W-EXP: A Routed Post-Decision Expectation Adapter for Dynamic Economic Solvers

Bo Li[a], Serguei Maliar[b,c], Peiyuan Zhang[a]

[a] *School of Economics, Peking University, Beijing, China*
[b] *Department of Economics, Santa Clara University, , USA*
[c] *Hoover Institution, Stanford University, , USA*

Abstract

Conditional expectations are repeatedly evaluated while policies and value approximations move during the solution of stochastic dynamic models. We study W-EXP, a fitted post-decision continuation adapter. Conditional on the current host-solver continuation, its training problem is a conventional conditional-mean regression. The contribution is the combination of a common adapter interface, online updating, explicit routing to a critic target or actor objective, and diagnostics for level error, derivative error, support, and drift. The interface is instantiated separately in tabular Q-learning, a deterministic actor–critic method, and soft actor–critic; trained weights and configurations are not transferred across solvers. We give conditional perturbation results for a contraction-based Bellman operator and a fixed-distribution deterministic actor, an idealized deterministic tracking lemma, and a one-call mean-squared-error comparison with fresh Monte Carlo. These results do not establish convergence of the nonlinear solvers or a computational break-even. In a descriptive consumption–saving study, tuned W-EXP-equipped packages have lower mean final policy error than direct MC50 packages in all 12 solver–environment cells after one million outer steps, with reductions of 23.5%–84.1% across three seeds. The comparison does not match optimizer work: the primary IAC and SAC W-EXPpackages use additional actor or actor–critic updates, and neural runtimes are 1.57–3.96 times the direct baselines. Mechanism diagnostics indicate that routing and training support can matter, but the small and partly unbalanced designs do not identify universal causal rankings. The evidence supports W-EXPas a testable adapter architecture, not as a generally faster or uniformly superior integration method.

Preprint submitted to Working paper *September 18, 2026*

Keywords: conditional expectations, post-decision states, dynamic economic models, reinforcement learning, Monte Carlo, actor–critic methods

1. Introduction

Stochastic dynamic economic models repeatedly evaluate conditional expectations. In a Bellman update, an expectation appears in the continuation value; in an actor–critic method, a related continuation can also shape the policy gradient. Fresh Monte Carlo draws provide an unbiased estimate conditional on a fixed integrand. Their effect persists through the solver update, but they are not retained as a separately fitted continuation surface that can be queried at neighboring post-decision states. This distinction motivates a statistical information-reuse problem, not an assumption that simulation draws otherwise have no effect.

This paper asks whether a moving, solver-induced continuation can be maintained behind a common adapter interface and routed to the parts of a host solver that consume it. We construct a post-decision function, denoted W-EXP, that predicts the discounted expected continuation generated by the host's current network or table. A critic may use its level in a target, while a deterministic actor may differentiate through its action-sensitive input. "Solver-adaptable" has a restricted meaning throughout: the mathematical input–output contract is shared, but the representation, route, training support, update schedule, and fitted weights are instantiated separately for each solver. We do not test cross-solver weight transfer or a universal configuration.

Conditional-mean approximation itself is established. PEA fits expectations in Euler or other expectational equations (Den Haan and Marcet, 1990; Duffy and McNelis, 2001). Post-decision ADP and fitted value methods learn value or continuation objects from simulated successors (Hull, 2015; Tsitsiklis and Van Roy, 2001). Least-squares Monte Carlo estimates continuation values by regression (Longstaff and Schwartz, 2001), and recent neural stochastic-control methods combine learned policies with Monte Carlo value regression (Huré et al., 2021). Precomputation instead integrates selected basis functions before the main iterations (Judd et al., 2017). Neural economic solvers reorganize equilibrium conditions into trainable operators (Maliar et al., 2021; Azinovic et al., 2022; Han et al., 2026), while bc-MC cor-

rects the estimator of a squared conditional-expectation loss (Pascal, 2024, 2026).

The primary research question is therefore deliberately architectural: what is learned, where is it routed, and which diagnostics reveal when a fitted post-decision continuation helps or fails inside different host solvers? Three secondary questions organize the evidence. First, under what conditional level, derivative, and drift errors can the adapter perturb a value or actor update in a controlled way? Second, can separately tuned W-EXP-equipped packages attain lower fixed-outer-step policy error than direct MC50 packages in controlled consumption–saving problems? Third, are observed outcomes associated with route, support, target drift, and integration structure in the manner predicted by the diagnostics? The experiments address the latter two questions descriptively; they do not isolate a universal treatment effect.

The contribution has four parts. First, we specify a common adapter contract for a fitted post-decision continuation and make actor/critic routing an explicit design variable. This places W-EXPinside the broad continuation-regression and post-decision-value family rather than presenting it as the first expectation parameterization. Second, we state conditional diagnostic results: a uniform level error perturbs a contraction-based Bellman fixed point, a local derivative error perturbs a fixed-distribution deterministic actor gradient, an idealized update displays a drift term, and a trained deterministic predictor has a one-call MSE threshold relative to fresh MC. Third, we report 12 proof-of-concept package comparisons under matched economic models, seed labels, outer steps, and nominal Monte Carlo draws. Optimizer work and time are not matched. Fourth, routing, support, gradient, frozen-value, and discrete-transition studies provide mechanism diagnostics and expose adverse cases.

Across four shock environments and three solvers, the W-EXP-equipped package has lower mean final policy error in all 12 cells and in 34 of 36 paired seed labels. These are configuration-specific observations from three seeds, not estimates of a population-wide dominance probability. The neural packages require more optimizer work and are slower; deterministic or structure-aware integration can also be more accurate and cheaper. The evidence supports feasibility and motivates the adapter diagnostics, while leaving reliability, causal isolation, and efficiency frontiers to larger confirmatory designs.

The rest of the paper is organized as follows. Section 2 locates W-

EXPamong adjacent methods. Section 3 defines the adapter and routing contract. Section 4 gives the conditional diagnostic results. Section 5 describes the experiments. Sections 6 and 7 report package comparisons and mechanism diagnostics. Section 8 states the interpretation and limitations, and Section 9 concludes. Proofs and reproducibility details are in the appendices.

2. Adjacent methods and the boundary of the contribution

2.1. Parameterized expectations and neural PEA

PEA starts from an expectational equation, typically an Euler equation, specifies the conditional expectation as a function of the state, and estimates the function's parameters along a simulated path (Den Haan and Marcet, 1990). Its strength is to move numerical integration away from a nested root-finding or optimization loop. Duffy and McNelis (2001) replace conventional polynomial specifications with neural networks and combine global and local parameter search. These papers preclude any claim that learning a conditional expectation with a neural network is itself new.

W-EXPretains the central PEA idea that a conditional expectation can be parameterized. Conditional on fixed continuation parameters, Eq. (6) is an ordinary conditional-mean regression. Its narrower distinction is where the target comes from and how the fitted object is consumed: the target is induced by the host solver's current continuation, the sampling distribution q_t is an explicit design choice, and the output can enter a critic, a deterministic actor, or both. Target movement alone is not unique to W-EXP, because PEA objects can also move with current policy or solution parameters.

2.2. Continuation regression and fitted value iteration

Regression-based Monte Carlo methods estimate conditional continuation values from simulated successors. Least-squares Monte Carlo uses this construction for optimal stopping (Longstaff and Schwartz, 2001); simulation-based approximate value iteration fits parameterized value functions on representative states (Tsitsiklis and Van Roy, 2001). Neural stochastic-control algorithms likewise combine learned controls with Monte Carlo regression of value functions (Huré et al., 2021). W-EXPbelongs to this broad fitted-continuation family. The object-level contribution is therefore not continuation regression itself. The paper studies a specific adapter construction in which the moving discounted continuation is maintained separately from the

host value or critic, its use by actor and critic is routed explicitly, and its level, derivative, support, and drift errors are diagnosed together.

2.3. Post-decision ADP and precomputation

Post-decision states are established in numerical dynamic programming and approximate dynamic programming (Judd, 1998; Powell, 2011). Hull (2015) rewrites dynamic economic models around them, allowing controls to be selected without explicitly computing an expectation. State-specific stochastic stepsizes aggregate information and implicitly perform integration. W-EXPis a post-decision continuation approximation in this broad sense. Its specific construction uses supervised shock-average labels from the host's moving continuation, maintains the fitted object as a separately addressable adapter, and exposes critic and actor routing as design variables. These are architectural and diagnostic choices, not a claim that the underlying post-decision object is mathematically new.

