Neural Networks for Efficient Estimation of High-Dimensional Dynamic Discrete Choice Models*

Hoang Nguyen†
Department of Economics, Georgetown University

January 10, 2026

Abstract

We propose the *Neural-Network Efficient Estimator* (NNES) for structural dynamic discrete choice models with high-dimensional state vectors. NNES replaces the policy evaluation step of Aguirregabiria and Mira [2002], which is expensive in high-dimensional settings, with a neural-network regression that minimises a penalized likelihood function. We prove that (i) the policy-iteration map retains a *zero Jacobian property*, (ii) the resulting likelihood score is *Neyman Orthogonal*, and therefore (iii) the estimator is $\sqrt{n}$-consistent and semiparametrically efficient while the information matrix remains block-diagonal. Here, n is the sample size. These properties hold under network approximation rates of order $o(n^{-1/4})$, attainable with over-parameterised Neural Nets for certain class of functions. We provide simulation evidence showing that NNES delivers the same precision as full-information maximum likelihood, demonstrating its attractiveness in high-dimensional settings.

1 Introduction

Dynamic discrete-choice (DDC) models are a cornerstone of modern structural econometrics, providing a framework to analyze forward-looking agents making sequential decisions under uncertainty. Although these models are fully parametric and, in principle, estimable by maximum likelihood, the dependence of the likelihood on the parameter of interest is not known a priori. This complexity arises because the likelihood depends on conditional choice probabilities, which in turn depend on value functions that are implicit functions of the parameter. While standard dynamic programming and implicit-function arguments ensure that the value functions vary smoothly with the parameter of interest, their functional form is not known a priori. The canonical nested fixed-point (NFXP) algorithm pioneered by Rust [1987] addresses this by repeatedly solving for fixed-point of the value function numerically at each candidate parameter vector. This approach is computationally feasible in models with small state spaces. However, in problems involving a large number of discrete states or continuous state variables, the value function becomes a high- or even infinite-dimensional object. In such settings, this approach is subject to the curse of dimensionality, where the computational cost grows exponentially with dimension, rendering it impractical for many modern applications.

*I am deeply grateful to John Rust, Dan Cao, and Whitney Newey for their guidance and support. I am also thankful to Alexandre Poirier, Ivana Komunjer, Marinho Bertanha, Wenlan Luo, Jeremy Rosen, Yao Luo, Amilcar Velez, Vira Semenova, Enoch Kang, Fedor Iskhakov, Zhaonan Qu, Kevin Chen, Han Xu, Ahnaf Rafi, Matthew Shum, Anna Bykhovskaya and Jiaying Gu for valuable comments and suggestions. Any and all errors are my own.

†Email: hhn10@georgetown.edu

To overcome this computational challenge, Hotz and Miller [1993] developed the *conditional choice probability* (CCP) approach, showing how one can invert observed choice probabilities to recover structural primitives without re-solving the dynamic program at each step. Building on CCPs, Aguirregabiria and Mira [2002] introduced the *nested pseudo-likelihood* (NPL) estimator, which iterates directly on CCPs rather than value functions, reducing per-iteration cost and enjoying a *zero-Jacobian property*. Thanks to this property, the NPL estimator's information matrix is block diagonal between the structural parameters in the utility function and the CCP pseudo parameters that are non-parametrically estimated in the first step and hence often quite noisy estimates. But the zero Jacobian property insulates the covariance of the structural parameters from the noise in the CCPs. Further identification and inference results for CCP-based estimators were developed by Magnac and Thesmar [2002] and Pesendorfer and Schmidt-Dengler [2008], while Arcidiacono and Miller [2011] extended CCP methods to accommodate unobserved heterogeneity and continuous state spaces via sieve approximations. However, these foundational methods typically rely on discretizing the state space or using low-order sieve approximations, which are themselves subject to the curse of dimensionality and are impractical for truly high-dimensional settings.

The recent integration of machine learning (ML) offers a promising path forward. Neural networks, with their well-documented approximation capabilities, can flexibly represent policy or value functions in high dimensions without requiring grids. While shallow networks are universal approximators (e.g., Barron [1993]), recent theoretical advances suggest that deep networks can also be particularly effective. For instance, Yarotsky [2017] establishes that deep networks with rectified linear unit (ReLU) activation functions can efficiently approximate functions from standard smoothness classes (e.g., Sobolev spaces), providing a foundation for their use with weaker assumptions than those required for the Barron class. However, naive "plug-in" ML methods lead to biased estimates of structural parameters. This bias arises from the regularization and overfitting inherent in many ML estimators and can contaminate the final estimates, preventing $\sqrt{n}$-consistency and valid inference (noted in Chernozhukov et al. [2022] and Chernozhukov et al. [2024a]). State-of-the-art solutions, such as the double/debiased machine learning (DML) framework Chernozhukov et al. [2022], Chernozhukov et al. [2024a], Adusumilli and Eckardt [2019], Farrell [2015] and Semenova [2018] correct this bias by constructing a Neyman-orthogonal score function. While theoretically sound, this often involves adding a complex influence function correction term, which can inflate the estimator's variance in finite samples and does not exploit any special structure of the economic model.

This paper proposes a different path. We demonstrate that for the canonical class of DDC models with additive, Type-I extreme value shocks, the path to first-order orthogonality does not require an additive bias correction term. The model's intrinsic structure, when leveraged correctly, provides a direct route to efficiency. We introduce the Neural-Network Efficient Estimator (NNES), an estimator that integrates neural network approximation into the NPL framework in a way that preserves the zero-Jacobian property. We show that the derivative of the policy-iteration map with respect to the Value function vanishes at the fixed-point. This property is particularly crucial in the modern machine learning context, as it allows NNES to accommodate overparameterized neural networks—where the number of parameters can exceed the sample size—without computational or statistical instability. It is also important to note that Aguirregabiria and Mira [2002] primarily consider a finite state space, where the conditional choice probabilities (CCPs) can be estimated non-parametrically at the standard $\sqrt{n}$ rate. In our work, which addresses continuous state spaces, non-parametric estimation of the CCPs necessarily occurs at a slower rate. This distinction underscores the importance of the Neyman orthogonality property. Orthogonality ensures that as long

as the first-stage CCP estimator converges faster than $n^{-1/4}$, its slower non-parametric rate does not contaminate the $\sqrt{n}$-consistency of the final structural parameter estimates.

Luo and Sang [2025] propose the SEES estimator, which extends the MPEC estimator in Su and Judd [2012] by approximating the equilibrium mapping by a finite-dimensional sieve and enforces the fixed-point condition via a large penalty. SEES attains $\sqrt{n}$-consistency and asymptotic efficiency, but the information matrix retains non-zero off-diagonal blocks. Consequently,the estimation of the variance requires inversion of large Hessians, unstable cross-derivative terms, which can be ill-conditioned in high-dimensional settings, especially when the nuisance parameter vector is large. Next, the sieve approximation suffers from the curse of dimensionality: to achieve a uniform approximation error of order $O(n^{-\alpha})$ in a d-dimensional state space, the number of basis functions K must grow at a rate that is at least exponential in d (Chen [2007]). This makes the optimization over the high-dimensional vector of sieve coefficients computationally challenging for large d. NNES, by contrast, is architecturally designed to preserve the block-diagonal information matrix structure of NPL, thereby avoiding this computational and statistical instability, even in the overparameterized regime.

In parallel, reinforcement-learning (RL) and empirical-risk-minimization (ERM) methods have been imported into structural econometrics. Adusumilli and Eckardt [2019] adapt temporal-difference (TD) learning and fitted value iteration to estimate DDC models without explicit transition-density estimation, achieving fast computation in moderately high dimensions. Foundational RL analyses (e.g., Tsitsiklis and Van Roy [1996]; Munos and Szepesvári [2008]) provide finite-sample convergence guarantees under linear-basis assumptions. More recently, Kang et al. [2025] cast DDC estimation as an ERM problem that jointly matches CCPs and minimizes the Bellman residual via gradient methods, establishing global convergence under Polyak–Łojasiewicz conditions for a broad class of function parameterizations, including neural networks. These methods are model-free since they do not require estimation of the transition densities. The NNES estimator is model-based, as it requires models of the state transition dynamics to compute the expected value function in the policy evaluation step. However, the NNES framework can potentially be extended to a model-free setting. Investigating the theoretical properties of such a model-free NNES estimator is an interesting direction for future research.

1.1 Contributions

We introduce the *Neural-Network Efficient Estimator* (NNES) estimator with:

1. **High-Dimensional Policy Evaluation with Neural Networks.** We replace grids and low-order sieves with a deep neural network for the value function, enabling policy evaluation in high-dimensional state spaces. Under standard Hölder/Sobolev smoothness, classical approximation results (e.g., Yarotsky [2017]; Langer [2021]) imply that ReLU or sigmoid networks can attain the $o(n^{-1/4})$ first-stage accuracy required for our asymptotics with logarithmic depth and polynomial size. This delivers scalable policy evaluation while preserving the rates needed for valid inference. In addition, we highlight the necessity of deep neural networks in the NNES framework, in contrast to shallow architectures. Depth delivers the required approximation accuracy with a parameter vector whose dimension grows slowly enough with n for the induced hypothesis class to be P-Donsker. This property underpins the empirical-process CLT, yielding root-n-consistency and attainment of the semiparametric efficiency bound for NNES.

2. **Zero-Jacobian preservation.** Extends the zero-Jacobian property of the policy-iteration map (Aguirregabiria and Mira [2002]) to the network-parameterized setting, showing that the derivative of the policy iteration and evaluation operators with respect to the Value function vanishes at the fixed point.

3. **Automatic Neyman Orthogonality and Simple, Efficient Inference.** The zero-Jacobian property directly implies that the likelihood score is Neyman-orthogonal to the first-stage network estimation error (in the sense of Chernozhukov et al. [2024a]). Consequently, NNES is $\sqrt{n}$-consistent and semiparametrically efficient under a mild network approximation rate of order $o(n^{-1/4})$. Interestingly, this efficiency is achieved without an explicit bias correction term (e.g., Newey [1994], Chernozhukov and Hansen [2004], Chernozhukov et al. [2022], Chernozhukov et al. [2023], Chernozhukov et al. [2024b], Chernozhukov et al. [2025]), avoiding potential variance inflation. Furthermore, the information matrix is block diagonal between the structural parameters in the utility function and the CCP pseudo and *value function* parameters that are non-parametrically estimated in the first step and hence often quite noisy estimates, ensuring that variance estimation is simple and stable even in high dimensions. In addition, we relax the linear-in-parameters utility assumption used in DML orthogonal-score constructions for DDC (e.g., Chernozhukov et al. [2022, 2023]) and establish orthogonality for general utility specifications.

NNES thus inherits the statistical efficiency and inferential simplicity of NPL while leveraging the power of neural networks to overcome the curse of dimensionality. Our simulation evidence confirms that NNES achieves the same precision as an oracle NFXP estimator, demonstrating its power and attractiveness for modern empirical research.

1.2 Outline

The rest of the paper is organized as follows. Section 2 presents the model and notation. Section 3 details the NNES algorithm. Section 4 develops the theoretical properties: zero-Jacobian, Neyman orthogonality, and asymptotic normality. Finally, Section 6 discusses implementation and simulation evidence.

2 Model and Notation

Environment. Time is discrete and infinite: $t = 0, 1, \ldots$. In each period the agent observes a *state* $X_t \in \mathcal{X} \subset \mathbb{R}^d$ and chooses one of $J \geq 2$ discrete actions $A_t \in \mathcal{A} := \{1, \ldots, J\}$. The per-period utility of choosing a in state x is

$$U_a(x; \theta_0, \varepsilon_t) \;=\; u_a(x; \theta_0) \;+\; \varepsilon_{a,t}, \qquad a \in \mathcal{A},$$

where:

- $u_a(\cdot; \theta)$ is the reward function, smooth in $(x, \theta) \in \mathcal{X} \times \Theta$;
- $\theta_0 \in \Theta \subset \mathbb{R}^{d_\theta}$ is the unknown structural parameter vector;
- $\varepsilon_t := (\varepsilon_{1,t}, \ldots, \varepsilon_{J,t})$ are normalized i.i.d. type-I extreme-value (T1EV) shocks, independent of $\{X_s\}_{s \leq t}$ and of past shocks.

State dynamics. Conditional on (X_t, A_t) the next state is drawn from a Markov kernel $f_{\kappa_0}(x' \mid X_t, A_t)$ satisfying $\sup_{(x,a)} \int_{\mathcal{X}} f_{\kappa_0}(x' \mid x, a)\, dx' < \infty$, where κ_0 are the parameters of the transition density. Assume the induced Markov chain on $\mathcal{X}$ is ergodic, with unique stationary distribution π_{κ_0}.

Value function and Bellman equation. Let $\beta \in (0,1)$ be the discount factor. The *choice-specific value functions* are

$$Q_a^*(x) \;=\; u_a(x;\theta_0) \;+\; \beta\, \mathbb{E}\big[V^*(X_{t+1}) \mid X_t = x,\, A_t = a\big],$$

and the ex-ante value function solves the Bellman equation

$$V^*(x) \;=\; \mathbb{E}_\varepsilon\big[\max_{a\in\mathcal{A}}\{Q_a^*(x) + \varepsilon_a\}\big].$$

Under the T1EV shocks, Rust [1987] shows that the value function satisfies the closed-form representation:

$$V^*(x) \;=\; \log\Big(\sum_{a=1}^{J} \exp Q_a^*(x)\Big), \tag{1}$$

Since ε_t are normalized, the Euler constant γ_{EM} is canceled out.

Conditional choice probabilities (CCPs). The optimal CCPs are

$$P^*(a \mid x) \;=\; \frac{\exp Q_a^*(x)}{\sum_{j=1}^{J} \exp Q_j^*(x)}, \qquad a \in \mathcal{A}. \tag{2}$$

We collect them in the vector $P^*(\cdot \mid x) \in \Delta^{J-1}$ and write P^* for the associated measurable map $x \mapsto (P^*(1 \mid x), \ldots, P^*(J \mid x))$.

Policy-iteration operator. For any admissible CCP-vector P define the *evaluation* operator

$$(\varphi_\theta[P,V])(x) \;=\; \sum_{a\in\mathcal{A}} P(a \mid x)\Big\{u_a(x;\theta) + \beta\, \mathbb{E}[V(X') \mid x, a] - \log P(a \mid x)\Big\}. \tag{3}$$

and the *improvement* operator

$$\Lambda_\theta[V]\,(a \mid x) \;=\; \frac{\exp\big\{u_a(x;\theta) + \beta\, \mathbb{E}[V(X') \mid x, a]\big\}}{\sum_j \exp\big\{u_j(x;\theta) + \beta\, \mathbb{E}[V(X') \mid x, j]\big\}}. \tag{4}$$

The composite map $\Psi_\theta := \Lambda_\theta \circ \varphi_\theta$ acts on the space of CCPs; a fixed point, $P = \Psi_\theta(P)$, solves the dynamic program Aguirregabiria and Mira [2002].

