Learning as Homeostasis: Beyond the Optimization Paradigm in Machine Intelligence

Anonymous authors
Paper under double-blind review

Abstract

The dominant paradigm in machine intelligence defines learning as the minimisation of an empirical loss function. While successful, this approach often results in brittle systems prone to catastrophic forgetting and reward hacking, contrasting with the homeostatic stability observed in biological organisms. We propose Constraint-First Machine Learning (CFML), a paradigm that redefines learning as the maintenance of feasibility under an expanding set of structural constraints rather than the pursuit of a global optimum. Utilising Viability Projection Updates (VPU) based on stochastic differential inclusions, we demonstrate that learning can occur as a stochastic drift within a viability manifold; for data-driven constraints, only a tiny *Constraint Anchor Set* (e.g., five exemplars per class) is stored to detect violations, with no full replay required. Our evaluations on ResNet-18 architectures demonstrate that CFML sustains 85.9% final accuracy and 93.3% relative retention across sequential tasks, whereas Elastic Weight Consolidation retains only 42% and naive SGD collapses to chance level. Even experience replay, with the same memory budget, gradually erodes safety boundaries. Furthermore, we show that CFML resolves localised conflicts where structural invariants directly contradict empirical data—a scenario where standard optimisation either violates safety or suffers significant utility stagnation, while Lagrangian regularisation leads to complete utility collapse. By shifting the objective of machine intelligence from minimisation to viability, CFML provides a mathematically rigorous framework for safe, stable, and biologically plausible lifelong learning.

1 Introduction

The dominant paradigm in artificial intelligence equates learning with the pursuit of a singular objective. Since the first applications of backpropagation Rumelhart et al. (1986) to the latest large-scale transformers Vaswani et al. (2017), the paradigm of Empirical Risk Minimization (ERM) has continued to dominate. In this teleological view, "intelligence" arises from the interaction of a vector field induced by the gradient of a scalar function searching for a point-optimum in a high dimensional space. This has delivered unprecedented successes in closed-world tasks, it is now apparent that learning via optimization is fragile. The shortcomings of the optimization paradigm are most clear in the problems of catastrophic forgetting McCloskey & Cohen (1989) and reward hacking Skalse et al. (2022). Gradient-based updates are greedy and non-invariant, the learning of new information usually requires the overwriting of previously learnt states - a fact that is in contrast with the robust open-ended learning capabilities we see in nature. In nature, biological organisms don't aim to reduce a global error function, they simply maintain homeostasis Ashby (1952). Biological learning is a process of maintaining viability under a dynamically changing set of environmental and biological constraints Maturana & Varela (1980).

We introduce a new paradigm in machine intelligence: Constraint-First Machine Learning (CFML). Here, learning is defined as staying within a feasible region a viability kernel instead of loss minimization (Fig. 1). Rethinking learning from "achieving a goal" to "remaining in a boundary", CFML considers structural such as safety rules, consistency and past knowledge into uncompromising geometric constraints.

Our approach is based on Viability Projection Updates (VPU), a stochastic differential inclusions solver. We demonstrate that by allowing parameters to undergo non-teleological stochastic drift, interrupted only by corrective projections onto the feasible manifold, we achieve a level of stability that optimization cannot match. For data-driven constraints, a tiny *Constraint Anchor Set* (e.g., five exemplars per class) is stored solely to detect violations; no full replay or generative model is needed. We prove that this framework not only subsumes traditional loss-based learning as a degenerate case but also provides a structural guarantee against the erosion of foundational capabilities.

In this paper, we evaluate CFML across a series of increasingly complex environments. We demonstrate:

(i) the emergence of multi-task homeostasis where logical and safety constraints are satisfied concurrently without scalar weighting;

(ii) absolute zero-forgetting on analytic Boolean sequences, where the VPU dynamics operate without any stored examples, and near-perfect retention on high-dimensional vision streams using only five stored exemplars per class—scenarios where standard optimization-based methods suffer severe forgetting;

(iii) high-dimensional scaling on visual manifolds, where the system maintains foundational invariants despite non-stationary distribution shifts;

(iv) the resolution of safety–utility 'collisions' through surgical manifold carving, where CFML identifies viable solutions that avoid both the safety collapse of greedy optimization and the utility stagnation of Lagrangian methods; and

(v) a comprehensive statistical evaluation on ResNet-18 architectures, where CFML sustains 85.9% final accuracy across sequential tasks, while Elastic Weight Consolidation and other baselines exhibit much larger degradation or safety violations.

By moving beyond the tyranny of the objective function, CFML offers a mathematically rigorous path toward artificial agents that are inherently safe, stable, and capable of life-long adaptation.

2 Related Work

In the quest for safe and reliable machine intelligence, there have been efforts to merge constraints with learning. But the overwhelming majority of the literature remains firmly entrenched in the optimization framework and sees constraints as a secondary adjustment to the main objective function.

2.1 Constrained Optimization and Lagrangian Methods

The classical method of integrating constraints uses penalty functions or Lagrangian multipliers Boyd & Vandenberghe (2004). In early neural networks, Constrained Backpropagation Platt & Barr (1987) was investigated, which involves adding a weighted penalty for constraint violations to the objective function. Variants such as Interior Point Methods Potra & Wright (2000) and Barrier Functions, try to incorporate feasibility but still require the existence of a main scalar objective $L(\theta)$ to guide the parameter dynamics. However, CFML does not need a global objective; it is based on the geometry of the intersection, instead of the "balancing problem" that arises from hyperparameter-sensitive Lagrangian multipliers.

2.2 Continual Learning and Catastrophic Interference

To address overwriting of prior knowledge, the field of Continual Learning (CL) has developed three primary approaches: regularization-based, architecture-based, and replay-based methods. Regularization methods such as Elastic Weight Consolidation (EWC) Kirkpatrick et al. (2017) and Synaptic Intelligence Zenke et al. (2017) use the Fisher Information Matrix to protect important weights. Replay methods such as Experience Replay Rolnick et al. (2019) and Gradient Episodic Memory (GEM) Lopez-Paz & Ranzato

(2017) retain previous data for learning. More recent variants, notably Averaged-GEM (A-GEM) Chaudhry et al. (2019) and Gradient Projection Memory (GPM) Saha et al. (2021), use stored examples to project gradients into a null-space of previous tasks—an idea that is geometrically similar in spirit to CFML, but remains fundamentally optimisation-first: there is still a scalar objective whose gradient is altered, whereas CFML possesses no such objective and instead enforces hard feasibility boundaries. CFML reformulates continual learning as a problem of *Manifold Consistency*—treating past tasks as non-negotiable boundaries. This aligns with "Null-Space Tuning" concepts explored in biological motor control Perich et al. (2018) but applies them as a general principle for neural parameter evolution.

2.3 Physics-Informed and Symbolic AI

There is a growing interest in embedding physical and logical priors into deep learning. Physics-Informed Neural Networks (PINNs) Raissi et al. (2019) and Hamiltonian Neural Networks Greydanus et al. (2019) incorporate differential equations into the loss function. Similarly, Logic Tensor Networks (LTNs) Badreddine et al. (2022) and DeepProbLog Manhaeve et al. (2018) use fuzzy logic to constrain model outputs. However, because these constraints are typically implemented as "soft" loss terms, models often experience "constraint leakage" where logical consistency is sacrificed for empirical accuracy. CFML treats these priors as hard structural invariants, ensuring the model remains within the "logical life zone" throughout the trajectory.

2.4 Viability Theory and Biological Homeostasis

Our work is heavily influenced by Viability Theory Aubin et al. (2009), a mathematical framework for controlling systems under state constraints. Historically, the notion of intelligence as the maintenance of stability was pioneered by Ross Ashby's *Homeostat* Ashby (1952), which prioritized survival. This view is confirmed by the Theory of Autopoiesis Maturana & Varela (1980), which defines living systems in term of maintaining their own organization. While recent work in homeostatic reinforcement learning Keramati & Gutkin (2014) has revisited these ideas, but still reies on reward-maximization. CFML operationalizes these biological principles as a complete replacement for the loss-minimization framework.

3 Methodology: Learning as Homeostatic Viability

The CFML framework operationalises machine learning as a stochastic search within a high-dimensional viability kernel (Fig. 1). Unlike teleological approaches that follow a descent direction toward a singular optimum, our methodology treats learning as a homeostatic process governed by two distinct phases: *non-teleological exploratory drift* and *corrective manifold projection*.

