{\#\{\text{samples of digit } d \text{ with true context C1, predicted as digit } d \text{ in C2}\}}{\#\{\text{samples of digit } d \text{ with true context C1}\}}.$$

This captures a context-only error in the C1→C2 direction where the digit is correctly identified but its context is misclassified as threatening (C2) when it should be benign (C1). The reverse direction of C2→C1 error rate, is defined analogously.

We also report overall accuracy (the proportion of test samples where both digit and context are correctly predicted), context-specific accuracy (computed separately on C1 and C2 samples), along with per-digit accuracy and outcome-category breakdowns.

Seeds and statistical reporting. Each experiment evaluates the same model architecture under multiple random seeds and reports means ± standard deviations across seeds. In Experiments 1 and 2 we use 5 seeds while Experiment 3 uses 10 seed-matched pairs across its two training conditions. Where formal hypothesis tests are reported (Chapter 6), they are paired t-tests on the per-seed C1→C2 rates, with Cohen's d as the effect-size measure.

Chapter 4

Experiment 1: Class Distribution Experiment

4.1 Introduction

In the Harding et al. (2024) account of delusions, it is understood that the amortized inference mappings are shaped by the statistical regularities an individual has encountered throughout their life. These learned mappings form what he describes as a "library" of possible input-cause interpretations/associations. The selection of appropriate input-cause mapping depends heavily on past inferences in similar contexts. The availability, preference and accessibility of different mappings within this library directly reflects individual's history of lived experiences making frequently encountered stimulus-context pairings strongly represented and readily activated, while rarely experienced pairs remain weakly represented Harding et al. (2024).

This computational principle ensures that two individuals encountering identical sensory input may end up with different inferences if their lived experiences (learned amortised mappings) differ substantially. This examined theoretical account explicitly suggests that trauma experience could reshape the library of mappings, creating worldviews where threatening interpretations become more accessible and more likely to be selected, even in objectively benign contexts. To operationalize this theoretical proposal, we translate life experiences into training data distributions. Each model we train represents different individual with a unique lived history. The proportion of times a given digit appears in Context 1 versus Context 2 during training directly corresponds to the proportion of times that individual has encountered that stimulus in benign versus threatening contexts throughout their life.

4.2 Experimental Design

4.2.1 Data setup

Training distribution

For training, we chose an imbalanced training distribution that was configured as follows:

- **Digit 3 (trauma associated):** 90% presented in threatening context (C2), 10% in benign context (C1)
- **All other digits (0-2, 4-9):** 10% presented in threatening context (C2), 90% in benign context (C1)

This training regime should simulate an individual who has overwhelmingly experienced a particular stimulus (digit 3) in threatening scenarios, while other stimuli have been predominantly encountered in benign contexts. The model was trained for 10 epochs with a batch size of 64. Weight updates were calculated using a local prediction error learning rule and applied using the Adam optimizer with a learning rate of 1×10^{-4}.

4.2.2 Predictions

- *If the bias is encoded in amortized mappings*, then the feedforward-only model should exhibit a digit 3 + C2 bias on the balanced test set because amortization is the only component being trained.
- *If the bias is encoded in generative associations*, then the generative-only model should also exhibit the bias since the generative weights have learned the same imbalanced training distribution.
- *If the bias arises from the interaction between amortization and PC inference*, then the hybrid model should show a stronger bias than either of the networks alone, and within the hybrid model, the bias should be reduced when amortization is bypassed.

4.3 Baseline model validation

Before introducing any imbalanced training, we verify that the three architectures produce balanced and accurate predictions when trained on balanced distribution.

We train all three configurations for 10 epochs on a balanced version of the task (each digit appearing equally often in C1 and C2) and evaluate on the balanced test

set. Across all three models there is comparable overall accuracy (Table 4.1) and more importantly, each configuration produces near symmetric performance across the two contexts (Table 4.2). Having established this baseline, the results in following sections can be read as deviations introduced by an imbalance in the training distribution and not as built-in tendency of the architectures.

Model	**Epochs**	**Hybrid Acc**	**PC Acc**	**Amort Acc**
Hybrid	10	85.26%	84.21%	84.52%
Generative	10	N/A	82.05%	N/A
Feedforward	10	N/A	N/A	79.61%

Table 4.1: Baseline overall accuracy on balanced distribution after 10 training epochs. The hybrid model is evaluated under all three inference modes. The generative and feedforward models are evaluated only under the inference mode that matches their training configuration.

Model	**Inference mode**	**C1 Accuracy**	**C2 Accuracy**	**Difference**
Hybrid	Hybrid	85.01%	84.86%	0.15%
Generative	PC	81.27%	81.78%	0.52%
Feedforward	Amort	79.01%	79.23%	0.22%

Table 4.2: Context-specific accuracy for the baseline models. All three configurations produce essentially symmetric performance across C1 and C2 when training is balanced (difference < 0.6 percentage points).

4.4 Results

We evaluated three architectures (hybrid, feedforward-only, generative-only) using different inference modes accordingly to model architecture on a balanced test set in which digit 3 appears equally often in Context 1 and Context 2, with $n = 505$ samples per context. This contrasts with the training distribution, where digit 3 appeared 90% of the time in C2 and only 10% in C1, while all other digits appeared 90% in C1 and 10% in C2.

4.4.1 Context confusion patterns accros three architectures

Hybrid model results

Under hybrid inference (amortisation initialization followed by 100 PC iterations), the pattern for digit 3 is the inverse of the pattern for every other digit. Digit 3 in C1 produces 301 C1→C2 errors out of 505 samples (59.56%), while digit 3 in C2 produces

0 C2→C1 errors. In contrast, every other digit shows 0% C1→C2 errors but elevated C2→C1 error rates ranging from 27.76% (digit 5) to 90.58% (digit 1), see Figure 4.1. This asymmetry directly reflects the training distribution: digits seen predominantly in one context are predicted in that context regardless the actual context signal.

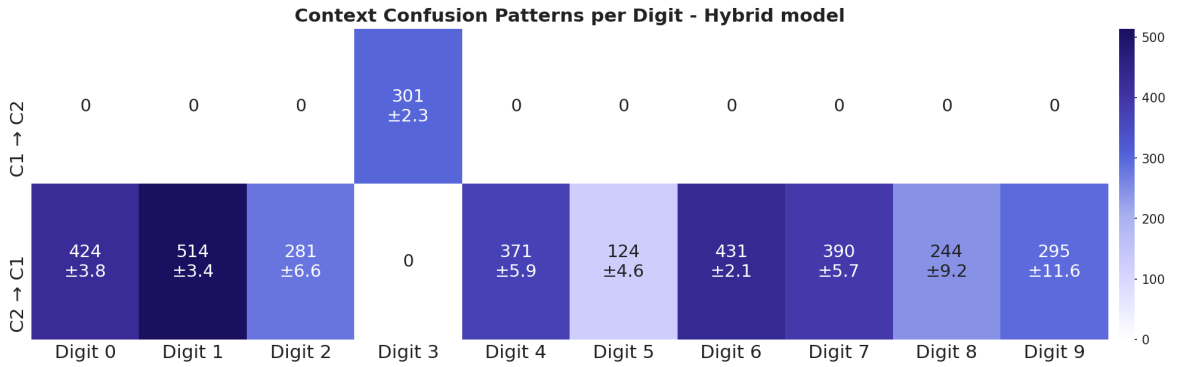

Figure 4.1: Context confusion patterns per digit for the hybrid model (amortization initialization followed by 100 PC iterations) on balanced test set ($n = 505$ per context). The top row counts C1→C2 errors (digit correct, context predicted as C2 instead of C1) and the bottom row counts C2→C1 errors. Cells show the mean count out of 505 with standard deviation across 5 seeds, shaded by magnitude.

Feedforward only model results

The feedforward model uses only the amortization network with no PC refinement. The qualitative pattern matches the hybrid model: digit 3 shows a C1→C2 bias ($24.16 \pm 1.27\%$, 122.0 ± 6.4 out of 505) with no errors in the opposite direction, while every other digit shows 0% C1→C2 errors and C2→C1 rates ranging from 15.28% (digit 8) to 77.62% (digit 6), see Figure 4.2. The bias direction is identical to the hybrid model's, and is consistent with the training distribution.

However, the digit 3 bias is smaller than in the hybrid model: 24.16% versus 59.56%. Amortization alone does encode a digit 3 C2 association, but its strength when expressed directly is less than half of the strength expressed when the same amortization output is trained and fed into PC inference (Figure 4.1). Adding PC inference on top of amortization initialization more than doubles the context error rate on digit 3 C1 samples, despite PC having access to generative weights that, under random std=0.01 initialization, produce 0.00% context errors (Figure 4.3).

Generative model results

The generative model produces essentially no context-only errors in either direction, for any digit. The only non-zero entry is a $0.4 \pm 0.08\%$ C1→C2 rate for digit 7, corresponding to a fraction of one error on average across seeds, see Figure 4.3. Errors

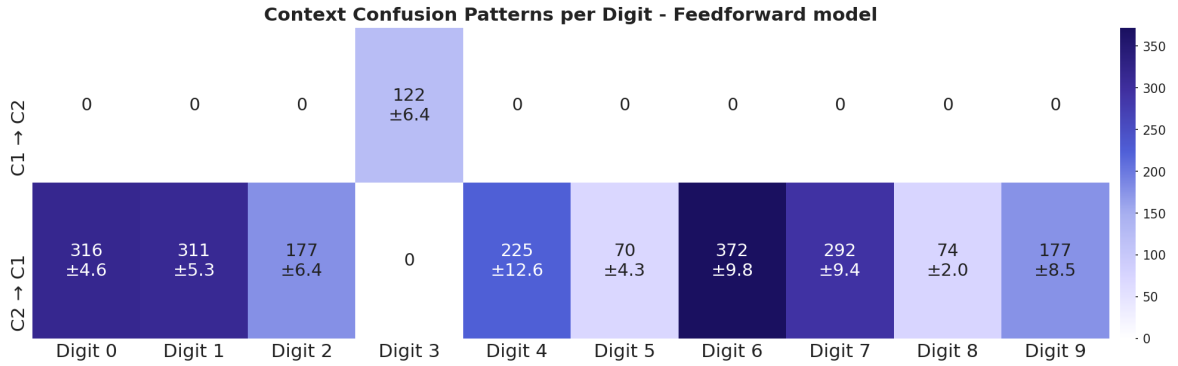

Figure 4.2: Context confusion patterns per digit for the feedforward model (amortization output only) on balanced test set ($n = 505$ per context). Rows and shading is as in Figure 4.1. The qualitative pattern matches the hybrid model: digit 3 is the only digit with C1→C2 errors (122, 24.16%) and shows 0 errors in the opposite direction, while every other digit shows 0 C1→C2 errors and elevated C2→C1 counts.

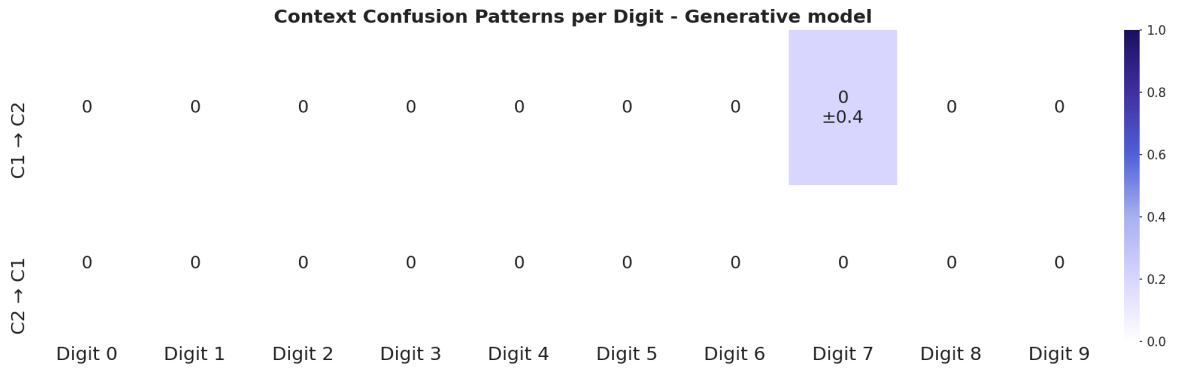

Figure 4.3: Context confusion patterns per digit for the generative model (PC inference from random std=0.01 initialization) on the balanced test set ($n = 505$ per context). Rows and shading as in Figure 4.1. The generative model produces essentially no context-only errors in either direction for any digit; the only non-zero entry is a $0.4 \pm 0.08\%$ C1→C2 rate for digit 7.

that do occur (reflected in the $< 100\%$ digit accuracy) are wrong digit predictions with correct context. This pattern is qualitatively different from both the hybrid and feedforward models, which both show a digit 3 C2 bias.

The result is consistent with the PC-only inference findings (Table 4.4) on the hybrid weights showing that when amortization is removed from the inference path, the generative weights produce balanced context predictions.

4.4.2 Iterative inference influence on Hybrid model

Effect of predictive coding power on biased inference

To examine whether the observed C1→C2 bias for digit 3 from Section 4.4.1 could be reduced through additional iterative inference, we evaluated the hybrid model under varying numbers of predictive coding iterations: 100 (hybrid baseline), 300, 500, and 1000. This manipulation tests whether increased number of PC iterations allows the system to escape biased amortized initializations and recover the correct context interpretation in a balanced test setting (digit 3, C1: $n = 505$, C2: $n = 505$).

Results show that the increased number of PC iterations does not reduce the C1→C2 bias but it amplifies it. Figure 4.4 shows that the C1→C2 error rate grows with iteration count, from $59.56 \pm 0.46\%$ at 100 iterations to $75.45 \pm 0.72\%$ at 1000 iterations a 15.89 percentage point increase. Table 4.3 reveals the mechanism behind this observed growth. The rise in context errors is matched almost exactly by a fall in digit errors (from 40.44% to 24.16%, a 16.28 pp decrease). Samples that the model originally got wrong by predicting an incorrect digit in the correct context are reinterpreted as the digit 3 in the wrong context (C2). In other words, additional PC iterations do not pull predictions toward the ground-truth C1 interpretation but they pull them toward the digit 3 C2 association, that is the dominant pattern in the training distribution.