Define the ex-ante value function as an *implicit function* of both the structural parameter and the CCP rule:

$$V_{\theta,P}(x) \;=\; \varphi_\theta[P,\, V_{\theta,P}](x), \qquad x \in \mathcal{X}. \tag{5}$$

For any admissible P, $\varphi_\theta[P,\cdot]$ is affine in V. Writing, for any bounded V,

$$c_{\theta,P}(x) \;\equiv\; \sum_{a\in\mathcal{A}} P(a|x)\{u_a(x;\theta) - \log P(a|x)\}, \qquad (T_P V)(x) \;\equiv\; \sum_{a\in\mathcal{A}} P(a|x)\, \mathbb{E}\big[V(X') \mid x, a\big]\,,$$

we have the decomposition

$$\varphi_\theta[P,V](x) \;=\; c_{\theta,P}(x) \;+\; \beta\,(T_P V)(x) \tag{6}$$

Denote $\mathcal{C}(\mathcal{X})$ as the space of continuous real-valued functions on $\mathcal{X}$, equipped with the supremum norm $\|\cdot\|_\infty$. By standard Dynamic Programming argument in Rust [1987], Aguirregabiria and Mira [2002], the map $V \mapsto \varphi_\theta[P,V]$ is a β-contraction on $(\mathcal{C}(\mathcal{X}), \|\cdot\|_\infty)$. Hence 5 admits a unique solution and, in the continuous-state case, is equivalently the linear functional equation:

$$(\mathrm{I} - \beta T_P)\, V_{\theta,P} \;=\; c_{\theta,P} \quad\Rightarrow\quad V_{\theta,P} \;=\; (\mathrm{I} - \beta T_P)^{-1} c_{\theta,P} \;=\; \sum_{t\geq 0} \beta^t T_P^t c_{\theta,P}. \tag{7}$$

Here T_P is a bounded linear *integral* operator, and thus $\mathrm{I} - \beta T_P$ is invertible. This is exactly the continuous-state analogue of the finite-state linear system highlighted by Aguirregabiria and Mira [2002]; only sums over states are replaced by integrals/conditional expectations. It reduces, when the Type-I extreme-value shocks are absent so that the optimal CCP degenerates on the maximizing alternative, to the ordinary policy-valuation operator for the deterministic policy that picks $\arg\max_a Q_a(x)$. Then the fixed-point reduces to the standard policy evaluation equation

$$V(x) \;=\; \max_{a\in A} \left\{ u_a(x;\theta) + \beta\, \mathbb{E}[V(X') \mid x, a] \right\}.$$

Thus, 3 generalizes the usual policy-evaluation operator to the logit DDC setting.

Smoothness in (θ, P) via the Implicit Function Theorem. Define the residual map:

$$r(\theta, P, V) := V - \varphi_\theta[P,V]$$

Its Fréchet derivative w.r.t. V is $D_V r(\theta,P,V) = \mathrm{I} - \beta T_P$, which is boundedly invertible on $\mathcal{C}(\mathcal{X})$. Since the utility function is smooth in θ, $\varphi_\theta[P,V]$ is continuously Fréchet-differentiable in (θ,P,V), and the Implicit Function Theorem yields a continuously Fréchet-differentiable solution map $(\theta,P) \mapsto V_{\theta,P}$. In particular, the partial derivative at fixed P solves the *gradient Bellman equation*

$$(\mathrm{I} - \beta T_P)\, \partial_\theta V_{\theta,P} \;=\; \sum_{a\in\mathcal{A}} P(a|\cdot)\, \partial_\theta u_a(\cdot;\theta), \qquad \partial_\theta V_{\theta,P} \;=\; (\mathrm{I} - \beta T_P)^{-1}\Big[\sum_a P(a|\cdot)\, \partial_\theta u_a(\cdot;\theta)\Big]. \tag{8}$$

Parameter of interest versus nuisance. Throughout the paper, θ is the structural parameter of interest, whereas P is a nuisance *for estimation*. At the true solution, both P and V are smooth functions of θ along the equilibrium manifold:

$$V_\theta \;=\; \varphi_\theta[P_\theta, V_\theta], \qquad P_\theta \;=\; \Lambda_\theta[V_\theta].$$

Thus, V is not a nuisance at the truth; it is a smooth function of θ defined implicitly by the fixed point.

An oracle one–step Nested Pseudo-Likelihood and its continuous–state extension. When the conditional choice probabilities used in the evaluation step coincide with the optimal CCPs at the truth, $P = P^*$ at $\theta = \theta_0$, the resulting *oracle* policy evaluation V_{θ,P^*} yields the same choice probabilities as the model itself, $\Lambda_{\theta_0}[V_{\theta_0,P^*}] = P^*$. In that (infeasible) case, maximizing the pseudo-likelihood formed from $\Lambda_\theta[V_{\theta,P^*}]$ delivers a consistent and efficient estimator for θ_0 within the partial-likelihood framework.

The practical analogue replaces P^* with a nonparametric estimate $\widehat{P}$ obtained from data. Because $V_{\theta,P}$ is a smooth function of (θ, P) by the Implicit Function Theorem (equation 8), a feasible *one-step* estimator that profiles the pseudo-likelihood through $V_{\theta,\widehat{P}}$ is consistent provided $\widehat{P}$ converges to P^* sufficiently fast; the key requirement is that the first-stage perturbation $\widehat{P} - P^*$ be orthogonal to the score for θ at the truth (a property shown by Aguirregabiria and Mira [2002] in finite state spaces and established in this paper for continuous states under our regularity and approximation conditions). This explains why, asymptotically, one iteration already targets the efficient limit, even when $\widehat{P}$ is learned at a nonparametric rate.

In finite samples, however, a single plug-in $\widehat{P}$ can be noisy. Iterating the policy-improvement mapping $P \mapsto \Psi_\theta(P) = \Lambda_\theta \circ \varphi_\theta[P, V]$ partial out this noise through the model: each iteration evaluates $V_{\theta,P}$ via 3 and updates the implied CCPs via 4, progressively aligning $\widehat{P}$ with model-consistent probabilities. In the finite-state analysis of Aguirregabiria and Mira [2002], these iterations converge to the NFXP partial MLE; Our proposed estimator (NNES) implements the same idea with continuous states by solving the valuation step with flexible neural networks that approximate $V_{\theta,P}$ via the linear functional equation, avoiding grids while preserving the likelihood-based updating. See 2.2 for a detailed comparison with NPL and SEES.

2.1 Primitive Assumptions and Zero–Jacobian Property

Assumption 1 (Additivity)**.** *The one–period utility is additively separable in observable and unobservable components:* $U_a(x; \theta_0, \varepsilon_t) = u_a(x; \theta_0) + \varepsilon_{a,t}$.

Assumption 2 (Conditional Independence)**.** *The state transition factors as* $p(X_{t+1}, \varepsilon_{t+1} \mid X_t, A_t, \varepsilon_t) = g(\varepsilon_{t+1} \mid X_{t+1})\, f_{\kappa_0}(X_{t+1} \mid X_t, A_t)$ *with* g *possessing finite first moments and two continuous derivatives in* ε.

Proposition 1 in Aguirregabiria and Mira [2002] shows that under assumptions 1–2, the smoothed Bellman operator is a contraction and has a unique fixed point $\{V^*, Q^*, P^*\}$.

Proposition 1 (Zero Jacobian for φ_θ and Ψ_θ)**.** *Under Assumptions 1–2:*

The Jacobian matrices $\partial\varphi_{\theta_0}(P)/\partial P$ *and* $\partial\Psi(P)/\partial P$ *evaluate to the zero matrix at* $P = P^*$*:*

$$\frac{\partial \varphi_{\theta_0}(P, V)}{\partial P}\Big|_{P=P^*, V=V^*} = 0, \qquad \frac{\partial \Psi_{\theta_0}(P)}{\partial P}\Big|_{P=P^*} = 0,$$

Proposition 1 is equivalent to Proposition 2 in Aguirregabiria and Mira [2002]. It establishes that at the fixed point it is not possible to increase expected utility by changing choice probabilities; that is, the optimal choice probabilities maximize the valuation operator locally. As a consequence, the Jacobian of the policy iteration operator is zero. This proposition extends naturally to the case of infinite state spaces.

2.2 Comparison with NPL and SEES Estimators

The landscape of structural estimation has been significantly shaped by powerful and innovative methods designed to overcome the computational burdens of traditional full-solution algorithms like the Nested Fixed-Point (NFXP) estimator Rust [1987]. Among these, the Nested Pseudo-Likelihood (NPL) estimator of Aguirregabiria and Mira [2002] and the Sieve-Based Efficient Estimator (SEES) of Luo and Sang [2025] represent two distinct and influential paths toward achieving both compu-

tational feasibility and statistical efficiency. NNES builds upon the foundational insights of both, combining the scalability of modern approximation methods with the statistical elegance derived from exploiting model structures.

The **Nested Pseudo-Likelihood (NPL)** estimator offers a seminal contribution for the class of dynamic discrete choice (DDC) models. Its core innovation is to reframe the estimation problem in the space of conditional choice probabilities (CCPs) rather than value functions. NPL "swaps" the traditional nested fixed-point (NFXP) algorithm by iterating between two steps: (i) an inner step that computes the structural parameters θ by maximizing a pseudo-likelihood function, taking the CCP vector P as given; and (ii) an outer step that updates the CCPs using the policy-iteration operator, $P^k = \Psi_{\theta^k}(P^{k-1})$. The cornerstone of NPL's statistical properties is the **zero-Jacobian property** of the policy-iteration map Ψ_θ for DDC models. As established by Aguirregabiria and Mira [2002], at the true CCP vector P^*, the Jacobian of this map is the zero matrix:

$$\left.\frac{\partial \Psi_{\theta_0}(P)}{\partial P}\right|_{P=P^*} = 0.$$

This property gives rise to a block-diagonal information matrix, which simplifies inference and ensures the estimator is asymptotically as efficient as the full-information maximum likelihood estimator (MLE).

More recently, the **Sieve-Based Efficient Estimator (SEES)** (Luo and Sang [2025]) was proposed as a general and robust framework for a broad class of structural models. SEES avoids repeated model solving by approximating the unknown equilibrium function p (e.g., CCPs, bid functions) using a flexible sieve basis, $p_\gamma(x) = \sum_k \gamma_k s_k(x)$. It then formulates a penalized objective function that jointly optimizes the structural parameters θ and the sieve coefficients γ:

$$(\hat{\gamma}, \hat{\theta}) = \arg\max_{\gamma,\theta} \left\{ \mathcal{L}_n(P_\gamma, \theta) - \omega_n \|P_\gamma - \Psi(P_\gamma, \theta)\|^2 \right\}$$

where ℓ_n is the log-likelihood, ω_n is a penalty measuring the deviation from the equilibrium condition $p = \Psi(p, \theta)$, and $\omega_n \to \infty$ is a tuning parameter that enforces the equilibrium constraint more strongly as the sample size grows. The contribution of SEES is its generality and its formulation as a single-level optimization problem, providing a direct and computationally convenient path to efficient estimates that are equivalent to MLE.

Both the **NPL** and our **NNES** estimators are explicitly designed to leverage the *Zero-Jacobian property*. By iterating on the policy operator Ψ_θ in the outer loop, they operate at a fixed point where this derivative vanishes. As a direct consequence, the resulting likelihood score is automatically Neyman-orthogonal to errors in the first-stage policy estimation. This structure ensures that $\sqrt{n}$-consistency and semiparametric efficiency are achieved without requiring an explicit bias-correction term, and it yields a block-diagonal information matrix, which simplifies variance estimation considerably.

In contrast, **SEES** finds the sieve coefficients γ and the structural parameters θ simultaneously, the first-order conditions for θ and γ are intrinsically linked. The derivative of the penalized objective with respect to the nuisance parameters γ does not vanish at the solution. Consequently, the likelihood score in the SEES framework is not, by construction, Neyman orthogonal. While the SEES estimator is proven to be fully efficient, its path to efficiency follows the theory for penalized sieve M-estimators, where the impact of the nuisance function estimation is handled through careful management of tuning parameters (ω_n and the sieve dimension K). The resulting

information matrix for (θ, γ) is generally not block-diagonal, reflecting the statistical dependence between the estimators for the structural and nuisance parameters.

NNES as a Synthesis. Our proposed NNES estimator is architecturally a synthesis of the insights from both NPL and SEES.

1. **Insight from SEES:** NNES adopts the core principle of using a flexible, high-dimensional approximator within a penalized estimation framework. We specifically use a neural network to approximate the value function to overcome the curse of dimensionality in the state space—a challenge that traditional NPL implementations face. Our inner-step objective, which minimizes a penalized likelihood to find the network weights is inspired by the SEES approach of balancing data fit with model-consistency.
2. **Insight from NPL:** Crucially, NNES embeds this penalized, neural-network-based estimation *within* an NPL-style outer loop. Instead of performing a single, joint optimization over all parameters like SEES, NNES performs policy iteration. This architectural choice is deliberate: it ensures that the algorithm converges to a fixed point of the policy-iteration map Ψ_θ.

By doing so, NNES inherits the best of both worlds. From SEES, it takes the power and scalability of using modern approximation methods to handle complex, high-dimensional problems. From NPL, it inherits the profound statistical benefit of the zero-Jacobian property. This synthesis allows NNES to remain computationally feasible in settings where NPL is not, while simultaneously achieving the statistical efficiency and inferential simplicity.

3 The Neural–Network Efficient Estimator (NNES)

The NNES algorithm iterates between *(i)* a **Inner Step** that fits a neural network that minimises a penalized likelihood function to construct the pseudo value function and Choice Probabilities; and *(ii)* a **Outer Step** step that updates the value of P and checks for convergence to a fixed point.

3.1 Neural-Network Approximation

Network class. For each sample size n fix depth–width schedules $L_n, W_n \to \infty$ and let $\mathcal{G}_{W_n,L_n}$ be the set of fully–connected ReLU networks that map states to un-anchored values:

$$G(x;\gamma) = W_{L_n}\sigma(W_{L_n-1}\sigma(\cdots\sigma(W_1 x + b_1)\cdots) + b_{L_n-1}) + b_{L_n},$$

with weights $\gamma = (W_\ell, b_\ell)_{\ell=1}^{L_n} \in \Gamma_n$, $\sigma(z) = \max\{z, 0\}$, and parameter dimension $d_\gamma(n) = O(L_n W_n^2)$.

To ensure numerical stability, we introduce an anchoring constraint. Let $x_0 \in \mathcal{X}$ be a fixed anchor point in the state space. The value function used in our estimator is the anchored neural network:

$$V_\gamma(x) := G(x;\gamma) - G(x_0;\gamma). \tag{9}$$

This construction enforces the normalization $V_\gamma(x_0) = 0$ for any γ. For the remainder of this paper, the notation V_γ refers to this anchored function. The optimization procedures described are implicitly performed over the parameter space Γ_n which defines this class of anchored functions. Appendix D provides an intuitive explanation of Value Function anchoring.

Intuition for anchoring. In logit DDC models, behavior depends on *differences* in choice–specific values, not on the absolute level of the value function. If we replace G by $G + C$, then every $Q_a(x) = u_a(x;\theta) + \beta\,\mathbb{E}[G(X') \mid x, a]$ is shifted by the same βC, which cancels in the soft–max; the CCPs and the likelihood are therefore invariant, and the level of G is not identified. Under an unanchored parameterization, a pure level shift C yields

$$(G + C) - \varphi_\theta[P, G + C] = (G - \varphi_\theta[P, G]) + (1 - \beta)C,$$

so as $\beta \to 1$ the objective becomes nearly flat along this spurious "level" direction, causing ill–conditioning and drift of the fitted value. The normalization in 9, $V_\gamma(x) = G(x;\gamma) - G(x_0;\gamma)$ with fixed anchor x_0, simply removes that redundant degree of freedom by setting $V(x_0) = 0$ and stabilizes the optimization. This normalization does not distort the structural estimates: the implied CCPs, likelihood, and score are exactly the same as with an unanchored Value function G, so $\hat{\theta}$ remains consistent. If one wants the absolute level, it can be recovered ex post from the anchored shape via the identity $(1 - \beta)G(x_0;\gamma) = \varphi_\theta[P, V_\gamma](x_0)$. Appendix D provides an detailed mathematical explanation of Value Function anchoring.

Likelihood contribution. Given (θ, γ) define

$$\ell_n(\theta, \gamma) := \frac{1}{n}\sum_{i=1}^{n} \log P_\gamma(A_i \mid X_i; \theta), \qquad P_\gamma := \Lambda_\theta[V_\gamma],$$

where Λ_θ is the soft-max improvement operator (4).

Bellman penalty. To enforce the Bellman equation, we evaluate the Bellman residual over a deterministic grid of points that becomes dense in the state space. Let $\mathcal{X}_{M_n} = \{x_m\}_{m=1}^{M_n} \subset \mathcal{X}$ be a deterministic grid of M_n points, where the number of points M_n may grow with the sample size n. Let π_{M_n} be the uniform probability measure on this grid, and $P_\gamma = \Lambda_\theta[V_\gamma]$ be the implied CCPs. We define the Bellman penalty as the mean squared Bellman residual on this grid:

$$\rho_n(\gamma, \theta, P) := r(\gamma, \theta)^2 := \frac{1}{M_n}\sum_{m=1}^{M_n} \left(V_\gamma(x_m) - (\varphi_\theta[P_\gamma, V_\gamma])(x_m)\right)^2 = \|V_\gamma - \varphi_\theta[P_\gamma, V_\gamma]\|^2_{L_2(\pi_{M_n})}. \quad (10)$$

Using a deterministic grid that becomes dense ensures that the Bellman equation is enforced uniformly across the state space, a crucial feature for robustness against sparse data regions. The formal requirements for this grid are stated in Assumption 10.