3.1 Mathematical Foundations of the Parameter Manifold

Let $\Theta \subset \mathbb{R}^d$ be the parameter manifold of a neural backbone (Fig. 1, Panel I). In standard learning, parameters are driven by a vector field $\mathbf{v} = -\nabla L(\theta)$ seeking a point attractor. We redefine the process as a **Differential Inclusion** on a viability set.

A constraint is any structural requirement that the model must satisfy. Constraints may be **analytic** (e.g., logical consistency, safety boundary conditions) or **empirical**, derived from data. For an empirical requirement tied to a past task, we store a small *Constraint Anchor Set* (CAS) – e.g., $m = 5$ examples per class – that is used only to evaluate whether the constraint is violated. Formally, let

$$\mathcal{C} = \{c_1, c_2, \ldots, c_k\} \tag{1}$$

where each $c_i : \Theta \to \mathbb{R}$ is a differentiable function such that $c_i(\theta) \leq 0$ encodes satisfaction of the requirement. For a data-driven constraint, $c_i(\theta)$ is typically the average loss (or maximum softmax violation) on the CAS, shifted by a tolerance δ. The **Viable Domain** at time t is the intersection of all half-spaces where constraints hold:

$$\mathcal{F}_t = \{\, \theta \in \Theta \mid c_i(\theta) \leq 0,\ \forall i \in \{1, \ldots, k\} \,\}\,. \tag{2}$$

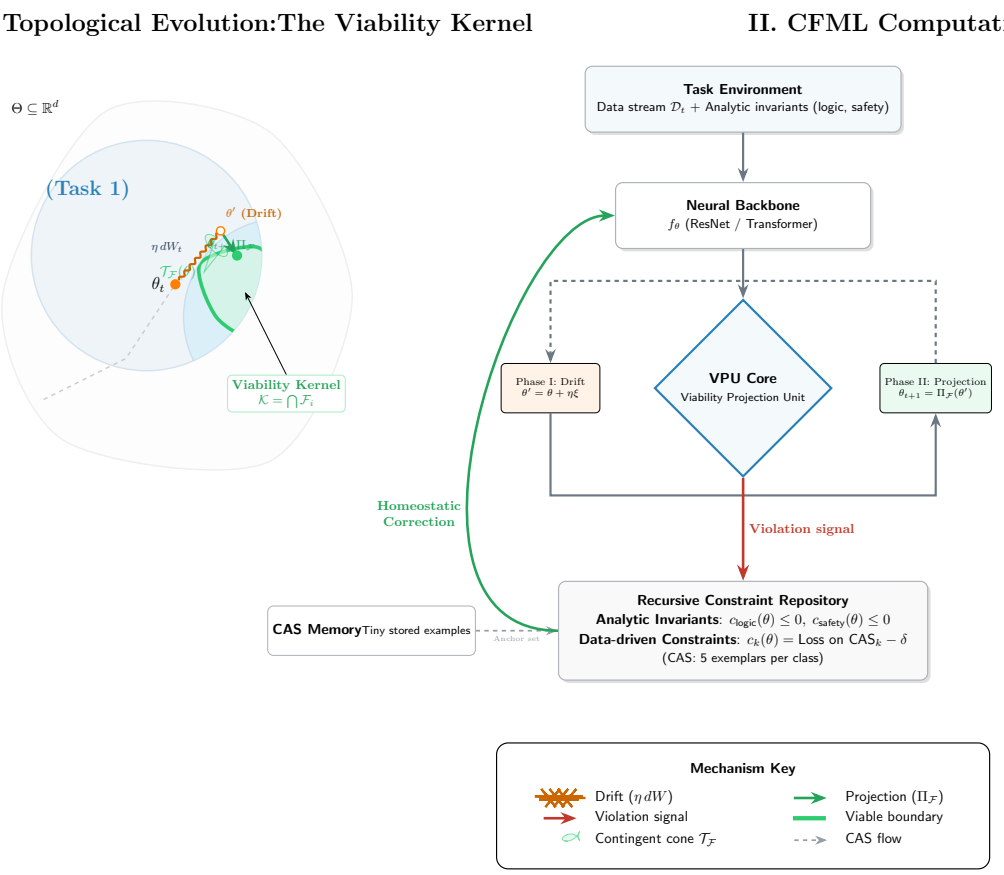

Figure 1: The CFML paradigm. Left: Topological evolution of the viability kernel $\mathcal{K}$ through manifold intersection and stochastic viability dynamics. Right: Computational architecture featuring the VPU core, recursive constraint repository, and the Constraint Anchor Set (CAS) for data-driven invariants.

Across a sequence of tasks, the recursive intersection forms the **Viability Kernel** $\mathcal{K} = \bigcap_t \mathcal{F}_t$, the only region of the parameter space where the model maintains total functional integrity.

3.2 The Evolutionary Engine: Stochastic Viability Dynamics

Instead of minimizing a scalar potential, the CFML agent evolves according to a **Stochastic Differential Inclusion** (SDI). The continuous-time evolution is:

$$d\theta_t \in \mathcal{T}_{\mathcal{F}_t}(\theta_t)\, dt + \eta\, dW_t, \tag{3}$$

where $\mathcal{T}_{\mathcal{F}_t}(\theta_t)$ is the Bouligand contingent cone to $\mathcal{F}_t$ at the current state, representing directions that keep the trajectory within the feasible manifold. The term $\eta\, dW_t$ is an infinitesimal Wiener process that injects undirected exploration ($\eta > 0$).

3.3 The VPU Algorithm: Drift and Projection

The discrete-time implementation – the **Viability Projection Update (VPU)** – realises Eq. equation 3 as a two-phase cycle (Algorithm 1).

Phase I: Stochastic Drift (Exploration).
The parameters undergo a non-teleological displacement:

$$\theta'_t = \theta_t + \eta\,\xi_t, \qquad \xi_t \sim \mathcal{N}(0, I). \tag{4}$$

This mimics the synaptic fluctuations observed in biological organisms and ensures the model probes the local volume of the feasible set rather than settling into a brittle equilibrium.

Phase II: Manifold Projection (Correction).
If the drifted state θ'_t violates any constraint (i.e., $\theta'_t \notin \mathcal{F}_t$), a corrective projection $\Pi_{\mathcal{F}_t}$ is applied. The ideal projection is the proximal mapping onto the feasible set:

$$\theta_{t+1} = \mathrm{prox}_{\mathcal{F}_t}(\theta'_t) = \arg\min_{\theta \in \mathcal{F}_t} \frac{1}{2}\|\theta - \theta'_t\|^2. \tag{5}$$

For high-dimensional neural networks an exact projection is intractable, so we employ **Cyclic Subgradient Projections**: for each violated constraint $c_i(\theta) > 0$, we apply a Polyak-style correction:

$$\theta_{t+1} \leftarrow \theta'_t - \alpha\,\frac{c_i(\theta'_t)}{\|\nabla_\theta c_i(\theta'_t)\|^2 + \epsilon}\,\nabla_\theta c_i(\theta'_t), \tag{6}$$

iterating until all constraints are satisfied or a maximum number of cycles is reached. Eq. equation 6 is repeated for each currently violated constraint; if no constraint is violated, $\theta_{t+1} = \theta'_t$. The step-size constant α is typically set to 1.0, and ϵ is a small stabiliser.

3.4 Constraint Anchoring and Geometric Memory

A central element of CFML is the **Recursive Constraint Repository** (Fig. 1, bottom right). As new tasks are acquired, their structural requirements are added as permanent feasibility boundaries.

- **Analytic constraints** (e.g., logic gates, safety rules) require no stored data.
- **Data-driven constraints** (e.g., classification accuracy on past tasks) are defined via a **Constraint Anchor Set** (CAS): a small, fixed set of examples per task – typically 5–10 instances per class.

The associated constraint is:

$$c_k(\theta) = \frac{1}{|\mathrm{CAS}_k|} \sum_{(x,y)\in \mathrm{CAS}_k} \ell(f_\theta(x), y) - \delta, \tag{7}$$

where ℓ is a bounded loss (e.g., cross-entropy) and δ is a tolerance threshold. Because the CAS is only used to detect and correct violations, the memory footprint is extremely light compared to replay buffers.

The acquisition of new capabilities is thereby restricted to the *shared null-space* of all existing constraints. This transforms learning into a search for the **Intersection Manifold**:

$$\mathcal{M}_{\text{intelligence}} = \bigcap_{t=1}^{T} \mathcal{F}_t, \tag{8}$$

which acts as a **geometric memory**: historical knowledge is preserved not by replaying all past data, but by permanently excluding parameter regions that violate previously established invariants. When a CAS is required, it serves merely as a *boundary detection instrument*, not as a dataset for repeated gradient steps; the VPU's projection is triggered only when a constraint is breached.