Precomputation addresses repeated integration in a different way. Accurate quadrature, monomial integration, and simulation-based approximation provide strong alternatives when the integration structure is tractable (Judd et al., 2011; Fernández-Villaverde and Levintal, 2018). Judd et al. (2017) integrate selected approximation bases once, before the main solution, so later iterations can proceed as if the model were deterministic. The key contrast is offline integration of separable or otherwise integrable bases versus online fitting of the composed host-solver continuation. Both approaches require a known or simulable transition mechanism. W-EXPreuses its surface across nearby queries only while target drift remains controlled; it does not automatically transfer across changes in the shock law, transition, policy, critic, or solver.

2.4. Neural residual operators and bc-MC

Deep learning methods have made high-dimensional global solution procedures practical by parameterizing decision and value functions and training them on simulated or sampled states (Maliar et al., 2021; Azinovic et al., 2022; Han et al., 2026). The all-in-one construction of Maliar et al. (2021) reorganizes stochastic equations into a non-nested learning problem. The bc-MC operator of Pascal (2024) corrects the small-sample bias that arises when a squared empirical mean estimates a square of expectations and can

yield a minimum-variance unbiased estimator under stated conditions. Pascal (2026) connects bc-MC to PEA and derives a multi-innovation, variance-reduced generalization.

These methods modify how a stochastic residual or squared-expectation loss is estimated. W-EXPinstead creates a persistent fitted continuation that can be queried outside its regression update. With unbiased Monte Carlo labels and fixed continuation parameters, label variance adds a term to the expected squared regression loss that is independent of the fitted prediction, so the population minimizer is already the conditional mean. A bc-MC hybrid would therefore require a specified squared-expectation objective or model-selection criterion; it is not, by itself, a label generator. Variance-reduced innovation estimators can still be used to construct the labels without changing the adapter interface.

Table 1: Position of W-EXPrelative to adjacent expectation methods

Method	Primary approximated object	Timing	Relation to W-EXP
PEA / neural PEA	Conditional expectation in an Euler or expectational equation	Along simulated solution iterations	Shares conditional-mean parameterization; differs in host equation, support choice, and routing
Continuation regression	Conditional continuation or value	Regress-now/value-iteration steps	Direct object-level ancestor; W-EXPadds a separate routed adapter and joint diagnostics
Post-decision ADP	Post-decision value or continuation	Stochastic post-state updates	Same broad object family; W-EXPisolates a supervised adapter inside several host solvers
Pre-computation	Integrated approximation bases	Once before main iterations	Offline basis integration rather than online fitting of the composed continuation
All-in-one / bc-MC	Stochastic residual or squared-expectation loss	Inside a non-nested training operator	Changes loss estimation; does not by itself define a separately routed surface
W-EXP	Discounted continuation induced by the host solver	Repeated online supervised updates	Separate adapter with explicit route, support, and drift choices

Note: The table states overlap as well as difference. Categories can be combined; for example, variance-reduced innovations can construct W-EXPlabels. The table does not assert priority or empirical superiority.

The contribution studied here is the joint construction of an online host-induced continuation adapter, explicit actor/critic routing, and diagnostics for level, derivative, support, and drift errors. Each ingredient has clear antecedents, and the paper does not make an exhaustive priority claim. The

empirical comparisons match economic models, seed labels, outer steps, and nominal MC draws; they do not match wall-clock time or total optimizer work.

3. The W-expframework

3.1. Dynamic problem and post-decision continuation

Let $s \in \mathcal{S}$ be a pre-decision state, $a \in \mathcal{A}(s)$ an action, and

$$x = h(s, a) \in \mathcal{X} \tag{1}$$

the post-decision state. A random element $\varepsilon \sim P(\cdot \mid x)$ produces the next pre-decision state

$$s' = \Phi(x, \varepsilon). \tag{2}$$

The solver supplies a continuation integrand $g_\vartheta(s', \varepsilon)$. Depending on the solver, this may be a state value, a target critic evaluated at a next-period policy, or an entropy-adjusted target value. We let ε include all randomness averaged by the continuation call, including reparameterized next-action noise in SAC, and absorb the corresponding frozen policy parameters into ϑ. Define the unscaled conditional expectation and discounted continuation by

$$G_\vartheta(x) = \mathbb{E}[g_\vartheta(\Phi(x, \varepsilon), \varepsilon) \mid x], \qquad C_\vartheta(x) = \beta G_\vartheta(x), \tag{3}$$

where $\beta \in (0, 1)$ is the discount factor. The implementation trains W_ω to approximate C_ϑ directly. Storing the discounted object makes the solver target simply

$$y^W(s, a) = u(s, a) + W_\omega(h(s, a)). \tag{4}$$

Direct Monte Carlo with K shocks computes a fresh estimate at every call,

$$\widehat{C}^{\mathrm{MC}}_{\vartheta,K}(x) = \frac{\beta}{K} \sum_{k=1}^{K} g_\vartheta(\Phi(x, \varepsilon_k), \varepsilon_k). \tag{5}$$

W-EXPuses the same form to construct supervised labels at sampled post-decision states $x_i \sim q_t$,

$$z_i = \widehat{C}^{\mathrm{MC}}_{\vartheta_t,K}(x_i), \qquad \mathcal{L}_W(\omega) = \frac{1}{B} \sum_{i=1}^{B} \left(W_\omega(x_i) - z_i\right)^2. \tag{6}$$

Here $\bar{\vartheta}_t$ denotes the continuation parameters used for label construction; a target network can make them move more slowly than the online critic. Information is reused through the fitted parameters ω, not through an assumption that past Monte Carlo labels remain valid forever. The current implementation generates fresh labels during each W-EXPupdate.

When the actor needs a pathwise derivative, an optional Sobolev term supervises the derivative with respect to the action-sensitive component $x^{(a)}$ of the post-decision state:

$$\mathcal{L}_W^{\nabla}(\omega) = \mathcal{L}_W(\omega) + \lambda_{\nabla} \frac{1}{B} \sum_{i=1}^{B} \left\| \nabla_{x^{(a)}} W_\omega(x_i) - \nabla_{x^{(a)}} z_i \right\|^2 . \tag{7}$$

The derivative label is available only when the conditional random element admits an x-independent base-noise reparameterization and the transition and frozen continuation are differentiable with an integrable derivative. Otherwise the adapter uses the value-only loss or requires a likelihood-ratio construction not studied here. The empirical results below show why this term is optional rather than defining: a better derivative fit does not guarantee a better final policy.

3.2. Routing to the critic and actor

The layer has two interfaces. In the critic route, W_{ω^-} replaces direct integration in a Bellman or temporal-difference target,

$$y^{\text{critic}} = u(s, a) + W_{\omega^-}(h(s, a)), \tag{8}$$

where ω^- may be a Polyak-averaged target. This route changes target noise and target lag but need not expose derivatives to the policy.

For a deterministic actor, the actor route maximizes an objective of the form

$$J_W(\phi) = \mathbb{E}_s \left[u(s, \mu_\phi(s)) + W_\omega(h(s, \mu_\phi(s))) \right], \tag{9}$$

The parameters ω are frozen during the actor update, but the derivative of W_ω with respect to x remains active. The SAC implementation uses a reparameterized stochastic action and an entropy term, and its routed objective differs from the ordinary Q-based SAC actor objective. The deterministic gradient result in Proposition 2 does not cover that stochastic objective. Actor-only routing changes the policy update without changing the critic target; both-routing changes both pathways.

The state distribution q_t used to train W is part of the method. Uniform sampling seeks a global surrogate over the bounded domain. On-policy sampling concentrates capacity near post-decision states produced by the current actor. Mixed sampling uses both. This choice determines where the approximation error and derivative error are small; it is not merely an implementation detail.

3.3. Adapter contract and scope of adaptability

The invariant contract is a typed query x, frozen host continuation state ϑ, a transition sampler, and adapter state ω, returning a discounted continuation estimate and, when the reparameterization conditions hold, a derivative with respect to selected components of x. The host solver specifies the route and supplies its ordinary state/action batches. Solver-specific adapter configuration consists of representation, support distribution, label estimator and count, batch size, update frequency and timing, warm-up or blending, target update, derivative loss, and the number of host optimizer updates. Contract conformance concerns whether these declared inputs, outputs, and routes are implemented. Diagnostic fidelity, policy accuracy at a fixed resource budget, and Pareto improvement are distinct, application-specific success criteria that should be declared before evaluation. The present V1 reports only final policy error for selected packages after fixed outer steps; it does not estimate a diagnostic tolerance or a cost frontier. Failure to improve an empirical criterion rejects that configuration's comparative case, not the existence of the interface. No trained-weight transfer, automatic solver discovery, or universal hyperparameter rule is claimed.