Algorithm 1: Neural-Network Efficient Estimator (NNES)

Input: Data $\{(X_i, A_i)\}_{i=1}^n$; action set $\mathcal{A}$; discount $\beta \in (0,1)$; transition density $f_{\kappa_0}(x'|x,a)$; anchor x_0; grid $\{x_m\}_{m=1}^{M_n}$; penalty ω_n; initial CCP $\widehat{P}^0$.

Output: $(\widehat{\theta}, \widehat{\gamma}, \widehat{P})$

Initialization. Start from an initial estimate $\widehat{P}^0$ of the CCP (using Neural Networks).

for $k = 1, 2, \ldots$ **do**

Inner step k**.**

(a) *Optimize the pseudo-likelihood*

$$\theta^k = \arg\max_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \log\Big\{ P_{\widehat{\gamma}(\theta; P^{k-1})}(A_i \mid X_i; \theta) \Big\}. \tag{11}$$

Here $P_{\widehat{\gamma}(\theta;P^{k-1})}$ is constructed in (13).

(b) *Policy-evaluation least squares (solved for every trial θ considered by the likelihood* (11)*)*:

$$\widehat{\gamma}(\theta; P^{k-1}) = \arg\min_{\gamma \in \Gamma_n} \Big\{ -\ell_n(\theta, \gamma) + \omega_n \, \rho_n(\gamma, \theta, P^{k-1}) \Big\}. \tag{12}$$

Then form the implied CCPs

$$P_{\widehat{\gamma}(\theta;P^{k-1})} := \Lambda_\theta \Big[V_{\widehat{\gamma}(\theta;P^{k-1})} \Big]. \tag{13}$$

In equation (12), a large ω_n enforces approximate Bellman consistency, whereas small ω_n lets the data shape V_γ;

(c) *Value update*

$$\gamma^k := \widehat{\gamma}(\theta^k; P^{k-1}), \qquad V^k := V_{\gamma^k}. \tag{14}$$

Outer step k **(policy improvement).**

$$P^k := \Lambda_{\theta^k}[\, V^k \,]. \tag{15}$$

Stopping rule.

$$\|\theta^k - \theta^{k-1}\| < \varepsilon_\theta \quad \text{and} \quad \|P^k - P^{k-1}\|_{L^2} < \varepsilon_p. \tag{16}$$

if (16) *holds* **then**

break

Output.

$$(\widehat{\theta}, \widehat{\gamma}, \widehat{P}) := (\theta^k, \gamma^k, P^k). \tag{17}$$

The NNES procedure involves two distinct uses of neural network estimators. First, an initial non-parametric estimate of the CCPs, $\widehat{P}^0$, is obtained once using a neural network classifier (as detailed in Section 4.4) to satisfy Assumption 8. Second, for all subsequent policy iterations $k \geq 1$, the CCPs P^{k-1} are treated as given, and the value function V is approximated by the network V_γ. The CCPs for the next iteration, P^k, are then implied by V_γ and are not separately estimated.

4 Asymptotic Theory

Mathematical Setup. Let $\mathcal{X} \subset \mathbb{R}^d$ be the compact state space. We define the following Banach spaces:

- $\mathcal{C}(\mathcal{X})$: The space of continuous real-valued functions on $\mathcal{X}$, equipped with the supremum norm $\|\cdot\|_\infty$.
- W = (X,A): The observed data.
- $\mathcal{P} \subset \mathcal{C}(\mathcal{X}, \Delta^{J-1})$: The space of continuous conditional choice probability (CCP) functions mapping from the state space to the $(J-1)$-dimensional unit simplex. This space inherits its topology from $\mathcal{C}(\mathcal{X})^J$.
- $\Gamma_n \subset \mathbb{R}^{d_\gamma}$: The compact set of admissible Value Function neural network parameters.
- $\Xi_n \subset \mathbb{R}^{d_\gamma}$: The compact set of admissible CCP neural network parameters.
- $\mathcal{S}_{V,n}, \mathcal{S}_{P,n}$ are Sieves for V and P, constructed from ReLU neural networks.
- p_n is the Maximal number of free parameters (weights) in the sieves.
- $\mathbb{P}_n$, $\mathbb{G}_n$ are Empirical measure and the centered empirical process $\sqrt{n}(\mathbb{P}_n - \mathbb{P})$.

The key operators are defined as maps between these spaces:

- **Neural Network Value Function:** $G : \Gamma_n \to \mathcal{C}(\mathcal{X})$, where $G(\gamma) = V_\gamma(\cdot)$. This map takes a parameter vector and produces a continuous value function.
- **Improvement Operator:** $\Lambda_{\theta_0} : \mathcal{C}(\mathcal{X}) \to \mathcal{P}$. Given a value function V, $\Lambda_{\theta_0}[V]$ computes the corresponding optimal CCPs.
- **Evaluation Operator:** $\varphi_{\theta_0} : \mathcal{P} \to \mathcal{C}(\mathcal{X})$. Given a CCP function P, $\varphi_{\theta_0}[P]$ computes the implied value function via the Hotz-Miller inversion (Equation (3)).
- **Policy-Iteration Operator:** $\Psi_{\theta_0} = \Lambda_{\theta_0} \circ \varphi_{\theta_0} : \mathcal{P} \to \mathcal{P}$.

At the true parameters (θ_0, V^*), the model is at a fixed point. We denote the true CCP function by $P^* \equiv P_{V^*} = \Lambda_{\theta_0}[V^*]$, which is the fixed point of the policy-iteration operator, i.e., $\Psi_{\theta_0}(P^*) = P^*$.

4.1 Assumptions and Regularity Conditions

Assumption 3 (Fréchet Differentiability)**.** *The following maps are continuously Fréchet differentiable in the interior of their domains:*

(a) The map from network weights to the value function, $G : \Gamma \to \mathcal{C}(\mathcal{X})$.

(b) The evaluation and improvement operators, $\varphi_{\theta_0} : \mathcal{P} \to \mathcal{C}(\mathcal{X})$ and $\Lambda_{\theta_0} : \mathcal{C}(\mathcal{X}) \to \mathcal{P}$.

This assumption is standard and is satisfied if the underlying primitives (e.g., utility function $u(x;\theta)$, transition density $f(x'|x,a;\theta)$) are sufficiently smooth in their arguments.

Assumption 4 (i.i.d. sampling)**.**

1. *The observable data*
$$\{(A_i, X_i, X_i')\}_{i=1}^n \overset{i.i.d.}{\sim} P_{A,X,X'}$$
are independent and identically distributed draws from a joint distribution $P_{A,X,X'}$ with support $\mathcal{A} \times \mathcal{X} \times \mathcal{X}$.

2. *The parameter space Θ for the structural parameters is compact.*

3. *The action set $\mathcal{A}$ is finite and the state space $\mathcal{X}$ is compact. Conditional choice probabilities are strictly interior, i.e. there exists $1 > p_{max} > p_{min} > 0$ such that*
$$p_{min} \;\le\; P(A_i = a \mid X_i = x) \;\le\; p_{max}, \quad \forall (a,x) \in \mathcal{A} \times \mathcal{X}.$$

Assumption 5 (Sieve Approximation Power)**.** *Let $\mathcal{S}_{V,n} = \{V_\gamma : \gamma \in \Gamma_n\}$ and $\mathcal{S}_{P,n} = \{P_\eta : \eta \in \Xi_n\}$ be sieves for the value and CCP functions. The neural network sieves $\mathcal{S}_{V,n}$ and $\mathcal{S}_{P,n}$ are sufficiently rich to approximate the true value function and CCP well. There exist approximation error rates $r_{P,n} = r_{V,n} = O(n^{-\delta})$ with exponents $1/2 > \delta > 1/4$, such that:*
$$\inf_{P_\eta \in \mathcal{S}_{P,n}} \|P_\eta - P^*\|_{\mathcal{L}_2(\mathbb{P}_X)} \le r_{P,n},$$
$$\sup_{\theta \in \Theta} \inf_{V_\gamma \in \mathcal{S}_{V,n}} \|V_\gamma - V_\theta^*\|_{\mathcal{L}_2(\mathbb{P}_X)} \le r_{V,n}.$$

This is an assumption on the richness of the sieve spaces, which grow with n, not a statement about the sampling error of the $\hat{V}$. The rate is achieved by appropriately choosing the network architecture as a function of the sample size n. For instance, if the true functions V_θ^* and P^* belongs to a Hölder /Sobolev class of smoothness p on an effective support of dimension d^*, Yarotsky [2017] shows that a ReLU network achieves L_2 error ε with depth $L = O(\log(1/\varepsilon))$ and total weights $W = O(\varepsilon^{-d/p}\log(1/\varepsilon))$; choosing $\varepsilon = n^{-1/4}$ yields the $o(n^{-1/4})$ approximation we require with $L_n = O(\log n)$ and $W_n \asymp n^{d/(4p)}$ (up to logs). Sigmoid networks admit an analogous trade-off; see Langer [2021]. Similarly, if V_θ^* and P^* lies in a Barron space, a rate of $O(n^{-1/2})$ is achievable with a single-hidden-layer network [Barron, 1993, Bach, 2017].

Assumption 6 (Operator and Function Regularity)**.** *(a)* ***Uniform Boundedness:*** *The functions in the sieve classes are uniformly bounded. There exists a constant $B < \infty$ such that $\sup_n \sup_{V_\gamma \in \mathcal{S}_{V,n}} \|V_\gamma\|_\infty \le B$. The true value functions V_θ^* are also assumed to be uniformly bounded by B.*

(b) ***Lipschitz Continuity:*** *The Bellman evaluation operator φ_θ and the policy improvement operator Λ_θ are Lipschitz continuous in their function arguments, uniformly in θ. There exist finite constants L_φ and L_Λ such that for all $P_1, P_2 \in \mathcal{S}_{P,n}$ and $V_1, V_2 \in \mathcal{S}_{V,n}$:*
$$\sup_{\theta \in \Theta} \|\varphi_\theta[P_1, V] - \varphi_\theta[P_2, V]\|_{\mathcal{L}_2(\mathbb{P}_X)} \le L_\varphi \|P_1 - P_2\|_{\mathcal{L}_2(\mathbb{P}_X)},$$
$$\sup_{\theta \in \Theta} \|\Lambda_\theta[V_1] - \Lambda_\theta[V_2]\|_{\mathcal{L}_2(\mathbb{P}_X)} \le L_\Lambda \|V_1 - V_2\|_{\mathcal{L}_2(\mathbb{P}_X)}.$$

Uniform boundedness is a condition required to control the behavior of empirical processes. It prevents the functions in our sieve from exploding and ensures that operators like the softmax function (used in Λ_θ) are well-behaved. Lipschitz continuity ensures that the model's operators are stable: small changes in the input functions (policies or value functions) lead to small, controlled changes in the output functions. This is critical for the error recursion step of the proof, as it allows us to propagate error bounds from one iteration to the next in a predictable way. Uniformity in θ is needed because our final result is a supremum over all possible θ. In Appendix G, we provide the analytical expressions for the Lipschitz constants L_φ and L_Λ.

Assumption 7 (Neural Network Sieve Complexity)**.** *The sieve classes for the initial CCPs and the value function are constructed from neural networks. They satisfy the following complexity and growth constraints.*

1. ***Initial CCP Sieve ($\mathcal{S}_{P,n}$):*** *Let Ξ_n be the parameter space for the initial CCP sieve $\mathcal{S}_{P,n}$, with dimension $p_{P,n} := \dim(\Xi_n)$. This sieve satisfies:*

   (a) ***Entropy Bound:*** *For a universal constant A_P, any probability measure $\mathbb{Q}$, and any $\varepsilon \in (0, B]$:*
   $$\log N(\varepsilon, \mathcal{S}_{P,n}, L_2(\mathbb{Q})) \leq A_P \, p_{P,n} \log(B/\varepsilon).$$

   (b) ***Parameter Growth Constraint:*** *The number of parameters is constrained by the approximation rate $\delta > 1/4$ required in Assumption 5:*
   $$p_{P,n} = o\left(\frac{n^{2\delta}}{\log n}\right).$$

2. ***Value Function Sieve ($\mathcal{S}_{V,n}$):*** *Let Γ_n be the parameter space for the value function sieve $\mathcal{S}_{V,n}$, with dimension $p_{V,n} := \dim(\Gamma_n)$. This sieve satisfies:*

   (a) ***Entropy Bound:*** *For a universal constant A_V, any probability measure $\mathbb{Q}$, and any $\varepsilon \in (0, B]$:*
   $$\log N(\varepsilon, \mathcal{S}_{V,n}, L_2(\mathbb{Q})) \leq A_V \, p_{V,n} \log(B/\varepsilon).$$

   (b) ***Parameter Growth Constraint:*** *The number of parameters is constrained by the approximation rate $\delta > 1/4$:*
   $$p_{V,n} = o\left(\frac{n^{2\delta}}{\log n}\right).$$

Assumption 7 is feasible under standard smoothness–over–dimension conditions. Let the target functions lie in a Hölder/Sobolev ball with smoothness p on an intrinsic $d^\star$–dimensional support. By Yarotsky [2017], for any $\varepsilon \in (0, 1)$ there exists a ReLU Neural Network of *depth* $L = O(\log(1/\varepsilon))$ and with *number of nonzero weights* $W = O(\varepsilon^{-d^\star/p} \log(1/\varepsilon))$ achieving L_2–error $O(\varepsilon)$. Setting $\varepsilon = n^{-\alpha_V}$ gives $L_n = O(\log n)$ and $W_n \asymp n^{\alpha_V d^\star/p}$ (up to log factors). So the growth condition $p_{V,n} = o(n^{2\delta}/\log n)$ holds whenever
$$\frac{2\delta d^\star}{p} < 2\delta \qquad \Longleftrightarrow \qquad p > d^\star.$$
With the benchmark $\delta > \frac{1}{4}$, this gives $p_{V,n} = o(n^{1/2}/\log n)$, exactly the complexity needed for the local Donsker bound used in Lemma 4.2. An analogous calibration holds for sigmoid networks:

Langer [2021] proves L_2–error of order $W^{-p/d^\star}$ at (fixed) depth, hence choosing $W_n \asymp n^{\delta d^\star/p}$ again yields $p_{V,n} = o\left(\frac{n^{2\delta}}{\log n}\right)$ and the same requirement $p > d^\star$.

Assumption 8 (Initial Nonparametric CCP)**.** *The intial estimator $\widehat{P}^0$ satisfies $\left\|\widehat{P}^0 - P^*\right\|_2 = o_p(n^{-\alpha_P})$ for $\alpha_P > \frac{1}{4}$.*

Assumption 9 (Penalty sequence)**.** *Let*

$$\omega_n = C\, n^{\delta}, \qquad \tfrac{1}{4} < \delta < \tfrac{1}{2},\ C > 0. \tag{18}$$

Intuition. $\delta > 1/4$ ensures $\rho_n(\widehat{\gamma}^k, \theta^{(k-1)}) = o_p(n^{-1/2})$, while $\delta < 1/2$ keeps the likelihood term from being negligible, preserving information on θ. Note that this delta is the same constant in Assumption 5.

Assumption 10 (Properties of the Penalty Grid)**.** *The sequence of deterministic grids $\mathcal{X}_{M_n} = \{x_m\}_{m=1}^{M_n}$ used in the Bellman penalty* (10) *is sufficiently regular such that for the class of functions $\mathcal{F}_n = \{(V - \varphi_\theta[P])^2 : V \in \mathcal{S}_{V,n}, P \in \mathcal{S}_{P,n}, \theta \in \Theta\}$, the sample average over the grid converges uniformly to the true integral with respect to the Lebesgue measure μ:*

$$\sup_{f \in \mathcal{F}_n} \left| \int f(x) d\pi_{M_n}(x) - \int f(x) d\mu(x) \right| = o(1/\omega_n). \tag{19}$$

Where $\omega_n = Cn^\delta$, $1/4 < \delta < 1/2$ (Assumption 9).