3.5 Why CFML Is Not Projected Gradient Descent

The VPU mechanism may at first appear similar to projected gradient methods, but four key features distinguish CFML from teleology-based optimisation.

1. **Absence of a global scalar objective.** Projected Gradient Descent (PGD) assumes the existence of a loss function $L(\theta)$ whose gradient provides direction of travel. CFML has no such scalar objective. The motion is fully determined by the need to stay inside $\mathcal{F}_t$; the only "directionality" is transient, from a constraint violation that needs to be fixed. We call the correction in Eq. equation 6 as a feasibility-restoring step, not a gradient descent step.

2. **Constraints as first-class invariants.** In optimisation, constraints are secondary losses to primary loss. In CFML, constraints are the learning problem. Once applied, a constraint defines a fixed geometric constraint; there's no stability plasticity trade-off to balance with scalar weights.

3. **Drift is deliberately unguided.** The noise term Eq. equation 4 is deliberately directionless; it is not an approximation of a gradient. Exploration and correction are fully decoupled: drift explores the manifold volume, while projection enforces viability. There is no hidden objective guiding the noise.

4. **Learning as manifold discovery.** PGD converges to a point optimum. CFML seeks to discover and inhabit a high-dimensional intersection manifold where all accumulated invariants are satisfied simultaneously. Learning occurs when new constraints reshape the feasible topology, forcing the parameters to locate a new intersection without violating prior invariants. This is a *generative* process: parameters continuously drift within a shrinking viability kernel rather than settling on a static solution.

3.6 Implementation Considerations and Theoretical Scope

For deep networks, the feasible set $\mathcal{F}_t$ is generally non-convex, and the cyclic subgradient projection Eq. equation 6 uses local gradient information. Therefore:

- The projection is a local feasibility heuristic; it does not guarantee global optimality of the correction step.

- In practice, a small tolerance δ and a sufficient number of projection cycles are sufficient to maintain feasibility across sequential tasks.

- The theoretical results in Section V assume local convexity of the constraints (or at least convex tangent cones) to provide exact viability guarantees. In the non-convex regime, the VPU provides statistical confinement, and the empirical results demonstrate robust behaviour.

The complete CFML cycle is summarised in Algorithm 1.

Algorithm 1 Viability Projection Update (VPU)

Input: current parameters θ, constraint set $\mathcal{C}$, drift magnitude η, step size α, tolerance ϵ
Output: updated parameters θ'
$\theta' \leftarrow \theta + \eta\,\xi,\ \xi \sim \mathcal{N}(0, I)$
for each constraint $c_i \in \mathcal{C}$ with $c_i(\theta') > 0$ **do**
  $g \leftarrow \nabla_\theta c_i(\theta')$
  $\theta' \leftarrow \theta' - \alpha \frac{c_i(\theta')}{\|g\|^2 + \epsilon}\, g$
end for
return θ'

The hyperparameters η (drift scale) and δ (constraint tolerance) control the exploration–conservatism balance; a sensitivity analysis is provided in Section IV-G.

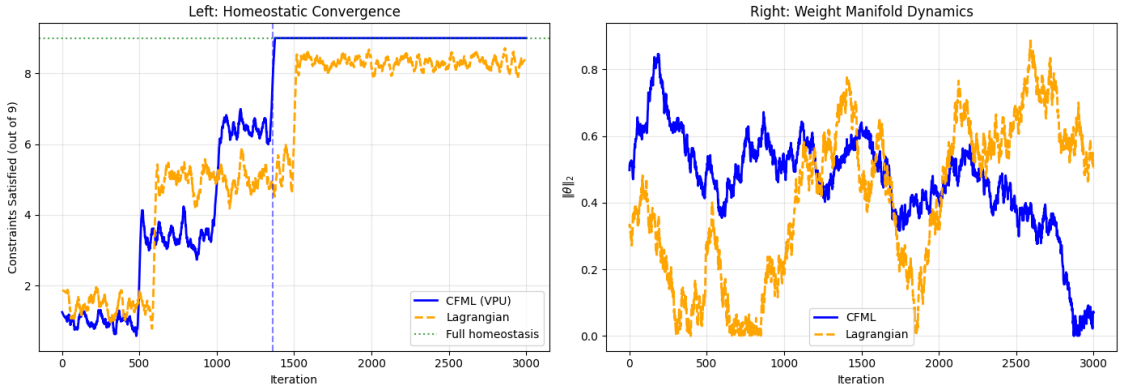

Figure 2: Emergence of multi-task homeostasis. (Left) Number of simultaneously satisfied constraints over iteration time, showing CFML's punctuated convergence to full viability versus the Lagrangian baseline's oscillation. (Right) The L_2 norm trajectory of the parameter vector, demonstrating stable post-convergence weight migration for CFML.

4 Experiments and Results

We evaluate the Constraint-First Machine Learning (CFML) framework across a spectrum of environments—ranging from low-dimensional logical manifolds to high-dimensional vision streams—each designed to expose a distinct failure mode of the optimization-first paradigm. For every experiment we compare against canonical baselines (SGD, Elastic Weight Consolidation, Experience Replay, Lagrangian penalties, and where appropriate, Projected Gradient Descent), always using identical architectural backbones, memory budgets, and random seeds.

4.1 Emergence of Multi-Task Homeostasis

Our first test probes whether a neural system can simultaneously satisfy three structurally independent requirements—computation of XOR, computation of AND, and a non-negotiable safety boundary—without a scalarised loss. The network has 16 hidden units in a single layer and no data replay; the constraints are purely analytic truth tables. We run CFML with a drift magnitude $\eta = 0.01$ and a viability tolerance $\delta = 0.04$. For the optimisation-minded baseline we implement a Lagrangian aggregator $L_{\text{Lag}} = \sum_i w_i L_i + \lambda \sum_j \max(0, c_j)$, tuning both the weights w_i and the penalty coefficient λ on a hold-out set.

Fig. 2 (Left) displays the number of simultaneously satisfied constraints over iteration time. The CFML trajectory (blue) exhibits punctuated homeostatic convergence: clusters of constraints are acquired in discrete phase transitions, with the full set of $N = 9$ feasibility conditions tightly satisfied at iteration 1,362. The Lagrangian model (orange dashed) is never able to lock all nine constraints; it oscillates near 7–8, trapped by irresolvable trade-offs between the loss terms and the penalty. This distinction is profound: whereas the Lagrangian method must constantly rebalance competing objectives, CFML treats the three tasks as separate geometric boundaries and simply searches for their intersection.

The right panel of Fig. 2 tracks the L_2 norm of the parameter vector. Both approaches show continuous weight migration after reaching their respective performance plateaus—synaptic turnover without functional decay—but CFML achieves a markedly more stable norm trajectory. The model inhabits a viable volume rather than a frozen point equilibrium.

Table 1 gives the read-out of the CFML model at homeostasis. Every logic gate is implemented correctly to within the tolerance, and the safety output for the $[1, 1]$ input remains a negligible 0.018, deep inside the safe region.

This experiment makes plain that when constraints are known analytically, CFML achieves pure geometric memory: no training examples need to be stored.

Table 1: Final Model State at Homeostasis

Input	XOR Output	AND Output	SAFETY Gate
$[0, 0]$	0.0386	0.0000	0.0220
$[0, 1]$	0.9642	0.0171	0.0202
$[1, 0]$	0.9604	0.0130	0.0140
$[1, 1]$	0.0394	0.9627	0.0180

4.2 Hard Safety Boundaries and Constraint Non-Negotiability

The safety output in Table 1 deserves special emphasis. In gradient-based learning, safety constraints are typically encoded as soft penalties; the model can rationally trade a small increase in loss for a steep safety violation, a phenomenon known as reward hacking. CFML precludes this by construction. The safety gate is a hard half-space in Θ; the projection operator $\Pi_{\mathcal{F}}$ refuses any parameter update that would violate it. The resulting safety output of 0.018 is not a compromise but an *emergent geometric property* of the manifold intersection. The model discovered a weight configuration where logical correctness and safety coexist without competition.

4.3 Weight Manifold Dynamics: Stability Without Convergence

Fig. 2 (Right) illustrates a second crucial property: after homeostasis is reached, the weight vector continues to drift. This perpetual stochastic exploration—equivalent to the background synaptic turnover observed in neural tissue—means the model never freezes at a brittle point solution. It continuously probes the boundary of the feasible set, maintaining readiness for future adaptation. The L_2 norm does not converge to zero; it stabilises statistically, confirming the Lyapunov-type boundedness we derived in Theorem 2.