A small rise in the "both wrong" rate for C2 samples (from 1.19% to 4.12%) accompanies the iteration increase, suggesting that even C2 predictions, which are correct under standard inference, become slightly less stable as iteration count grows. C2 "Perfect" accordingly falls from 98.81% to 95.64%.

These taken together indicate that when PC inference is initialized from an amortization output, additional iterations do not act as corrective process one might be thinking but instead it moves the predictions further into the dominant association from training distribution.

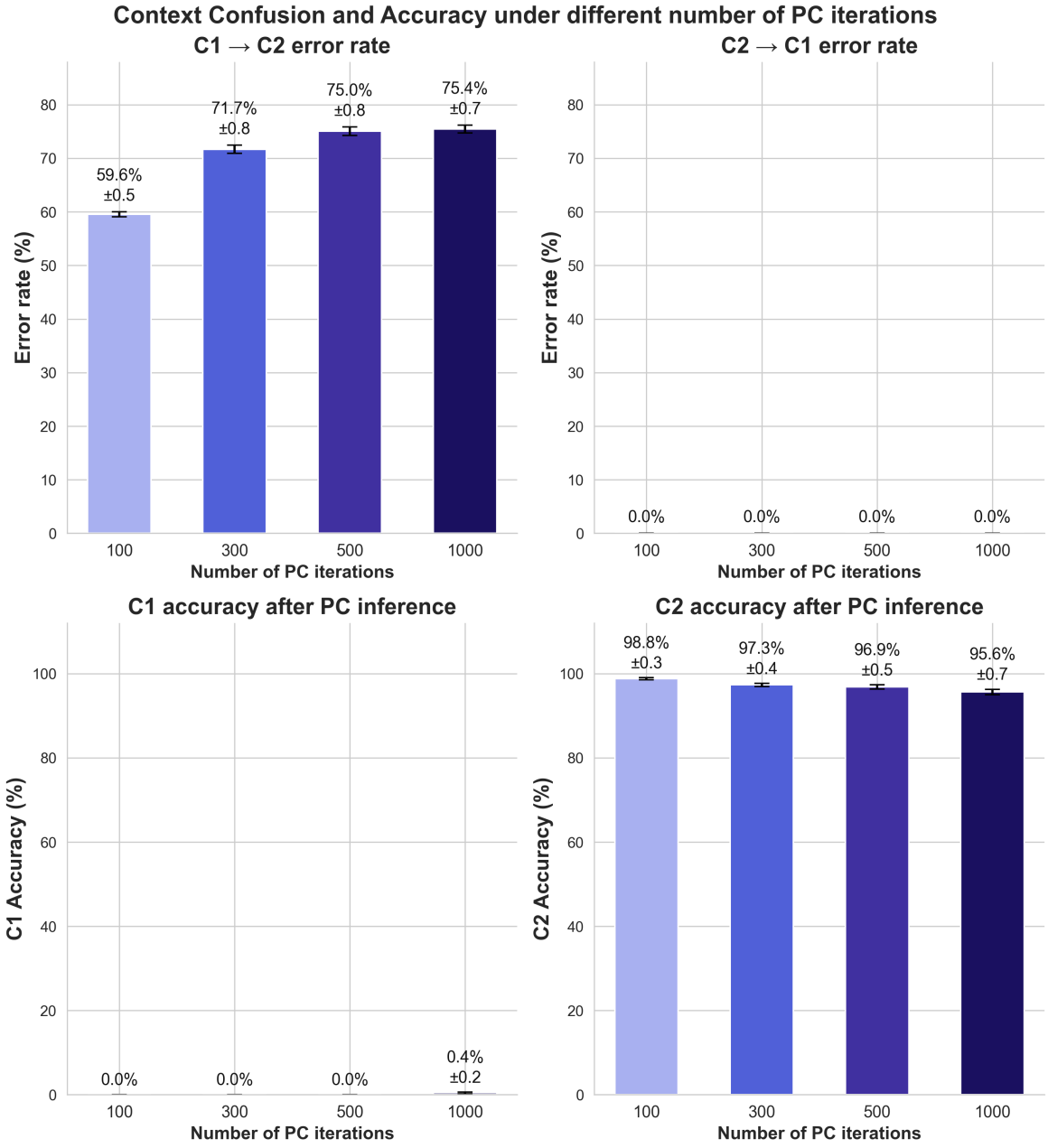

Figure 4.4: Effect of PC iteration count on digit 3 inference in the hybrid model. Top: C1→C2 error rate rises with iteration count, while the C2→C1 error rate remains at 0%. Bottom: C1 accuracy stays at (or near) 0% across all iteration counts; C2 accuracy declines slightly. Bars show mean ± standard deviation across 5 random seeds.

PC iterations	Context	Perfect	Context error	Digit error	Both wrong
100 (baseline)	C1	**0.00%**	**59.56 ± 0.46%**	**40.44 ± 0.46%**	**0.00%**
	C2	**98.81 ± 0.25%**	**0.00%**	**0.00%**	**1.19 ± 0.25%**
300	C1	0.00%	71.68 ± 0.76%	28.32 ± 0.76%	0.00%
	C2	97.35 ± 0.37%	0.00%	0.00%	2.65 ± 0.37%
500	C1	0.00%	75.05 ± 0.81%	24.95 ± 0.81%	0.00%
	C2	96.87 ± 0.49%	0.00%	0.00%	3.13 ± 0.49%
1000	C1	0.40 ± 0.22%	75.45 ± 0.72%	24.16 ± 0.75%	0.00%
	C2	95.64 ± 0.65%	0.00%	0.24%	4.12 ± 0.68%

Table 4.3: Full outcome breakdown for digit 3 samples by PC iteration count(hybrid model, amortization initialization, balanced test, $n = 505$ per context). Values are mean ± standard deviation across 5 random seeds; cells without ± are either structurally zero or residuals with negligible variance. *Perfect* = both digit and context correct. *Context error* = digit correct, context wrong. *Digit error* = digit wrong, context correct. *Both wrong* = digit and context both incorrect.

PC inference from random initialization

The previous result raises a question: does the hybrid model's failure on digit 3 C1 samples reflect a property of the trained generative weights or a property of inference dynamics initialized from amortization? The difference between these two is that the first one would mean that the generative weights have learned the digit 3 in C2 while the second one would mean that the generative weights still support correct C1 predictions but cannot be reached from amortization's starting point. To distinguish and answer these, we run PC inference on the same trained hybrid weights while bypassing amortization entirely and initializing hidden states from Gaussian noise at three scales (std $\in \{0.01, 0.1, 1.0\}$).

The starting questions are leaving us with different predictions for the small-noise condition: i) If the bias lives in the generative weights, std = 0.01 should behave like the amortization initialized baseline, ii) but if the bias lives in amortization's output, std = 0.01 should recover balanced performance across the two contexts. Figure 4.5 and Table 4.4 supports the second prediction showing that under random std=0.01 initialization, the C1→C2 error rate drops from 59.56 ± 0.46% to 0.00%, and perfect classification rises from 0.00% to 87.52 ± 0.83% in C1 and stays high at 84.16 ± 0.82% in C2. The same generative weights that produce a near-total C1 failure under amortization initialization produce near-balanced performance under small random initialization. The bias is therefore a property of where inference starts, not a property of what it optimizes.

Noticing further, the larger initialization scales degrade performance, but in a qual-

itatively different way than amortization initialization does. At std = 0.1, both contexts converge to roughly 40–43% perfect classification, with small and roughly same context-error rates in each direction (9.74 ± 1.38% C1→C2, 10.65 ± 1.70% C2→C1) and digit-error rates above 40% on both sides. At std = 1.0, classification collapses: perfect classification falls below 10% in both contexts, "both wrong" rates exceed 35%, and the symmetry between C1 and C2 persists. This looks qualitatively different than a context specific failure produced by amortization initialization. This contrast is summarized in Figure 4.6, which plots C1→C2 error rate against C1 accuracy across all four initialization conditions.

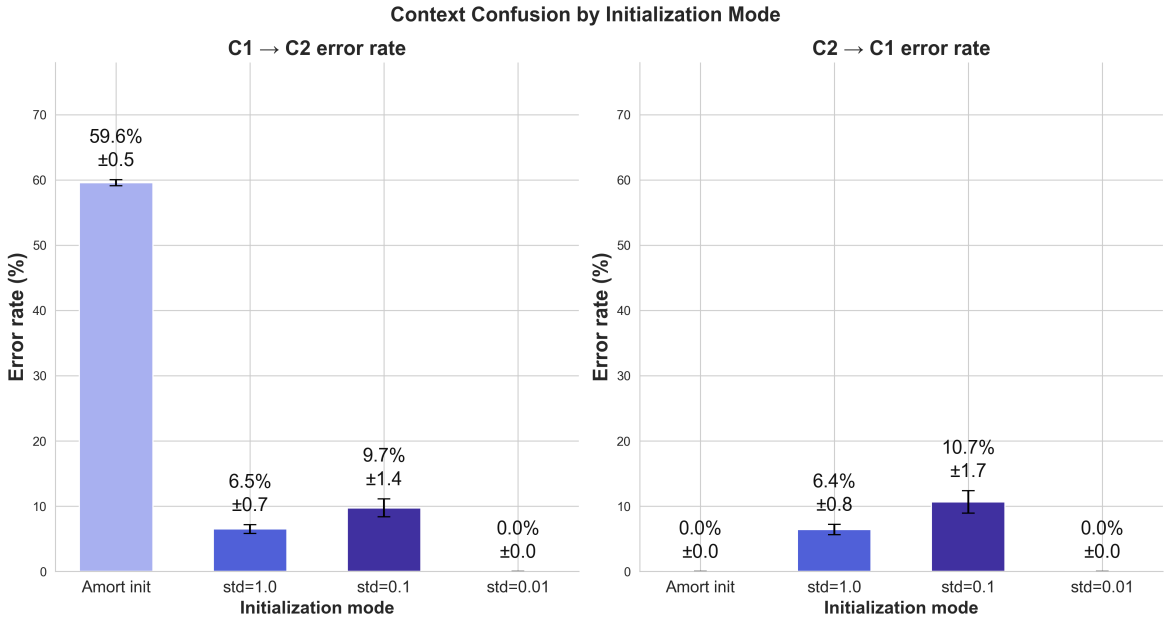

Figure 4.5: C1→C2 (left) and C2→C1 (right) context-error rates for digit 3 under PC inference with four initialization modes (amortization, random std=1.0, 0.1, 0.01). Amortization initialization produces a large and asymmetric C1→C2 bias absent under all random initializations. Bars show mean ± standard deviation across 5 random seeds; 100 PC iterations in all conditions.

Per layer error dynamics during inference

The previous results show that inference outcomes depend on where inference starts. Amortization initialization produces a 59.56% C1→C2 on digit 3 C1 samples, while initialization from random std=0.01 on the same generative weights shows balanced performance. To further examine this we look at inference dynamics themselves. We recorded the L2 magnitude of the prediction error at each layer of the hierarchy across 100 PC iterations, separately under amortization and random initialization. We additionally tracked the trajectory of the top level class belief μ_0 over the same iteration

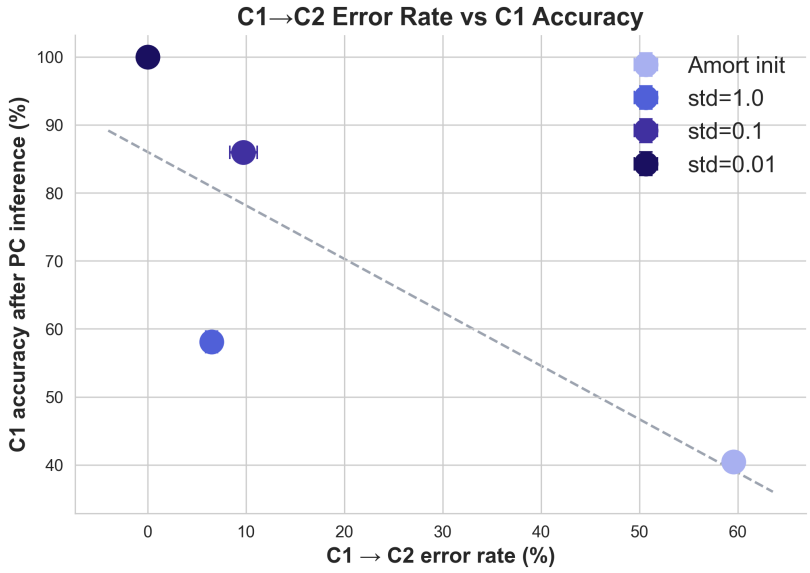

Figure 4.6: Figure shows C1→C2 context-error rate against C1 accuracy after PC inference for digit 3, across the four initialization modes (amortization, random std=1.0, 0.1, 0.01). Each point is the mean across 5 random seeds with error bars showing standard deviation; the dashed line marks the inverse relationship between context-error rate and C1 accuracy. Random std=0.01 initialization (top-left) achieves the highest C1 accuracy with a 0% C1→C2 rate, while amortization initialization (bottom-right) sits at the opposite corner with a ∼60% C1→C2 rate and the lowest C1 accuracy. The intermediate random scales (std=0.1 and std=1.0) fall between these extremes.

Initialization	Context	Perfect	Context error	Digit error	Both wrong
Amortization	C1	0.00%	$59.56 \pm 0.46\%$	$40.44 \pm 0.46\%$	0.00%
	C2	$98.81 \pm 0.25\%$	0.00%	0.00%	$1.19 \pm 0.25\%$
Random std=0.01	C1	$87.52 \pm 0.83\%$	0.00%	$12.48 \pm 0.83\%$	0.00%
	C2	$84.16 \pm 0.82\%$	0.00%	$15.84 \pm 0.82\%$	0.00%
Random std=0.1	C1	$43.29 \pm 2.26\%$	$9.74 \pm 1.38\%$	$42.65 \pm 1.91\%$	$4.32 \pm 0.49\%$
	C2	$40.12 \pm 2.17\%$	$10.65 \pm 1.70\%$	$44.99 \pm 1.97\%$	$4.24 \pm 0.72\%$
Random std=1.0	C1	$8.00 \pm 1.62\%$	$6.50 \pm 0.66\%$	$50.10 \pm 3.12\%$	$35.41 \pm 1.29\%$
	C2	$7.17 \pm 0.88\%$	$6.42 \pm 0.78\%$	$46.38 \pm 1.64\%$	$40.04 \pm 1.43\%$

Table 4.4: Full outcome breakdown for digit 3 samples under PC inference with different initializations on the hybrid model (balanced test, $n = 505$ per context, 100 iterations). Perfect = both digit and context correct. Context error = digit correct, context wrong (C1→C2 for C1 rows; C2→C1 for C2 rows). Digit error = correct context, wrong digit. Both wrong = digit and context both incorrect. Values are mean ± standard deviation across 5 random seeds.

window. We aggregated all trajectories across 5 seeds and ∼505 samples per context condition.