3.4. Solver-specific realizations

The same interface admits different representations. In tabular Q-learning, W is a table over savings and the exogenous income state; linear interpolation supplies continuation values between savings grid points. The table is updated toward Monte Carlo labels with a stochastic stepsize. In IAC, the actor, state-value critic, and post-decision layer are two-hidden-layer networks with 64 tanh units per layer. The actor can differentiate through either direct integration or W. In SAC, W is a two-hidden-layer multilayer perceptron with 256 SiLU units per layer; a Polyak target with coefficient 0.01 supplies critic targets, while the online W supplies actor derivatives. These representations differ, but all approximate the same mathematical object in Eq. (3).

Algorithm 1 Generic solver update with a W-EXPlayer

Require: Host parameters (ϕ, ϑ) and optional target $\bar{\vartheta}$; adapter state (ω, ω^-); route r; support q_t; label count K and batch size B; update frequency, multiplicity, and before/after timing; warm-up/blend; target rate; derivative weight; host-update multiplicity n_t

for $t = 1, \ldots, T$ **do**
  Sample solver states and actions under the solver's ordinary training rule.
  if a W-EXPupdate is scheduled before the host update **then**
    Draw labeled post-states and apply the configured adapter update.
  end if
  if critic route is enabled and warm-up is complete **then**
    Build critic targets with $W_{\omega^-}(h(s, a))$.
  else
    Use the solver-defined critic fallback recorded in the configuration.
  end if
  if actor route is enabled and warm-up is complete **then**
    Differentiate the actor objective through $W_\omega(h(s, \mu_\phi(s)))$.
  else
    Use the solver-defined actor fallback recorded in the configuration.
  end if
  Apply n_t configured host updates to (ϕ, ϑ) and their target copies.
  if a W-EXPupdate is scheduled after the host update **then**
    Draw $x_i \sim q_t$ and $\varepsilon_{ik} \sim P(\cdot \mid x_i)$ for $i = 1{:}B$, $k = 1{:}K$.
    Set $z_i \leftarrow \beta K^{-1} \sum_{k=1}^{K} g_{\bar{\vartheta}}(\Phi(x_i, \varepsilon_{ik}), \varepsilon_{ik})$.
    Apply the configured adapter updates using Eq. (6) or Eq. (7); update ω^- if used.
  end if
end for

4. Error propagation and moving-target theory

The results in this section are conditional diagnostic lemmas. They do not establish convergence of a nonlinear actor–critic implementation or explain how a finite label budget attains the assumed errors; classical convergence results for temporal-difference learning with function approximation require substantially more specific sampling and approximation conditions (Tsitsiklis and Van Roy, 1997; Sutton and Barto, 2018). The results instead state how a supplied continuation error enters a contraction-based value update, a fixed-distribution deterministic actor update, and an idealized moving-target recursion.

Assumption 1 (Bounded contraction setting)**.** *Let $\mathcal{B}(\mathcal{S})$ be the bounded measurable real functions on $\mathcal{S}$ with the sup norm, and let $\mathcal{V} \subseteq \mathcal{B}(\mathcal{S})$ contain the functions considered below. One-period utility is bounded, $\beta \in (0,1)$, and the post-decision map and transition kernel are measurable. The exact Bellman operator T maps $\mathcal{V}$ to itself and is a β-contraction in the sup norm. All action extrema below are written as suprema; no attainment claim is needed.*

For a candidate value V, write

$$C_V(x) = \beta \mathbb{E}[V(\Phi(x,\varepsilon)) \mid x], \qquad (TV)(s) = \sup_{a \in \mathcal{A}(s)} \{u(s,a) + C_V(h(s,a))\}. \tag{10}$$

Let $\widetilde{T}$ replace C_V by an approximation W_V.

Proposition 1 (Bellman perturbation)**.** *Under Assumption 1, suppose*

$$\sup_{V \in \mathcal{V}} \sup_{x \in \mathcal{X}} |W_V(x) - C_V(x)| \le \epsilon_0. \tag{11}$$

Then $\left\| \widetilde{T}V - TV \right\|_\infty \le \epsilon_0$ for every $V \in \mathcal{V}$. If V^* is the fixed point of T, $\widetilde{V}$ is a fixed point of $\widetilde{T}$, and $\widetilde{T}$ is also a β-contraction, then

$$\left\| \widetilde{V} - V^* \right\|_\infty \le \frac{\epsilon_0}{1-\beta}. \tag{12}$$

If the approximation error is instead measured against the unscaled expectation G_V, replace ϵ_0 by $\beta\epsilon_0$.

The proposition gives a worst-case level-error channel. It says neither that the bound is tight nor that minimizing global W error is sufficient for a nonconvex actor–critic algorithm.

For deterministic actor routing, define the exact objective

$$J_C(\phi) = \mathbb{E}_{s\sim d}\left[u(s, \mu_\phi(s)) + C(h(s, \mu_\phi(s)))\right] \tag{13}$$

for a fixed state distribution d, and let J_W replace C by W.

Proposition 2 (Fixed-distribution deterministic actor-gradient perturbation)**.** *Suppose d is fixed with respect to ϕ. For d-almost every s, let C, W, h, and μ_ϕ be differentiable on the policy support. Assume their derivative products are dominated by a d-integrable envelope in a neighborhood of ϕ, so differentiation and expectation can be interchanged. Under compatible Euclidean and induced operator norms, suppose*

$$\begin{aligned} \sup_{x\in\mathcal{X}_\phi} \|\nabla_x W(x) - \nabla_x C(x)\| &\le \epsilon_1, \\ \sup_{s\in\mathrm{supp}(d)} \|D_a h(s, \mu_\phi(s)) D_\phi \mu_\phi(s)\| &\le L_{h\mu}, \end{aligned} \tag{14}$$

where $\mathcal{X}_\phi = \{h(s, \mu_\phi(s)) : s \in \mathrm{supp}(d)\}$. Then

$$\|\nabla_\phi J_W(\phi) - \nabla_\phi J_C(\phi)\| \le L_{h\mu}\epsilon_1. \tag{15}$$

The relevant derivative accuracy is local to the policy-induced support. This observation motivates on-policy or mixed training for a deterministically actor-routed layer, while also explaining why an on-policy surrogate can be poor on a uniform diagnostic grid. The proposition holds at a fixed d and does not bound changes in the endogenous occupancy distribution, a stochastic-policy gradient, or the trajectory of a coupled optimizer.

The continuation target moves because the critic and policy move. The next result isolates that tracking problem in function space.

Proposition 3 (Idealized deterministic moving-target tracking)**.** *Let C_t be the discounted continuation target at iteration t. Consider the idealized update*

$$W_{t+1} = (1-\eta)W_t + \eta\widetilde{C}_t, \qquad 0 < \eta \le 1, \tag{16}$$

under any norm for which

$$\left\|\widetilde{C}_t - C_t\right\| \le \rho_t, \qquad \|C_{t+1} - C_t\| \le \delta_t. \tag{17}$$

Writing $e_t = \|W_t - C_t\|$ *gives*

$$e_{t+1} \le (1-\eta)e_t + \eta\rho_t + \delta_t. \tag{18}$$

If $\rho_t \le \rho$ *and* $\delta_t \le \delta$*, then*

$$\limsup_{t\to\infty} e_t \le \rho + \frac{\delta}{\eta}. \tag{19}$$

The bound isolates deterministic approximation and drift terms. Under this uniform-error statement, the steady-state contribution of ρ is independent of η, while the drift contribution is δ/η. The result therefore does not prove that a smaller update rate suppresses stochastic label noise; that claim would require a filtration and a second-moment or high-probability analysis of the implemented stochastic update. Warm-up, targets, blending, and update frequency can change the empirical counterparts of ρ, δ, and the effective update rate.

Finally, consider direct Monte Carlo at a fixed x. Let

$$\sigma_g^2(x) = \mathrm{Var}[g_\vartheta(\Phi(x,\varepsilon),\varepsilon) \mid x,\vartheta], \qquad e_W(x) = W_\omega(x) - C_\vartheta(x). \tag{20}$$

Proposition 4 (Conditional one-call mean-squared-error comparison)**.** *Conditional on fixed continuation parameters* ϑ*, trained adapter parameters* ω*, and a fixed evaluation distribution* d_x*, assume the evaluation draws* $\varepsilon_1,\dots,\varepsilon_K$ *are conditionally iid given* x*, independent of the adapter training history, and* $g_\vartheta(\Phi(x,\varepsilon),\varepsilon)$ *is square-integrable. The direct estimator in Eq. (5) is unbiased and has conditional mean squared error*

$$\mathrm{MSE}\Big(\widehat{C}^{\mathrm{MC}}_{\vartheta,K}(x) \mid x,\vartheta,\omega\Big) = \frac{\beta^2\sigma_g^2(x)}{K}. \tag{21}$$

Conditional on the trained parameters, $W_\omega(x)$ *is deterministic and has integrated mean squared error* $\mathbb{E}_{d_x}[e_W(x)^2]$*. Hence it has lower integrated one-call MSE precisely when*

$$\mathbb{E}_{d_x}[e_W(x)^2] < \frac{\beta^2}{K}\mathbb{E}_{d_x}[\sigma_g^2(x)]. \tag{22}$$

This identity is an accuracy threshold for one evaluation call after training. It contains no adapter-fitting cost, optimizer work, query-count breakeven, or learning-theoretic bound on e_W. Increasing K lowers direct Monte Carlo variance. Additional W-EXPupdates may lower approximation error but can add lag, computation, or optimization bias. Neither side uniformly dominates, and the final policy comparison is more demanding because continuation error enters a coupled nonlinear training system.