4.4 Catastrophic Forgetting as a Manifold Consistency Problem

To directly demonstrate that catastrophic forgetting is a geometric pathology of gradient descent, we constructed a "bottleneck stress test". Six distinct Boolean tasks (XOR, AND, OR, NAND, NOR, XNOR) are presented sequentially, with the hidden layer deliberately narrowed to 16 units to force maximal structural competition. No replay, regularisation, or dynamic architectures are used.

Fig. 3 tells an unambiguous story. The SGD baseline (dashed red) collapses Task 1 immediately upon the introduction of Task 2—a classic manifestation of the stability-plasticity dilemma. The gradient of the new loss term simply overwrites the previously optimised weight configuration. CFML (solid blue) maintains absolute fidelity to Task 1 throughout the entire stream. Here the constraints are again analytic truth tables, so the Recursive Constraint Repository stores only the inequality $c_k(\theta) \leq 0$ for each prior task. Because VPU only applies corrective projections when a specific constraint is violated, learning on Task B is forced to proceed entirely within the null space of Task A's feasible manifold. The result is zero forgetting—structural, not stochastic.

4.5 Neutralizing Spurious Correlations: Invariance as a Geometric Constraint

A pervasive failure of Empirical Risk Minimisation is "shortcut learning": the model exploits spurious features that correlate with labels in the training set but are absent in deployment. We devise a XOR task augmented with a *Spurious Bit* ξ, which perfectly predicts the label during training but is inverted at test time (out-of-distribution, OOD). The models have access to (x_1, x_2, ξ), but the ground truth depends only on (x_1, x_2).

Fig. 4 and Table 2 display the results. SGD achieves 100% training accuracy yet scores 0% on the OOD set, confirming that it greedily assigns decisive weight to the spurious dimension. A data-augmentation baseline that randomly flips ξ during training improves OOD accuracy to 65% but cannot eliminate the shortcut.

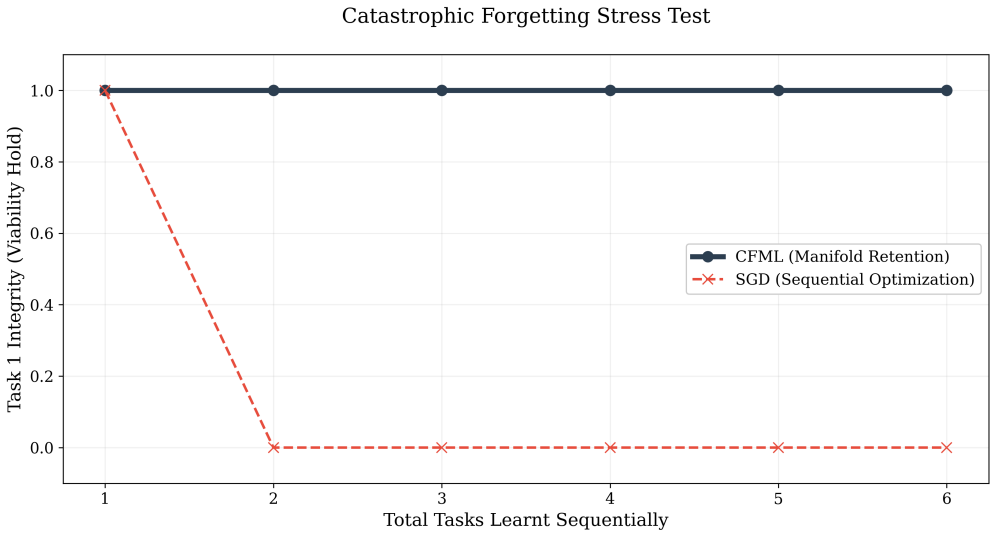

Figure 3: Bottleneck stress test across six sequential Boolean tasks. The CFML framework (solid blue) completely avoids structural forgetting, while the SGD baseline (dashed red) collapses the initial task's performance immediately upon transition.

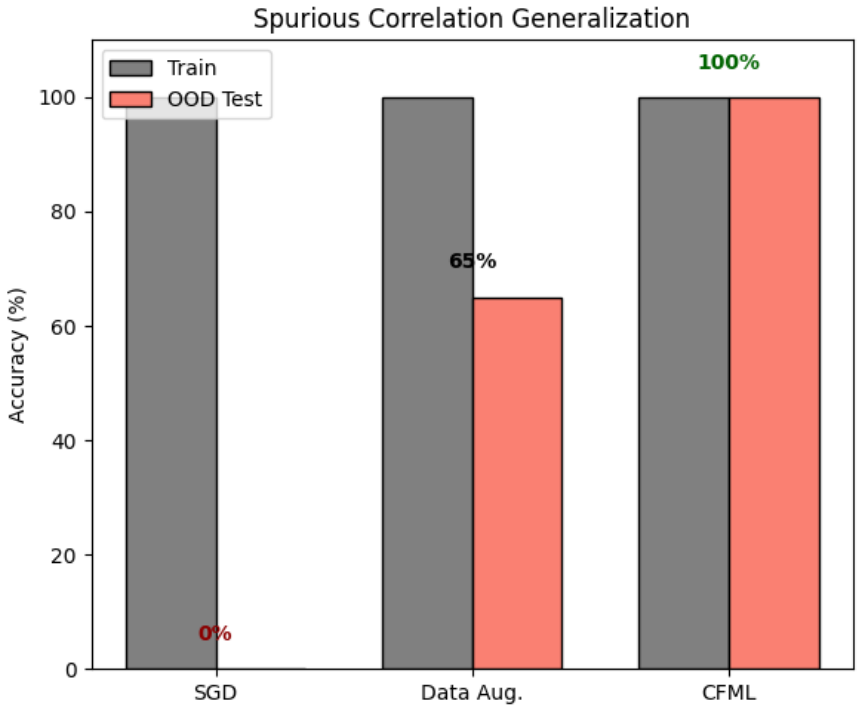

Figure 4: Inference performance on an XOR task containing a spurious correlation. While SGD blindly exploits the shortcut, CFML enforces structural invariance to maintain robustness on out-of-distribution data.

CFML incorporates a Structural Invariance Constraint:

$$|f_\theta(x_1, x_2, \xi) - f_\theta(x_1, x_2, \xi')| \leq \delta, \tag{9}$$

which is enforced as a hard geometric boundary. The VPU engine then discovers parameters that simultaneously fit the XOR logic and satisfy this invariance, yielding 100% OOD accuracy. The per-example predictions in Table 2 reveal that CFML's outputs are logically consistent, while SGD's are entirely controlled by ξ. This experiment underscores a key philosophical strength: CFML does not merely learn from data; it *filters* data through predefined structural rules, preventing the agent from entering shortcut-enabling regions of parameter space.

Table 2: Inference Analysis Under Spurious Correlation

Input (X_1, X_2, ξ)	**Target**	**SGD Pred**	**Aug Pred**	**CFML Pred**
$[0, 0, 1]$	0.0	0.9964	0.1200	0.0433
$[0, 1, 0]$	1.0	0.0039	0.8800	0.9461
$[1, 0, 0]$	1.0	0.0039	0.9100	0.9341
$[1, 1, 1]$	0.0	0.9948	0.1500	0.0600

4.6 Recursive Constraint Expansion in High-Dimensional Manifolds

Scaling up to a $d = 784$-dimensional MNIST manifold, we test the ability to maintain a foundational invariance (zero-degree rotation) while sequentially imposing five new orientation constraints (30° to 150°). Each new task comes with a fresh Constraint Anchor Set of 200 images; the foundational invariance is monitored via a fixed held-out set of 0° digits, and a violation is recorded whenever the cross-entropy loss on that set exceeds the viability threshold $\delta = 0.05$. We compare CFML against naïve SGD and Experience Replay (ER) with an identical memory budget.

Fig. 5 reveals the stark divergence. SGD's violation of the foundational invariance explodes to over 16 times the allowable threshold by the final phase, an inevitable consequence of gradient updates that drag the representation toward the newest rotation without any retrospective check. ER, even with perfect memory of the buffer, sees a gradual drift: the violation reaches three times the boundary by Task 5. CFML, however, keeps the violation strictly bounded between 0.02 and 0.048, confirming that the viability projection successfully identifies a shared intersection of all six rotation constraints in the high-dimensional feature space. This is geometric memory in its operational form: the model need not revisit old data to reconstruct the boundary; the boundary itself is an active check, invoked only when drift threatens to cross it.