Convergence of per-layer prediction errors In the Figure 4.7 we show the per-layer L2 prediction error across iterations. Under amortization initialization (top row), we can see that all three layer errors start near their final values and barely change across 100 PC iterations. The input-layer error (Layer 3) sits at ∼1.6, the intermediate layer (Layer 2) at ∼0.8, and the top-level error (Layer 1) at ∼0.7 showing no visible descent. On the other hand, under random initialization (std=0.01) at the bottom row, the picture is qualitatively different. The input-layer error (Layer 3) starts at ∼12 and decreases over the first 60 iterations before plateauing near 1.6, which is the same final values reached by using amortization initialization. The intermediate layer follows a similar descent. Both initializations therefore converge to indistinguishable final per-layer errors despite producing different classification outcomes. Based on this, we can rule out an interpretation in which the bias reflects incomplete optimization because PC inference reaches an equally good fit from both starting points, but the two fits correspond to different internal explanations of the same input.

The convergence pattern also reveals that all per-layer errors plateau by iteration ∼60 in both initialization conditions making the choice of 100 PC iterations justified.

Class belief trajectories Further we look how the class belief changes across 100 PC iterations (Figure 4.8). In the top-left panel we can see the biased case, where under amortization initialization the digit 3 C1 samples (correct class 3) the model

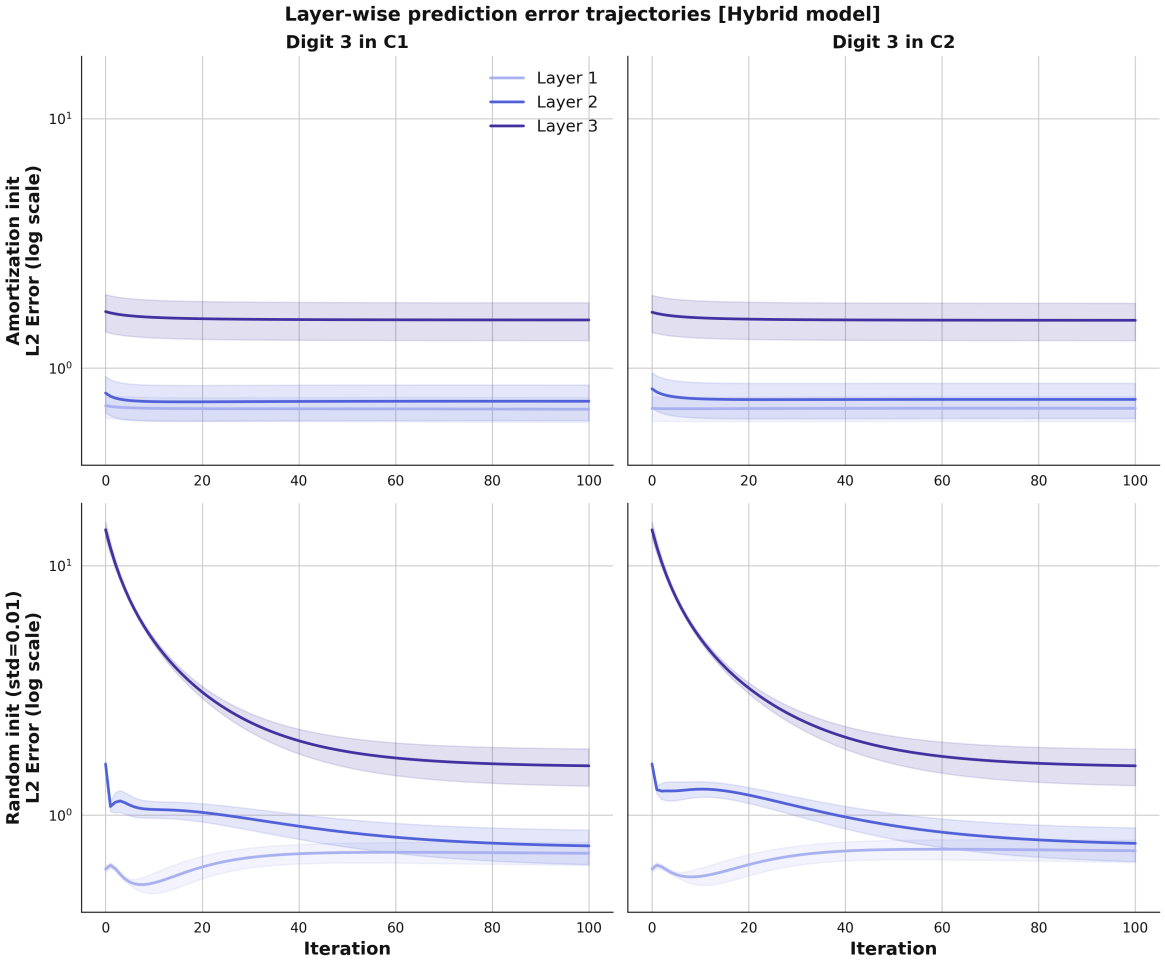

Figure 4.7: Per layer L2 prediction error for digit 3 samples during PC inference in the hybrid model across 100 iterations (log scale). Each row represent different initialization mode: amortization initialization top row versus random initialization bottom row. Columns represent digit 3 in C1 (left column) versus digit 3 in C2 (right column). Under amortization initialization we can see that all three layer errors start near their final values and barely change while under random initialization the input layer error starts at ∼12 and descends over the first ∼60 iterations before plateauing near the same ∼1.6 reached under amortization initialization. Both initializations therefore converge to similar final per layer errors despite producing different classification outcomes. Lines show the mean across 5 random seeds with shaded standard deviation.

starts from the start with class 13 already winning on ∼42% of samples, while class 3 (the correct class) is at ∼0%. Over the course of other 100 PC iterations, the fraction assigned to class 13 grows to ∼60%, while the fraction assigned to class 3 remains at ∼0% throughout. This shows as that PC inference does not pull samples toward class 3 (correct), but it pulls them further into class 13. Note that this iteration window underestimates the full extent of the drift where at 1000 iterations (Table 4.3) the C1→C2 rate reaches 75.45%.

The bottom-left panel (random init, digit 3 in C1) shows the opposite trajectory. The model starts near uniform on all 20 classes, and over the first ∼20 iterations class 3 grows rapidly from ∼5% to ∼80%, plateauing near 87% by iteration 60. The wrong-context class (class 13) remains close to 0% throughout. This shows us that the same generative weights that produce a drift toward wrong class 13 under amortization initialization produce on reversely a drift toward correct class 3 under random initialization (std=0.01).

The right column shows the case with digit 3 in C2 samples which is consistent with training data the model was trained on (meaning model sees digit 3 in C2 from 90% of the time). Under both initializations, class 13 (the correct class for these samples) wins from iteration 0 or grows rapidly and dominate. The two initialization outcomes agree when the training-distribution bias and the ground-truth label point in the same direction.

Effect of general prior on generative model's balanced context performance

The same dissociation holds when we run PC inference on the generative model's own weights using the hybrid model's general prior as initialization. Under this initialization, the digit 3 C1→C2 bias reappears: 76.40±1.35% context errors in C1, with 0.00% perfect classification in C1 and $98.18 \pm 0.15\%$ perfect classification in C2. Under random std=0.01 initialization on the same generative weights both context show 0.00% context errors and balanced perfect-classification rates (82.65±0.80% C1, 87.88±1.01% C2). The Table 4.5 shows that the hybrid model's general prior is sufficient to recreate the bias observed under amortization initialization in the hybrid model.

4.5 Evaluation under unclear context signal

For further exploration, we manipulate the context signal at test time to examine how each architecture behaves when context information is absent or ambiguous. Two conditions are evaluated and both are applied by overwriting the context dimensions of the test inputs before they are passed to a model that has been trained under the standard $[10, 0]$ / $[0, 10]$ scheme:

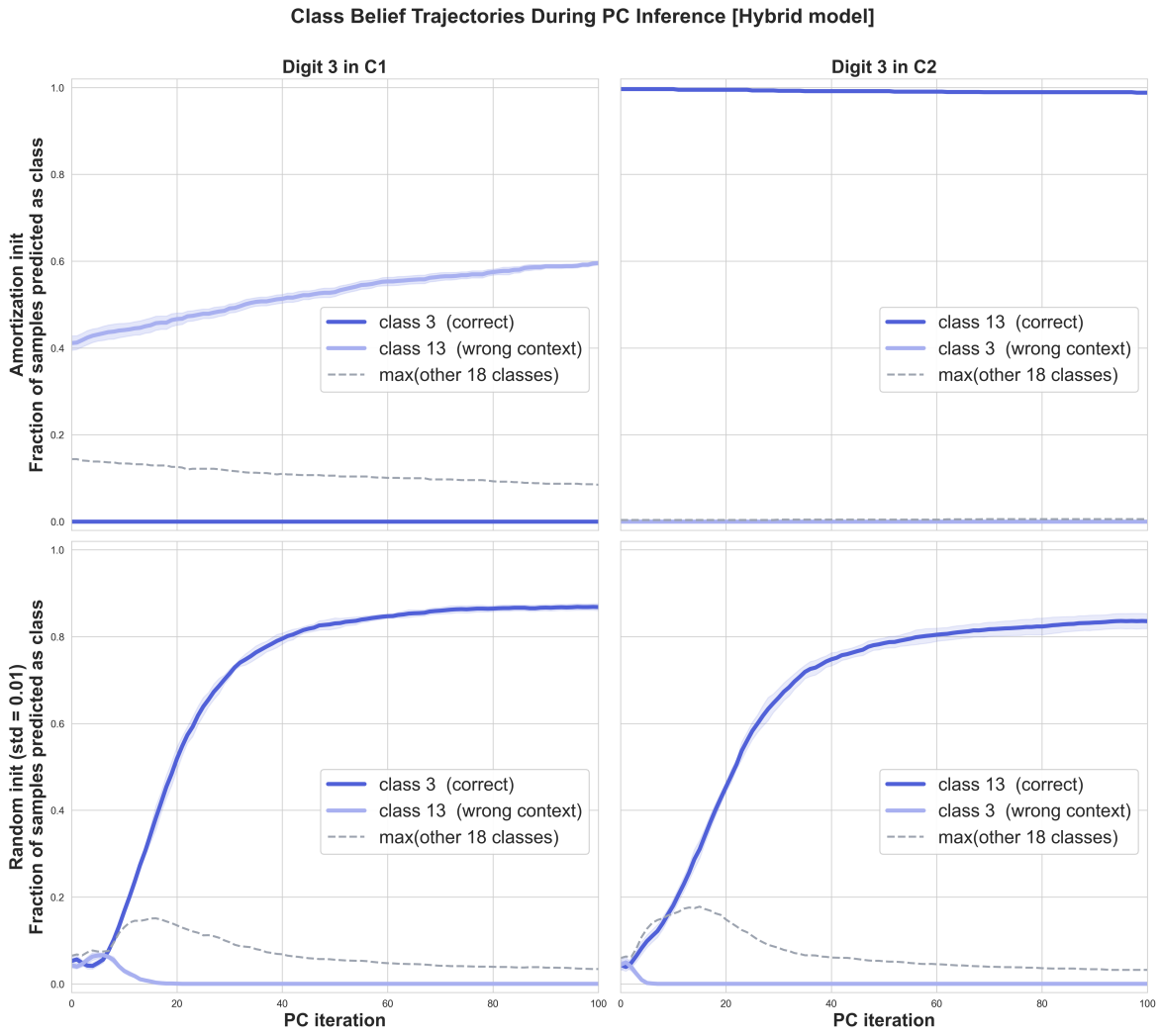

Figure 4.8: Class belief trajectories for digit 3 samples during PC inference in the hybrid model under different initializations and 100 PC iterations. Rows reflect the initialization used: amortization initialization (top row) versus random initialization (bottom row). Columns: digit 3 in C1 (left, correct class 3) versus digit 3 in C2 (right, correct class 13). Solid lines show the fraction of samples assigned to the correct class and to the wrong class (d versus d+10); the dashed line is used to track the maximum fraction held by any of the remaining 18 classes. Under amortization initialization on digit 3 C1 samples (top left), class 13 (wrong context) already wins on ∼42% of samples at iteration 0 and grows to ∼60%, while class 3 (correct) stays at ∼0% throughout. Under random initialization on the same samples (bottom left), the trajectory reverses: class 3 grows from ∼5% to ∼87% while class 13 stays near 0%. For digit 3 C2 samples (right column), where the training-distribution bias and the ground-truth label agree, class 13 dominates under both initializations.

Initialization	Context	Perfect	Context error	Digit error	Both wrong
Random std=0.01	C1	$82.65 \pm 0.80\%$	0.00%	$17.35 \pm 0.80\%$	0.00%
	C2	$87.88 \pm 1.01\%$	0.00%	$12.12 \pm 1.01\%$	0.00%
General prior	C1	0.00%	$76.40 \pm 1.35\%$	$23.60 \pm 1.35\%$	0.00%
	C2	$98.18 \pm 0.15\%$	0.00%	$1.82 \pm 0.15\%$	0.00%

Table 4.5: The comparison of generative model performance on digit 3 samples under PC inference with different initializations (baseline std=0.01 and general prior). Values are mean ± standard deviation across 5 random seeds. The general prior used here is taken from the **hybrid** model, not the generative model's own prior.

- $[0, 0]$ **(no context signal):** both context dimensions are set to zero. The model receives a digit image with no contextual information.

- $[10, 10]$ **(ambiguous context signal):** both context dimensions are set to 10. Because training used $[10, 0]$ for C1 and $[0, 10]$ for C2, the $[10, 10]$ input provides equally strong evidence for both contexts simultaneously.

Once the presented versions of a given digit become the same input after the original context signal is overwritten, the question shifts from context accuracy to context preference. When context signal is absent or ambiguous, which interpretation does each architecture default to? In this setup, we retain digit identity as the only quantity that still has ground truth, and for each digit 3 sample we report the fraction assigned to class 3 (the C1 reading), class 13 (the C2 reading), or some other class.