5. Experimental design

5.1. Consumption–saving benchmark

The primary experiments use a finite-domain infinite-horizon consumption–saving problem. The pre-decision state is $s_t = (w_t, y_t)$, where w_t is available wealth and y_t is log income. The household chooses a consumption share $a_t \in [0.001, 0.999]$,

$$c_t = a_t w_t, \qquad b_t = w_t - c_t, \tag{23}$$

and receives CRRA utility

$$u(c_t) = \frac{c_t^{1-\gamma} - 1}{1-\gamma}. \tag{24}$$

The post-decision state is $x_t = (b_t, y_t)$. The next state satisfies

$$y_{t+1} = \rho y_t + \xi_{t+1}, \qquad w_{t+1} = R_{t+1} b_t + \exp(y_{t+1}). \tag{25}$$

All environments use $\beta = 0.9$, $\gamma = 2$, $\rho = 0$, and $w \in [0.1, 4]$. Thus the innovations are independent over time in the present design. The four environments in Table 2 vary shock scale, integration dimension, or tail shape while holding the preference and wealth parameters fixed.

5.2. Solvers, package matching, and evaluation

The formal suite includes tabular Q-learning (Watkins and Dayan, 1992), IAC, and SAC (Haarnoja et al., 2018). IAC uses a deterministic consumption-share actor, a state-value critic, and pathwise differentiation through the known transition. It is included as a solver architecture rather than presented as an external methodological contribution. DDPG-style runs (Silver et al., 2014) remain archived as stability diagnostics but are excluded because the observed direct baseline is unstable across seeds. This exclusion and the selected adapter configurations were determined after development runs, not in a preregistered confirmatory protocol. The controlled household problem is intentionally smaller than the heterogeneous-agent benchmark of Krusell and Smith (1998); that literature motivates reporting policy accuracy across algorithms rather than relying only on aggregate fit (Den Haan, 2010).

For each environment and solver, a direct MC50 package is compared with a separately tuned W-EXP-equipped package trained with 50-draw labels.

Table 2: Consumption–saving environments

Environment	Income innovation ξ_{t+1}	Gross return R_{t+1}	Feature isolated
Classic Gaussian	$0.10\varepsilon_y$	1.04	One-dimensional symmetric benchmark
High-vol Gaussian	$0.20\varepsilon_y$	1.04	Larger sampling dispersion, same integration dimension
Joint income–return	$0.10\varepsilon_y$	$1.04\exp[-0.5(0.15)^2 + 0.15\varepsilon_R]$	Independent income and return shocks; two-dimensional expectation
Rare disaster	$0.094249\varepsilon - b(D - p_d)$	1.04	Mean-zero, variance-0.10^2 innovation with negative skewness and a heavy left tail

Note: $\varepsilon_y, \varepsilon_R \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$. In the rare-disaster model, $D \sim \text{Bernoulli}(p_d)$, $p_d = 0.0043$, and $b = 0.5108$. The probability that an ordinary 50-draw batch contains at least one disaster is 19.38%. High volatility is defined in log income; because $\mathbb{E}[\exp(y)]$ rises with its variance, it is not a strictly mean-preserving income-level experiment.

Both use seed labels 42, 123, and 2026, one million outer steps, the same economic model, solver class, and evaluation rule. The shared integer seed does not guarantee common shock paths because the code paths consume random numbers differently. The comparison matches neither optimizer work nor time. In particular, IAC W packages use four actor updates after step 50,000 versus one for direct MC, and SAC W packages use three actor–critic updates after step 10,000 versus one for direct MC. Some SAC routes and W objectives also differ by environment. The main table therefore estimates package-level fixed-step performance; it does not isolate the causal effect of replacing one expectation call.

Table 3: Primary W-EXPpackage configurations and unmatched host updates

Host	Environments	Route / support	Adapter schedule	Host-update difference from direct
Q-learning	All four	Q target; tabular	MC50 labels; update every 8 steps; 8× state coverage; one W batch	None recorded; both packages use one host update per outer step
IAC	All four	Both; on-policy; gradient weight 1	MC50; 2× label-state coverage; update every 4 steps with four W batches; 10k warm-up; after-host timing	Four actor updates after step 50k versus one in direct MC
SAC	Classic	Critic; on-policy; value loss	MC50; every step; 5k warm-up; target rate 0.01	Three actor–critic updates after step 10k versus one in direct MC
SAC	High-vol / joint	Both; on-policy; gradient weight 1	MC50; 2× label-state coverage; every step; 5k warm-up; target rate 0.01	Three actor–critic updates after step 10k versus one in direct MC
SAC	Rare disaster	Critic; on-policy; value loss	MC50; every step; 5k warm-up; target rate 0.01	Three actor–critic updates after step 10k versus one in direct MC

Note: The table records archived primary configurations, not a universal recipe. The Classic SAC archive omits an explicit usage key; the implementation's legacy default resolves to critic-only routing. Full JSON configurations are authoritative.

Final accuracy is the mean squared error of consumption on a common validation grid against a separately computed high-accuracy value-function-iteration (VFI) policy,

$$\mathrm{MSE}_C = \frac{1}{N_{\mathrm{eval}}} \sum_{j=1}^{N_{\mathrm{eval}}} \left(\widehat{c}(w_j, 0) - c^{\mathrm{VFI}}(w_j, 0) \right)^2 . \tag{26}$$

The primary ratio first averages final MSE across seeds and then divides,

$$\mathcal{R}_{W/MC} = \frac{\overline{\mathrm{MSE}}_W}{\overline{\mathrm{MSE}}_{MC50}}, \qquad \text{reduction} = 100(1 - \mathcal{R}_{W/MC}). \tag{27}$$

Paired-seed ratios are reported separately because the ratio of means need not equal the mean of within-seed ratios. The experimental unit for algorithmic variability is an end-to-end training run, not a validation-grid point or checkpoint. Three runs per cell are insufficient to estimate tail failure probabilities or support confirmatory reliability claims. We report magnitudes, paired directions, and adverse runs without asymptotic significance language. Reusing the same three seed labels across conditions does not create 36 independent replications.

Development screens also include GH, moment, antithetic, control-variate, RQMC, and mixture-aware rules. Several outperform W-EXPin selected seed-42 or frozen-continuation diagnostics, especially when low-dimensional distributional structure is directly exploitable. Because these alternatives were screened on one development seed and do not share one information regime, they provide context rather than a formal ranking. Multi-seed Pareto comparisons remain necessary for a competitiveness claim.

Table 4: High-accuracy numerical references

Environment	State grid	Nodes per state	VFI iterations	Bellman residual
Classic Gaussian	4,001	81	183	8.73×10^{-12}
High-vol Gaussian	8,001	61	164	8.27×10^{-12}
Joint income–return	8,001	3,721	183	8.91×10^{-12}
Rare disaster	4,001	162	185	8.26×10^{-12}

Note: The joint reference uses 61×61 tensor Gauss–Hermite nodes. The rare-disaster reference uses 81 nodes in each mixture branch. Reference residuals are below 9×10^{-12} in every environment.

5.3. Mechanism-diagnostic designs

Five auxiliary designs probe competing explanations.

1. A 3×2 SAC routing-by-budget screen compares critic-only, actor-only, and both-routing at two W-EXPstate-label budgets, plus direct MC50. Each cell uses three seeds and 20,000 steps.
2. One-million-step SAC and IAC variants compare routing, warm-up, on-policy sampling, and gradient supervision.
3. Saved W-EXPcheckpoints are evaluated by 21-node Gauss–Hermite quadrature on uniform and on-policy post-state distributions, recording level and savings-gradient errors.
4. Frozen-value experiments hold the continuation function fixed, removing moving-target and actor–critic feedback.
5. A two-seed Krusell–Smith experiment enumerates four discrete transitions exactly and measures both policy error and Bellman-expectation error. It asks whether policy improvements require superior pointwise integration accuracy.

These designs are complementary but not jointly identifying. The 20,000-step route-by-budget screen controls a limited set of short-run contrasts, but route also changes compute and, for SAC, the actor objective. Frozen-value tests remove drift and policy feedback but omit online coupling. Krusell–Smith adds a different model and exact discrete integration but has only two seeds. The long-run variants are selected rather than a balanced factorial design.