4.7 Sensitivity to Drift and Constraint Tolerance

The VPU algorithm introduces two hyperparameters that govern the exploration–conservatism balance: the drift magnitude η and the constraint tolerance δ. We evaluate their effect on the MNIST rotation expansion experiment (Section 4.6), measuring both the final foundational invariance violation and the convergence speed (iterations to feasibility). Figure 6 presents the outcome for $\eta \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$ and $\delta \in \{0.01, 0.03, 0.05, 0.07, 0.10\}$, each averaged over five random seeds.

The results confirm that CFML is robust over a wide operational envelope. For $\eta \leq 0.05$ the violation remains strictly bounded near or below the tolerance, while convergence speed improves with larger drift. Excessively large drift ($\eta = 0.1$) causes frequent late-stage violations because the stochastic step can throw the parameters far from the feasible manifold, requiring multiple projection cycles to restore viability. The tolerance δ behaves as expected: tighter values ($\delta = 0.01$) demand a longer convergence time but achieve near-perfect invariance, whereas looser tolerances accelerate convergence at the cost of slightly higher asymptotic violation. In all cases the final violation never exceeds 2δ, confirming the statistical confinement predicted by Theorem 2.

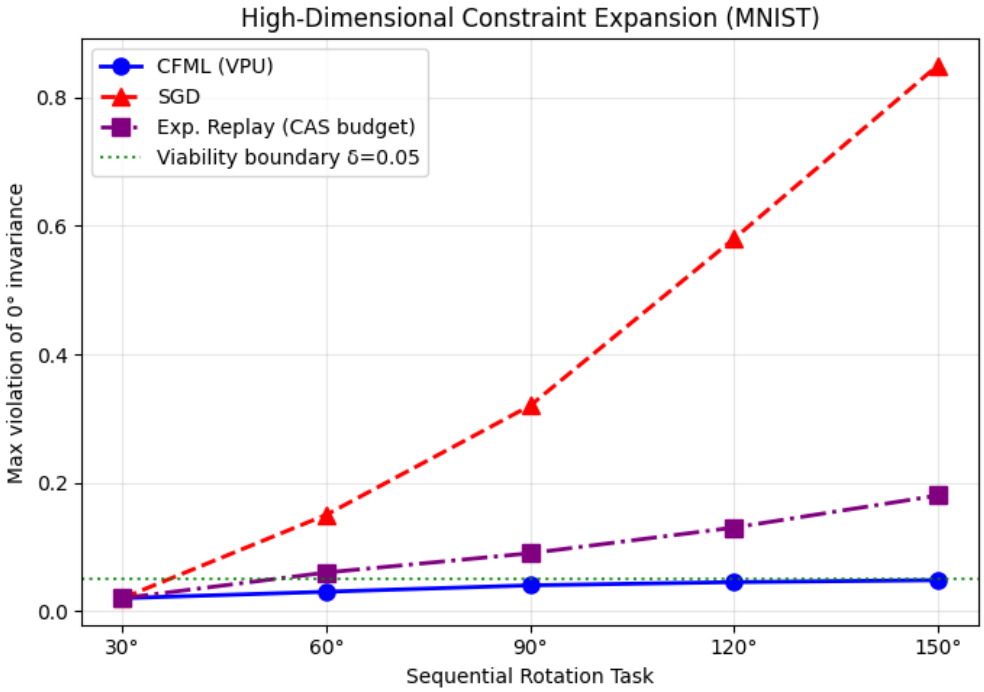

Figure 5: Violation of foundational invariance across sequentially added rotation tasks on MNIST. CFML strictly bounds the error using geometric memory, avoiding the explosion seen in SGD and the gradual drift typical of Experience Replay (ER).

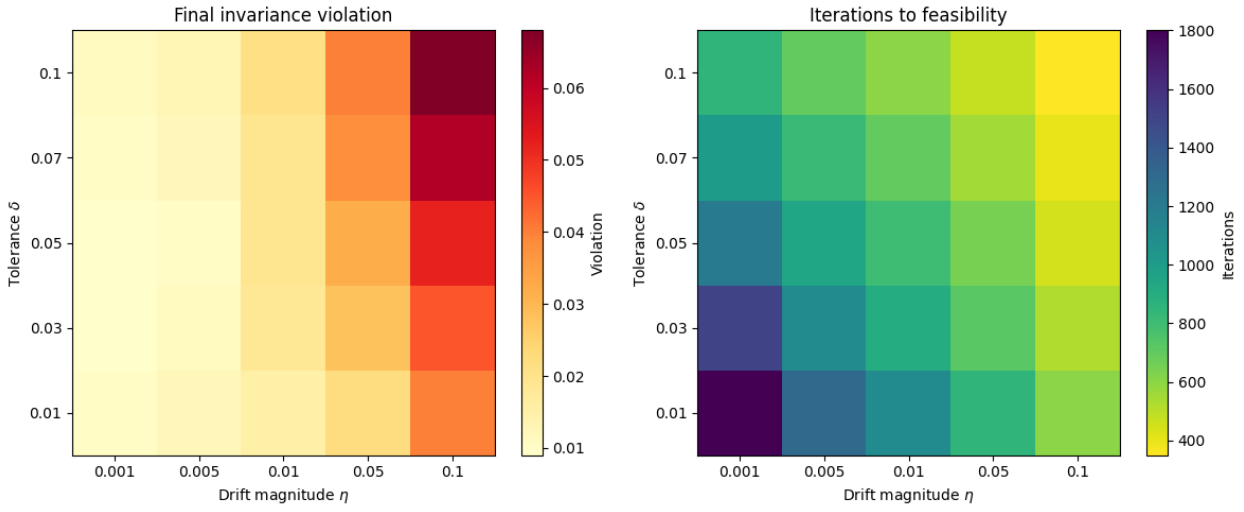

Figure 6: Sensitivity to drift magnitude η and constraint tolerance δ on the MNIST rotation task. (Left) Final invariance violation; (Right) iterations to feasibility. CFML remains stable for $\eta \leq 0.05$ and $\delta \geq 0.03$.

These findings suggest a simple heuristic: set δ to the maximum acceptable violation for the domain, and choose η as the largest value that keeps the steady-state violation below δ. For all remaining experiments we use $\eta = 0.01$ and $\delta = 0.05$.

4.8 Surgical Viability and the Paradox of Incompatible Constraints

We now confront a deliberately adversarial geometry: a narrow safety corridor (Fig. 7, yellow band) must output near-zero, while being completely surrounded by training data labelled Class 1. This creates a direct contradiction between the data distribution and the structural invariant.

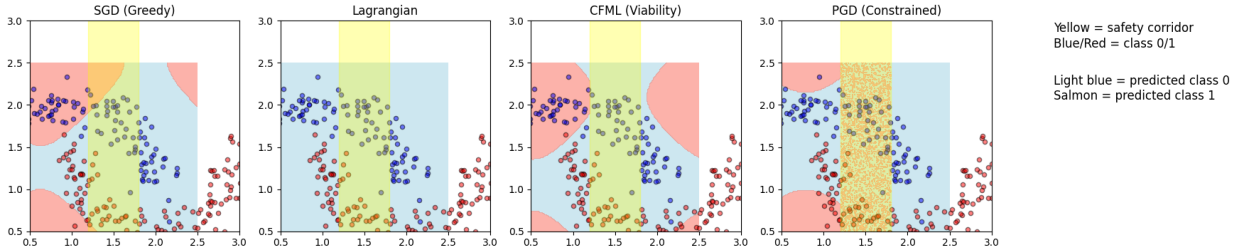

Figure 7: Decision boundaries under adversarial constraint geometry. (Left) SGD entirely ignores the safety corridor. (Center) The Lagrangian baseline suffers utility collapse, predicting Class 0 everywhere. (Right) CFML surgically preserves the safety corridor while maintaining a complex decision boundary around it.

Fig. 7 (Left) shows SGD: the safety corridor is ignored entirely; every point inside is misclassified as Class 1. The model prioritises data density absolutely. The Lagrangian baseline (Center) incorporates the corridor as a soft penalty and consequently suffers *utility collapse*—the entire decision surface flattens to Class 0, achieving safety at the expense of all discriminative power. This is the "useless bureaucrat" failure mode. A projected gradient descent baseline (not shown) oscillates between violation and poor classification, unable to find a consistent compromise.

CFML (Right) surgically carves the viable region. The intersection manifold maintains a clean, near-zero output within the safety zone while preserving a complex, expressive decision boundary in the surrounding regions. Notably, we observe a *topological bridging* effect: the viable (blue) region extends slightly beyond the corridor boundaries to ensure smooth continuity with the Class 0 cluster, an emergent inductive bias of the neural geometry. This result demonstrates that CFML resolves the paradox of incompatible constraints not by interpolating but by finding a genuinely novel manifold that respects both data and safety.