Assumption 10 implies that the number of grid points $M_n \to \infty$ as $n \to \infty$. It ensures that the error from using a grid is asymptotically negligible relative to the statistical estimation error, which is of order $O_p(n^{-1/2})$. This condition can be satisfied by choosing a grid sequence that is sufficiently space-filling. In practice, low-discrepancy sequences (e.g., Sobol or Halton sequences) are often used as they can achieve better integration accuracy than simple rectangular grids.

4.2 Convergence of the Value Function

For the brevity of notations, denote $\ell_n(V)$ as the log-likelihood function $\log \Lambda_\theta[V](A_i|X_i)$.

Lemma 4.1 (Log-Likelihood is Lipschitz in V)**.** *Under Assumptions 4–6, for every $\theta \in \Theta$, the mapping $V \mapsto \mathbb{P}[\ell_n(V)]$ is Lipschitz continuous in the L_2-norm:*

$$|\mathbb{P}[\ell_n(V_1)] - \mathbb{P}[\ell_n(V_2)]| \ \le\ C_\ell \, \|V_1 - V_2\|_2, \tag{20}$$

for a finite constant C_ℓ.

Proof. The proof is provided in the Appendix A.2. □

Lemma 4.2 (Local Donsker Property of the Value Function Classes)**.** *Let $r_{V,n} = O(n^{-\delta})$ with $\delta > \frac{1}{4}$. Set*

$$\mathcal{H}_{V,n} := \{\ell_n(V) - \ell_n(V^*) : V \in \mathcal{S}_{V,n},\ \|V - V^*\|_2 \le r_{V,n}\},$$

Under Assumptions 4-7, the class $\mathcal{H}_{V,n}$ is asymptotically P-Donsker.

Proof. The proof is provided in the Appendix A.3. □

This lemma formally proves a stochastic equicontinuity property for the empirical process indexed by our Value Function classes within a shrinking neighborhood of the true functions. This result allows us to rigorously control the stochastic remainder terms that arise from estimating the Value Function from the same data. It is the key to establishing that these remainder terms are asymptotically negligible in the subsequent proofs of convergence and asymptotic normality.

Proposition 2. *(Rate of Convergence of the Value Function) Fix any integer $k \geq 1$ and $1/2 > \delta > 1/4$ as in Assumption 5, let $\widehat{V}^k$ be the estimated Value function after iteration k. Under Assumptions 4- 9,*

$$\sup_{\theta \in \Theta} \|\widehat{V}^k - V_\theta^*\|_2 = o_p(n^{-\delta}).$$

Proof. The proof is provided in the Appendix A.3. □

This result demonstrates that the first-stage Value Function estimator $\widehat{V}^k$ converges sufficiently fast. The rate of $o_p(n^{-1/4})$ is a key condition for the main theorem on the $\sqrt{n}$-consistency of $\hat{\theta}$, as it ensures that the first-stage estimation error does not contaminate the asymptotic distribution of the structural parameters when using a Neyman-orthogonal score function.

4.3 Neyman Orthogonality of the Score

We first show that, at the fixed-point $V = V^*$ and $P = P^*$, the derivatives of the evaluation operator φ_{θ_0} and the policy–iteration operator $\Psi_{\theta_0} = \Lambda_{\theta_0} \circ \varphi_{\theta_0}$ with respect to the Value function V vanish.

Proposition 3. *Under Assumptions 1–2, let V^* be unique solution to the Bellman equation 1, P^* is the true CCP, and $P_V = \Lambda_{\theta_0}[V]$. Then*

$$\frac{\partial}{\partial V}\left(\varphi_{\theta_0}[\Lambda_{\theta_0}[V], V]\right)\Big|_{V=V^*} = 0 \quad \textit{and} \quad \frac{\partial}{\partial V}\left(\Psi_{\theta_0}(P_V)\right)\Big|_{V=V^*} = 0.$$

Proof. The proof is provided in the Appendix A.1. □

The implications of Proposition 3 is that any small first-stage error in approximating the Value Function only affects the likelihood score at second order. This is precisely the definition of a Neyman-orthogonal score.

Proposition 4 (Neyman Orthogonality of the Likelihood Score)**.** *Let $\mathcal{V}$ be a suitable Banach space of continuous functions containing the value functions (e.g., a subset of $\mathcal{C}(\mathcal{X})$). Define the score function $\psi : \mathcal{W} \times \Theta \times \mathcal{V} \to \mathbb{R}^{d_\theta}$ as:*

$$\psi(W; \theta, V) := \frac{\partial}{\partial \theta} \log P(A \mid X; \theta, V), \tag{21}$$

where $W = (A, X)$ and the CCPs are generated from the value function V via the policy iteration operator, $P(\cdot \mid \cdot; \theta, V) := \Psi_\theta(\Lambda_\theta[V])$.

Let $\theta_0 \in \Theta$ be the true structural parameter and let $V^* \in \mathcal{V}$ be the corresponding true value function, which is the unique solution to the Bellman equation at θ_0. Suppose the following regularity conditions hold:

(i) ***Moment and boundedness condition:*** *For some neighborhood $\mathcal{N}$ of V^* in $\mathcal{V}$:*

$$\mathbb{E}\left[\sup_{V\in\mathcal{N}}\|\psi(W;\theta_0,V)\|\right]<\infty.$$

(ii) ***Pathwise Differentiability:*** *The map $t\mapsto\mathbb{E}[\psi(W;\theta_0,V_t)]$ is differentiable at $t=0$ for any path $V_t=V^*+t(V-V^*)$ where $V\in\mathcal{V}$.*

Then, the score ψ is Neyman orthogonal with respect to the Value Function V at the true values (θ_0,V^*). Specifically, the Gâteaux derivative of the expected score with respect to V in any valid direction $(V-V^*)\in\mathcal{V}$ is zero:

$$\frac{d}{dt}\mathbb{E}[\psi(W;\theta_0,V^*+t(V-V^*))]\Big|_{t=0}=0.$$

Proof. The proof is provided in Appendix A.5. □

Connection to the Estimator. This proposition provides the theoretical foundation for the robustness of our NNES estimator. While our estimator uses a finite-dimensional approximation $V_{\widehat{\gamma}}$ for V^*, the orthogonality property ensures that small errors in this approximation do not create a first-order bias in the estimation of θ_0. This property is attractive since we only enforce the Bellman penalty softly in 12. This allows for $\sqrt{n}$-consistent estimation of θ_0 even when V^* is estimated with flexible, high-dimensional machine learning models, provided the approximation rate is faster than $n^{-1/4}$.

4.4 Root-n Consistency and Semiparametric Efficiency

Theorem 4.3 (Asymptotic Normality and Semiparametric Efficiency)**.** *Let $k\geq 1$ be a fixed number of outer-loop iterations. Let $\widehat{P}^0$ be an initial non-parametric estimator of the true Conditional Choice Probability (CCP) function P^*. Assume this initial estimator converges in the L^2-norm at a rate faster than $n^{-1/4}$:*

$$\|\widehat{P}^0-P^*\|_{L^2}=o_p(n^{-1/4}).$$

The NNES algorithm is initiated with $\widehat{P}^0$, and let $(\widehat{\theta},\widehat{\gamma})$ denote the final parameter estimates for the structural parameters and the value-function network, respectively, after k iterations have completed. Under Assumptions 1–9 and the moment conditions in Proposition 4, the NNES estimator $\widehat{\theta}$ is $\sqrt{n}$-consistent and asymptotically normal, achieving the semiparametric efficiency bound:

$$\sqrt{n}(\widehat{\theta}-\theta_0)\xrightarrow{d}\mathcal{N}\left(0,\mathcal{I}_{\theta\theta}^{-1}\right).$$

Proof. The proof is provided in Appendix B. □

Rate requirement. Because the score function in 21 is Neyman–orthogonal to the first–stage errors, root-n inference for θ is valid provided that the estimated conditional choice probabilities satisfy

$$\|\widehat{P}^0-P^*\|_{L^2}=o_{\mathrm{p}}(n^{-1/4}),$$

thereby forcing the second–order remainder to be $o_{\mathrm{p}}(n^{-1/2})$. The threshold rate $n^{-1/4}$ is known to be *sharp* for semiparametric GMM with non-parametric estimation of the nuisance parameters; see Chernozhukov et al. [2024a] for a general discussion.

Convergence rate Achievability for initial nonparametric estimation of CCPs. If the true CCPs P^* belongs to a Hölder /Sobolev class of smoothness p on an effective support of dimension d^*, Yarotsky [2017] shows that a ReLU network achieves L_2 error ε with depth $L = O(\log(1/\varepsilon))$ and total weights $W = O(\varepsilon^{-d/p}\log(1/\varepsilon))$; choosing $\varepsilon = n^{-1/4}$ yields the $o(n^{-1/4})$ approximation we require with $L_n = O(\log n)$ and $W_n \asymp n^{d/(4p)}$ (up to logs). Sigmoid networks admit an analogous trade-off; see Langer [2021]. Similarly, if P^* lies in a Barron space, a rate of $O(n^{-1/2})$ is achievable with a single-hidden-layer network [Barron, 1993, Bach, 2017].

Practical implementation (soft-max network). We construct the initial estimator $\widehat{P}^0$ by posing a multi-class classification problem. Let a neural network $f_\eta\colon \mathcal{X} \to \mathbb{R}^J$, with parameter η, map a state x to logits $z(x;\eta) = (z_1, \ldots, z_J)$ produced by one hidden ReLU layer of width $W_n = \lfloor n^{1/2} \rfloor$. The soft-max layer converts logits into probabilities,

$$\widehat{P}^0(a \mid x;\eta) = \frac{\exp\big(z_a(x;\eta)\big)}{\sum_{j=1}^{J}\exp\big(z_j(x;\eta)\big)}. \tag{22}$$

Given observations $\{(A_i, X_i)\}_{i=1}^n$, we estimate $\widehat{\eta}$ by minimising the empirical cross-entropy loss

$$\mathcal{L}_n(\eta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{a=1}^{J}\mathbf{1}\{A_i = a\}\,\log \widehat{P}_0(a \mid X_i;\eta),$$

using stochastic gradient descent. The fitted network $\widehat{P}^0(\cdot \mid \cdot;\widehat{\eta})$ provides the non-parametric CCP required as the starting point of the NNES algorithm.

5 Computation of the Profile Cross-Term

5.1 Numerical Derivative

Recall that the inner step solves, for fixed P^{k-1},

$$\hat{\gamma}(\theta; P^{k-1}) \in \arg\min_{\gamma\in\Gamma_n}\Big\{-\,\ell_n(\theta,\gamma)\;+\;\omega_n\,\rho_n(\gamma,\theta,P^{k-1})\Big\}, \tag{23}$$

and the profile log-likelihood at inner iteration k is $L_n(\theta) = \ell_n\big(\theta, \hat{\gamma}(\theta; P^{k-1})\big)$. By the chain rule:

$$\frac{d}{d\theta}L_n(\theta) = \underbrace{\frac{\partial}{\partial\theta}\,\ell_n\big(\theta, \hat{\gamma}(\theta; P^{k-1})\big)}_{\text{direct term}} \;+\; \underbrace{D_V\ell_n\big(\theta, V_{\hat{\gamma}(\theta;P^{k-1})}\big)\big[\,\partial_\theta V_{\hat{\gamma}(\theta;P^{k-1})}\,\big]}_{\text{cross term}}. \tag{24}$$

Here $D_V\ell_n(\theta, V)[H]$ denotes the Gâteaux derivative of $V \mapsto \ell_n(\theta, V)$ in the direction H. The direct term in (24) is the derivative of ℓ_n with respect to θ in the current period utility function, and the indirect term is where θ enters in the expected value functions.

Functional form of the cross-term. Write $P(\cdot \mid \cdot) = \Lambda_\theta[V]$ and recall that $\ell_n(\theta, V) = \frac{1}{n}\sum_{i=1}^n \log P(A_i \mid X_i)$, with

$$Q_a(x;\theta,V) = u_a(x;\theta) + \beta\,\mathbb{E}[\,V(X') \mid x, a\,], \qquad P(a \mid x) = \frac{\exp Q_a(x;\theta,V)}{\sum_{b\in\mathcal{A}}\exp Q_b(x;\theta,V)}.$$

A standard differentiation of the log-softmax yields, for any direction $H \in C(\mathcal{X})$,

$$D_V \ell_n(\theta, V)[H] = \frac{\beta}{n} \sum_{i=1}^{n} \sum_{a \in \mathcal{A}} \Big(\mathbf{1}\{A_i = a\} - P(a \mid X_i) \Big) \, \mathbb{E}\big[\, H(X') \mid X_i, a \,\big] \,. \tag{25}$$

Consequently, the cross-term in (24) is the inner product of (25) with the value-function derivative $\partial_\theta V$:

$$\boxed{\mathrm{Cross}(\theta) = \frac{\beta}{n} \sum_{i=1}^{n} \sum_{a \in \mathcal{A}} \Big(\mathbf{1}\{A_i = a\} - P(a \mid X_i) \Big) \, \mathbb{E}\Big[\, \partial_\theta V_{\hat{\gamma}(\theta; P^{k-1})}(X') \mid X_i, a \,\Big] \,,} \tag{26}$$

where $P = \Lambda_\theta\big[V_{\hat{\gamma}(\theta;P^{k-1})}\big]$ and the conditional expectation is taken under the model transition $f_{\kappa_0}(\cdot \mid X_i, a)$.

Numerical computation of $\partial_\theta V$ at fixed P^{k-1}. The derivative $\partial_\theta V_{\hat{\gamma}(\theta;P^{k-1})}$ is computed by a symmetric finite-difference protocol that refits the inner problem (23) at $\theta \pm h\, e_j$ while *holding P^{k-1} fixed*. This keeps the computation aligned with the definition of $\hat{\gamma}(\theta; P^{k-1})$ and with the policy-iteration structure of Algorithm 1. The procedure is:

(i) **Baseline fit.** Solve (23) at the current θ and P^{k-1} to obtain $V^0 := V_{\hat{\gamma}(\theta;P^{k-1})}$. Use the anchored value network $V_\gamma(x) = G(x;\gamma) - G(x_0;\gamma)$ and train until the Bellman-residual MSE in ρ_n is below a tolerance $\varepsilon_{\mathrm{res}}$.

(ii) **Warm-started $\pm h$ refits.** For a candidate step $h > 0$, form $\theta_\pm = \theta \pm h\, e_j$. With the same fixed P^{k-1}, resolve (23) at $\theta_\pm$, warm-starting from the baseline weights, and enforce the *same* residual tolerance $\varepsilon_{\mathrm{res}}$. Denote the resulting value functions by V^+ and V^-.

(iii) **Central difference.** Define the numerical value-derivative

$$\dot{V}^{(h)}(x) \;:=\; \frac{V^+(x) - V^-(x)}{2h}. \tag{27}$$

(iv) **Blind step selection.** Evaluate (27) on a decreasing grid $h_1 > h_2 > \cdots > h_K$. Choose the smallest h_k such that both $\pm h_k$ refits meet the residual tolerance and the estimate is stable:

$$\frac{\|\dot{V}^{(h_k)} - \dot{V}^{(h_{k-1})}\|_{L^2(\pi_{M_n})}}{\|\dot{V}^{(h_k)}\|_{L^2(\pi_{M_n})}} \;\le\; \tau,$$

for a preset $\tau > 0$. Set $\partial_\theta V_{\hat{\gamma}(\theta;P^{k-1})} \equiv \dot{V}^{(h_k)}$.

This finite-difference protocol is symmetric, uses identical training settings across the $\pm h$ branches, and selects h via residual and stability criteria only.