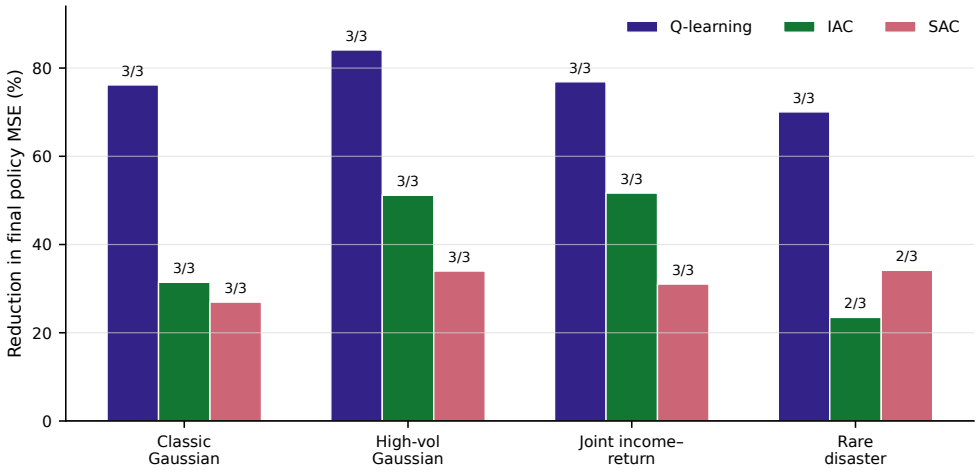
6. Main consumption–saving evidence

6.1. Policy accuracy across environments and solvers

Figure 1 and Table 5 summarize the one-million-outer-step package comparisons. The tuned W-EXP-equipped package has lower mean final policy MSE in all 12 environment–solver cells. Q-learning reductions range from 70.1% to 84.1%; IAC reductions range from 23.5% to 51.7%; and SAC reductions range from 26.9% to 34.1%. These configuration-specific differences combine the adapter with the routing and update schedules in Table 3; they are not isolated effects of expectation replacement.

The paired results qualify the all-cell mean pattern. In the high-vol and joint environments, the W-equipped package finishes below the direct package in all 18 observed seed labels. Under rare disasters, the counts are 3/3 for Q-learning and 2/3 for both IAC and SAC. One adverse path appears in each of the latter cells. These counts describe the observed runs and do not estimate a population win probability.

Figure 1: Fixed-step policy-MSE difference between W-EXP-equipped and direct MC50 packages



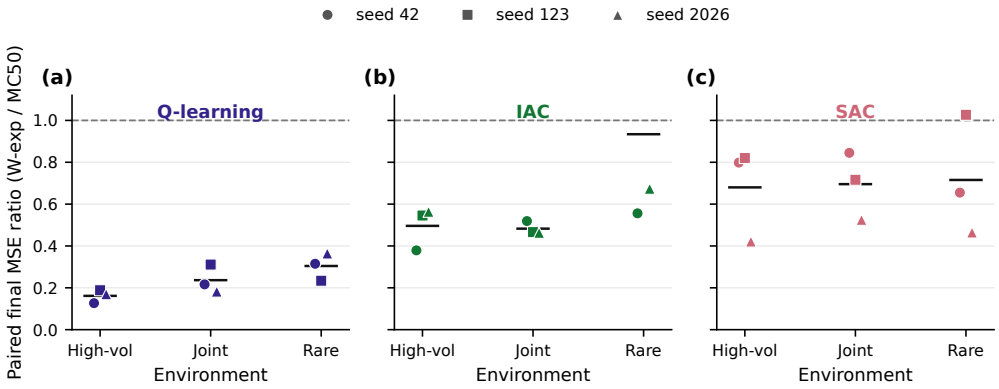
Note: Each bar uses the ratio of mean final errors from three shared seed labels after one million training steps. Labels above bars show the number of seed-label pairs in which W-EXPhas lower final error. The figure contains no statistical confidence intervals; seed-level results for the three stress environments appear in Fig. 2.

Table 5: Primary one-million-step policy-accuracy results

Environment	Solver	Direct MC50 MSE	W-EXPMSE	$\mathcal{R}_{W/MC}$	Reduction
Classic Gaussian	Q-learning	3.343×10^{-4}	7.966×10^{-5}	0.238	76.17%
	IAC	7.020×10^{-6}	4.814×10^{-6}	0.686	31.43%
	SAC	1.653×10^{-4}	1.208×10^{-4}	0.731	26.94%
High-vol Gaussian	Q-learning	5.057×10^{-4}	8.042×10^{-5}	0.159	84.10%
	IAC	8.066×10^{-6}	3.940×10^{-6}	0.488	51.16%
	SAC	1.667×10^{-4}	1.101×10^{-4}	0.660	33.99%
Joint income–return	Q-learning	7.008×10^{-4}	1.620×10^{-4}	0.231	76.88%
	IAC	2.314×10^{-5}	1.119×10^{-5}	0.483	51.65%
	SAC	1.711×10^{-4}	1.180×10^{-4}	0.690	31.03%
Rare disaster	Q-learning	3.126×10^{-4}	9.360×10^{-5}	0.299	70.06%
	IAC	8.889×10^{-6}	6.803×10^{-6}	0.765	23.47%
	SAC	1.737×10^{-4}	1.144×10^{-4}	0.659	34.15%

Note: Values are means over seeds 42, 123, and 2026. The ratio divides the W-EXP-package mean by the direct-package mean. Every W-EXPlabel uses 50 Monte Carlo draws. IAC and SAC packages also differ in host-update multiplicity and routing as shown in Table 3; the table is not a matched-compute causal comparison.

Figure 2: Paired-seed error ratios in the stress environments



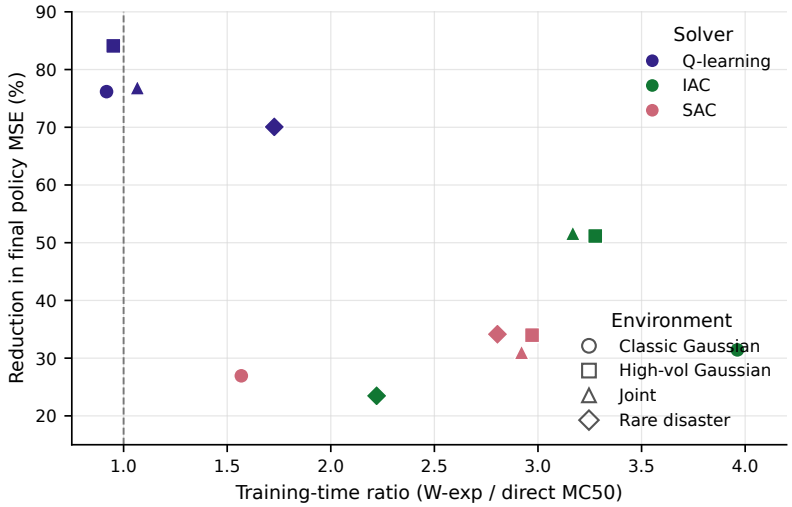
Note: Points are final W-EXP-to-MC50 MSE ratios for shared seed labels; short horizontal segments are within-cell means. The dashed line at one denotes equal error. Marker shape identifies the seed label. All high-vol and joint ratios are below one; rare-disaster IAC and SAC each have two wins in three pairs.

6.2. Accuracy does not imply lower computational cost

The primary experiment fixes training steps, not time. Figure 3 shows the resulting trade-off. Tabular Q-learning is close to computational parity: its time ratios are 0.92, 0.95, 1.07, and 1.73 across the four environments, with the same recorded number of value evaluations as MC50. The neural adapters are more expensive. IAC time ratios range from 2.22 to 3.96 despite value-evaluation ratios near 1.01. SAC time ratios range from 1.57 to 2.97, while recorded value evaluations are approximately equal in the classic and rare-disaster cases and about double in the high-vol and joint cases.

The figure rejects a simple fewer-calls interpretation of the package differences. The neural W-equipped packages can finish with lower policy error while consuming substantially more time, which is a package-level accuracy–cost trade-off at fixed outer steps rather than wall-clock acceleration. The frozen-continuation diagnostics separately provide evidence consistent with statistical information reuse. Fixed-time and matched-optimizer-work benchmarks remain necessary.