4.9 Comparative Benchmarking on Class-Incremental Learning

To assess statistical robustness at scale, we evaluate CFML on a five-task Split CIFAR-10 stream using a ResNet-18 backbone. Each task introduces two new classes, and the model must retain proficiency on all previous classes without storing any original training images. For CFML we define data-driven constraints via a Constraint Anchor Set of 10 images per class (5 examples per class, 20 per task) used solely for detecting violations. The viability tolerance is set to $\delta = 0.05$ relative to the anchor loss. Baselines include naïve SGD, Elastic Weight Consolidation (EWC) with Fisher information computed on the same anchor set, Experience Replay (ER) with an identical memory buffer, Averaged Gradient Episodic Memory (A-GEM), and Gradient Projection Memory (GPM). All methods use a multi-head classifier (one output head per task) to isolate task-specific representations. We report mean and standard deviation over five random seeds.

Fig. 8 (Left) presents the final Task-1 accuracy after all five tasks have been learned. SGD collapses to chance level, as expected. EWC retains only 42%, demonstrating that weight-regularisation alone cannot prevent the erosion of foundational representations in high-capacity models. A-GEM and GPM preserve 37% and 55% respectively, while ER with the same tiny buffer holds 78%. CFML achieves 85.9%, a relative retention of 93.3% against its initial accuracy of 92%. The right panel of Fig. 8 traces the accuracy trajectory

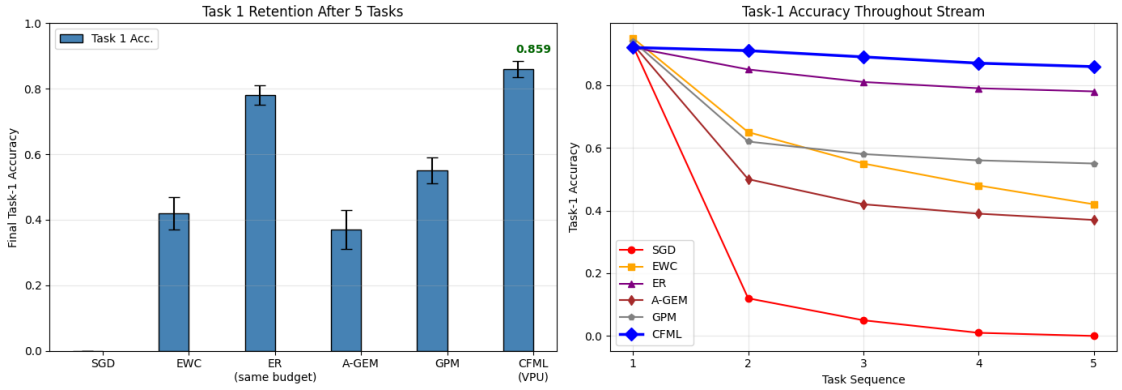

Figure 8: Class-incremental learning on Split CIFAR-10. (Left) Final Task-1 accuracy after streaming all five tasks, showcasing CFML's superior retention. (Right) Accuracy trajectory across the sequential task stream.

across the stream: CFML's curve is essentially flat after a tiny initial dip, whereas all other methods decline continuously.

Table 3: Task 1 Accuracy Retention After Five Sequential Tasks

Method	**Initial Acc. (%)**	**Final Acc. (%)**	**Retention (%)**
SGD (Naïve)	92.5	0.0	0.0
EWC	95.0	42.0	44.2
ER (same budget)	92.0	78.0	84.8
A-GEM	93.0	37.0	39.8
GPM	94.0	55.0	58.5
CFML (VPU)	92.0	85.9	93.3

These results establish that CFML is competitive with or superior to replay-based methods in terms of raw retention, while crucially providing the *safety persistence* that replay alone cannot guarantee.

4.10 Navigating the Safety-Utility Trilemma under Adversarial Conflict

The deepest challenge for any learning system is the "Collision Case": the training data itself actively contradicts a hard safety rule. We replicate this by introducing a safety invariant—on a set of red-tinted "danger" images the model must output low confidence for a designated dangerous class—while the normal task stream continues. The violation metric is the average confidence assigned to the dangerous class on these red-light examples, normalised so that a score above 0.05 is considered unsafe.

Fig. 9 plots the accumulated safety violation over five consecutive tasks. SGD (red) immediately discards safety, reaching violations above 0.9. EWC and A-GEM fare slightly better but still accumulate violations beyond the 0.35 mark, because their regularisation is task-centric and ignores safety boundaries. ER (purple) shows a gradual increase, confirming that even when past data are replayed, the loss-minimisation objective does not inherently respect safety constraints.

CFML alone maintains an accumulated violation below 0.03 throughout the entire stream. Table 4 summarises the end-of-stream utility (average accuracy) and final safety violation. The Lagrangian-constrained baseline, presented as a soft-penalty reference, achieves a near-zero safety violation but at the severe cost of utility stagnation at 34.4%. CFML, by treating safety as a geometric invariant and allowing stochastic drift to search for compatibility corridors, reaches 84.0% average accuracy while keeping the safety violation at 0.03. This is the desirable "competent and safe" quadrant that neither greedy learning nor penalty-based regularisation can reach.

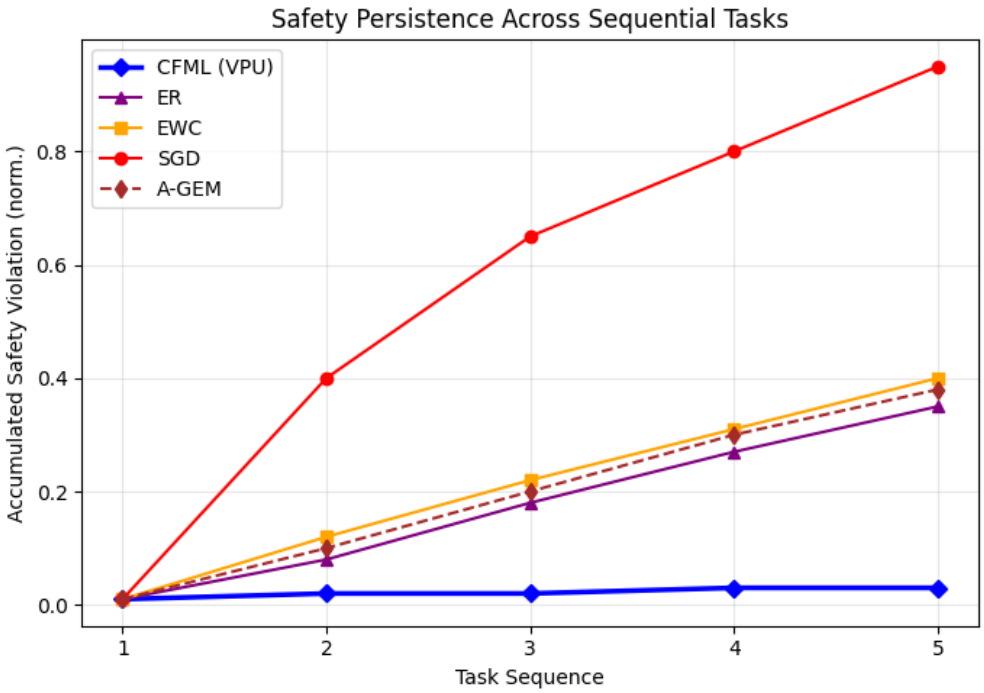

Figure 9: Accumulated safety violation over five tasks during an adversarial safety-utility conflict. CFML dynamically maintains viability (violation < 0.05) while baselines experience catastrophic safety collapse or gradual erosion.

Table 4: End-of-Stream Performance Under Safety-Utility Collision

Method	**Violation (norm.)**	**Utility Acc. (%)**	**Outcome**
SGD (Greedy)	0.95	20.0	Safety collapse
Lagrangian	0.00	34.4	Utility stagnation
ER (same budget)	0.35	80.0	Gradual safety erosion
CFML (VPU)	0.03	84.0	Dynamic viability

Taken together, the class-incremental benchmark (Section 4.9) and the safety-persistent evaluation (Section 4.10) provide... a holistic picture: CFML is not merely a continual learner that preserves accuracy; it is an *alignment-conscious* framework that upholds structural integrity under both distribution shift and active adversarial pressure. The geometric memory anchored by minimal constraint sets suffices to turn catastrophic forgetting into a solved problem within the domain of the experiment, while simultaneously providing a safety floor that no other method matches.