Table 4.6 and Figure 4.9 summarize digit 3 final class predictions across the three signal conditions. The results show that the dominant pattern is that the C2 reading of digit 3 (class 13) is the default across nearly every architecture and condition, while the C1 reading (class 3) is essentially unreachable for the two architectures that use amortization.

Both hybrid and feedforward architectures assign 0.00% of digit 3 samples to class 3 under every signal condition. For the hybrid model, class 13 accounts for 79.19% of digit 3 samples under original context signals, rising to 86.04% under $[0, 0]$ and falling to 68.02% under $[10, 10]$. The feedforward model shows even more commitment under $[10, 10]$, where class 13 climbs to 95.45%. In no condition does either of these two architecture produce the C1 reading of the digit 3.

On the other hand, the generative model under original context splits digit 3 samples near-evenly between the two readings (41.29% class 3, 43.88% class 13). This near-50/50 split reflects its unbiased random-initialization prior and its ability to follow the context signal in both directions, and it is the pooled view of the balanced per-context performance reported in Section 4.4.1 (82.65% C1, 87.88% C2 perfect classification)

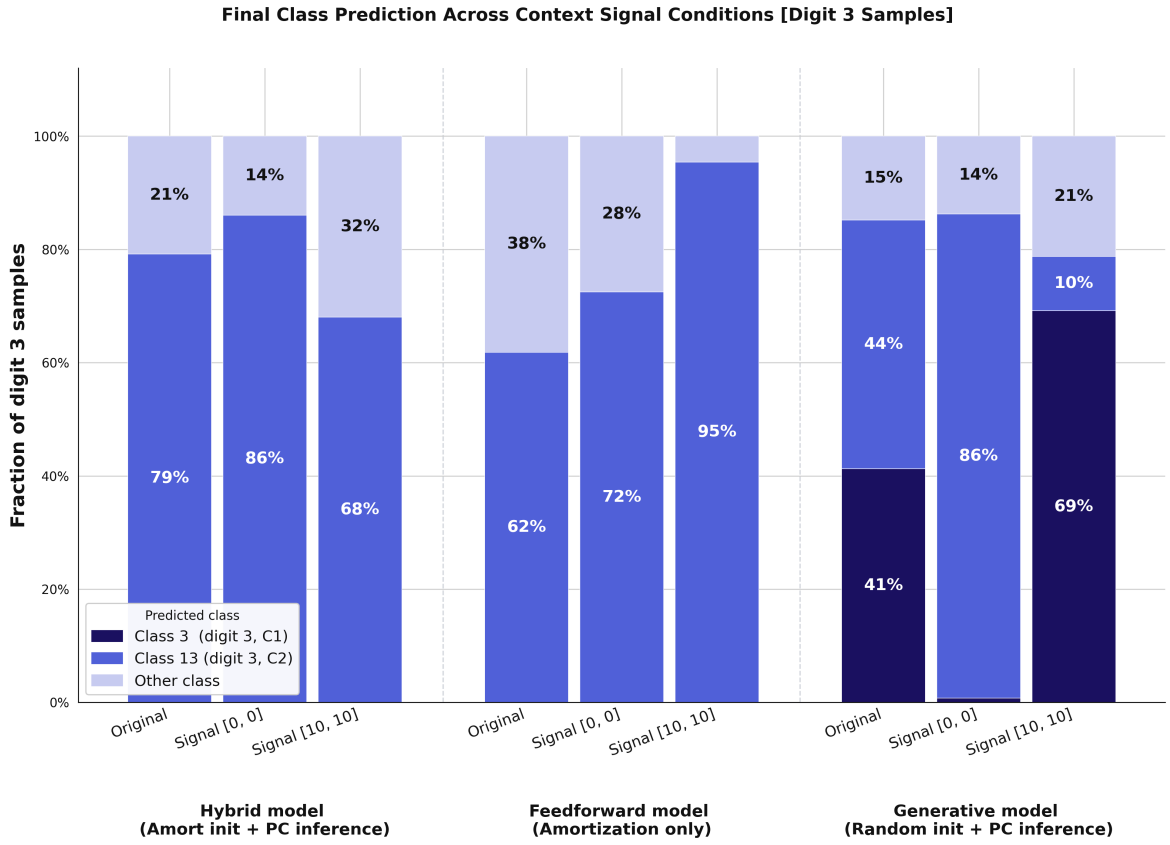

Figure 4.9: Final class predictions for digit 3 samples across the three architectures and the three context signal conditions. Bars show the fraction of digit 3 samples assigned to class 3 (the C1 reading), class 13 (the C2 reading), or any other class. Training used original $[10, 0]$ for C1 and $[0, 10]$ for C2; the test set is balanced (50% C1 / 50% C2 per digit). The C2 reading dominates across almost every architecture and condition; only the generative model has assigned some to the correct C1 reading under original and $[10, 10]$ context signal.

Model	Signal	Class 3 (C1)	Class 13 (C2)	Other
Hybrid	Original (baseline)	0.00%	$79.19 \pm 0.32\%$	20.81%
	$[0, 0]$	0.00%	86.04%	13.96%
	$[10, 10]$	0.00%	68.02%	31.98%
Feedforward	Original (baseline)	0.00%	$61.84 \pm 0.70\%$	38.16%
	$[0, 0]$	0.00%	72.48%	27.52%
	$[10, 10]$	0.00%	95.45%	4.55%
Generative	Original (baseline)	$41.29 \pm 0.39\%$	$43.88 \pm 0.74\%$	14.83%
	$[0, 0]$	$0.75 \pm 0.18\%$	$85.52 \pm 0.50\%$	13.72%
	$[10, 10]$	$69.15 \pm 0.91\%$	$9.62 \pm 0.85\%$	21.23%

Table 4.6: Digit 3 final class prediction rates across context signal conditions, by model. Balanced test set: 50% C1 / 50% C2 per digit. Baseline uses the original context signal. Mean ± std across 5 seeds (std omitted where $< 0.01\%$). Full per-digit breakdowns for all three architectures are in Appendix C.3.

rather than a new result. Contrary, removing the original context signal damage this balance. Under $[0, 0]$ the C1 reading collapses to 0.75% and class 13 rises to 85.52%, essentially matching the hybrid model's $[0, 0]$ default of 86.04%. Removing the context signal therefore causes the generative model to default to the C2 reading regardless of the underlying digit presentation. Under $[10, 10]$ the direction reverses, where the generative model now favors the C1 reading (69.15% class 3, 9.62% class 13).

4.5.1 Class belief dynamics under different initializations in hybrid model

To see how hybrid models class predictions changes using different initializations, we further track the fraction of digit 3 samples assigned to class 3 and class 13 using amortization initialization and random initialization.

In the Figure 4.10 we can see that under amortization initialization (left column), the C2 reading (class 13) dominates from iteration 0 and barely changes across 100 PC iterations, in both the $[0, 0]$ (top) and $[10, 10]$ (bottom) conditions making the model to prefer class 13 from the start. The C1 reading (class 3) stays at 0% throughout in both conditions. This is the same behavior we observed under clear context in Section 4.4.2, where amortization initialization locked the model onto class 13 and additional PC iterations only deepened that commitment. On the other hand, under random std=0.01 initialization (right column), the picture is qualitatively different. The model starts near uniform and the C2 reading emerges over the first $\sim$40 iterations before plateauing.

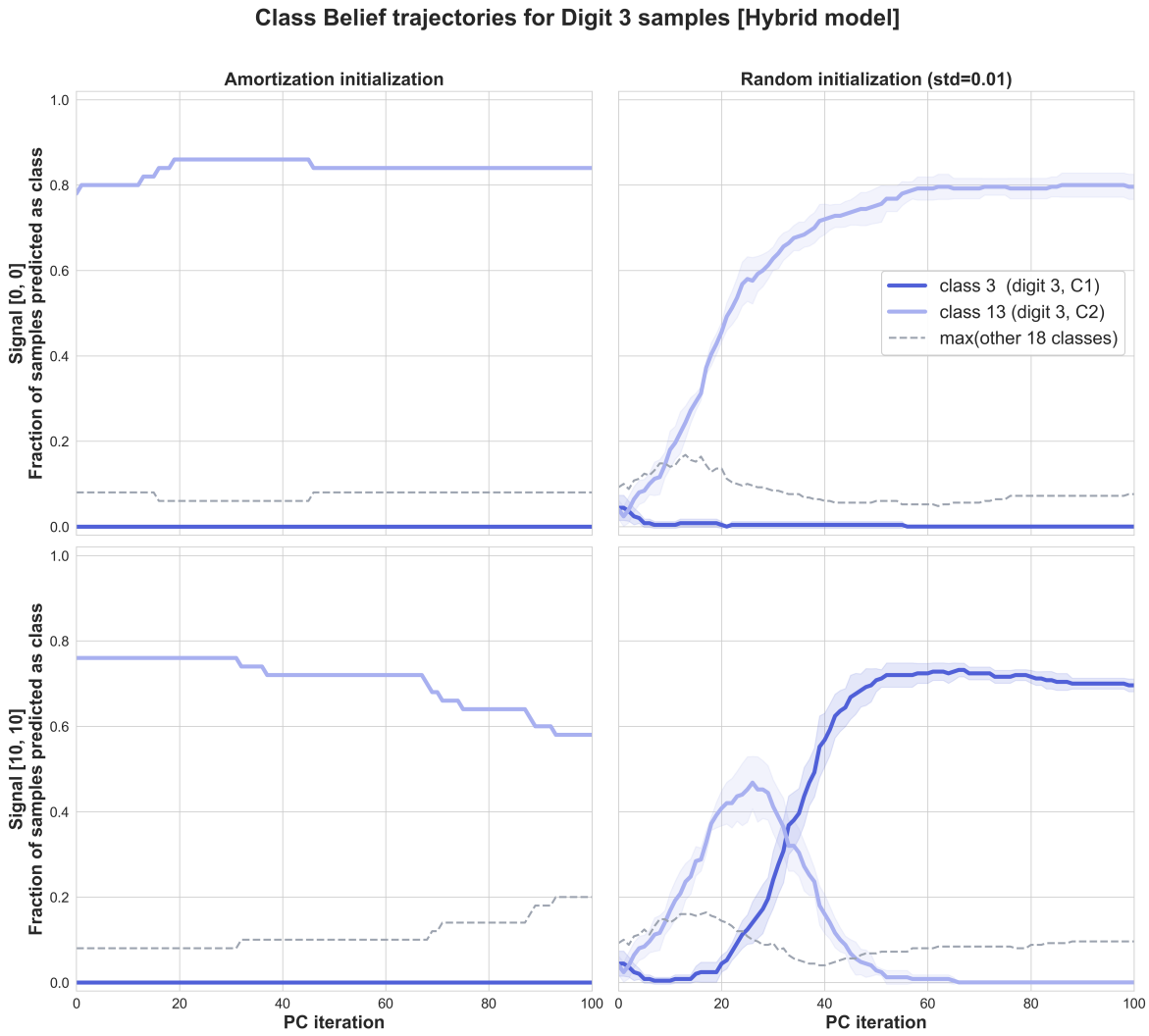

Figure 4.10: Class-belief trajectories for digit 3 samples in the hybrid model across 100 PC iterations. Columns refer to different initializations: amortization initialization (left) versus random initialization (right). Rows: $[0, 0]$ context signal (top) versus $[10, 10]$ (bottom). Solid lines show the fraction of samples assigned to class 3 (the C1 reading) and class 13 (the C2 reading); the dashed line tracks the maximum fraction held by any of the remaining 18 classes. Lines show the mean across 5 random seeds with shaded standard deviation.

On the other hand, under random std=0.01 initialization (right column), the picture is qualitatively different, and the two context conditions now diverge. The model starts near uniform in both rows. Under $[0, 0]$ (top), the C2 reading emerges over the first $\sim$40 iterations and plateaus near 0.80, while the C1 reading stays close to 0%, so the random-init endpoint matches the amortization-init endpoint even though it is reached by a different path. Under $[10, 10]$ (bottom), however, the preference reverses and the C2 reading rises first and peaks near iteration 25, but is then overtaken by the C1 reading, which crosses above it near iteration 35 and grows to dominance ($\sim$0.70) by the end of inference, while the C2 reading collapses toward 0%.

Taken together, these results show that the hybrid model with amortisation initialization favors the class 13 under different context signals while the preference of initialization from small random noise depends on concrete context signal option.

4.5.2 Summary of findings

Model	Context	Perfect	Context error	Digit error	Both wrong
Hybrid	C1	0.00%	59.56±0.46%	40.44±0.46%	0.00%
	C2	98.81±0.25%	0.00%	0.00%	1.19±0.25%
Feedforward	C1	0.00%	24.16±1.27%	75.84±1.27%	0.00%
	C2	99.52±0.20%	0.00%	0.00%	0.48±0.20%
Generative	C1	82.65±0.77%	0.00%	17.43±0.77%	0.00%
	C2	87.88±1.48%	0.00%	12.24±1.48%	0.00%

Table 4.7: Digit 3 outcome breakdown by context and model (balanced test set, n=505 for each context), mean ± std over 5 seeds. Hybrid uses 100 PC iterations from amortization initialization; feedforward uses amortization output only; generative uses 100 PC iterations from random initialization at std=0.01.

Table 4.7 summarizes digit 3 performance across the three architectures and show three patterns.

First, only the generative model achieves balanced context performance on digit 3 C1 samples (82.65% perfect in C1, 87.88% in C2). Both the hybrid and feedforward models achieve 0.00% perfect classification for digit 3 C1 samples, making every digit 3 C1 sample misclassified, either as digit 3 in C2 (context error) or as a different digit in C1 (digit error). The split between these two error types differs by architecture: the feedforward model leans toward digit errors (75.84%), while the hybrid model lean toward context errors (59.56%).

Second, the hybrid model exhibits a stronger context bias than the feedforward model (59.56% versus 24.16% C1$\rightarrow$C2 errors). Adding PC inference to the same amor-

tized output increases the context error rate by a factor of 2.5, despite the fact that PC inference, when accessed via random std=0.01 initialization (Section 4.4.2), supports balanced predictions on the same hybrid weights. The combination of biased amortization initialization with PC dynamics produces worse context performance than amortization alone.