Figure 3: Policy-accuracy gains and training-time ratios



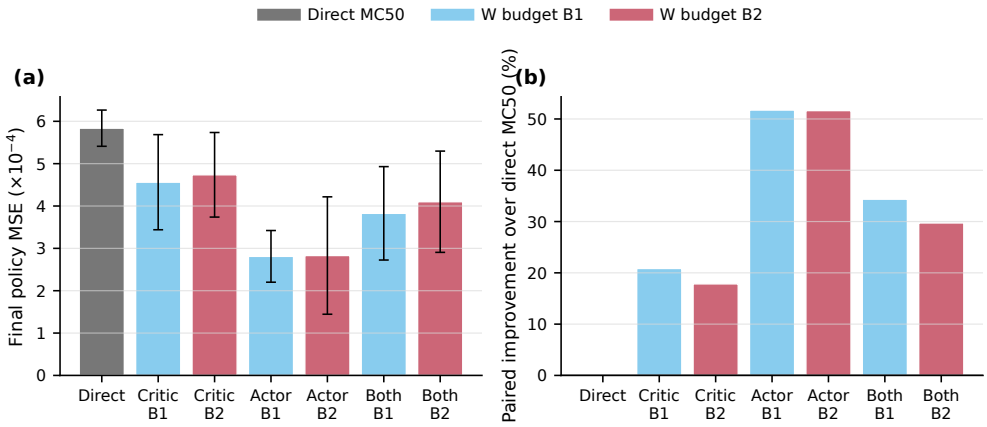
Note: Each point is one solver–environment cell from the one-million-step experiment. The vertical dashed line marks equal mean training time. Ratios compare W-EXPwith direct MC50 within the same solver and environment; they should not be used to rank hardware or solvers across environments.

7. Mechanism diagnostics

7.1. A short SAC screen varies route and label budget

The route-by-budget screen changes where the learned continuation enters SAC while holding the economic environment and 20,000-step horizon fixed. Figure 4 reports the result. Critic-only routing has final MSE 20.9% lower at budget B1 and 17.8% lower at B2 than direct MC50. Actor-only routing is 51.7% lower at both budgets. Both-routing gives intermediate differences of 34.4% and 29.7%. All six W-EXPcells finish below the direct baseline for the three observed seed labels.

Figure 4: SAC routing-by-budget mechanism screen



Note: Panel (a) reports mean final policy MSE with one-standard-deviation error bars across three seeds. Panel (b) reports the mean within-seed-label difference relative to direct MC50. B2 doubles the W-EXPstate-label coverage relative to B1. All cells use 20,000 training steps; the screen records short-horizon configuration differences and is not a replacement for the one-million-step evidence.

Within each budget, actor-only finishes below critic-only for all three seed labels, by 36.7% at B1 and 39.6% at B2 on average. Adding the critic route to actor-only gives a worse final result in every observed seed: the actor-to-both contrast is −34.9% at B1 and −54.1% at B2. Increasing label-state budget from B1 to B2 is not consistently helpful. These contrasts bundle route with compute and signal construction: B1 critic, actor, and both cells use approximately 268.8, 524.8, and 281.6 million value evaluations, respectively. The screen is therefore consistent with route alignment mattering, but it does not isolate a route effect under equal work.

7.2. Selected long runs show solver-dependent route patterns

The longer variants prevent the short SAC screen from being overinterpreted. Table 6 compares selected one-million-step configurations. In SAC, critic-only routing is close to direct MC50, while both-routing with on-policy, value-only training reaches 8.598×10^{-5}, 48.0% below the direct mean. Adding gradient supervision and warm-up raises the mean to 9.604×10^{-5}. In IAC, the ranking is different: critic-only routing reaches 3.754×10^{-6} and is substantially better than both-routing with either value-only or value-plus-gradient training.

Table 6: Selected one-million-step routing and supervision variants

Solver	Continuation use	W-EXPtraining	Mean final policy MSE
SAC	Direct MC50	none	1.653×10^{-4}
	Critic only	on-policy, value	1.599×10^{-4}
	Actor and critic	on-policy, value	8.598×10^{-5}
	Actor and critic	on-policy, value + gradient, warm-up	9.604×10^{-5}
IAC	Direct MC50	none	7.020×10^{-6}
	Critic only	on-policy, value	3.754×10^{-6}
	Actor and critic	on-policy, value	8.440×10^{-6}
	Actor and critic	on-policy, value + gradient	5.502×10^{-6}

Note: Means use three seeds. The table selects mechanism variants from the archived long-run experiments; it is not a balanced factorial table. Comparisons should be made within solver. The IAC direct row is the corresponding selected primary baseline.

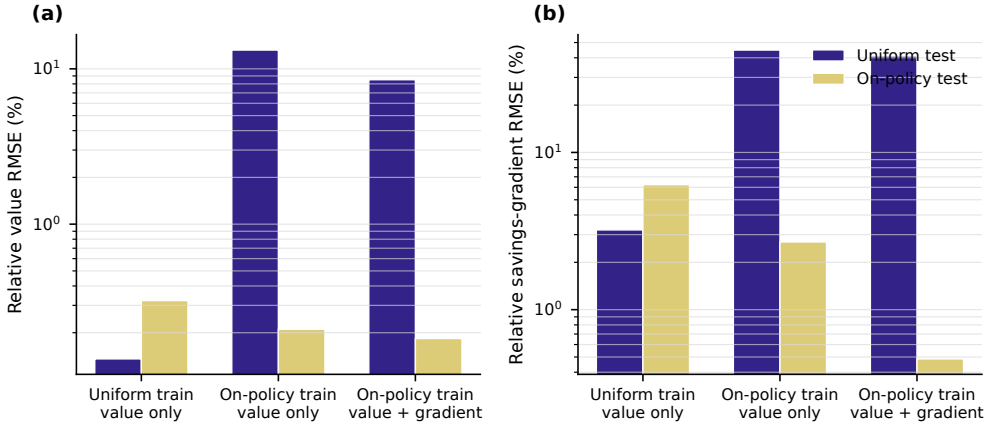
The apparent reversal is suggestive rather than an identified route-by-solver interaction. Actor access is associated with lower error in the short SAC screen, while selected long IAC runs favor critic-only. The table is not balanced across route, support, gradient loss, warm-up, and horizon. A continuation route therefore cannot be ranked independently of the host representation and schedule.

7.3. Training support and derivative quality

Figure 5 evaluates saved SAC layers using a 21-node Gauss–Hermite oracle. Uniform training has relative value RMSE 0.136% on a uniform grid and

0.322% on its on-policy states. An on-policy value-only checkpoint has on-policy value and gradient errors of 0.211% and 2.69%, while its uniform-grid errors are 13.21% and 44.78%. Each on-policy test set is generated by the checkpoint's own actor, so these values show specialization to each observed policy support but do not isolate the training-support intervention.

Figure 5: Distribution support and derivative diagnostics for SAC W-EXPcheckpoints



Note: Bars report relative RMSE against a 21-node Gauss–Hermite oracle; both vertical axes are logarithmic. "Uniform train" and "on-policy train" describe the W-EXPstate-sampling distribution. Value-only versus value-plus-gradient describes the training loss. Uniform and on-policy test bars evaluate the same checkpoint on different supports.

The gradient-supervised checkpoint has on-policy savings-gradient relative RMSE 0.486%, compared with 2.69% for the value-only checkpoint. The configurations also differ in other choices, and the corresponding long-run SAC policy MSE is 11.7% higher for the gradient-supervised variant. The actor-gradient bound in Proposition 2 is one-sided: small derivative error limits one source of distortion but does not determine a coupled optimizer's trajectory. For the on-policy value-only checkpoint, the descriptive target-network gap is 26.9% of level RMSE and 2.6% of gradient RMSE.

7.4. Frozen continuation and policy–Bellman decoupling

Frozen-value experiments remove target drift and policy feedback. In the classic Gaussian case, direct MC50 has continuation RMSE 6.626×10^{-3} and W-EXPhas 2.566×10^{-3}, a 61.3% difference. In the rare-disaster case, the corresponding RMSE changes from 6.552×10^{-3} to 4.368×10^{-3}, a 33.3% difference. In the rare-disaster low-savings region it changes from $1.082 \times$

10^{-2} to 1.783×10^{-3}. This is evidence of information pooling relative to fresh MC50, not a best-method result: mixture-aware GH21 has rare-disaster RMSE 5.259×10^{-5} in the same archived diagnostic, about 83 times below W-EXP, because it uses the known mixture law directly.

The Krusell–Smith diagnostic shows that this channel is not the whole explanation. With four discrete transitions enumerated exactly, W-EXPreduces policy MSE by 42.9% relative to direct exact integration, but its pointwise Bellman-expectation MAE is 1.3% higher. Table 7 records the two metrics.

Table 7: Krusell–Smith policy and Bellman-expectation diagnostics

Method	Policy MSE	Bellman-expectation MAE	Seeds
Direct exact	1.496×10^{-3}	4.984×10^{-3}	2
W-EXPwith exact labels	8.537×10^{-4}	5.048×10^{-3}	2
Relative change	-42.9%	$+1.3\%$	

Note: Every method is evaluated by exact enumeration of the four discrete transitions. The experiment uses 50 heterogeneous households and an IAC solver. It is a two-seed mechanism diagnostic, not part of the 12-cell primary consumption–saving table.