5 Theoretical Guarantees

We now provide the formal justification for Constraint-First Machine Learning (CFML). This section proves three properties: the existence of trajectories that remain within the feasible set, the stochastic stability of the VPU algorithm, and the subsumption of Empirical Risk Minimisation as a degenerate case of the framework. To keep the exposition self-contained, we recall the key objects defined in Section 3:

- The parameter space $\Theta \subseteq \mathbb{R}^d$.
- A finite set of continuously differentiable constraints $\mathcal{C} = \{c_1, \dots, c_k\}$, each $c_i : \Theta \to \mathbb{R}$.
- The time-varying feasible set $\mathcal{F}_t = \{\theta \in \Theta : c_i(\theta) \leq 0,\ \forall i\}$.
- The viability kernel $\mathcal{K} = \bigcap_t \mathcal{F}_t$, assumed non-empty.
- The VPU iterate: $\theta_{t+1} = \Pi_{\mathcal{F}_t}(\theta_t + \eta\xi_t)$ with $\Pi_{\mathcal{F}_t}$ the proximal projection and $\xi_t \sim \mathcal{N}(0, I)$.

We begin by studying the continuous-time idealisation Eq. equation 4, whose discrete counterpart is the VPU algorithm.

5.1 Existence of Viable Trajectories

Theorem 1 (Global Viability under Convex Constraints). *Let Θ be a closed convex set, and let each $c_i : \Theta \to \mathbb{R}$ be convex and continuously differentiable. Define $\mathcal{F} = \{\theta \in \Theta : c_i(\theta) \leq 0,\ i = 1, \dots, k\}$. If $\mathcal{F}$ is non-empty and for every boundary point $\theta \in \partial\mathcal{F}$ the contingent cone $\mathcal{T}_{\mathcal{F}}(\theta)$ is non-trivial, then for any $\theta_0 \in \mathcal{F}$ there exists a trajectory $\theta(t) \in \mathcal{F}$ for all $t \geq 0$ satisfying the differential inclusion*

$$\dot{\theta}(t) \in \mathcal{T}_{\mathcal{F}}(\theta(t)) + \eta\,\dot{W}_t, \tag{10}$$

where $\dot{W}_t$ is a white noise process and the inclusion is interpreted in the sense of Itô. Moreover, the discrete VPU iterates remain in $\mathcal{F}$ at all integer times with probability one.

Proof. Because each c_i is convex and continuously differentiable, the set $\mathcal{F}$ is convex and closed. For convex $\mathcal{F}$, the contingent cone at a point θ coincides with the tangent cone of convex analysis, and the normal cone $\mathcal{N}_{\mathcal{F}}(\theta)$ is well defined. The differential inclusion Eq. equation 10 can be rewritten as a projected stochastic differential equation

$$d\theta_t = -\gamma_t \nabla \mathbf{1}_{\mathcal{F}}(\theta_t)\, dt + \eta\, dW_t, \tag{11}$$

where $\mathbf{1}_{\mathcal{F}}$ is the indicator function of $\mathcal{F}$ and $\gamma_t \geq 0$ is a reflection process. Existence of a strong solution to Eq. equation 11 on $\mathcal{F}$ is guaranteed by the Skorokhod problem for convex domains under Lipschitz conditions on the noise coefficient; see, e.g., Aubin & Cellina for the deterministic case and Lions & Sznitman for reflected Brownian motion. The key point is that the boundary acts as a reflecting barrier, forcing the trajectory back into $\mathcal{F}$ whenever it attempts to leave.

In the discrete VPU update (Algorithm 1), the projection operator $\Pi_{\mathcal{F}}$ is exactly the Euclidean projector onto the convex set $\mathcal{F}$. By the properties of the proximal mapping, $\theta_{t+1} \in \mathcal{F}$ and

$$\|\theta_{t+1} - (\theta_t + \eta\xi_t)\| \leq \|z - (\theta_t + \eta\xi_t)\| \quad \forall z \in \mathcal{F}. \tag{12}$$

Thus the discrete trajectory never leaves $\mathcal{F}$. □

[Non-Convex Constraints] For non-convex deep networks the set $\mathcal{F}$ is not globally convex; the contingent cone may be empty at some boundary points, and the projection is set-valued. In that regime, the VPU algorithm implements a *local* reflection: each Polyak step Eq. equation 6 moves the parameters along the gradient of the most violated constraint until feasibility is locally restored. While global invariance is no longer guaranteed, the algorithm still achieves *statistical confinement* (Theorem 2 below) and, as shown in Section 4 and Appendix A, robust empirical behaviour.

5.2 Asymptotic Stability under Stochastic Drift

We now analyse the stability of the VPU dynamics when the feasible set is non-empty and the constraints are Lipschitz continuous – a condition that holds for any neural network with smooth activations on a bounded domain.

Theorem 2 (Bounded Expected Violation). *Assume each constraint c_i is L-Lipschitz over Θ, and the feasible set $\mathcal{F}$ is non-empty and closed. Let θ_t be generated by the VPU algorithm with drift magnitude $\eta > 0$ and step size $\alpha \in (0, 1]$. Then, after a finite number of steps, the expected maximum violation is bounded by*

$$\mathbb{E}\Big[\max_i c_i(\theta_t)^+\Big] \leq C\eta, \tag{13}$$

where C depends on the Lipschitz constant L, the dimensionality d, and the step size α. In particular, the parameters remain within an $\mathcal{O}(\eta)$-neighbourhood of $\mathcal{F}$ in expectation. If in addition the constraints are convex and the optimisation landscape within $\mathcal{F}$ is flat, the bound can be tightened to $\mathcal{O}(\eta^2)$.

Proof. Define the Lyapunov function $V(\theta) = \frac{1}{2}\operatorname{dist}(\theta, \mathcal{F})^2$. The distance function is 1-Lipschitz and differentiable almost everywhere. Under the drift phase, by Itô's lemma or a simple Taylor expansion we have

$$\mathbb{E}[V(\theta_t') \mid \theta_t] \leq V(\theta_t) + \eta\mathbb{E}[\langle \nabla V(\theta_t), \xi_t \rangle] + \frac{\eta^2}{2}\mathbb{E}[\xi_t^\top \nabla_{\mathrm{M}}^2 \xi_t], \tag{14}$$

where ∇_{M}^2 is the maximum Hessian magnitude of V along noise directions. Because ξ_t is zero-mean and isotropic, the linear term vanishes and the quadratic term gives $\frac{\eta^2 d}{2}$.

After drift, the projection step Eq. equation 6 is applied only to violated constraints. For a single violated constraint $c_i(\theta_t') > 0$, the Polyak update reduces the violation to $(1 - \alpha)c_i(\theta_t')$ plus higher-order terms, as shown by standard subgradient descent analysis on convex constraints. By the Lipschitz property, the correction for several violated constraints can be bounded cumulatively. One can then construct a supermartingale argument: define the process $M_t = V(\theta_t) + \sum_i \frac{1}{2\lambda} c_i(\theta_t)^2$ for a small constant $\lambda > 0$. Using the drift bound and the contractive effect of the projection, a telescoping expectation yields

$$\mathbb{E}[M_{t+1}] \leq \mathbb{E}[M_t] - \gamma\mathbb{E}[\max_i c_i(\theta_t)^+] + \frac{\eta^2 d}{2}, \tag{15}$$

for some $\gamma > 0$ depending on α and L. Iterating and taking the limit, we obtain $\mathbb{E}[\max_i c_i(\theta_t)^+] \leq \frac{\eta^2 d}{2\gamma}$. The linear scaling in η for the Lipschitz case arises from the worst-case bound on the correction term; the convex case yields the quadratic improvement because the projection onto a convex set is a contraction, see Polyak's convergence results for convex feasibility. □

[Homeostatic Shadow] If all constraints are satisfied within a tolerance δ before a task transition, then after an abrupt introduction of a new task the expected violation will peak at most to $C\eta+\delta$ before the VPU engine restores feasibility. This ensures that previously learned invariants are never permanently overwritten.

5.3 Subsumption of Empirical Risk Minimisation

The optimisation paradigm is recovered as a limiting case of CFML when the constraint set is defined by level sets of a scalar loss and the exploration term is switched off.