Third, the hybrid and feedforward models show near-perfect performance on digit 3 C2 samples (98.81% and 99.52% perfect, respectively), while the generative model is noticeably lower at 87.76%. The generative model's lower C2 accuracy reflects the tradeoff that comes with not being biased toward C2. It achieves comparable performance in both contexts (82.57% C1 vs. 87.76% C2) rather than ceiling performance in one and floor performance in the other.

Taken together, these results localize the source of the digit 3 C2 bias to the amortization component and its interaction with PC inference. The generative weights are not themselves biased. The generative model produces 0.00% context errors in either direction, and PC inference on the hybrid model's generative weights from random std=0.01 initialization also produces 0.00% context errors (Table 4.4). The bias appears at the amortization output and is amplified, rather than corrected, by PC iterations starting from that output.

Chapter 5

Experiment 2: Healing Experiment

5.1 Simulating recovery from biased experience through sequential training phases

The previous chapter showed that a hybrid predictive coding model trained on imbalanced distribution develops a strong context bias for the overrepresented stimulus. This chapter asks a follow-up question: once such a bias is established, can it be reduced through subsequent exposure to a corrective distribution, and if not, where does the failure lie?

The experiment addresses three questions: i) After 10 epochs of biased training, does subsequent training on a flipped distribution restore balanced performance across both contexts? ii) If the model cannot restore the balanced performance, is this due to the amortization component, the generative component, or the interaction of the two during inference? iii) If it fails, does this failure reflect the loss of learned associations, or is it a failure to access associations the model still represents?

The three training configurations introduced below (hybrid, amortization-only and generative-only) are chosen so that each question can be addressed by comparing the appropriate pair of conditions.

The experimental design consists of two phases. Phase 1 simulates the development of trauma-based delusional beliefs, whereas Phase 2 simulates the process of healing through corrective experience.

5.1.1 Phase 1: Establishing a context bias

To establish the context bias whose reversibility we will test in Phase 2, we train each model on the imbalanced distribution presented in the previous chapter: digit 3 appears 90% in Context 2 and 10% in Context 1, while all other digits (0–2, 4–9) appear 90% in Context 1 and 10% in Context 2. Training proceeds for 10 epochs (epochs 1–10).

Stimulus	Context 1	Context 2	Samples/epoch	Total (10 epochs)
Digit 3	10%	90%	≈6,100	≈61,000
Other digits (0–2, 4–9)	90%	10%	≈53,900	≈539,000
Total			**60,000**	**600,000**

Table 5.1: Phase 1 training distribution. Sample counts are approximate because digit frequencies in MNIST are not exactly equal; exact counts vary by epoch.

5.1.2 Phase 2: Inverted distribution as corrective exposure

After Phase 1, we invert the digit 3 training distribution: for the next 5 epochs (epochs 11–15), digit 3 appears 90% in Context 1 and 10% in Context 2, matching the distribution of all other digits. The distribution for digits other than 3 is unchanged.

5.2 Results

5.2.1 Trajectory of context errors

Each model's context-only errors paired for context accuracy regarding digit 3 samples across Phase 1 (epoch 9, end of biased training) and Phase 2 (epochs 10–14, corrective exposure) is shown in Figures 5.1, 5.2, and 5.3. The three figures use identical layout to make the cross model comparison easier for the reader. Each figure has two subplots. The *left* subplot pairs the C1→C2 error rate with C2 accuracy and the *right* subplot pairs the C2→C1 error rate with C1 accuracy. The two plots together reveals whether a model's bias inverts, partially shifts, or remains stable across the phase transition.

Hybrid model

For the hybrid model, the two subplots in Figure 5.1 mirror each other across the phase boundary. At the end of Phase 1 the model exhibits a strong C2 bias: 0% C1 accuracy paired with 98.7% C2 accuracy, with 56.5% of digit 3 C1 samples misclassified as C2. A single training epoch of inverted digit 3 distribution collapses the C1→C2 rate to 0% and lifts C1 accuracy to 67.5%. But this initial shift does not stabilize. Over the next four epochs C2 accuracy decays steeply from 84.3% to 0.7%, while C2→C1 errors goes up from near zero to 82.4%. The model converges to a state that is the mirror image of where it began showing that the hybrid model does not preserve accuracy on both context simultaneously but rather flip the dominant direction of context confusion. Table 5.2 further reports the underlying counts.

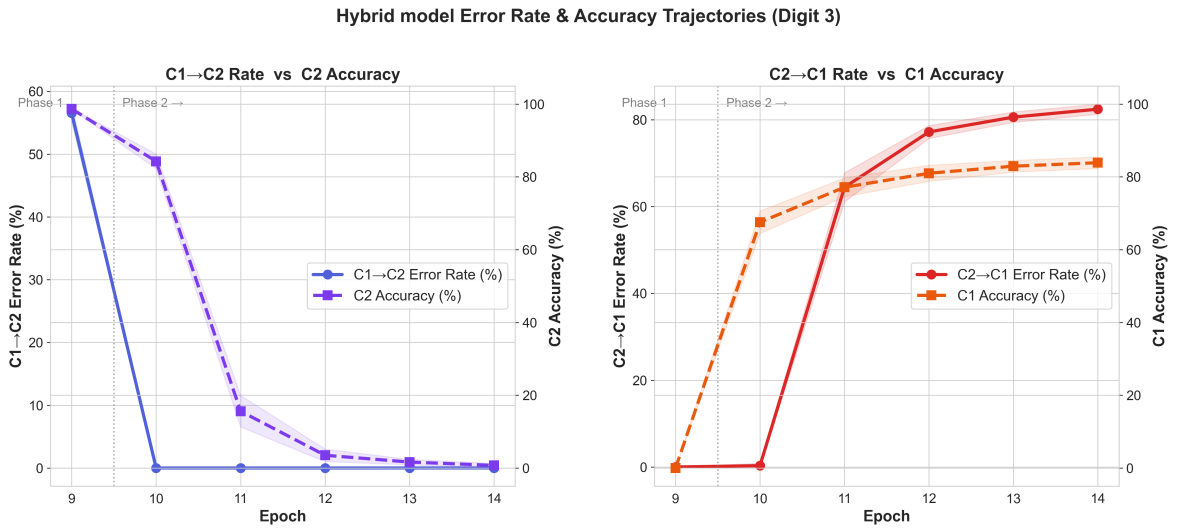

Figure 5.1: Trajectory of digit 3 context errors and complementary context accuracy for the hybrid model across Phase 1 (epoch 9, end of biased training) and Phase 2 (epochs 10–14, inverted digit 3 distribution). Left: C1→C2 error rate (solid) paired with C2 accuracy (dashed). Right: C2→C1 error rate (solid) paired with C1 accuracy (dashed). The two panels mirror each other across the boundary.

Epoch	Phase	C1 Accuracy	C1→C2	C2 Accuracy	C2→C1
9	Phase 1	$0/504$ $(0.0 \pm 0.0\%)$	285.0 ± 6.9	$497/503$ $(98.7 \pm 0.2\%)$	0
10	Phase 2	$341/504$ $(67.5 \pm 3.1\%)$	0	$424/503$ $(84.3 \pm 2.0\%)$	1.6 ± 2.7
11	Phase 2	$389/504$ $(77.1 \pm 2.6\%)$	0	$79/503$ $(15.6 \pm 4.3\%)$	324.4 ± 16.8
12	Phase 2	$408/504$ $(81.0 \pm 2.2\%)$	0	$18/503$ $(3.5 \pm 1.7\%)$	388.6 ± 7.3
13	Phase 2	$418/504$ $(82.9 \pm 1.6\%)$	0	$8/503$ $(1.6 \pm 0.7\%)$	405.8 ± 5.8
14	Phase 2	$423/504$ $(83.9 \pm 1.6\%)$	0	$4/503$ $(0.7 \pm 0.6\%)$	415.0 ± 5.7

Table 5.2: Hybrid inference results (hybrid model) across a 10-epoch Phase 1 followed by a 5-epoch Phase 2 in which the digit 3 training distribution is inverted between contexts. Test set contains digit 3 in balanced distribution (50% C1, 50% C2, $n = 504$ and $n = 503$ respectively). Counts are means; percentages and error counts show mean $\pm$ std across 5 independent runs.

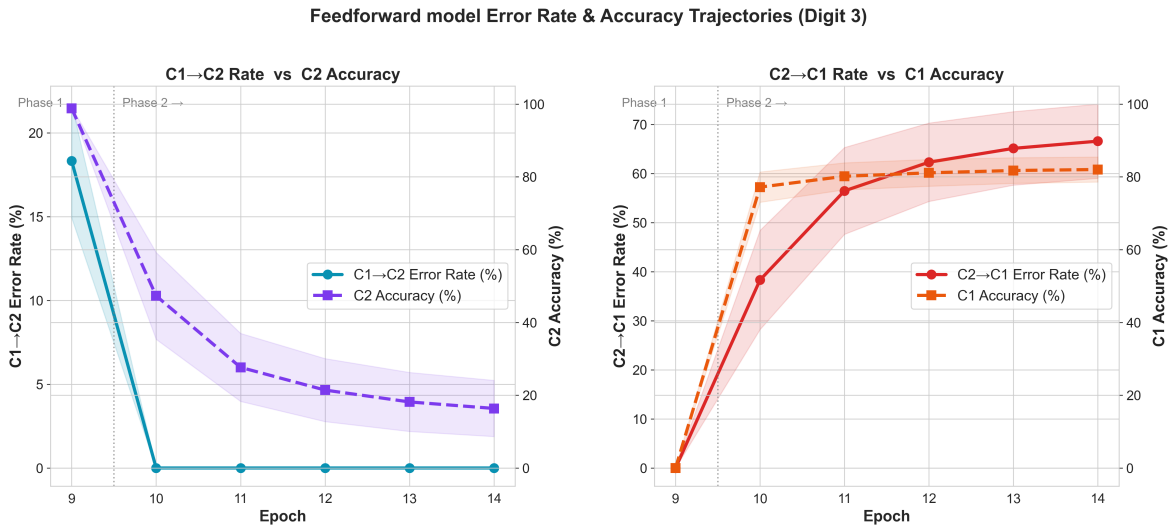

Figure 5.2: Trajectory of digit 3 context errors and complementary context accuracy for the feedforward model (amortization only), with layout identical to Figure 5.1.

Feedforward model

When we trained only the amortization weights, we can see that the qualitative pattern mostly matches the hybrid model but the collapse of C2 accuracy is smaller (Figure 5.2). The last epoch of Phase 1 is practically identical with the hybrid model's performance (0% C1 accuracy, 98.8% C2 accuracy) with a smaller initial C1→C2 rate (18.4% vs. the hybrid model's 56.5%). After the first Phase 2 epoch the C1→C2 rate goes down to 0% and C1 accuracy rises (plateauing around 82%). But C2 accuracy decays only as far as 16.4% by epoch 14, not 0.7%, and the C2→C1 error rate plateaus around 67% rather than the hybrid model's 82%. The bias inverts, as in the hybrid case, but the inverted state retains some C2 accuracy at the end of the Phase 2. More detailed counts we report in Table 5.3.

Epoch	Phase	C1 Accuracy	C1→C2	C2 Accuracy	C2→C1
9	Phase 1	0/504 (0.0 ± 0.0%)	92.4 ± 17.1	497/503 (98.8 ± 0.3%)	0
10	Phase 2	389/504 (77.2 ± 4.2%)	0	238/503 (47.3 ± 11.9%)	192.8 ± 50.7
11	Phase 2	404/504 (80.2 ± 3.7%)	0	139/503 (27.6 ± 9.3%)	284.2 ± 44.4
12	Phase 2	409/504 (81.1 ± 3.7%)	0	108/503 (21.4 ± 8.7%)	313.6 ± 40.1
13	Phase 2	412/504 (81.7 ± 3.5%)	0	91/503 (18.2 ± 8.1%)	327.8 ± 37.6
14	Phase 2	414/504 (82.0 ± 3.4%)	0	82/503 (16.4 ± 7.7%)	335.2 ± 38.0

Table 5.3: Amortization predictions (amortization-only training mode). Test set contains digit 3 in balanced distribution (C1: $n = 504$, C2: $n = 503$). Counts are means; percentages and error counts show mean ± std across 5 independent runs.

Generative model results

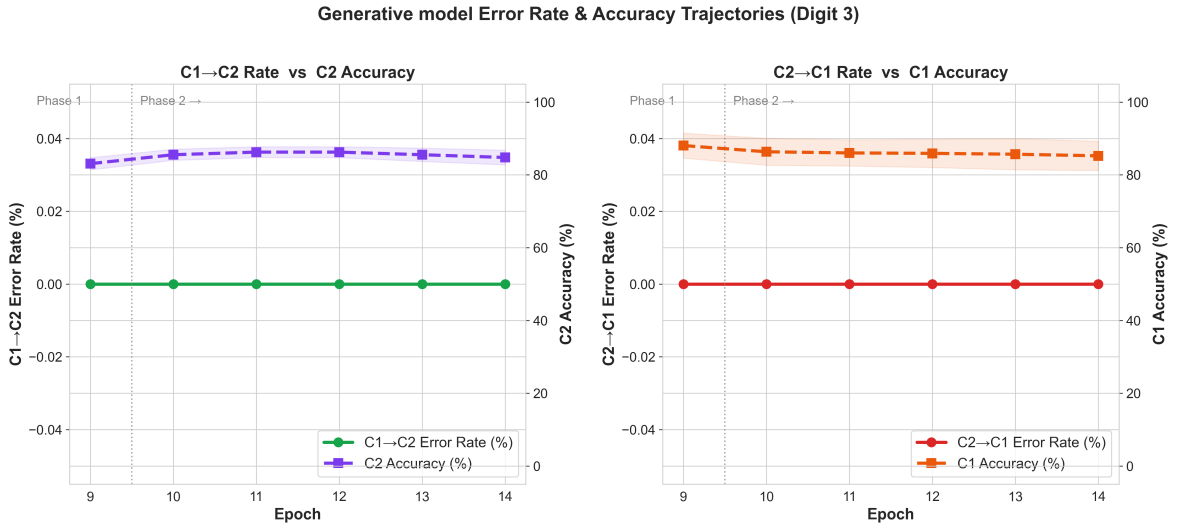

Figure 5.3: Trajectory of digit 3 context errors and complementary context accuracy for the generative model (PC inference from random std=0.01 initialization), with layout identical to Figure 5.1. Left: C1→C2 error rate (solid) with C2 accuracy (dashed). Right: C2→C1 error rate (solid) with C1 accuracy (dashed). In contrast to the other two architectures, context error rates stay at zero in both directions across all epochs (note the error-rate axis spans only ±0.04%), and C1 and C2 accuracy track each other closely between roughly 83% and 88% throughout.