A lower final policy error therefore does not require a lower global level error for the Bellman expectation in these two observed runs. Plausible explanations include altered optimization geometry, smoother updates, and different allocation of approximation accuracy over visited states, but this design does not identify among them. Reporting only continuation fit or training loss is insufficient.

7.5. Mechanism synthesis

Taken together, the experiments provide evidence consistent with four interacting channels.

1. *Statistical information reuse.* Frozen-value tests show lower pointwise RMSE than fresh MC50 for the fitted surface because labels are pooled across updates and neighboring states, although structure-aware quadrature is much better in the tractable disaster diagnostic.
2. *Route alignment.* Actor and critic consume different aspects of the surface. Short SAC and selected long IAC configurations have different route rankings, without an equal-work factorial identification.

3. *Support allocation.* On-policy checkpoints are more accurate on their own visited supports and worse on a uniform grid. Because each variant supplies its own policy distribution, the comparison mixes training and evaluation support.
4. *Optimization-path effects.* Better derivative fit is not sufficient for a better final policy, and two Krusell–Smith runs pair a better policy with slightly worse expectation MAE. The evidence motivates, but does not identify, an optimization-geometry explanation.

The target-network gap is an additional implementation diagnostic. It is nonzero in selected checkpoints, but the experiment does not estimate the terms of Proposition 3 on a common support or verify that recursion.

8. Discussion and limitations

The main contribution of W-EXPis an explicit adapter construction inside an established family. PEA parameterizes expectations in expectational equations; continuation regression and post-decision ADP fit related conditional value objects; precomputation integrates selected bases offline; and bc-MC changes estimation of a squared-expectation loss. W-EXPstudies the combination of an online host-induced post-decision continuation, a common interface instantiated separately within different solvers, explicit actor/critic routing, and joint diagnostics for level, derivative, support, and drift errors.

The implementation evidence answers a limited feasibility question. The adapter contract can be instantiated in a table, a deterministic actor–critic solver, and SAC, and the selected W-equipped packages finish with lower mean policy error than their direct MC50 packages in the 12 observed cells. Because optimizer updates, some routes, and tuning histories are not matched, the main table cannot attribute those differences to the expectation adapter alone. The route and support studies are diagnostic rather than causal. Structure-aware rules also remain decisive: when a low-dimensional shock law is tractable, quadrature or moment-based integration can use that information without fitting an additional function.

Several limitations bound the claims.

1. The primary cells contain three training runs, and the Krusell–Smith diagnostic contains two. The evidence is descriptive and does not characterize tail failure probabilities or support confirmatory reliability claims.

2. Outer steps are matched, but optimizer work and wall-clock time are not. Primary IAC and SAC W packages use additional host updates, and the neural variants are materially slower. Fixed-update, fixed-time, and fixed-value-evaluation frontiers are needed.
3. MC50 is the sole formal three-seed comparator. Stronger quadrature, moment, variance-reduction, and quasi-Monte Carlo alternatives are only single-seed or frozen-function diagnostics, so the study does not establish a competitive method ranking.
4. The four primary models are deliberately controlled consumption–saving environments with $\rho = 0$. They do not test persistent stochastic volatility, time-varying disaster risk, or a high-dimensional aggregate state.
5. High-volatility log income is not mean-preserving in income levels. A $-\sigma^2/2$ correction would separate dispersion from the change in mean income.
6. Configurations were selected after development screens, the same seed labels recur, and common random-number paths are not guaranteed. The retrospective ledger documents the retained search but cannot remove selection bias; new held-out seeds are needed for confirmatory inference.
7. Solver-specific adapters differ in representation, timing, routing, support, and update count. The interface is shared, but weights and hyperparameters are not transferred.
8. The theory consists of conditional perturbation and idealized tracking results. It neither predicts achieved approximation error nor proves global convergence of the nonlinear implementations.
9. Uniform and on-policy diagnostics use selected checkpoints and partly variant-specific supports. They do not establish accuracy on every endogenous distribution or identify a support intervention.
10. The literature comparison is focused rather than systematic. Additional continuation-regression or auxiliary-critic constructions may further narrow the architectural contribution.

The immediate empirical priorities are matched host-update comparisons, more independent held-out seeds, and multi-seed accuracy–cost frontiers against structure-aware integration. A balanced route-by-budget design can then test solver interactions. Higher-dimensional and persistent-shock models are needed only after these identification and precision gaps are closed.

9. Conclusion

W-EXPrepresents the discounted expected continuation of a stochastic dynamic model as a separate post-decision adapter. It belongs to the broad conditional-mean, fitted-continuation, and post-decision-value tradition. Its contribution is the particular interface that maintains the host-induced object online, routes it explicitly to a critic or actor, and makes support and drift part of the diagnosis. The adapter can be tabular or neural, but its weights and configuration are solver-specific.

The conditional theory separates four questions. Uniform level error bounds a contraction-based Bellman perturbation. Local derivative error bounds a deterministic actor-gradient perturbation for a fixed state distribution. An idealized deterministic update exposes approximation and drift terms but does not analyze stochastic-gradient noise. A trained prediction has lower one-call MSE than fresh K-draw Monte Carlo only when its integrated squared error lies below direct sampling variance. The experiments provide diagnostics related to these channels; they do not verify the sufficient conditions throughout nonlinear training.

The resulting claim is limited. A fitted continuation can retain statistical information across queries and can be organized as a routed component of several host solvers. The present package comparisons motivate that design, but they do not establish isolated causality, cross-solver transfer, reliability beyond three runs, or computational superiority. Those questions require the matched and multi-seed experiments listed above.

Appendix A. Proofs

The propositions depend only on Assumption 1 and the additional assumptions stated in their own text. This appendix is the authoritative theory record for the manuscript. The older `wexp_amortized_expectations.tex` file in the proof-source directory is a superseded development draft with a different scaling convention, and `bellman_error_ks_integration_plan.md` is an implementation plan rather than a proof.

Proof of Proposition 1. For any state s, let $F(a) = u(s,a) + C_V(h(s,a))$ and $\widetilde{F}(a) = u(s,a) + W_V(h(s,a))$. The elementary inequality

$$\left|\sup_a \widetilde{F}(a) - \sup_a F(a)\right| \le \sup_a \left|\widetilde{F}(a) - F(a)\right|$$

and Eq. (11) imply $\left|(\widetilde{T}V)(s) - (TV)(s)\right| \le \epsilon_0$. Taking the supremum over s gives the operator bound. For the fixed points,

$$\begin{aligned}\left\|\widetilde{V} - V^*\right\|_\infty &= \left\|\widetilde{T}\widetilde{V} - TV^*\right\|_\infty \\ &\le \left\|\widetilde{T}\widetilde{V} - T\widetilde{V}\right\|_\infty + \left\|T\widetilde{V} - TV^*\right\|_\infty \\ &\le \epsilon_0 + \beta\left\|\widetilde{V} - V^*\right\|_\infty .\end{aligned}$$

Rearrangement yields Eq. (12). If W approximates the unscaled G, the target difference is multiplied by β. □

Proof of Proposition 2. The utility term is identical in J_W and J_C and cancels in their gradient difference. The domination assumption permits differentiation under the d-expectation. The chain rule then gives

$$\begin{aligned}\nabla_\phi J_W - \nabla_\phi J_C = \mathbb{E}_{s\sim d}\big[&D_\phi \mu_\phi(s)^\top D_a h(s, \mu_\phi(s))^\top \\ &\times \{\nabla_x W(x_\phi(s)) - \nabla_x C(x_\phi(s))\}\big],\end{aligned}$$

where $x_\phi(s) = h(s, \mu_\phi(s))$. Applying the triangle inequality and the two sup-norm bounds in Eq. (14) gives

$$\|\nabla_\phi J_W - \nabla_\phi J_C\| \le \mathbb{E}_{s\sim d}[L_{h\mu}\epsilon_1] = L_{h\mu}\epsilon_1.$$

□

Proof of Proposition 3. Subtract C_{t+1} from Eq. (16), add and subtract $(1-\eta)C_t + \eta C_t$, and apply the triangle inequality:

$$\begin{aligned}e_{t+1} &\le (1-\eta)\,\|W_t - C_t\| + \eta\left\|\widetilde{C}_t - C_t\right\| + \|C_{t+1} - C_t\| \\ &\le (1-\eta)e_t + \eta\rho_t + \delta_t.\end{aligned}$$