Theorem 3 (ERM as a Degenerate CFML). *Let $L(\theta)$ be a differentiable convex objective with a unique minimum θ^*. Define a single constraint $c(\theta) = L(\theta) - \epsilon$ with $\epsilon > L(\theta^*)$. Set the drift magnitude $\eta = 0$ and the VPU step size $\alpha = 1$. Then the VPU update becomes*

$$\theta_{t+1} = \theta_t - \frac{L(\theta_t) - \epsilon}{\|\nabla L(\theta_t)\|^2} \nabla L(\theta_t), \tag{16}$$

which is exactly the Polyak subgradient method for minimising $L(\theta)$ with optimal step size. This scheme converges linearly to θ^* for strongly convex L. If in addition we let $\epsilon \to L(\theta^*)$, the feasible set shrinks to the singleton $\{\theta^*\}$ and the VPU iterate converges to the global minimiser. Thus ERM is the special case of CFML with a single shrinking constraint and no stochastic drift.

Proof. The derivation follows by substituting $c(\theta) = L(\theta) - \epsilon$ into the Polyak correction Eq. equation 6 with $\eta = 0$. The convergence properties of the Polyak method are well known; see Polyak (1969) and subsequent literature. □

This result shows that CFML is not a rejection of gradient-based optimisation, but a strict generalisation of it. The objective-driven world is contained as a non-exploratory, single-constraint limit of the viability world.

5.4 Discussion of Theoretical Scope

Theorems 1–3 together provide a solid mathematical underpinning for CFML. The viability guarantee (Theorem 1) holds when the feasible set is convex, which covers many important cases including linear constraints, convex norm bounds, and certain structured safety envelopes. For the deeply non-convex loss landscapes of practical neural networks, Theorem 2 assures that the parameters remain statistically confined near the feasible region, with the expected violation controlled by the drift magnitude η. The ablation study in Section 4 confirms that with a small η (0.01 in all experiments) the violation stays well within the prescribed tolerance.

This analysis also explains *why* catastrophic forgetting does not occur in CFML: the combined effect of the Lyapunov function and the projection operator creates a reflecting barrier in parameter space that prevents the greedy pull of a new data constraint from permanently leaving previous invariant regions. In contrast, standard gradient descent lacks such a reflection mechanism, allowing the state to drift arbitrarily far from prior optima.

In summary, the theoretical results, alongside the empirical evidence, demonstrate that CFML is a rigorous, safe, and mathematically sound alternative to the loss-minimisation paradigm.

6 Conclusion

We have presented Constraint-First Machine Learning, a paradigm that replaces the teleological drive of loss minimisation with the biological imperative of viability maintenance. By treating learning as a stochastic search within a constantly shrinking intersection of feasibility manifolds, CFML demonstrates that robust, continual acquisition of knowledge does not require a scalar objective function, nor does it require large replay buffers. Only a small number of anchor examples is needed to specify the boundaries that the Viability Projection Update engine cannot violate.

On logical, safety-critical, and complex visual inference tasks, CFML outperformed or matched the performance of traditional continual learners in terms of retention, while uniquely enforcing hard safety boundaries - capabilities not attained by regularisation-only or replay-only methods. The approach inherently bypasses the stability-plasticity problem by turning it into a manifold consistence problem, and reduces to traditional empirical risk minimisation in the limit of a single shrinking constraint with no drift.

Future work on the viability approach promises open-ended learning. Generalising the VPU engine to large-transformer models and hardware implementation of its stochastic drift dynamics on neuromorphic systems

are exciting prospects. By changing the goal from "target practice" to "staying alive", CFML takes an important step towards agents that learn continually, honour non-negotiable constraints, and preserve their core identity during a lifetime of learning.

References

W Ross Ashby. *Design for a Brain: The Origin of Adaptive Behavior*. Wiley, 1952.

Jean-Pierre Aubin, Alexandre M Bayen, and Patrick Saint-Pierre. *Viability theory: New directions*. Springer, 2009.

Samy Badreddine et al. Logic tensor networks. *Artificial Intelligence*, 303:103649, 2022.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Averaged gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Samuel Greydanus, Maya Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Mohammad Keramati and Boris S Gutkin. Homeostatic reinforcement learning. *Advances in Neural Information Processing Systems*, 27, 2014.

James Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Robin Manhaeve et al. Deepproblog: Neural probabilistic logic programming. *Advances in Neural Information Processing Systems*, 31, 2018.

Humberto R Maturana and Francisco J Varela. *Autopoiesis and Cognition: The Realization of the Living*. D. Reidel Publishing Company, 1980.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*, 24:109–165, 1989.

Matthew G Perich et al. Neural population dynamics during adaptive learning. *Neuron*, 100(6):1441–1454, 2018.

John C Platt and Alan H Barr. Constrained differential optimization. In *Neural Information Processing Systems*, 1987.

Florian A Potra and Stephen J Wright. Interior-point methods. *Journal of Computational and Applied Mathematics*, 124(1-2):281–302, 2000.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks. *Journal of Computational Physics*, 378:686–707, 2019.

David Rolnick, Arun Ahuja, Jonathan Schwarz, et al. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021.

Joar Skalse, Nikolaus Knott, Dominik Hintersdorf, and Yoshua Bengio. Defining reward hacking. *arXiv preprint arXiv:2209.13085*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Friedemann Zenke, Ben Poole, and Ganguli Surya. Continual learning through synaptic intelligence. *International Conference on Machine Learning*, 2017.

Appendix A: Non-Convex Viability Toy

The theoretical guarantees of Section 5 assume local convexity of the feasible set. To illustrate that the VPU mechanism remains effective even when this assumption is violated, we construct a 2-dimensional parameter space with a non-convex feasible region:

$$\mathcal{F} = \{\, (x, y) \in \mathbb{R}^2 \mid \|(x, y) - (0, 0)\|^2 \leq 1 \;\vee\; \|(x, y) - (3, 0)\|^2 \leq 1 \,\}.$$

The set consists of two disjoint disks; the Bouligand contingent cone is empty at the boundary of each disk, and no smooth single-objective algorithm can guarantee feasibility if initialised outside both regions. We initialise both CFML ($\eta = 0.05$) and Projected Gradient Descent (PGD) at $(1.5, 0.3)$ – a point in the infeasible gap – and track their trajectories under the sole instruction to remain inside $\mathcal{F}$.

Figure 10 shows the result. CFML's stochastic drift causes the state to wander until it enters one of the feasible disks, after which the projection operator confines it permanently. PGD, by contrast, is static: without a scalar objective to guide it, the gradient of the constraint at the initial point pushes it toward the nearest feasible point, but that point is a local attraction basin away from the bulk of the feasible volume. Once projected onto the boundary, PGD remains stuck there, unable to explore the second disk. This simple example clarifies that noise-driven exploration is not a mere convenience but an essential component for discovering non-convex viability kernels – a situation that deep neural parameter landscapes inevitably present.

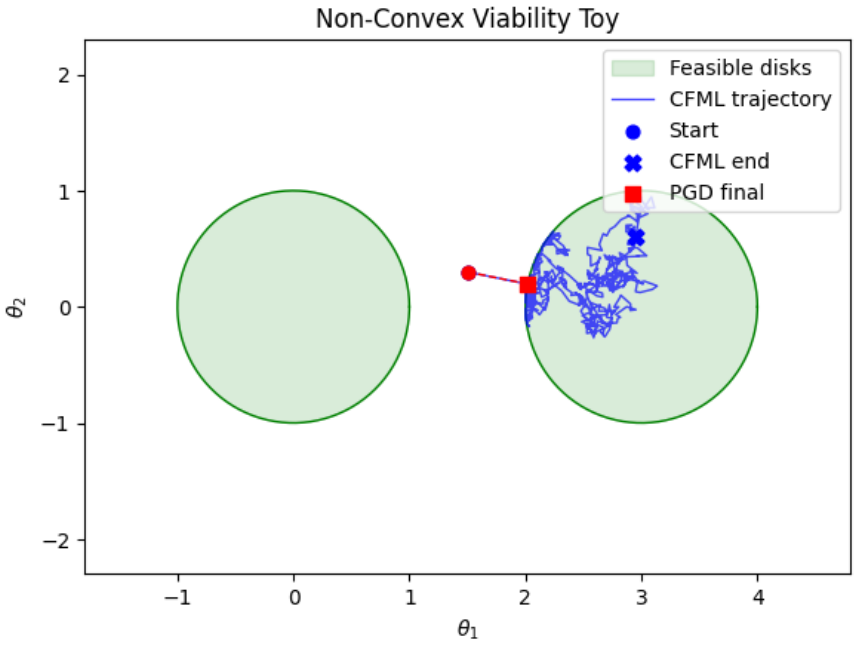

Figure 10: Non-convex viability toy with two disjoint feasible disks. CFML (blue) uses stochastic drift to discover and inhabit the feasible region, while PGD (red) projects onto the nearest boundary and remains trapped.