When only the generative weights are trained and the amortization network is bypassed at inference, we can see that the trajectory is qualitatively different from both previous models (Figure 5.3). C1 and C2 accuracy track each other closely between 80% and 88% throughout both phases, and the context error rates remain at zero in both directions across all epochs. Phase 1 produces no asymmetry to begin with, and Phase 2 induces only a 4.0 percentage drop in C1 accuracy and a 2.2 percentage drop in C2 accuracy, which is of the same order as seed-level noise, and an order of magnitude smaller than the 82.4 percentage C2 drop in the feedforward model and the 98.0 percentage C2 drop in the hybrid model. The contrast across the three figures is the central finding of this specific result section: The generative weights, when accessed without amortization, do not develop the asymmetric C2 bias in Phase 1 and do not undergo a bias inversion in Phase 2.

5.3 Effect of different inference modes on Hybrid model

The previous Section 5.2.1 shows that the generative model (trained without amortization network) produces stable, balanced predictions under PC-only inference throughout both phases. This section asks whether the hybrid-trained model preserves the

Epoch	Phase	C1 Accuracy	C2 Accuracy	C1→C2	C2→C1
9	Phase 1	437/504 (86.7 ± 6.1%)	442/503 (87.8 ± 1.7%)	0	0
10	Phase 2	431/504 (85.4 ± 6.9%)	446/503 (88.5 ± 1.4%)	0	0
11	Phase 2	424/504 (84.2 ± 7.8%)	445/503 (88.5 ± 2.2%)	0	0
12	Phase 2	422/504 (83.7 ± 7.9%)	442/503 (87.8 ± 3.1%)	0	0
13	Phase 2	420/504 (83.3 ± 7.4%)	437/503 (86.9 ± 4.3%)	1.0 ± 1.5	0
14	Phase 2	417/504 (82.7 ± 6.3%)	431/503 (85.6 ± 5.7%)	3.8 ± 6.2	0

Table 5.4: Underlying counts for Figure 5.3. PC-only inference with random initialization at std=0.01, 100 iterations. Balanced test set, $n = 504$ for C1 and $n = 503$ for C2. Mean ± std across 5 seeds.

same property we saw while using random initialization and PC-only inference. The hybrid model's generative weights were trained jointly with an amortization network that develops the bias; whether joint training transfers the bias into the generative weights, or leaves them uncontaminated, is our question.

5.3.1 C1→C2 errors and C2 accuracy

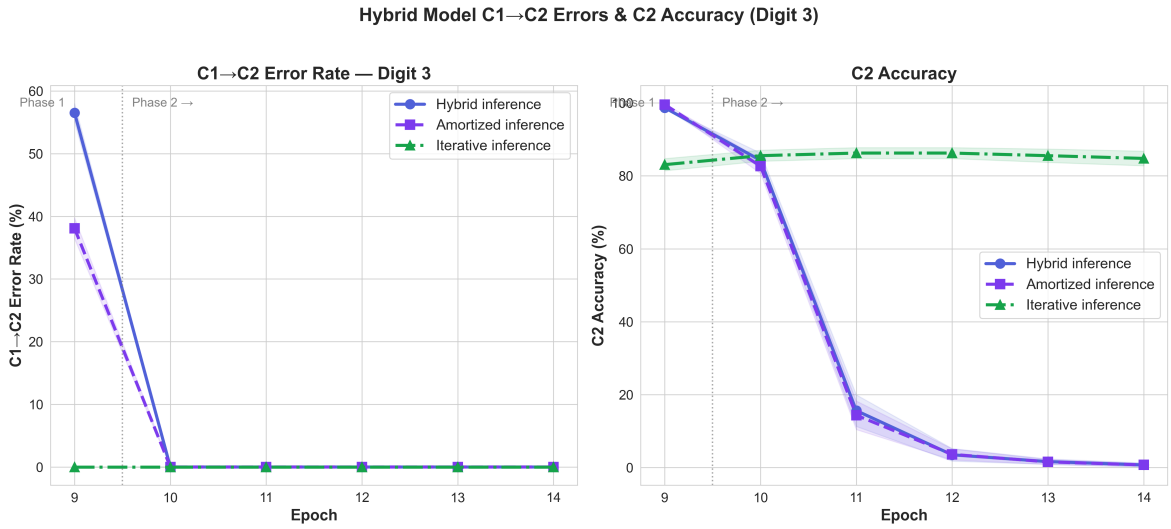

Figure 5.4: C1→C2 error rate (left) and C2 accuracy (right) across Phase 1 (epoch 9) and Phase 2 (epochs 10–14) on the hybrid-trained model, evaluated under three inference modes. The dotted vertical line marks the Phase 1 / Phase 2 boundary. Shaded bands show ±1 std across 5 seeds.

At the end of Phase 1, iterative inference produces a 0% C1→C2 error rate with 83.1% C2 accuracy. The hybrid model's generative weights, when queried in isolation, do not encode the bias and they reproduce the behavior we observed in Section 5.2.1 on the generative model's weights. Despite the generative model's weights having been

trained alongside an amortization network that produces a 56.5% C1→C2 error rate at the same epoch joint training with biased amortization does not transfer its bias into the generative weights.

Looking at the two inference modes dependent on amortization initialization they look very different from iterative inference at this point. Amortized inference produces a 38% C1→C2 error rate with near-ceiling C2 accuracy while hybrid inference, which adds 100 PC iterations on top of that initialization, produces an even higher 56.5% error rate with 98.7% C2 accuracy.

After the first Phase 2 epoch the C1→C2 rates of all three modes converge to 0% and remain there. Regarding C2 accuracy trajectory, hybrid and amortized inference collapse together from 98–99% to below 1% by epoch 14. Iterative inference holds between 83% and 86% across both phases, tracking the same range it occupied at the end of Phase 1.

5.3.2 C2→C1 errors and C1 accuracy

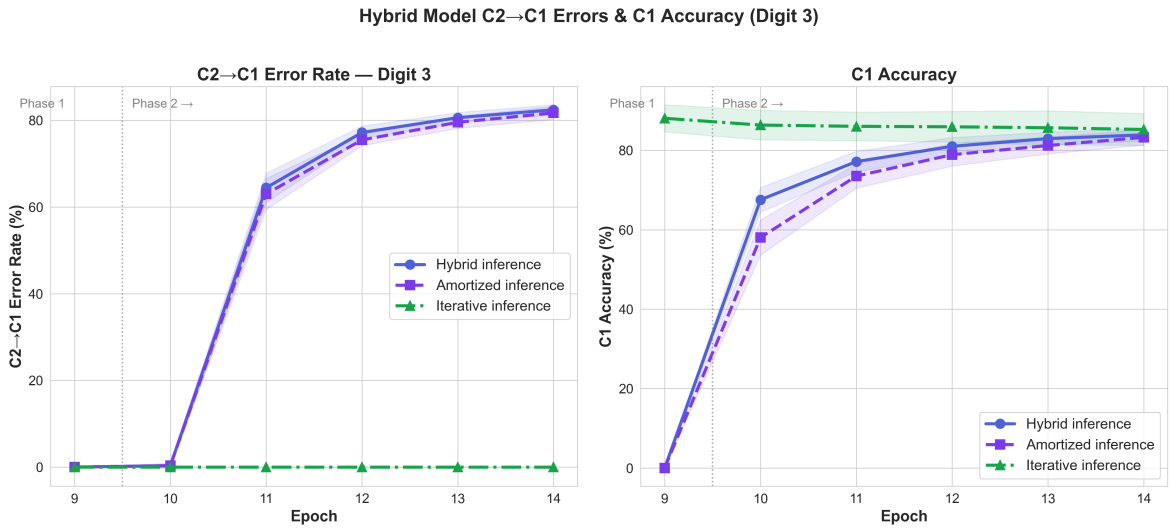

Figure 5.5: C2→C1 error rate (left) and C1 accuracy (right) across Phase 1 (epoch 9) and Phase 2 (epochs 10–14) on the hybrid-trained model, evaluated under three inference modes. Conventions as in Figure 5.4.

The same qualitative pattern of dissociation appears on the other side of the bias (Figure 5.5). Iterative inference shows no trajectory at all keeping C1 accuracy steady at 85–88% across all epochs while the C2→C1 rate stays at 0% throughout. The hybrid model's generative weights continue to support both contexts across Phase 2 just as the generative model's weights did in Section 5.2.1.

Hybrid and amortized inference look again entirely different. At epoch 9 both produce 0% C1 accuracy and 0% C2→C1 errors. After the first Phase 2 epoch, C1

accuracy rises under both inference modes (58% under amortized inference and 67% under hybrid inference). By epoch 11 we can see that the C2→C1 rate goes from near zero to roughly 64% and proceeds to 80–82% by epoch 14.

Chapter 6

Experiment 3: Bias Transfer Experiment

6.1 Motivation and research question

The previous chapter showed that an imbalanced training distribution for a stimulus (digit 3) produces a context bias for that stimulus, and that the bias is encoded in amortized inference rather than in the generative weights.

This chapter asks different question: does an environmental distribution affect predictions for a stimulus whose own training is balanced toward the opposite context? Concretely, if a model is trained in an environment where almost every stimulus appears in the threatening context, does it also predict a specific stimulus (digit 3) as threatening, even when 3's own training distribution placed it predominantly in the benign context (C1) during training?

Two possible outcomes i) If environmental bias transmits to digit 3, this would suggest that amortized inference develops a generic context prior that overrides stimulus specific associations. The amortization network would have learned that this environment is threatening as a default, and would apply it even to stimuli whose own training distribution contradicts it. ii) If environmental bias does not transmit to digit 3, this would suggest that amortized inference learns stimulus-specific associations that resist environmental override. Digit 3's own training distribution would be preserved as a separate mapping.

6.2 Experimental design

We use a controlled two-condition design. The conditions differ only in the training distribution of non-3 digits across contexts; digit 3's distribution is held identical in

both conditions.

- **Experimental condition (threatening environment):** Digits 0–2 and 4–9 appear 90% in C2 (threatening), 10% in C1 (benign).
- **Control condition (benign environment):** Digits 0–2 and 4–9 appear 90% in C1 (benign), 10% in C2 (threatening).
- **Digit 3 identical in both conditions:** 90% in C1, 10% in C2.

6.3 Hypotheses

- **Null hypothesis:** No difference in digit 3 C1→C2 rate between conditions (no environmental influence on digit 3).
- **H1:** Digit 3 C1→C2 rate is higher in the experimental condition than in the control condition (environmental influence on digit 3).

We are testing these hypotheses by using paired t-tests with seed-matched pairs ($n = 10$ per condition, df $= 9$) and report Cohen's d for effect size.

6.4 Results

This section presents the findings from the bias transfer experiment, which examines whether the broader environmental context can influence predictions for specific stimuli that were trained with opposing context distribution.

6.4.1 Primary hypothesis test

The central hypothesis tested whether hybrid model trained in a predominantly threatening environment would exhibit bias toward threatening predictions when processing digit 3 that was predominantly associated with benign context during training.

Table 6.1 reports the C1→C2 error rate for digit 3 in three inference modes and under both conditions.

Digit 3 shows 0.00% C1→C2 errors in both conditions across all three inference modes and all 10 seeds. Because digit 3's own training distribution is identical across conditions, any condition-dependent difference in its predictions would have to come from the surrounding environmental distribution but there is no such difference.

Looking at the results, the null hypothesis cannot be rejected. The data is consistent with the view where digit 3's predicted context is determined by its own training distribution, regardless of how the surrounding stimuli were distributed during training.

Inference Mode	**Experimental Mean ± SD**	**Control Mean ± SD**	**Difference (pp)**	t**(9)**	p**-value**	**Cohen's** d
Amortized	0.00 ± 0.00	0.00 ± 0.00	0.00	—	—	—
Hybrid	0.00 ± 0.00	0.00 ± 0.00	0.00	—	—	—
PC-only	0.00 ± 0.00	0.00 ± 0.00	0.00	—	—	—

Table 6.1: Digit 3 C1→C2 error rates across inference modes and conditions. Values are means ± standard deviation across 10 paired models.

The threatening environment in the experimental condition does not pull digit 3's predictions toward C2.

6.4.2 Confirmation of the experimental manipulation

Table 6.2 report C1→C2 error rates for all digits under amortized and hybrid inference.

	Amortized				**Hybrid**			
Digit	**Exp. (%)**	**Ctrl. (%)**	$t(9)$	**Cohen's** d	**Exp. (%)**	**Ctrl. (%)**	$t(9)$	**Cohen's** d
0	74.25	0.00	184.08	82.32	86.62	0.00	341.86	152.88
1	80.33	0.00	155.24	69.43	90.79	0.00	250.04	111.82
2	38.44	0.00	54.43	24.34	55.58	0.00	95.64	42.77
3	**0.00**	**0.00**	—	—	**0.00**	**0.00**	—	—
4	53.15	0.00	86.97	38.90	75.20	0.00	259.06	115.86
5	11.99	0.00	20.91	9.35	27.11	0.00	54.35	24.31
6	80.64	0.00	124.75	55.79	89.98	0.00	210.07	93.95
7	62.51	0.00	99.41	44.46	77.31	0.00	274.84	122.91
8	25.01	0.00	27.60	12.34	48.10	0.00	47.74	21.35
9	23.71	0.00	18.20	8.14	57.89	0.00	60.79	27.19

Table 6.2: C1→C2 error rates by digit under amortized vs. hybrid inference. All p-values are < 0.001 except for digit 3, where both conditions are zero and the test is not defined. Difference (pp) equals the Exp. value since all Ctrl values are zero. Paired t-tests across 10 seed-matched model pairs.

The non-3 digits show large condition differences in C1→C2 rates: in the experimental condition, rates range from 11.99% (digit 5) to 80.64% (digit 6) under amortized inference, and from 27.11% to 89.98% under hybrid inference. In the control condition, all non-3 digits show 0.00% C1→C2 rates. All differences are statistically significant ($p < 0.001$) with very large effect sizes (Cohen's d from 8.14 to 152.88).