Profile gradient used in the θ-update. With $\mathrm{Cross}_j(\theta)$ from (26), the FOC of the inner problem 11 is:

$$\frac{d}{d\theta} L_n(\theta) = \frac{\partial}{\partial \theta}\, \ell_n\big(\theta, V^0\big) \;+\; \mathrm{Cross}(\theta), \tag{28}$$

where the partial derivative is obtained by standard backpropagation through the explicit dependence of Λ_θ on θ at fixed V^0, and the cross-term is computed by plugging the numerical $\partial_\theta V$ into (26). Additional simulation results in Appendix C.2 show that the numerical derivative can be a reliable approximation of the analytical derivative.

5.2 Analytical Derivative of $\partial_\theta V$ via a Gradient Bellman Equation

This section replaces the finite-difference protocol used for $\partial_\theta V$ in section 5.1 with an analytical step based on differentiating the model's fixed point. Let V_θ denote the (ex-ante) value function that solves the log-sum-exp Bellman equation in 1:

$$V_\theta(x) = \log\Big(\sum_{a\in\mathcal{A}} \exp\{u_a(x;\theta) + \beta\,\mathbb{E}[V_\theta(X') \mid x, a]\}\Big),$$

Differentiating equation 1 with respect to θ yields the *gradient Bellman equation* for $G_\theta(x) := \partial_\theta V_\theta(x) \in \mathbb{R}^{d_\theta}$:

$$G_\theta(x) = \sum_{a\in\mathcal{A}} P(a|x)\,\partial_\theta u_a(x;\theta) + \beta \sum_{a\in\mathcal{A}} P(a|x)\,\mathbb{E}\big[G_\theta(X') \mid x, a\big]. \tag{29}$$

This is a linear fixed point in the function G_θ. At the inner step of Algorithm 1, we fix the previous iteration policy P^{k-1}. We approximate G_θ by a separate neural network $G_w : \mathcal{X} \to \mathbb{R}^{d_\theta}$ trained to solve (29) with $P(a|x)$ replaced by P^{k-1}. Here w is the parameter of the Neural Network. G_w is then estimated by minimizing the mean-squared *gradient Bellman residual*

$$\min_w\ \mathbb{E}_x \left\| G_w(x)\ -\ \sum_a P^{k-1}(a \mid x)\,\partial_\theta u_a(x;\theta)\ -\ \beta \sum_a P^{k-1}(a \mid x)\,\mathbb{E}\big[G_w(X') \mid x, a\big] \right\|^2. \tag{30}$$

Recall the profile objective $L_n(\theta) = \ell_n\big(\theta, \hat\gamma(\theta; P^{k-1})\big)$ and its chain-rule decomposition in 24:

$$\frac{d}{d\theta} L_n(\theta) = \frac{\partial}{\partial\theta}\,\ell_n(\theta, V) +\ D_V \ell_n(\theta, V)\big[\partial_\theta V\big],$$

where $V = V_{\hat\gamma(\theta;P^{k-1})}$. Replacing the unknown $\partial_\theta V$ by $G_{\hat w}$ from (30) gives the implementable profile gradient

$$\frac{d}{d\theta} L_n(\theta) = \frac{\partial}{\partial\theta}\,\ell_n(\theta, V)\ +\ \frac{\beta}{n}\sum_{i=1}^{n}\sum_{a\in\mathcal{A}} \Big(\mathbf{1}\{A_i = a\} - P(a|X_i)\Big)\,\mathbb{E}\big[G_{\hat w}(X') \mid X_i, a\big]. \tag{31}$$

Score with an additional gradient nuisance. Fix a value proxy V and write the choice-specific value

$$Q_a(x;\theta,V)\ :=\ u_a(x;\theta) + \beta\,\mathbb{E}[V(X') \mid x, a], \qquad V\ :=\ \varphi_\theta\big[\Lambda_\theta[V],\, V\big],$$

so that the CCPs entering the likelihood are $P_{\theta,V}(\cdot \mid x) = \Psi_\theta\big(\Lambda_\theta[V]\big)(\cdot \mid x)$. The per-observation log-likelihood is:

$$\ell(A, X; \theta, V) = \log P_{\theta,V}(A \mid X) = Q_A(X;\theta,V) - \log \sum_{j\in\mathcal{A}} e^{Q_j(X;\theta,V)}.$$

By the standard log-softmax identity,

$$\partial_\theta \ell(A, X; \theta, V) = \sum_{a\in\mathcal{A}} \big\{\mathbb{1}\{A = a\} - P_{\theta,V}(a \mid X)\big\}\,\partial_\theta Q_a(X;\theta,V), \tag{32}$$

$$\partial_\theta Q_a(X;\theta,V) = \partial_\theta u_a(X;\theta)\ +\ \beta\,\mathbb{E}\big[\partial_\theta V(X') \mid X, a\big]. \tag{33}$$

At the truth, $\partial_\theta V = G_\theta$. We therefore define the *practical* score that treats a fitted gradient network G_w as a nuisance:

$$\psi(W;\theta,V,G_w) \;:=\; \sum_{a\in\mathcal{A}} \{\mathbb{1}\{A=a\} - P_{\theta,V}(a \mid X)\}\Big[\partial_\theta u_a(X;\theta) \;+\; \beta\,\mathbb{E}\big[\,G_w(X') \mid X,a\big]\Big], \qquad (34)$$

where $W = (A, X)$. Equation (34) is the score formula obtained when a separate network is used to approximate $\partial_\theta V$. Let $M(\theta; V, G) := \mathbb{E}[\psi(W;\theta,V,G)]$. We now establish that M is first-order insensitive to local perturbations of G at the truth.

Assumption 11 (Pathwise regularity in G)**.** *For any measurable direction $H : X \to \mathbb{R}^{d_\theta}$ with $\mathbb{E}\|H(X')\| < \infty$, the path $G_t := G^* + tH$ satisfies dominated convergence so that differentiation under the expectation is valid.*

Proposition 5 (Neyman orthogonality of the score with respect to G)**.** *Let θ_0 denote the true parameter and V^* the associated value, with optimal CCPs $P^*(\cdot \mid X)$. Suppose $P_{\theta_0,V^*} = P^*$ and Assumption 11 holds. Then the moment $M(\theta; V, G)$ defined from* (34) *is Neyman-orthogonal in G at (θ_0, V^*, G^*):*

$$D_G M(\theta_0; V^*, G^*)[H] := \frac{d}{dt}\,\mathbb{E}[\psi(W;\theta_0,V^*,G^*+tH)]\Big|_{t=0} = 0 \qquad \textit{for all directions } H.$$

Proof. Only the continuation term in (34) depends on G. Differentiating at $t = 0$ and applying the law of iterated expectations yields

$$D_G M(\theta_0; V^*, G^*)[H] = \mathbb{E}\Bigg[\sum_{a\in\mathcal{A}} \{\mathbb{1}\{A=a\} - P_{\theta_0,V^*}(a \mid X)\}\,\beta\,\mathbb{E}\big[H(X') \mid X,a\big]\Bigg].$$

Condition on X and use that, at the truth, $\mathbb{E}[\mathbb{1}\{A=a\} \mid X] = P^*(a \mid X) = P_{\theta_0,V^*}(a \mid X)$:

$$\mathbb{E}\Big[\{\mathbb{1}\{A=a\} - P_{\theta_0,V^*}(a \mid X)\}\,\beta\,\mathbb{E}[H(X') \mid X,a] \;\Big|\; X\Big] = 0, \quad \forall a \in \mathcal{A}.$$

Summing over a and integrating over X gives $D_G M(\theta_0; V^*, G^*)[H] = 0$. Hence the score is Neyman-orthogonal with respect to G. □

Proposition 5 implies that any perturbation of G, including those induced by noise in a first-stage CCP, do not have a first order effect on the score function for θ.

6 Monte-Carlo Study

The simulation reproduces Rust [1987]'s bus-engine environment with two independent buses, but the NNES estimator is not told that the true value function is additively separable. This design highlights NNES's ability to recover structure that is unknown to the approximating class.

6.1 Dynamic Discrete-Choice Model

State space. At the start of each period t the continuous mileage vector is

$$x_t = (m_{1,t}, m_{2,t}) \in \mathcal{X} = [0, M_{\max}]^2, \quad M_{\max} = 100.$$

Actions. The decision maker chooses $a_t = (a_{1,t}, a_{2,t}) \in \{0,1\}^2$, where $a_{j,t} = 1$ denotes *replace* module j and $a_{j,t} = 0$ denotes *keep*.

Structural parameter.

$$\theta = (c_{\text{rep},1}, c_{\text{rep},2}, c_1, c_2)^\top \in \Theta \subset \mathbb{R}^4, \qquad \beta = \{0.9, 0.999\} \ .$$

Flow utility. For module j and action a_j

$$u_j(m_j, a_j; \theta) = -\left[c_j m_j (1 - a_j) + c_{\text{rep},j} a_j\right] + \varepsilon_{j,a_j}, \ \varepsilon_{j,a} \overset{i.i.d.}{\sim} \text{T1EV}. \tag{35}$$

Total utility is additive: $u(x, a; \theta) = u_1 + u_2$.

Continuous transition. Conditional on (m_j, a_j)

$$m_j' = \begin{cases} \min\{m_j + y_j^{(K)}, M_{\max}\}, & a_j = 0, \\ \min\{y_j^{(R)}, M_{\max}\}, & a_j = 1, \end{cases} \quad y_j^{(K)}, y_j^{(R)} \overset{i.i.d.}{\sim} \exp(\lambda), \ \lambda = \tfrac{1}{5}.$$

The two modules evolve independently.

Choice-specific values and inclusive value. For $j \in \{1, 2\}$,

$$v_{j,a}(m_j; \theta) = u_j(m_j, a; \theta) + \beta \, \mathbb{E}\big[V_j(m_j') \mid m_j, a\big], \tag{36}$$

$$V_j(m_j; \theta) = \log\big[e^{v_{j,0}(m_j;\theta)} + e^{v_{j,1}(m_j;\theta)}\big]. \tag{37}$$

Additive fixed point (unknown to NNES). Because the shocks and transition kernel factorise across j, the aggregate Bellman fixed point is $V_\theta(x) = V_1(m_1; \theta) + V_2(m_2; \theta)$. *NNES is nonetheless trained with a single 2-D network that is not forced to satisfy this additivity.*

Closed-Form Conditional Choice Probabilities

Because each pair of private shocks $(\varepsilon_{j,0}, \varepsilon_{j,1})$ is i.i.d. Type-I Extreme Value, the difference $\varepsilon_{j,1} - \varepsilon_{j,0}$ follows a standard logistic distribution. Hence module j's *structural* probability of replacement has the familiar logit form

$$p_{j,\theta}(1 \mid m_j) = \frac{\exp\big[v_{j,1}(m_j; \theta)\big]}{\exp\big[v_{j,0}(m_j; \theta)\big] + \exp\big[v_{j,1}(m_j; \theta)\big]}, \tag{38}$$

where $v_{j,a}$ are the choice-specific value functions. The complementary probability of keeping the module is $p_{j,\theta}(0 \mid m_j) = 1 - p_{j,\theta}(1 \mid m_j)$.

Because the two modules' shocks are independent *across* j, the joint probability of any action vector $a = (a_1, a_2) \in \{0,1\}^2$ factorises:

$$P_\theta(a \mid x) = \prod_{j=1}^{2} p_{j,\theta}(a_j \mid m_j). \tag{39}$$

6.2 Numerical DP Solver

1. *Uniform grid.* Define mileage grid $\mathcal{G} = \{0, \Delta, \ldots, M_{\max}\}$ with $\Delta = 0.5$ (so $G = 201$ nodes).

2. *Gauss–Laguerre quadrature.* Precompute 20 pairs $\{(w_q, z_q)\}_{q=1}^{20}$ so that for *any* smooth function $\phi(\cdot)$,
$$\int_0^\infty \phi(y)\, \lambda e^{-\lambda y}\, dy \;\approx\; \sum_{q=1}^{20} w_q\, \phi\big(z_q/\lambda\big).$$
In particular, in value iteration we take $\phi(y) = V_j\big(\min\{m_g + y, M_{\max}\}\big)$ when the module is kept, and $\phi(y) = V_j\big(\min\{y, M_{\max}\}\big)$ when it is replaced.

3. *Value iteration (one module at a time).* Initialize $V_j^0 \equiv 0$. At each grid point m_g:
$$\mathbb{E}\big[V_j(m') \mid m_g, 0\big] \approx \sum_{q=1}^{20} w_q\, V_j^{(k)}\big(\min\{m_g + z_q/\lambda, M_{\max}\}\big),$$
$$\mathbb{E}\big[V_j(m') \mid m_g, 1\big] \approx \sum_{q=1}^{20} w_q\, V_j^{(k)}\big(\min\{z_q/\lambda, M_{\max}\}\big),$$
where off-grid values of $V_j^{(k)}$ are obtained by *linear* interpolation. Update until $\max_g |V_j^{(k+1)} - V_j^{(k)}| < 10^{-8}$.

4. *Store CCP grid.* After convergence compute
$$P_j(m_g) \;=\; p_{j,\theta^\star}(1 \mid m_g) \;=\; \frac{\exp[v_{j,1}(m_g)]}{\exp[v_{j,0}(m_g)] + \exp[v_{j,1}(m_g)]}$$
at each $m_g \in \mathcal{G}$.

6.3 Panel Simulation

Using the true CCPs (39):

1. Simulate $N = 50$ buses for $B = 10$ burn-in periods (state discarded) and $T = 20$ kept periods.

2. Each step draws $a_{j,t} \sim \text{Bernoulli}\big(\texttt{interp}(m_j, P_j)\big)$, then updates mileage via the exponential increment, capping at $M_{\max}$.

One replication contains $n = N \times T = 1000$ observations. The final dataset returned to the estimators is a tensor $\big\{(x_{i,t}, a_{i,t}) \mid i = 1, \ldots, N,\ t = 0, \ldots, T-1\big\}$, where each mileage entry is a *continuous* realisation of the stochastic law, not a grid index. Grid interpolation is used only when functions V_j or P_j must be evaluated.

6.4 Implementation of the Neural–Network Efficient Estimator (NNES)

Network architecture

Inputs and scaling. Each state is $x = (m_1, m_2) \in [0, M]^2$ with $M = 100$. We normalise to $\tilde{x} = x/M \in [0, 1]^2$.

Neural Network specification. We use a single–hidden–layer ReLU network of width $m = W_n = 8$. For an input $\tilde{x} \in \mathbb{R}^d$ define

$$h_j(\tilde{x}) \;=\; \sigma(w_{1j}^T \tilde{x} + b_{1j}), \qquad j = 1, \ldots, m, \quad \sigma(z) = \max\{z, 0\},$$

and let the output layer be

$$f_\gamma(\tilde{x}) \;=\; b_2 + \sum_{j=1}^{m} a_j \, h_j(\tilde{x}).$$

The network output f_γ represents the un-anchored component of the value function. As established in our theoretical discussion, to ensure numerical stability, the value function V_γ used throughout the model is the anchored version of this network. We define the anchor point at the origin of the scaled state space, $\tilde{x}_0 = (0, 0)$. The value function is therefore constructed as:

$$V_\gamma(x) := f_\gamma(\tilde{x}) - f_\gamma(\tilde{x}_0).$$

This construction ensures that $V_\gamma(x_0) = 0$ for any parameter vector γ, stabilizing the estimation procedure. The vector γ contains all trainable parameters of the underlying network f_γ. The full parameter vector is $\gamma = (w_{11}, \ldots, w_{1m}, b_{11}, \ldots, b_{1m}, a_1, \ldots, a_m, b_2)$, so the total number of trainable parameters is $dm + m + m + 1 = (d + 2)m + 1 = 33$ when $d = 2$ and $m = 8$.

Four-action Q-function & quadrature

For joint action $a = (a_1, a_2) \in \{0, 1\}^2$, instantaneous utility is

$$U(x, a; \theta) = \sum_{j=1}^{2} \Big[-c_j \, m_j (1 - a_j) - c_{\text{rep},j} \, a_j \Big].$$

Given (γ, θ), the choice-specific value is

$$Q_{\gamma,\theta}(x, a) = U(x, a; \theta) + \beta \, \mathbb{E}\big[V_\gamma(X') \mid X = x, A = a\big],$$

where the expectation is approximated by a 20×20 Gauss–Laguerre tensor rule on the exponential increments $\exp(\lambda)$. All 400 draws (x', a) are passed jointly through the network, so the Neural Network never knows of any additive structure.