If $\rho_t \le \rho$ and $\delta_t \le \delta$, iterating the recursion yields

$$e_t \le (1-\eta)^t e_0 + (\eta\rho + \delta)\sum_{j=0}^{t-1}(1-\eta)^j.$$

The geometric sum converges to $1/\eta$, giving Eq. (19). □

Proof of Proposition 4. Conditional iid sampling, square integrability, and independence from the fixed training history give

$$\mathbb{E}[\widehat{C}^{\mathrm{MC}}_{\vartheta,K}(x) \mid x, \vartheta, \omega] = C_\vartheta(x), \qquad \mathrm{Var}(\widehat{C}^{\mathrm{MC}}_{\vartheta,K}(x) \mid x, \vartheta, \omega) = \frac{\beta^2 \sigma_g^2(x)}{K}.$$

Unbiasedness makes conditional MSE equal variance. Conditional on the trained parameters, $W_\omega(x)$ is a deterministic prediction, so its pointwise squared error is $e_W(x)^2$. Integrating both expressions over the fixed d_x and comparing gives Eq. (22). □

Appendix B. Implementation and reproducibility details

Table B.8: Core implementation details

Component	Representation	W-EXPintegration
Q-learning	Q table and value table on wealth, income, and action grids	A savings–income W table; stochastic updates toward discounted expectation labels; linear interpolation in savings
IAC	64–64 tanh actor and state-value critic	64–64 tanh post-decision network; uniform, on-policy, or mixed state sampling; optional actor/critic routing and savings-gradient loss
SAC	256–256 SiLU stochastic actor and twin Q critics	256–256 SiLU post-decision network; Adam with initial learning rate 10^{-3}; Polyak target coefficient 0.01; optional routing, warm-up, blending, and gradient loss

Note: These are the canonical representations in the archived implementations. Individual mechanism variants change update frequency, label-state budget, routing, or warm-up as recorded in their JSON configurations.

The figure generator asserts the expected row counts and the positivity of the 12 mean primary improvements before plotting. It writes both PDF and 300-dpi PNG files and a machine-readable figure manifest that maps each figure to its source table and limitation. Figures contain axis labels and legends but no embedded titles; captions and notes are supplied by the manuscript.

The source map below identifies the immediate inputs, not a complete computational archive. The figure manifest records input/output hashes, the rebuild command, and software versions. Some upstream report builders still contain hard-coded derived endpoints, and the million-step jobs were audited from retained results rather than rerun for this revision. A public archive and frozen end-to-end training environment remain pre-submission tasks recorded in the accompanying backlog and provenance note.

Appendix C. Source-to-claim map

Table C.9: Project artifacts supporting the principal empirical claims

Claim or exhibit	Source artifact	Scope
Twelve primary CS comparisons	`report/V2/data/primary_four_models.csv`	Three seeds, one million steps, four environments, three solvers
Classic-Gaussian seed pairs	`paper/V1/data/primary_classic_seed_rows.csv`	Nine pairs with exact raw-summary paths
Paired stress-seed ratios	`report/V2/data/paired_new_models.csv`	27 paired rows
SAC routing-by-budget screen	`mechanism_identification_20260901/results/analysis/new_role_budget_summary.csv`	Seven cells, three seeds, 20,000 steps
Factorial routing contrasts	`mechanism_identification_20260901/results/analysis/new_factorial_contrasts.csv`	Within-seed final and tail contrasts
SAC checkpoint diagnostics	`mechanism_identification_20260901/results/analysis/checkpoint_diagnostics_summary.csv`	Uniform/on-policy level, gradient, and target-lag errors
SAC selected long-run diagnostics	`mechanism_identification_20260901/results/analysis/sac_1m_method_metrics.csv`	Three seeds, one million steps
IAC selected long-run diagnostics	`mechanism_identification_20260901/results/analysis/iac_1m_mechanism_summary.csv`	Three seeds, one million steps
Krusell–Smith decoupling	`mechanism_identification_20260901/results/analysis/ks_summary.csv`	Two seeds, exact four-outcome evaluation
Frozen-value diagnostics	`report/V2/data/frozen_v_summary.csv` and `report/V2/data/rare_frozen_v_estimators.csv`	Fixed continuation function
Configuration selection	`paper/V1/TUNING_SELECTION_LEDGER.md`	Retrospective retained-run inventory; not preregistration
Figure provenance	`paper/V1/figure_manifest.json`	Commands, versions, SHA-256 input/output hashes
Provenance and rebuild note	`paper/V1/ARTIFACT_PROVENANCE.md`	Commands, immediate inputs, and unresolved archive boundary

Note: Raw run folders, histories, policies, and configurations are retained elsewhere in the same repository. Figure inputs and outputs are hashed; the full upstream training chain is not yet a frozen public archive.

References

Azinovic, M., Gaegauf, L., Scheidegger, S., 2022. Deep equilibrium nets. International Economic Review 63, 1471–1525. doi:10.1111/iere.12575.

Den Haan, W.J., 2010. Comparison of solutions to the incomplete markets

model with aggregate uncertainty. Journal of Economic Dynamics and Control 34, 4–27. doi:10.1016/j.jedc.2008.12.010.

Den Haan, W.J., Marcet, A., 1990. Solving the stochastic growth model by parameterizing expectations. Journal of Business & Economic Statistics 8, 31–34. doi:10.1080/07350015.1990.10509770.

Duffy, J., McNelis, P.D., 2001. Approximating and simulating the stochastic growth model: Parameterized expectations, neural networks, and the genetic algorithm. Journal of Economic Dynamics and Control 25, 1273–1303. doi:10.1016/S0165-1889(99)00077-9.

Fernández-Villaverde, J., Levintal, O., 2018. Solution methods for models with rare disasters. Quantitative Economics 9, 903–944. doi:10.3982/QE744.

Haarnoja, T., Zhou, A., Abbeel, P., Levine, S., 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: Proceedings of the 35th International Conference on Machine Learning, pp. 1861–1870.

Han, J., Yang, Y., E, W., 2026. DeepHAM: A global solution method for heterogeneous agent models with aggregate shocks. Quantitative Economics 17, 297–341. doi:10.3982/QE2190.

Hull, I., 2015. Approximate dynamic programming with post-decision states as a solution method for dynamic economic models. Journal of Economic Dynamics and Control 55, 57–70. doi:10.1016/j.jedc.2015.03.008.

Huré, C., Pham, H., Bachouch, A., Langrené, N., 2021. Deep neural networks algorithms for stochastic control problems on finite horizon: Convergence analysis. SIAM Journal on Numerical Analysis 59, 525–557. doi:10.1137/20M1316640.

Judd, K.L., 1998. Numerical Methods in Economics. MIT Press.

Judd, K.L., Maliar, L., Maliar, S., 2011. Numerically stable and accurate stochastic simulation approaches for solving dynamic economic models. Quantitative Economics 2, 173–210. doi:10.3982/QE14.

Judd, K.L., Maliar, L., Maliar, S., Tsener, I., 2017. How to solve dynamic stochastic models computing expectations just once. Quantitative Economics 8, 851–893. doi:10.3982/QE329.

Krusell, P., Smith, A.A., 1998. Income and wealth heterogeneity in the macroeconomy. Journal of Political Economy 106, 867–896. doi:10.1086/250034.

Longstaff, F.A., Schwartz, E.S., 2001. Valuing american options by simulation: A simple least-squares approach. The Review of Financial Studies 14, 113–147. doi:10.1093/rfs/14.1.113.

Maliar, L., Maliar, S., Winant, P., 2021. Deep learning for solving dynamic economic models. Journal of Monetary Economics 122, 76–101. doi:10.1016/j.jmoneco.2021.07.004.

Pascal, J., 2024. Artificial neural networks to solve dynamic programming problems: A bias-corrected Monte Carlo operator. Journal of Economic Dynamics and Control 162, 104853. doi:10.1016/j.jedc.2024.104853.

Pascal, J., 2026. A generalization of the parameterized expectations algorithm. Economics Letters 259, 112790. doi:10.1016/j.econlet.2025.112790.

Powell, W.B., 2011. Approximate Dynamic Programming: Solving the Curses of Dimensionality. 2 ed., Wiley.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M., 2014. Deterministic policy gradient algorithms, in: Proceedings of the 31st International Conference on Machine Learning, pp. 387–395.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. 2 ed., MIT Press.

Tsitsiklis, J.N., Van Roy, B., 1997. An analysis of temporal-difference learning with function approximation. IEEE Transactions on Automatic Control 42, 674–690. doi:10.1109/9.580874.

Tsitsiklis, J.N., Van Roy, B., 2001. Regression methods for pricing complex american-style options. IEEE Transactions on Neural Networks 12, 694–703. doi:10.1109/72.935083.

Watkins, C.J.C.H., Dayan, P., 1992. Q-learning. Machine Learning 8, 279–292. doi:10.1007/BF00992698.