The role of these results is that they confirm that the training manipulation produced models that differ as expected between conditions, and they establish that the

architecture is fully capable of producing context errors at high rates. Against this baseline, digit 3's 0% rate in both conditions is meaningful.

A secondary observation in this table is that hybrid inference increases the per digit C1→C2 rates relative to amortized inference (e.g., digit 0: 74.25% → 86.62%; digit 6: 80.64% → 89.98%). For each non-3 digit, PC iterations starting from amortization output move predictions further toward the more frequent training context rather than away from it. This replicates the observation of the previous chapter, where PC iterations amplify whatever bias the amortization carries.

PC-only inference mode

In the Table 6.2 the results for PC-only inference are ommited because it produces 0.00% C1→C2 error rates for digit 3 and for 8 other digits, in both conditions, across all seeds. The single deviation is digit 0 in the control condition, which shows a 0.16% error rate ($t(9) = -2.23$, $p = 0.053$, Cohen's $d = -1.00$). We treat this deviation as noise rather than as evidence of partial influence, for three reasons: it represents approximately one sample per seed, it is not statistically significant at $\alpha = 0.05$, and the direction is opposite to what an environmental account would predict (control slightly higher than experimental).

The contrast with amortized and hybrid inference is informative. Under those modes, the non-3 digits showed large condition dependent C1→C2 rates that reflected each digit's own training distribution. Under PC-only inference, those condition-dependent differences disappear: every digit produces near-zero rates regardless of training condition. This means that the per stimulus context mappings revealed by amortized and hybrid inference are not encoded in the generative weights because when the generative model is accessed via random initialization PC does not produce context errors.

6.4.3 Feedforward model results

When only the amortization weights are trained and the generative weights remain frozen, the same pattern holds: digit 3 shows 0.00% C1→C2 errors in both conditions, while the non-3 digits show large condition-dependent differences, with experimental-condition rates ranging from 12.28% (digit 5) to 78.05% (digit 6) and control-condition rates of 0.00% across all non-3 digits (all $p < 0.001$, Cohen's d from 4.57 to 19.62). Full per-digit statistics are reported in Appendix D.4.1 (Table 13).

6.4.4 Generative model results

When we look at the results of the generative model, where generative weights are trained and the amortization weights remain at their random initialization, all digits including digit 3 produce near-zero C1→C2 rates under PC-only inference. The largest entry across the 20 cells (10 digits × 2 conditions) is 0.02% (digits 0, 2, and 3, in the control condition), corresponding to roughly one error every five seeds; no comparison reaches statistical significance ($p \geq 0.343$ for the non-zero entries). Full per-digit statistics are reported in Appendix D.4.2 (Table 14).

This mirrors the PC-only finding earlier in this section (Section 6.4.2) on the hybrid-trained model and is consistent with the previous chapter's observation that the generative weights, when accessed via small random initialization (init_std=0.01), do not produce context errors.

Conclusion

Harding et al. (2024) extended the hybrid predictive coding architecture with idea that the amortized network encodes a library of mappings shaped by an individual's experiental history, and they use this framing to ilustrate an account of delusion formation. They are explicit that the proposal is preliminary and call for formal computational modelling of two things in particular: how the amortized mapping is learned and selected, and how the amortized and iterative parts of the system interact during inference. This thesis took a step toward that formalisation. Using the Tschantz et al. (2023) HPC architecture trained on a context augmented MNIST classification task, in which each digit is paired with a benign (C1) or threatening (C2) context signal and imbalanced training distributions stand in as an analogue of imbalanced experiental history, we investigated three computational properties motivated by this account.

The first experiment established that imbalanced experience leaves a visible, stimulus specific mark and that the mark is localised in the amortization component of the hybrid architecture and the iterative inference amplifies it rather than corrects it. For digit 3 the C1→C2 error rate rose from 24.16% under amortization alone to 59.56% once PC inference was run on top of the amortized initialization, and continued climbing to 75.45% at 1000 PC iterations. The cleanest evidence that this is a property of inference and not of the learned generative model came from changing only the starting point. The same hybrid generative weights that produce a near total C2 failure under amortization initialization produce 0.00% context errors and balanced performance under random std=0.01 initialization.

The second experiment sharpened the question of where the failure lives by asking whether an established bias can be undone by corrective experience, in this case flipped distribution for digit 3 samples. Under an inverted Phase 2 distribution the hybrid model did not recover balanced performance but flipped the dominant direction of context confusion, moving from a 56.5% C1→C2 rate at the end of Phase 1 to an 82.4% C2→C1 rate and by the end of Phase 2, converging to the mirror image of where it began. On the other hand, the generative weights behaved entirely different. C1 and C2 accuracy tracked each other between roughly 83% and 88% throughout both phases, with context error rates at zero in either direction. This dissociation held even though those generative weights had been trained jointly with the biased

amortization network, which establishes that joint training does not transfer the bias into the generative weights. The bias is therefore an intrinsic phenomenon to the amortization initialization.

In the third experiment we confirmed that the bias is stimulus specific rather than a general environmental prior. Digit 3, whose own training distribution was held identical across conditions, produced 0.00% C1→C2 errors in both a predominantly threatening and a predominantly benign environment, across all three architectures (hybrid, feed-forward and generative model) and all ten seeds, while the non-3 digits showed large condition dependent rates that followed each digit's own training distribution.

When we look back on the two Harding et al. (2024) targets, mentioned findings are in account's favor, although the support is partial in ways worth stating. Regarding the interaction between the two inference components, the results are clear. Amortization carries biases shaped by the statistics of experience, and PC inference initialized from a biased amortized output cannot escape it but instead amplifies it. Regarding the learning and selection of the amortized mapping, the picture is more nuanced. The present design speaks to how the library is formed by experience, since imbalanced training produces stimulus-specific biases localised in the amortized network, but it does not speak to how a mapping is selected from the library at inference, because in our task the context is given to the model as an observed input rather than something it has to infer. The triggering mechanism that Harding et al. (2024) place at the heart of proposal, namely the misapplication of a mapping under a perturbation in the selection process, is therefore not something the present model could itself enact. A further point worth noting is that in Harding et al. (2024), a misapplied amortized guess persists uncorrected specifically because prodromal priors are weakened but in our model, the amortized bias persists without any such manipulation. PC inference initialized from a biased amortized output cannot escape it even under fully intact dynamics of the hybrid model. This is closer to the local minima self-reinforcement that Harding et al. (2024) also propose than to the prodromal perturbation. After this clarification, the three stakes we set out at the start of the thesis in Section 2.2.4 can now be closed in turn: i) stimulus specific bias was produced, so the library-of-mappings framing retains its accountability for the formation of mappings, ii) the wrong amortized output was not corrected by iterative inference but rather amplified by it, so the account of the fixedness of delusional beliefs retains its mechanism, and finally, iii) the bias did not transfer across stimuli.

Moving forward, there are several other limitations that bound the scope of the claims made in this thesis. First, for our purposes we used MNIST toy domain which is not a model of clinical delusions. Also the terminology of trauma, healing, threatening and benign contexts that recurs throughout is just a deliberate analogy to the framing in Harding et al. (2024) rather than the actual clinical description. The find-

ings therefore should be read as tests of computational properties the account requires, and not as demonstrations about patients.

To fully capture the account Harding proposed, future work could take several directions. Making context a latent variable that the model must infer rather than observe as an input would engage Harding et al. (2024) selection target directly, since it would allow the model to commit the misapplication errors. Additionally, manipulating the precision of priors during inference would let the prodromal disturbance route to fixedness be tested alongside the local minima route engaged with here.

To conclude, the central contribution of this thesis is to show that the fixedness Harding et al. (2024) attributes to delusion can be reproduced in a trained HPC model as an initialization and amplification dynamic. It is a property of where inference begins and the two inference modes interact.

Bibliography

Adams, R. A., Stephan, K. E., Brown, H. R., Frith, C. D., and Friston, K. J. (2013). The computational anatomy of psychosis. *Frontiers in Psychiatry*, 4:47.

Ahissar, M. and Hochstein, S. (2004). The reverse hierarchy theory of visual perceptual learning. *Trends in Cognitive Sciences*, 8(10):457–464.

American Psychiatric Association (2013). *Diagnostic and Statistical Manual of Mental Disorders*. 5 edition.

Bailey, T., Alvarez-Jimenez, M., Garcia-Sanchez, A. M., Hulbert, C., Barlow, E., and Bendall, S. (2018). Childhood trauma is associated with severity of hallucinations and delusions in psychotic disorders: A systematic review and meta-analysis. *Schizophrenia Bulletin*, 44(5):1111–1122.

Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., and Friston, K. J. (2012). Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711.

Beal, M. J. (2003). *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University College London.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Buckley, C. L., Kim, C. S., McGregor, S., and Seth, A. K. (2017). The free energy principle for action and perception: A mathematical review. *Journal of Mathematical Psychology*, 81:55–79.

Carlson, T., Tovar, D. A., Alink, A., and Kriegeskorte, N. (2013). Representational dynamics of object vision: The first 1000 ms. *Journal of Vision*, 13(10):1–1.

Clark, A. (2015). *Surfing Uncertainty: Prediction, Action, and the Embodied Mind*. Oxford University Press.

Corlett, P. R. and Fletcher, P. C. (2021). Modelling delusions as temporally-evolving beliefs (commentary on coltheart and davies). *Cognitive Neuropsychiatry*, 26(4):231–241.

Corlett, P. R., Horga, G., Fletcher, P. C., Alderson-Day, B., Schmack, K., and Powers, A. R. (2019). Hallucinations and strong priors. *Trends in Cognitive Sciences*, 23(2):114–127.

Croft, J., Heron, J., Teufel, C., et al. (2019). Association of trauma type, age of exposure, and frequency in childhood and adolescence with psychotic experiences in early adulthood. *JAMA Psychiatry*, 76(1):79–80.

Croft, J., Teufel, C., Heron, J., et al. (2022). A computational analysis of abnormal belief updating processes and their association with psychotic experiences and childhood trauma in a uk birth cohort. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, 7(7):725–734.

Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. (1995). The Helmholtz machine. *Neural Computation*, 7(5):889–904.

Fletcher, P. C. and Frith, C. D. (2009). Perceiving is believing: A bayesian approach to explaining the positive symptoms of schizophrenia. *Nature Reviews Neuroscience*, 10:48–58.

Fox, C. W. and Roberts, S. J. (2012). A tutorial on variational bayesian inference. *Artificial Intelligence Review*, 38(2):85–95.

Friston, K. (2005). A theory of cortical responses. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1456):815–836.

Friston, K. (2008). Hierarchical models in the brain. *PLoS Computational Biology*, 4(11):e1000211.

Friston, K. and Kiebel, S. (2009). Predictive coding under the free-energy principle. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521):1211–1221.

Gershman, S. and Goodman, N. (2014). Amortized inference in probabilistic reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36.

Harding, J. N., Wolpe, N., Brugger, S. P., Navarro, V., Teufel, C., and Fletcher, P. C. (2024). A new predictive coding model for a more comprehensive account of delusions. *The Lancet Psychiatry*, 11(4):295–302.

Hasson, U., Yang, E., Vallines, I., Heeger, D. J., and Rubin, N. (2008). A hierarchy of temporal receptive windows in human cortex. *Journal of Neuroscience*, 28(10):2539–2550.

Hohwy, J. (2013). *The Predictive Mind*. Oxford University Press.

Jensen, G. (2022). Delusion and reason: An argument for a phenomenological model for understanding schizophrenic delusion. *Schizophrenia Bulletin*, page sbac185.

Keysers, C., Xiao, D., Földiák, P., and Perrett, D. I. (2001). The speed of sight. *Journal of Cognitive Neuroscience*, 13(1):90–101.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes.

Knill, D. C. and Pouget, A. (2004). The bayesian brain: The role of uncertainty in neural coding and computation. *Trends in Neurosciences*, 27(12):712–719.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Millidge, B., Tschantz, A., Seth, A., and Buckley, C. (2021). Neural Kalman filtering.

Mishara, A. L. (2010). Klaus conrad (1905–1961): Delusional mood, psychosis, and beginning schizophrenia. *Schizophrenia Bulletin*, 36(1):9–13.

Rao, R. P. N. and Ballard, D. H. (1999). Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286.

Sass, L. and Byrom, G. (2015). Phenomenological and neurocognitive perspectives on delusions: A critical overview. *World Psychiatry*, 14(2):164–173.

Sips, R. (2019). Psychosis as a dialectic of aha- and anti-aha-experiences. *Schizophrenia Bulletin*, 45(5):952–955.

Teufel, C. and Fletcher, P. C. (2016). The promises and pitfalls of applying computational models to neurological and psychiatric disorders. *Brain*, 139(10):2600–2608.

Thorpe, S., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, 381(6582):520–522.

Thunell, E. and Thorpe, S. J. (2019). Memory for repeated images in rapid-serial-visual-presentation streams of thousands of images. *Psychological Science*, 30(7):989–1000.

Tschantz, A., Millidge, B., Seth, A. K., and Buckley, C. L. (2023). Hybrid predictive coding: Inferring, fast and slow. *PLOS Computational Biology*, 19(8):e1011280.

van Bergen, R. S. and Kriegeskorte, N. (2020). Going in circles is the way forward: The role of recurrence in visual inference.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.

Appendix

A.1 The source code

The source code is available at `https://github.com/alibdnrc/hybridpc-delusions`.

B.2 Per-digit context confusion patterns across three architectures

This appendix reports the full per-digit context confusion breakdowns for the three architectures evaluated in Section 4.4.1 on the balanced test set. Tables 3, 4, and 5 correspond to the hybrid, feedforward, and generative models respectively.