The four-alternative logit is

$$P_{\gamma,\theta}(a \mid x) = \frac{\exp\{Q_{\gamma,\theta}(x, a)\}}{\sum_{a' \in \{0,1\}^2} \exp\{Q_{\gamma,\theta}(x, a')\}}.$$

Equivalently—since the DGP is separable—the code also computes module-wise logits

$$p_{j,\gamma,\theta}(1 \mid m_j) = \frac{\exp Q_{j,1}(x)}{\exp Q_{j,0}(x) + \exp Q_{j,1}(x)}, \quad P_{\gamma,\theta}(a \mid x) = \prod_{j=1}^{2} p_{j,\gamma,\theta}(a_j \mid m_j).$$

Empirical objectives: inner vs. outer

Let $\{(x_i, a_i)\}_{i=1}^n$ be the sample ($n = 1000$). Define the negative log-likelihood

$$\ell_n(\gamma, \theta) = -\frac{1}{n}\sum_{i=1}^{n} \log P_{\gamma,\theta}(a_i \mid x_i).$$

On a fixed grid $\{x^{(\ell)}\}_{\ell=1}^G$ of size $G = 101^2$, define the Bellman residuals

$$r_\ell(\gamma, \theta) = V_\gamma(x^{(\ell)}) \; - \; (\varphi_\theta[P_\gamma, V_\gamma])(x^{(\ell)}), \quad \rho_n(\gamma, \theta) = \frac{1}{G}\sum_{\ell=1}^{G} r_\ell^2.$$

Where $\varphi_\theta[P, V])(x)$ is defined as:

$$(\varphi_\theta[P, V])(x) \; = \; \sum_{a\in\mathcal{A}} P(a \mid x)\Big\{u_a(x;\theta) + \beta\,\mathbb{E}[V(X') \mid x, a] - \log P(a \mid x)\Big\}.$$

Then the inner and outer steps of the NNES estimator are implemented following the strategy in section 3.

Automatic Debiased Machine Learning (ADML) estimator

We implemented the ADML as in Nguyen [2025] to estimate the structural parameters θ. We first estimate the CCPs using single–hidden–layer ReLU networks with the same architecture in 6.4. For the output layer, we apply a soft-max transformation as illustrated in section 4.4 to ensure that the estimated CCPs are between 0 and 1. We then apply the Automatic Debiasing method introduced in Nguyen [2025] to correct the potential biases arise from the nonparametric estimation of the CCPs.

Benchmark: Oracle NFXP

The Oracle NFXP estimator solves the two 1-D DPs on $\mathcal{G}$; outer loop maximises the log-likelihood using the true separability.

Table 1: Monte-Carlo estimates (100 replications, $n = 1000$)

Method	$\hat{c}_{\text{rep},1}$	$\hat{c}_{\text{rep},2}$	$\hat{c}_1$	$\hat{c}_2$	Avg. sec
True value	2.000	2.500	0.0500	0.0800	—
NNES (k = 10)	1.9421	2.5825	0.0515	0.0878	1315.3
(numerical derivative)	(0.1755)	(0.1903)	(0.0103)	(0.0141)	
NNES (k = 10)	1.9443	2.5823	0.0515	0.0872	680.6
(analytical derivative)	(0.1754)	(0.1903)	(0.0103)	(0.0141)	
Oracle NFXP	1.9454	2.5135	0.0509	0.0843	208.3
	(0.1746)	(0.1812)	(0.0103)	(0.0134)	
ADML	1.8801	2.3622	0.0632	0.0701	222.1
	(0.1912)	(0.2012)	(0.0141)	(0.0155)	

Table 1 averages 100 replications; CPU times are per replication; Discount factor $\beta = 0.9$; $\hat{\theta}$ are all initialized at 0[1]. NNES closely matches as an oracle estimator that knows the separable structure

[1] In Appendix C.3, we conduct additional simulation results using different initial guesses of $\hat{\theta}$ to test the robustness

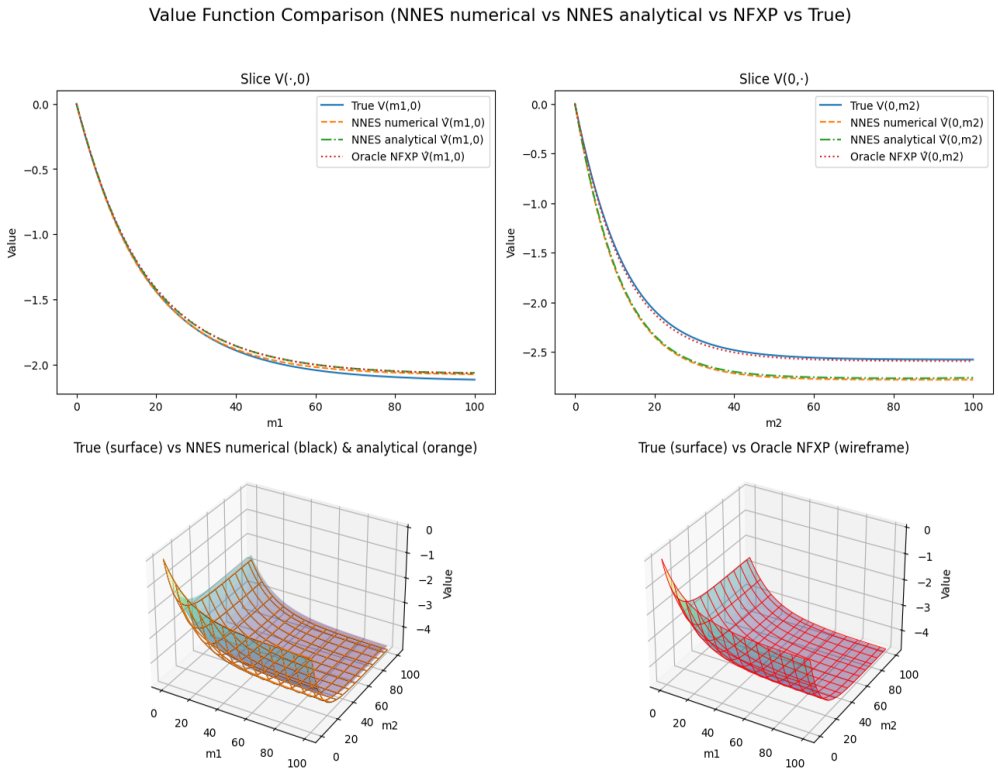


Figure 1: Comparison of true and estimated value functions. The NNES estimated function matches the performance of the Oracle estimator closely. For ADML, we can recover only differences in the expected value functions; consequently, we cannot visualize the level of the value functions for this method.

ex ante, while solving a 33-parameter optimisation and avoiding any dynamic-programming grid search. With analytical derivative, NNES's CPU cost about three times higher than the oracle because the network must learn separability from data—but remains negligible for empirical sample sizes. In this Monte Carlo design, NNES produces parameter estimates that are closer to the oracle benchmark than ADML. That said, ADML remains competitive from a computational perspective: as a two-step semiparametric estimator, it avoids forward-iteration and incurs lower CPU cost. Overall, the results suggest a clear accuracy–computation trade-off between NNES and ADML.

6.5 Four-Dimensional State (d = 4)

Design. To probe performance beyond the two-dimensional benchmark, we extend the replacement model to *four* independent modules. The state is $x = (m_1, m_2, m_3, m_4) \in [0, M_{\max}]^4$ and the action is $a = (a_1, \ldots, a_4) \in \{0, 1\}^4$, where $a_j = 1$ replaces module j. Per-period utility for module j is

$$u_j(m_j, a_j; \theta) = -\{c_j m_j (1 - a_j) + c_{\text{rep},j}\, a_j\},$$

with i.i.d. T1EV shocks and exponential mileage increments as in the two-dimensional design; the value function is represented by the anchored network used throughout the paper. We set $\beta = 0.9$, simulate $n = 1200$ observations per replication, and repeat the experiment 100 times. The oracle benchmark solves four one-dimensional DPs exploiting independence across modules; NNES uses

of NNES. We also compare its performance against the oracle NFXP and the SEES estimator in Luo and Sang [2025].

a single 4-D value net with the same anchoring and inner/outer schedule as in the $d = 2$ runs.

Table 2: Monte–Carlo results, $d = 4$ (100 replications, $n = 1200$, $\beta = 0.9$, k = 12).

	True	NNES (numerical)		NNES (analytical)		Oracle NFXP		ADML	
Parameter	value	mean	s.e.	mean	s.e.	mean	s.e.	mean	s.e.
$c_{\text{rep},1}$	2.0000	1.8947	0.1401	1.8933	0.1403	1.9733	0.1366	1.8311	0.1653
$c_{\text{rep},2}$	2.5000	2.4393	0.1665	2.4411	0.1662	2.4533	0.1658	2.3443	0.2002
$c_{\text{rep},3}$	1.5000	1.5111	0.1301	1.5109	0.1301	1.5102	0.1295	1.6010	0.1432
$c_{\text{rep},4}$	1.8000	1.9023	0.1503	1.8871	0.1501	1.8222	0.1467	1.7102	0.1545
c_1	0.0500	0.0489	0.0055	0.0482	0.0055	0.0501	0.0053	0.0662	0.0084
c_2	0.0700	0.0794	0.0070	0.0771	0.0069	0.0715	0.0067	0.0812	0.0072
c_3	0.0900	0.0890	0.0085	0.0892	0.0085	0.0897	0.0085	0.0824	0.0088
c_4	0.1100	0.1171	0.0106	0.1168	0.0105	0.1104	0.0098	0.1258	0.0153

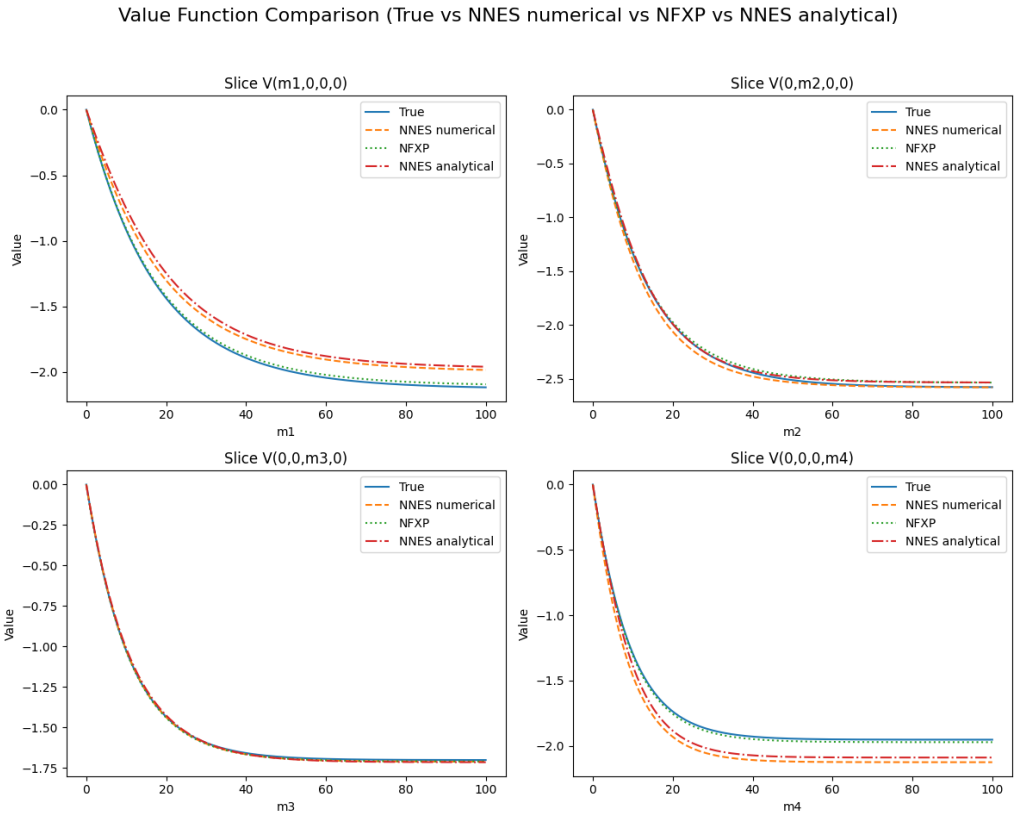


Figure 2: **Value-function cross-sections in $d = 4$.** Each panel plots $V_j(m_j)$ holding the other coordinates fixed at zero, comparing truth (solid), NNES (dots/wireframe), and the oracle. For ADML, we can recover only differences in the expected value functions; consequently, we cannot visualize the level of the value functions for this method.

Results. Table 2 summarizes Monte–Carlo means with standard errors; All $\hat{\theta}$ are initialized at 0. NNES matches the oracle closely across all eight structural parameters; cross-sectional plots of the recovered value surfaces overlay the truth and the oracle closely along each coordinate (Figure 2). Average runtime per replication is 3279.2s for NNES with numerical derivative (numerical); 1659.3s for NNES with analytical derivative (analytical) versus 428.0s for the oracle and 561.0s for ADML;

the gap reflects the oracle's bus-wise decomposition rather than a statistical disadvantage for NNES. In this Monte Carlo design, we observe the same accuracy–computation trade-off between NNES and ADML as in the two-dimensional exercise in the previous section.

7 Conclusion

This paper introduces the Neural-Network Efficient Estimator (NNES) for high-dimensional dynamic discrete choice models. NNES embeds a flexible neural network approximation of the value function within a policy-iteration outer loop. This design preserves the zero-Jacobian property of the policy map and, in turn, delivers an automatically orthogonal likelihood score and a block-diagonal information matrix. Under mild approximation requirements, these features yield $\sqrt{n}$-consistent and semiparametrically efficient inference while allowing the value function to be learned with modern neural networks. In Monte Carlo designs with multi-dimensional state spaces, NNES matches the precision of oracle NFXP estimators at a fraction of the programming complexity, illustrating its practicality for empirical work.

Extensions. Beyond the present results, we see three natural directions for extending this work.

(i) Model-free NNES. Our implementation is model-based because policy evaluation uses the transition law. A model-free variant that relies on observed transitions or simulation-based moment matching could potentially retain the same outer-loop architecture while relaxing transition-density modeling. Analyzing orthogonality, efficiency, and stability for such a model-free NNES is an important next step.

(ii) Alternative gradient computation via equilibrium propagation. NNES currently computes gradients of the profiled criterion using differentiation of the value approximation. An appealing alternative is *equilibrium propagation* in Zucchet and Sacramento [2022], which forms an augmented objective that blends the "inner" fit (policy/value consistency) and the "outer" likelihood with a small nudging weight, then estimates the outer gradient by contrasting partial derivatives at the baseline and the nudged equilibria. This approach avoids explicit Hessian–vector products, may be easier to implement with complex approximators, and could be faster or more robust in practice.

(iii) Deep surrogate structural model for smoothness. A complementary route to smooth, stable estimation is to train *one* global surrogate $V_w(x, \theta)$ that maps states and parameters jointly to the value function, keeping w fixed during estimation. This is similar to the idea of Deep Surrogates for option pricing as in Chen et al. [2023]. Treating parameters as additional (time-invariant) "pseudo-states" yields an off-the-shelf, differentiable surface whose inherent neural-network smoothness can regularize estimation and simplify gradient computation. The trade-off is the up-front cost and potentially slower approximation rates due to the enlarged input dimension, but once trained, the surrogate can make parameter searches extremely fast.

(iv) NNES for Dynamic Games. Another interesting extension is to develop an NNES-type estimator for dynamic discrete games that leverages the sequential pseudo-likelihood ideas of Aguirregabiria and Mira [2007] and the efficient, stable k-EPL sequence of Dearing and Blevins [2024] to update equilibrium objects iteratively rather than nesting a full equilibrium solver. Both contributions are formulated under a finite, discrete observed state space, making them most tractable in low-dimensional environments where conditional choice probabilities or value functions can be represented in tabular form. By replacing these discrete-state representations with neural-network approximations of players' value functions and best-response mappings, NNES's high-dimensional

continuous-state machinery and orthogonality-based inference could be brought to dynamic games, substantially widening the class of empirically relevant models that can be estimated and analyzed.