Digit	C1 Samples	C1→C2 error	C2 Samples	C2→C1 error
0	490	0.00% (0)	490	424.0 ± 3.8 (86.53 ± 0.77%)
1	568	0.00% (0)	567	513.6 ± 3.4 (90.58 ± 0.60%)
2	516	0.00% (0)	516	280.6 ± 6.6 (54.38 ± 1.28%)
3	**505**	300.8 ± 2.3 **(59.56 ± 0.46%)**	**505**	**0.00% (0)**
4	491	0.00% (0)	491	370.6 ± 5.9 (75.48 ± 1.21%)
5	446	0.00% (0)	446	123.8 ± 4.6 (27.76 ± 1.04%)
6	479	0.00% (0)	479	431.4 ± 2.1 (90.06 ± 0.43%)
7	514	0.00% (0)	514	390.4 ± 5.7 (75.95 ± 1.10%)
8	487	0.00% (0)	487	244.4 ± 9.2 (50.18 ± 1.88%)
9	505	0.00% (0)	504	294.8 ± 11.6 (58.49 ± 2.31%)

Table 3: Per-digit context confusion pattern (hybrid model, 100 PC iterations from amortization initialization). C1→C2 rate: errors among C1 samples where the digit is correctly identified but the context is wrong. C2→C1 rate: same, among C2 samples. Values are mean ± std across 5 seeds.

Digit	C1 Samples	C1→C2 error	C2 Samples	C2→C1 error
0	490	0.00% (0)	490	316.4 ± 4.6 (64.57 ± 0.94%)
1	568	0.00% (0)	567	311.2 ± 5.3 (54.89 ± 0.94%)
2	516	0.00% (0)	516	177.2 ± 6.4 (34.34 ± 1.25%)
3	**505**	122.0 ± 6.4 **(24.16 ± 1.27%)**	**505**	**0.00% (0)**
4	491	0.00% (0)	491	224.8 ± 12.6 (45.78 ± 2.57%)
5	446	0.00% (0)	446	70.0 ± 4.3 (15.70 ± 0.97%)
6	479	0.00% (0)	479	371.8 ± 9.8 (77.62 ± 2.04%)
7	514	0.00% (0)	514	292.0 ± 9.4 (56.81 ± 1.83%)
8	487	0.00% (0)	487	74.4 ± 2.0 (15.28 ± 0.40%)
9	505	0.00% (0)	504	177.0 ± 8.5 (35.12 ± 1.69%)

Table 4: Per-digit context confusion pattern (feedforward model, balanced test). Values are mean ± std across 5 seeds.

Digit	C1 Samples	C1→C2 error	C2 Samples	C2→C1 error
0	490	0.00% (0)	490	0.00% (0)
1	568	0.00% (0)	567	0.00% (0)
2	516	0.00% (0)	516	0.00% (0)
3	**505**	**0.00% (0)**	**505**	**0.00% (0)**
4	491	0.00% (0)	491	0.00% (0)
5	446	0.00% (0)	446	0.00% (0)
6	479	0.00% (0)	479	0.00% (0)
7	514	0.04 ± 0.08% (0.2 ± 0.4)	514	0.00% (0)
8	487	0.00% (0)	487	0.00% (0)
9	505	0.00% (0)	504	0.00% (0)

Table 5: Per-digit context confusion pattern (generative model, 100 PC iterations from random initialization at std=0.01, balanced test). Values are mean ± std across 5 seeds.

C.3 Evaluation under unclear context signal

C.3.1 Hybrid model results

[0, 0] context signal

Digit	Class C1	Class C2	Other
0 (C1=0, C2=10)	$94.29 \pm 0.46\%$	0.00%	5.71%
1 (C1=1, C2=11)	$69.07 \pm 0.46\%$	0.00%	30.93%
2 (C1=2, C2=12)	$81.10 \pm 0.34\%$	0.00%	18.90%
3 (C1=3, C2=13)	**0.00%**	**86.04 ± 0.67%**	**13.96%**
4 (C1=4, C2=14)	$84.11 \pm 0.58\%$	0.00%	15.89%
5 (C1=5, C2=15)	$18.27 \pm 0.48\%$	0.00%	81.73%
6 (C1=6, C2=16)	$91.96 \pm 0.43\%$	0.00%	8.04%
7 (C1=7, C2=17)	$81.42 \pm 0.64\%$	0.00%	18.58%
8 (C1=8, C2=18)	$94.05 \pm 0.37\%$	0.00%	5.95%
9 (C1=9, C2=19)	$70.66 \pm 1.79\%$	0.00%	29.34%

Table 6: Per-digit final class prediction rates for the **Hybrid** model under context signal $[0, 0]$. C1 = correct context 1 class (digit index), C2 = correct context 2 class (digit + 10), Other = all remaining classes. Mean ± std across 5 seeds (std omitted where $< 0.01\%$).

[10, 10] context signal

Digit	Class C1	Class C2	Other
0 (C1=0, C2=10)	$92.96 \pm 0.30\%$	0.00%	7.04%
1 (C1=1, C2=11)	$99.21 \pm 0.06\%$	0.00%	0.79%
2 (C1=2, C2=12)	$56.78 \pm 0.69\%$	0.00%	43.22%
3 (C1=3, C2=13)	**0.00%**	**68.02 ± 0.76%**	**31.98%**
4 (C1=4, C2=14)	$75.87 \pm 0.88\%$	0.00%	24.13%
5 (C1=5, C2=15)	$80.38 \pm 0.82\%$	0.00%	19.62%
6 (C1=6, C2=16)	$90.19 \pm 0.40\%$	0.00%	9.81%
7 (C1=7, C2=17)	$87.06 \pm 0.19\%$	0.00%	12.94%
8 (C1=8, C2=18)	$35.93 \pm 1.31\%$	0.00%	64.07%
9 (C1=9, C2=19)	$73.64 \pm 1.50\%$	0.00%	26.36%

Table 7: Per-digit final class prediction rates for the **Hybrid** model under context signal $[10, 10]$. C1 = correct context 1 class (digit index), C2 = correct context 2 class (digit + 10), Other = all remaining classes. Mean ± std across 5 seeds (std omitted where $< 0.01\%$).

C.3.2 Feedforward model results

[0, 0] context signal

Digit	Class C1	Class C2	Other
0 (C1=0, C2=10)	$90.92 \pm 0.75\%$	0.00%	9.08%
1 (C1=1, C2=11)	$85.73 \pm 0.42\%$	0.00%	14.27%
2 (C1=2, C2=12)	$78.29 \pm 0.91\%$	0.00%	21.71%
3 (C1=3, C2=13)	**0.00%**	**72.48 ± 0.81%**	**27.52%**
4 (C1=4, C2=14)	$73.73 \pm 1.29\%$	0.00%	26.27%
5 (C1=5, C2=15)	$44.17 \pm 0.63\%$	0.00%	55.83%
6 (C1=6, C2=16)	$91.44 \pm 0.62\%$	0.00%	8.56%
7 (C1=7, C2=17)	$81.52 \pm 0.70\%$	0.00%	18.48%
8 (C1=8, C2=18)	$87.89 \pm 0.62\%$	0.00%	12.11%
9 (C1=9, C2=19)	$83.35 \pm 1.49\%$	0.00%	16.65%

Table 8: Per-digit final class prediction rates for the **Feedforward** model under context signal $[0, 0]$. C1 = correct context 1 class (digit index), C2 = correct context 2 class (digit + 10), Other = all remaining classes. Mean ± std across 5 seeds (std omitted where $< 0.01\%$).

[10, 10] context signal

Digit	Class C1	Class C2	Other
0 (C1=0, C2=10)	$90.10 \pm 0.77\%$	0.00%	9.90%
1 (C1=1, C2=11)	$94.18 \pm 0.19\%$	$0.09 \pm 0.06\%$	5.73%
2 (C1=2, C2=12)	$44.57 \pm 0.48\%$	0.00%	55.43%
3 (C1=3, C2=13)	**0.00%**	**95.45 ± 0.19%**	**4.55%**
4 (C1=4, C2=14)	$92.97 \pm 0.45\%$	0.00%	7.03%
5 (C1=5, C2=15)	$55.61 \pm 0.27\%$	0.00%	44.39%
6 (C1=6, C2=16)	$77.77 \pm 1.38\%$	0.00%	22.23%
7 (C1=7, C2=17)	$84.44 \pm 0.65\%$	0.00%	15.56%
8 (C1=8, C2=18)	$22.69 \pm 0.62\%$	0.00%	77.31%
9 (C1=9, C2=19)	$36.07 \pm 1.50\%$	0.00%	63.93%

Table 9: Per-digit final class prediction rates for the **Feedforward** model under context signal $[10, 10]$. C1 = correct context-1 class (digit index), C2 = correct context-2 class (digit + 10), Other = all remaining classes. Mean ± std across 5 seeds (std omitted where $< 0.01\%$).

C.3.3 Generative model results

[0, 0] context signal

Digit	Class C1	Class C2	Other
0 (C1=0, C2=10)	$1.73 \pm 0.30\%$	$93.33 \pm 0.36\%$	4.94%
1 (C1=1, C2=11)	$0.58 \pm 0.12\%$	$64.28 \pm 0.72\%$	35.14%
2 (C1=2, C2=12)	$2.02 \pm 0.31\%$	$70.52 \pm 0.59\%$	27.46%
3 (C1=3, C2=13)	$0.75 \pm 0.17\%$	$85.52 \pm 0.82\%$	13.72%
4 (C1=4, C2=14)	$1.14 \pm 0.29\%$	$56.52 \pm 1.41\%$	42.34%
5 (C1=5, C2=15)	$0.40 \pm 0.25\%$	$23.05 \pm 0.71\%$	76.55%
6 (C1=6, C2=16)	$9.98 \pm 0.47\%$	$80.21 \pm 0.83\%$	9.81%
7 (C1=7, C2=17)	$2.35 \pm 0.37\%$	$71.98 \pm 0.84\%$	25.66%
8 (C1=8, C2=18)	$2.46 \pm 0.63\%$	$90.47 \pm 0.77\%$	7.06%
9 (C1=9, C2=19)	$0.04 \pm 0.06\%$	$75.76 \pm 1.31\%$	24.20%

Table 10: Per digit final class prediction rates for the **Generative** model under context signal $[0, 0]$. C1 = correct context 1 class (digit index), C2 = correct context 2 class (digit + 10), Other = all remaining classes. Mean ± std across 5 seeds (std omitted where $< 0.01\%$).

[10, 10] context signal

Digit	Class C1	Class C2	Other
0 (C1=0, C2=10)	$95.31 \pm 0.29\%$	$2.47 \pm 0.35\%$	2.22%
1 (C1=1, C2=11)	$91.05 \pm 0.78\%$	$7.67 \pm 0.81\%$	1.29%
2 (C1=2, C2=12)	$67.05 \pm 0.84\%$	$0.23 \pm 0.13\%$	32.71%
3 (C1=3, C2=13)	$69.15 \pm 0.82\%$	$9.62 \pm 0.93\%$	21.23%
4 (C1=4, C2=14)	$43.22 \pm 0.81\%$	$45.15 \pm 1.07\%$	11.63%
5 (C1=5, C2=15)	$43.61 \pm 1.33\%$	$4.19 \pm 0.63\%$	52.20%
6 (C1=6, C2=16)	$93.59 \pm 0.65\%$	$1.17 \pm 0.26\%$	5.24%
7 (C1=7, C2=17)	$52.55 \pm 1.43\%$	$33.85 \pm 1.39\%$	13.60%
8 (C1=8, C2=18)	$67.13 \pm 1.12\%$	$0.23 \pm 0.12\%$	32.65%
9 (C1=9, C2=19)	$75.04 \pm 1.53\%$	$1.37 \pm 0.37\%$	23.59%

Table 11: Per digit final class prediction rates for the **Generative** model under context signal $[10, 10]$. C1 = correct context 1 class (digit index), C2 = correct context 2 class (digit + 10), Other = all remaining classes. Mean ± std across 5 seeds (std omitted where $< 0.01\%$).

D.4 Bias transfer experiment: per-digit statistics

Here is the full per-digit statistical result overview for the bias transfer experiment referenced in the main text: PC-only inference on the hybrid-trained model, and the two alternative training architectures (amortization-only and PC-only).

Digit	**Exp. (%)**	**Ctrl. (%)**	**Diff. (pp)**	t**(9)**	p**-value**	**Cohen's** d
0	0.00	0.16	−0.16	−2.23	0.053	−1.00
1	0.00	0.00	0.00	—	—	—
2	0.00	0.00	0.00	—	—	—
3	**0.00**	**0.00**	**0.00**	—	—	—
4	0.00	0.00	0.00	—	—	—
5	0.00	0.00	0.00	—	—	—
6	0.00	0.00	0.00	—	—	—
7	0.00	0.00	0.00	—	—	—
8	0.00	0.00	0.00	—	—	—
9	0.00	0.00	0.00	—	—	—

Table 12: C1→C2 error rates per digit under PC-only inference (random Gaussian initialization at std=0.01, 100 iterations). Paired t-tests across 10 seed-matched model pairs. Both conditions trained for 10 epochs.

D.4.1 Feedforward model

Digit	**Exp. (%)**	**Ctrl. (%)**	**Diff. (pp)**	$t(9)$	p**-value**	**Cohen's** d
0	67.77	0.00	67.77	26.77	<0.001	11.97
1	70.59	0.00	70.59	18.77	<0.001	8.39
2	41.65	0.00	41.65	20.49	<0.001	9.16
3	**0.00**	**0.00**	**0.00**	—	—	—
4	57.91	0.00	57.91	22.79	<0.001	10.19
5	12.28	0.00	12.28	10.78	<0.001	4.82
6	78.05	0.00	78.05	43.88	<0.001	19.62
7	59.63	0.00	59.63	26.28	<0.001	11.75
8	25.87	0.00	25.87	10.23	<0.001	4.57
9	35.01	0.00	35.01	12.63	<0.001	5.65

Table 13: C1→C2 error rates per digit under amortized inference for the amort-only trained model. Paired t-tests across 10 seed-matched model pairs.

D.4.2 Generative model

Digit	**Exp. (%)**	**Ctrl. (%)**	**Diff. (pp)**	$t(9)$	p**-value**	**Cohen's** d
0	0.00	0.02	−0.02	−1.00	0.343	−0.45
1	0.00	0.00	0.00	—	—	—
2	0.00	0.02	−0.02	−1.00	0.343	−0.45
3	**0.00**	**0.02**	**−0.02**	**−1.00**	**0.343**	**−0.45**
4	0.00	0.00	0.00	—	—	—
5	0.00	0.00	0.00	—	—	—
6	0.00	0.00	0.00	—	—	—
7	0.00	0.00	0.00	—	—	—
8	0.00	0.00	0.00	—	—	—
9	0.00	0.00	0.00	—	—	—

Table 14: C1→C2 error rates per digit under PC-only inference for the PC-only trained model. Paired t-tests across 10 seed-matched model pairs.