(v) Higher-order asymptotic refinement of NNES under slow first-stage/nuisance learning rates. A natural theoretical extension is to develop higher-order asymptotics for NNES in regimes where the curse of dimensionality in the continuous state space implies that first-stage policy/value learners converge slowly, so the Neyman-orthogonal structure no longer renders the second-order remainder negligible under standard inference. This would call for a refined von Mises/Bahadur expansion of the NNES estimating equation that explicitly tracks the leading higher-order bias terms induced by approximation and regularization. A promising direction could be exploiting additional outer-loop iterations as an algorithmic bias-reduction device, in the spirit of the higher-order improvements and local contraction to the finite-sample MLE shown for k-step efficient pseudo-likelihood in dynamic games (Dearing and Blevins [2024]). Such results would complement the double/debiased machine learning literature by clarifying when NNES can still deliver reliable inference even when nuisance learning rates fall below the classical fourth-root threshold (Chernozhukov et al. [2024a]).

References

K. Adusumilli and D. Eckardt. Temporal-difference estimation of dynamic discrete choice models. *arXiv preprint arXiv:1912.09509*, 2019.

V. Aguirregabiria and P. Mira. Swapping the nested fixed point algorithm: A class of estimators for discrete markov decision models. *Econometrica*, 70(4):1519–1543, 2002.

V. Aguirregabiria and P. Mira. Sequential estimation of dynamic discrete games. *Econometrica*, 75(1):1–53, 2007. doi: https://doi.org/10.1111/j.1468-0262.2007.00731.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0262.2007.00731.x`.

P. Arcidiacono and R. A. Miller. Conditional choice probability estimation of dynamic discrete choice models with unobserved heterogeneity. *Econometrica*, 79(6):1823–1867, 2011.

F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

P. Bühlmann and S. Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

H. Chen, A. Didisheim, and S. Scheidegger. Deep surrogates for finance: With an application to option pricing. *Available at SSRN 3782722*, 2023.

X. Chen. Large sample sieve estimation of semi-nonparametric models. *Handbook of econometrics*, 6:5549–5632, 2007.

V. Chernozhukov and C. Hansen. The impact of 401 (k) participation on the wealth distribution: An instrumental quantile regression analysis. *Review of Economics and statistics*, 86(3):735–751, 2004.

V. Chernozhukov, J. C. Escanciano, H. Ichimura, W. K. Newey, and J. M. Robins. Locally robust semiparametric estimation. *Econometrica*, 90(4):1501–1535, 2022.

V. Chernozhukov, W. Newey, R. Singh, and V. Syrgkanis. Automatic debiased machine learning for dynamic treatment effects and general nested functionals, 2023.

V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins. Double/debiased machine learning for treatment and causal parameters, 2024a. URL `https://arxiv.org/abs/1608.00060`.

V. Chernozhukov, W. K. Newey, V. Quintas-Martinez, and V. Syrgkanis. Automatic debiased machine learning via riesz regression, 2024b. URL `https://arxiv.org/abs/2104.14737`.

V. Chernozhukov, W. Newey, and V. Semenova. Welfare analysis in dynamic models, 2025. URL `https://arxiv.org/abs/1908.09173`.

A. Dearing and J. R. Blevins. Efficient and convergent sequential pseudo-likelihood estimation of dynamic discrete games. *The Review of Economic Studies*, 92(2):981–1021, 05 2024. ISSN 0034-6527. doi: 10.1093/restud/rdae050. URL https://doi.org/10.1093/restud/rdae050.

M. H. Farrell. Robust inference on average treatment effects with possibly more covariates than observations. *Journal of Econometrics*, 189(1):1–23, 2015.

S. Gunasekar, J. D. Lee, D. Soudry, and N. Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in neural information processing systems*, 31, 2018.

M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016.

V. J. Hotz and R. A. Miller. Conditional choice probabilities and the estimation of dynamic models. *The Review of Economic Studies*, 60(3):497–529, 1993.

E. H. Kang, H. Yoganarasimhan, and L. Jain. An empirical risk minimization approach for offline inverse rl and dynamic discrete choice model, 2025. URL https://arxiv.org/abs/2502.14131.

S. Langer. Approximating smooth functions by deep neural networks with sigmoid activation function. *Journal of Multivariate Analysis*, 182:104696, 2021. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2020.104696. URL https://www.sciencedirect.com/science/article/pii/S0047259X20302773.

Y. Luo and P. Sang. Efficient estimation of structural models via sieves, 2025. URL https://arxiv.org/abs/2204.13488.

T. Magnac and D. Thesmar. Identifying dynamic discrete decision processes. *Econometrica*, 70(2): 801–816, 2002.

R. Munos and C. Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

W. K. Newey. The asymptotic variance of semiparametric estimators. *Econometrica: Journal of the Econometric Society*, pages 1349–1382, 1994.

W. K. Newey and D. McFadden. Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245, 1994.

B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

H. Nguyen. Automatic debiased machine learning for dynamic discrete choice. https://huuhoang2211.github.io/hoangnguyen.com/WorkingPaper_ADML_for_DDC.pdf, Oct. 2025. URL https://huuhoang2211.github.io/hoangnguyen.com/WorkingPaper_ADML_for_DDC.pdf. Working paper. Version dated 2025-10-01.

M. Pesendorfer and P. Schmidt-Dengler. Asymptotic least squares estimators for dynamic games. *The Review of Economic Studies*, 75(3):901–928, 2008.

J. Rust. Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica: Journal of the Econometric Society*, pages 999–1033, 1987.

J. Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875 – 1897, 2020. doi: 10.1214/19-AOS1875. URL https://doi.org/10.1214/19-AOS1875.

V. Semenova. Machine learning for dynamic discrete choice. *arXiv preprint arXiv:1808.02569*, 2018.

D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57, 2018.

C.-L. Su and K. L. Judd. Constrained optimization approaches to estimation of structural models. *Econometrica*, 80(5):2213–2230, 2012.

J. Tsitsiklis and B. Van Roy. Analysis of temporal-diffference learning with function approximation. *Advances in neural information processing systems*, 9, 1996.

A. W. Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

A. W. Van Der Vaart and J. A. Wellner. Weak convergence. In *Weak convergence and empirical processes: with applications to statistics*, pages 16–28. Springer, 1996.

S. Wager, S. Wang, and P. S. Liang. Dropout training as adaptive regularization. *Advances in neural information processing systems*, 26, 2013.

D. Yarotsky. Error bounds for approximations with deep relu networks. *Neural networks*, 94: 103–114, 2017.

N. Zucchet and J. Sacramento. Beyond backpropagation: Bilevel optimization through implicit differentiation and equilibrium propagation. *Neural Computation*, 34(12):2309–2346, 11 2022. ISSN 0899-7667. doi: 10.1162/neco_a_01547. URL https://doi.org/10.1162/neco_a_01547.

A Appendix

A.1 Proof of Proposition 3

Proof of Proposition 3. Let D denote the Fréchet derivative of an operator. The proof proceeds by applying the chain rule for Fréchet derivatives, leveraging the zero-Jacobian property.

(i) Derivative of the Composite Evaluation Operator

$$D_V[\varphi_{\theta_0} \circ \Lambda_{\theta_0}](V^*) = D_P\varphi_{\theta_0}(P^*) \circ D_V\Lambda_{\theta_0}(V^*)$$

The crucial insight, established in Proposition 2 of Aguirregabiria and Mira [2002], is that the evaluation operator's derivative is zero at the optimal policy. In our notation, this is precisely:

$$D_P\varphi_{\theta_0}(P^*) = \mathbf{0},$$

where $\mathbf{0}$ is the zero operator mapping from the space of CCPs to the space of value functions.

Substituting this into the chain rule expression, we have:

$$D_V[\varphi_{\theta_0} \circ \Lambda_{\theta_0}](V^*) = D_P\varphi_{\theta_0}(P^*) \circ D_V\Lambda_{\theta_0}(V^*) = \mathbf{0} \circ D_V\Lambda_{\theta_0}(V^*) = \mathbf{0}.$$

(ii) Derivative of the Composite Policy-Iteration Operator

A direct consequence of the zero-Jacobian property for the evaluation operator is that the full policy-iteration operator $\Psi_{\theta_0} = \Lambda_{\theta_0} \circ \varphi_{\theta_0}$ also has a zero derivative at its fixed point $P^* = P_{V*}$. Formally:

$$D_V\Psi_{\theta_0}(P_{V^*}) = D_V\Lambda_{\theta_0}(\varphi_{\theta_0}(P^*)) \circ D_V[\varphi_{\theta_0} \circ \Lambda_{\theta_0}](V^*) = D_V\Lambda_{\theta_0}(V^*) \circ \mathbf{0} = \mathbf{0}.$$

This demonstrates that the mapping from Value Function to the next-iterate CCPs is also flat at the fixed point.

□

A.2 Proof of Lemma 4.1

Proof. Let $\ell(V) := \mathbb{P}[\ell_n(V)] = \mathbb{E}[\log \Lambda_\theta[V](A|X)]$. Define the policies $P_1 = \Lambda_\theta[V_1]$ and $P_2 = \Lambda_\theta[V_2]$. The functional $\log(\cdot)$ is differentiable, so by the Mean Value Theorem, there exists a $\bar{t} \in (0,1)$ such that for the path $\bar{P} = (1-\bar{t})P_1 + \bar{t}P_2$, we have:

$$\ell(V_1) - \ell(V_2) = \mathbb{P}[\log P_1] - \mathbb{P}[\log P_2] = \mathbb{E}_{A,X}\left[\nabla_P \log P(A|X)|_{P=\bar{P}} \cdot (P_1(A|X) - P_2(A|X))\right].$$

The expectation $\mathbb{E}_{A,X}$ is taken over the true data-generating process. We can decompose this expectation using the law of iterated expectations, where the inner expectation is over actions A conditional on the state X. By definition, this conditional expectation weights each action a by the true conditional choice probability, $P^*(a|X)$.

$$\begin{aligned}\ell(V_1) - \ell(V_2) &= \mathbb{E}_X\left[\mathbb{E}_{A|X}\left[\frac{1}{\bar{P}(A|X)}(P_1(A|X) - P_2(A|X))\right]\right] \\ &= \mathbb{E}_X\left[\sum_{a\in\mathcal{A}} P^*(a|X)\frac{1}{\bar{P}(a|X)}(P_1(a|X) - P_2(a|X))\right].\end{aligned}$$

By Assumption 4, P^* is bounded. We also assume that any policy $P \in \mathcal{S}_{P,n}$ (and any convex combination thereof, like $\bar{P}$) is bounded away from zero. Thus, the fraction $|P^*(a|X)/\bar{P}(a|X)|$ is uniformly bounded by a constant $M_1 = p_{max}/p_{min}$. Taking absolute values:

$$\begin{aligned}
|\ell(V_1) - \ell(V_2)| &\leq M_1 \mathbb{E}_X \left[\sum_{a \in \mathcal{A}} |P_1(a|X) - P_2(a|X)| \right] \\
&\leq M_1 \mathbb{E}_X \left[\sqrt{J} \left(\sum_{a \in \mathcal{A}} (P_1(a|X) - P_2(a|X))^2 \right)^{1/2} \right] \qquad \text{(Cauchy-Schwarz in } \mathbb{R}^J\text{)} \\
&\leq M_1 \sqrt{J} \left(\mathbb{E}_X \left[\sum_{a \in \mathcal{A}} (P_1(a|X) - P_2(a|X))^2 \right] \right)^{1/2} \qquad \text{(Jensen's Inequality)} \\
&= M_1 \sqrt{J} \|P_1 - P_2\|_2.
\end{aligned}$$

Finally, applying the Lipschitz property of the policy operator from Assumption 6, we have $\|P_1 - P_2\|_2 \leq L_\Lambda \|V_1 - V_2\|_2$. This gives the final result with constant $C_\ell := M_1 \sqrt{J} L_\Lambda$. □

A.3 Proof of Lemma 4.2

Proof. In the following proofs, we use the notation $a_n \lesssim b_n$ to denote $a_n \leq C \cdot b_n$ for some universal constant$C > 0$ that does not depend on the sample size n. According to Dudley's entropy integral criterion [Van Der Vaart and Wellner, 1996, e.g.,], a class $\mathcal{F}$ is P-Donsker if its entropy integral is finite: $\int_0^\infty \sqrt{\sup_{\mathbb{Q}} \log N(\varepsilon, \mathcal{F}, L_2(\mathbb{Q}))}\, d\varepsilon < \infty$. For an asymptotically Donsker property of a shrinking class $\mathcal{H}_{V,n}$, we must show that this integral vanishes as $n \to \infty$.

The entropy of our class $\mathcal{H}_{V,n}$ is bounded by the entropy of the underlying sieve, $\mathcal{S}_{V,n}$, due to the Lipschitz continuity of the map $V \mapsto \ell_n(V)$ (proved in lemma 4.1). Applying the entropy bound from Assumption 7, the integral is bounded by:

$$J_n := \int_0^{\text{diam}(\mathcal{H}_{V,n})} \sqrt{\sup_{\mathbb{Q}} \log N(\varepsilon, \mathcal{H}_n(r_{V,n}), L_2(\mathbb{Q}))}\, d\varepsilon \lesssim \int_0^{C_\ell r_{V,n}} \sqrt{p_n \log(B/\varepsilon)}\, d\varepsilon.$$

We use the substitution $u = \varepsilon / r_{V,n}$, so $du = d\varepsilon / r_{V,n}$.

$$J_n \lesssim \sqrt{p_n} \int_0^{C_\ell} \sqrt{\log(B/(u r_{V,n}))}\, r_{V,n} du = r_{V,n} \sqrt{p_n} \int_0^{C_\ell} \sqrt{\log(B/u) + \log(1/r_{V,n})}\, du.$$

Since $r_{V,n} \to 0$, $\log(1/r_{V,n}) \sim \log\left(n^\delta\right) \sim \log(n)$. The term $\log(B/u)$ is a fixed function integrated over a fixed interval. Therefore, the integral is dominated by the $\log(n)$ term:

$$J_n \lesssim r_{V,n} \sqrt{p_n \log n}.$$

For the class to be asymptotically Donsker, we need this integral to vanish. Substituting $r_{V,n} \sim n^{-\delta}$:

$$J_n \lesssim n^{-\delta} \sqrt{p_n \log n} \to 0.$$

This holds if and only if $p_n = o(n^{2\delta} / \log n)$, which is precisely the condition imposed by Assumption 7. Thus, the the class $\mathcal{H}_{V,n}$ is asymptotically P-Donsker. □

A.4 Proof of Proposition 2

Proof. Let $R_n := n^{-\delta} + \sqrt{\frac{p_n \log n}{n}}$. By our assumptions, $R_n = o(n^{-1/4})$. The proof proceeds by induction. Assume for $k-1$ that $\sup_\theta \|\widehat{V}^{k-1} - V_\theta^*\|_2 = O_p(R_n)$. Let $\hat{V} := \widehat{V}^k$, $V^* := V_\theta^*$, $\hat{P} := P^{k-1}$, and $\tilde{V}$ be a best-in-sieve approximation to V^*.

Step 1: Basic Inequality. The optimality of $\hat{V}$ for the objective function implies that for any given θ:

$$-\mathbb{P}_n \ell_\theta(\hat{V}) + \omega_n \rho_n(\hat{V}, \hat{P}) \leq -\mathbb{P}_n \ell_\theta(\tilde{V}) + \omega_n \rho_n(\tilde{V}, \hat{P}). \tag{40}$$

where $\rho_n(V, P) := \|V - \varphi_\theta[P]\|^2_{L_2(\pi_{M_n})}$ is the Bellman penalty evaluated on the deterministic grid measure π_{M_n}.

Step 2: Decomposition and Bounding. Rearranging (40) yields:

$$\begin{aligned}\omega_n \rho_n(\hat{V}, \hat{P}) \leq \omega_n \rho_n(\tilde{V}, \hat{P}) + \mathbb{P}[\ell_\theta(\hat{V}) - \ell_\theta(\tilde{V})] \\ + (\mathbb{P}_n - \mathbb{P})[\ell_\theta(\hat{V}) - \ell_\theta(\tilde{V})].\end{aligned} \tag{41}$$

Now, we use Assumption 10 to relate the grid-based penalty ρ_n to the true integrated squared Bellman residual, which we denote by $\rho(V, P) := \|V - \varphi_\theta[P, V]\|^2_{L_2(\mu)}$. The assumption implies:

$$\rho_n(V, P) = \rho(V, P) + o(1/\omega_n).$$

Substituting this into (41) for both $\rho_n(\hat{V}, \hat{P})$ and $\rho_n(\tilde{V}, \hat{P})$:

$$\begin{aligned}\omega_n \left(\rho(\hat{V}, \hat{P}) + o_p(n^{-1/2})\right) \leq \omega_n \left(\rho(\tilde{V}, \hat{P}) + o(n^{-1/2})\right) + \mathbb{P}[\ell_\theta(\hat{V}) - \ell_\theta(\tilde{V})] \\ + (\mathbb{P}_n - \mathbb{P})[\ell_\theta(\hat{V}) - \ell_\theta(\tilde{V})].\end{aligned}$$

Assuming $\omega_n = o(n^{1/2})$, the $\omega_n \cdot o(n^{-1/2})$ terms are $o(1)$ and can be absorbed. This simplifies the expression to:

$$\omega_n \rho(\hat{V}, \hat{P}) \leq \omega_n \rho(\tilde{V}, \hat{P}) + \mathbb{P}[\ell_\theta(\hat{V}) - \ell_\theta(\tilde{V})] + (\mathbb{P}_n - \mathbb{P})[\ell_\theta(\hat{V}) - \ell_\theta(\tilde{V})] + o(1). \tag{42}$$

We now bound the terms on the right-hand side of (42).

- **Population Bellman Residual of $\tilde{V}$:** $\rho(\tilde{V}, \hat{P}) = \left\|\tilde{V} - \varphi_\theta[\hat{P}, \hat{V}]\right\|_2^2 \leq (\|\tilde{V} - V^*\|_2 + \|V^* - \varphi_\theta[\hat{P}, \hat{V}]\|_2)^2 \leq (o_p(n^{-\alpha_V}) + L_\phi L_\Lambda O_p(R_n))^2 = O_p(R_n^2)$.
- **Population Likelihood**: $|\mathbb{P}[\ell_\theta(\hat{V}) - \ell_\theta(\tilde{V})]| \leq C_\ell(\left\|\hat{V} - V^*\right\|_2 + \|\tilde{V} - V^*\|_2) \leq C_\ell(\left\|\hat{V} - V^*\right\|_2 + o_p(n^{-\alpha_V}))$.
- **Empirical Process Term**: Because $\hat{V}, \tilde{V}$ lie in a shrinking neighborhood of V^*, we can apply the local Donsker result from Lemma 4.2 to the class of functions $\{\ell_\theta(V) : V \in \mathcal{S}_{V,n}\}$. The term is stochastically equicontinuous, so $(\mathbb{P}_n - \mathbb{P})[\ell_\theta(\hat{V}) - \ell_\theta(\tilde{V})] = o_p(n^{-1/2})$.

Substituting these bounds into (42):

$$\omega_n \rho(\hat{V}, \hat{P}) \leq \omega_n O_p(R_n^2) + C_\ell(\left\|\hat{V} - V^*\right\|_2 + o_p(R_n)) + o(1).$$

Since $\omega_n \to \infty$, for large n we can divide by ω_n:

$$\rho(\hat{V}, \hat{P}) = \left\|\hat{V} - \varphi_\theta[\hat{P}, \hat{V}]\right\|_2^2 \leq O_p(R_n^2) + \frac{C_\ell}{\omega_n}\left\|\hat{V} - V^*\right\|_2 + o(\omega_n^{-1}). \tag{43}$$

Step 3: Final Error Recursion Let

$$\varepsilon_k := \left\|\hat{V}^k - V^*\right\|_2, \qquad \hat{P} := \Lambda_\theta[\hat{V}^{\,k-1}], \qquad P^* := \Lambda_\theta[V^*].$$

By definition $V^* = \varphi_\theta[P^*, V^*]$. Using Lipschitz continuity (Assumption 6), there exists a constant $L := L_\varphi L_\Lambda$ such that

$$\left\|\varphi_\theta[\hat{P}, \hat{V}] - V^*\right\|_2 = \left\|\varphi_\theta[\hat{P}, \hat{V}] - \varphi_\theta[P^*, V^*]\right\|_2 \leq L\,\varepsilon_{k-1}.$$

Decompose the L_2 error at iteration k by the triangle inequality:

$$\varepsilon_k \leq \left\|\hat{V}^k - \varphi_\theta[\hat{P}, \hat{V}]\right\|_2 + \left\|\varphi_\theta[\hat{P}, \hat{V}] - V^*\right\|_2 \leq \left\|\hat{V}^k - \varphi_\theta[\hat{P}, \hat{V}]\right\|_2 + L\,\varepsilon_{k-1}.$$

Note that the norm $\|\cdot\|_2$ is the true integrated norm, matching the term in our residual bound (43). Squaring and applying $(a+b)^2 \leq 2a^2 + 2b^2$ gives:

$$\varepsilon_k^2 \leq 2\left\|\hat{V}^k - \varphi_\theta[\hat{P}, \hat{V}]\right\|_2^2 + 2L^2\,\varepsilon_{k-1}^2.$$

Invoke the residual bound from Step 2 (equation (43)):

$$\varepsilon_k^2 \leq 2\Big(O_p(R_n^2) + \frac{C_\ell}{\omega_n}\,\varepsilon_k + o_p(\omega_n^{-1})\Big) + 2L^2\,\varepsilon_{k-1}^2.$$

Using the induction hypothesis $\varepsilon_{k-1} = O_p(R_n)$ and absorbing constants:

$$\varepsilon_k^2 - \frac{2C_\ell}{\omega_n}\,\varepsilon_k - O_p(R_n^2) \leq 0.$$

The positive root of $x^2 - bx - c \leq 0$ is $x \leq \frac{1}{2}(b + \sqrt{b^2 + 4c})$. Here $b = O(n^{-\delta})$ and $c = O_p(R_n^2)$. Since $\delta > 1/4$ and $R_n = o(n^{-\delta})$, we have $b = o(R_n)$, hence

$$\varepsilon_k = O_p(n^{-\delta}) + O_p(R_n) = O_p(R_n).$$

This bound holds uniformly in θ. By construction, $R_n = o(n^{-\delta}) = o(n^{-1/4})$. This completes the induction and the proof. □

A.5 Proof of Proposition 4 (Neyman Orthogonality)

Proof. Let the value function be parameterized by a neural network $V_\gamma(x) = G(x;\gamma)$. The CCPs are generated from this value function via the improvement operator, Λ_θ:

$$P(a|x;\theta,V) := (\Psi_\theta(\Lambda_\theta[V]))\,(a|x).$$

The score function for the structural parameter θ, given an observation $W = (A, X)$, is the derivative of the log-likelihood contribution with respect to θ:

$$\psi(W;\theta,V) := \frac{\partial}{\partial\theta}\log P(A|X;\theta,V).$$

We need to show that this score is Neyman orthogonal with respect to V at the true parameters (θ_0, V^*). Here, V^* is the true value function, which is the unique solution to the Bellman equation. Formally, we must show that the gradient of the expected score with respect to V vanishes:

$$\nabla_V \mathbb{E}\left[\psi(W;\theta_0,V)\right]\Big|_{V=V^*} = 0.$$

The expectation $\mathbb{E}[\cdot]$ is taken over the true data-generating process. We introduce the solution manifold. For any θ in a neighborhood of θ_0, there exists a unique value function V_θ that solves the Bellman equation. This relationship defines the solution manifold. To ensure this manifold is well-behaved, we posit an additional regularity condition standard for such proofs:

(iii) **Non-singularity of the Bellman Residual Jacobian.** The Bellman residual operator $r(\theta, V) = V - \varphi_\theta[\Lambda_\theta[V]]$ has a Fréchet derivative with respect to V, $D_V r(\theta_0, V^*)$, that is invertible.

Under this condition, the Implicit Function Theorem guarantees the existence of a unique, continuously differentiable map $\theta \mapsto V(\theta)$ in a neighborhood of θ_0 such that $r(\theta, V(\theta)) \equiv 0$. This function $V(\theta)$ describes the value function that solve the model for any given θ. The proof proceeds by analyzing the derivatives along this manifold.

Step 1: Interchange of Differentiation and Expectation.

The Dominated Convergence Theorem allows us to interchange the gradient operator ∇_γ and the expectation operator $\mathbb{E}[\cdot]$:

$$\nabla_V \mathbb{E}\left[\psi(W; \theta_0, V)\right]\Big|_{V=V^*} = \mathbb{E}\left[\nabla_V \psi(W; \theta_0, V)\Big|_{V=V^*}\right].$$

Our goal is to show that the term inside the expectation, $\nabla_V \psi(W; \theta_0, V^*)$, is equal to zero.

Step 2: Applying the Chain Rule to the Score.

By definition, $\psi(W; \theta, V) = \frac{\partial}{\partial \theta} \log P(A|X; \theta, V)$. We compute its gradient with respect to V. Assuming sufficient smoothness to apply Clairaut's Theorem, we can interchange the order of differentiation:

$$\begin{aligned}\nabla_V \psi(W; \theta, V) &= \frac{\partial}{\partial V}\left(\frac{\partial}{\partial \theta} \log P(A|X; \theta, V)\right) \\ &= \frac{\partial}{\partial \theta}\left(\frac{\partial}{\partial V} \log P(A|X; \theta, V)\right).\end{aligned}$$

Using the chain rule for logarithms on the inner derivative gives the full expression:

$$\nabla_V \psi(W; \theta, V) = \frac{\partial}{\partial \theta}\left[\frac{1}{P(A|X; \theta, V)} \nabla_V P(A|X; \theta, V)\right].$$

Step 3: The Zero-Jacobian Property on the Solution Manifold. The Zero-Jacobian Property 1 states that the derivative of the policy-iteration operator $\Psi_\theta := \Lambda_\theta \circ \varphi_\theta$ with respect to the policy is zero when evaluated at a fixed point.

For any θ in the neighborhood of θ_0, the pair $(\theta, V(\theta))$ lies on the solution manifold, meaning $P_{V(\theta)}$ is a fixed point of Ψ_θ. Therefore, the ZJP (established in Proposition 1) holds for any such point:

$$D_P \Psi_\theta(P)\big|_{P=P_{V(\theta)}} = 0.$$

Now consider the fixed-point identity, which holds for all θ on the manifold:

$$P(A|X; \theta, V(\theta)) = \left(\Psi_\theta[P_{V(\theta)}]\right)(A|X).$$

Differentiating this identity with respect to V (holding θ fixed) and evaluating at $V = V(\theta)$ yields:

$$\nabla_V P(A|X; \theta, V)\Big|_{V=V(\theta)} = \left(D_P \Psi_\theta(P)\big|_{P=P_{V(\theta)}}\right)\left[\nabla_V P(A|X; \theta, V)\Big|_{V=V(\theta)}\right].$$

Since $D_P\Psi_\theta(P_{V(\theta)}) = 0$, the right-hand side is zero. This gives the crucial intermediate result:

$$\nabla_V P(A|X;\theta,V(\theta)) = 0, \quad \text{for all } \theta \text{ in a neighborhood of } \theta_0. \tag{44}$$

Crucially, this result demonstrates that the derivative of the CCPs with respect to the $V$is zero not just at the single point (θ_0, V^*), but all along the path where the V co-move with θ to satisfy the model's equilibrium condition.

Step 4: Assembling the Final Result.

We now substitute this result back into our expression for $\nabla_V \psi$ from Step 2. We evaluate the expression at (θ_0, V^*), which lies on the path $(\theta, V(\theta))$.

$$\nabla_V \psi(W;\theta_0,V^*) = \frac{\partial}{\partial\theta}\left[\frac{1}{P(A|X;\theta,V)}\nabla_V P(A|X;\theta,V)\right]\Big|_{(\theta_0,V^*)}.$$

Consider the function inside the brackets, evaluated along the solution manifold:

$$f(\theta) := \frac{1}{P(A|X;\theta,V(\theta))}\nabla_V P(A|X;\theta,V(\theta)).$$

From our result in Step 3, equation (44), we have shown that $\nabla_V P(A|X;\theta,V(\theta))$ is identically zero for all θ in a neighborhood of θ_0. Consequently, $f(\theta) \equiv 0$ in this neighborhood.

The derivative of a function that is identically zero is also zero. Therefore, when we evaluate the derivative with respect to θ at θ_0, we find:

$$\nabla_V \psi(W;\theta_0,V^*) = \frac{d}{d\theta} f(\theta)\Big|_{\theta=\theta_0} = 0.$$

Finally, substituting this result into the expression from Step 1 completes the proof:

$$\nabla_V \mathbb{E}\left[\psi(W;\theta_0,V)\right]\Big|_{V=V^*} = \mathbb{E}\left[0\right] = 0.$$

This demonstrates that the score function ψ for the structural parameters θ is orthogonal to the V at the true value. □

B Proof of Theorem 4.3

Proof. The proof establishes the asymptotic distribution of the NNES estimator $\widehat{\theta}$. It is structured to first prove the consistency of $\widehat{\theta}$, then establish the convergence rates of the nuisance estimators, and finally establish asymptotic normality and efficiency by analyzing its influence function.

Step 1: Consistency of $\widehat{\theta}$. Before establishing asymptotic normality, we must prove that $\widehat{\theta} \xrightarrow{p} \theta_0$. We verify the conditions for consistency of sieve M-estimators [Newey and McFadden, 1994, Chen, 2007]. The estimator $\widehat{\theta}$ maximizes the sample profile log-likelihood $\mathcal{L}_n(\theta) := \mathbb{P}_n[\log P_{\theta,\widehat{\gamma}(\theta)}(A|X)]$. The required conditions are:

1. The parameter space Θ is compact.
2. The population objective function $Q(\theta) := \mathbb{E}[\log P_{\theta,\gamma^*(\theta)}(A|X)]$ is uniquely maximized at the true parameter θ_0.
3. The sample objective function converges uniformly in probability to the population objective: $\sup_{\theta\in\Theta} |\mathcal{L}_n(\theta) - Q(\theta)| \xrightarrow{p} 0$.

(a) Verification of Compactness. The parameter space Θ for the structural parameters is assumed to be a compact by Assumption 4. This condition is satisfied.

(b) Verification of Population Identification. This condition ensures that the true parameter θ_0 is distinguishable from any other parameter $\theta \in \Theta$ at the population level. The difference between the population objective at θ_0 and at any other θ can be written as:

$$\begin{aligned} Q(\theta_0) - Q(\theta) &= \mathbb{E}[\log P_{\theta_0,\gamma(\theta_0)}(A|X)] - \mathbb{E}[\log P_{\theta,\gamma(\theta)}(A|X)] \\ &= \mathbb{E}\left[\log \frac{P^*(A|X)}{P_{\theta,\gamma(\theta)}(A|X)}\right], \end{aligned}$$

where we have used the fact that at the true parameter, $P_{\theta_0,\gamma^*(\theta_0)} = P^*$. The final expression is the Kullback-Leibler (KL) divergence between the true data generating process P^* and the model-implied process $P_{\theta,\gamma(\theta)}$. By Gibbs' inequality, the KL divergence is always non-negative, and is zero if and only if the two distributions are identical almost everywhere.

$$Q(\theta_0) - Q(\theta) = D_{KL}(P^*||P_{\theta,\gamma(\theta)}) \geq 0.$$

For unique identification, this inequality must be strict for all $\theta \neq \theta_0$. This requires that for any $\theta \neq \theta_0$, the CCP function $P_{\theta,\gamma(\theta)}$ implied by the model is distinct from the true CCP function P^* on a set of positive measure. This high-level identification condition is met following the results from Rust [1987] and Hotz and Miller [1993].

(c) Verification of Uniform Convergence. To prove that $\sup_{\theta\in\Theta} |\mathcal{L}_n(\theta) - Q(\theta)| \xrightarrow{p} 0$, we use the triangle inequality to decompose the uniform deviation into a stochastic error term (Term I) and a non-stochastic approximation error term (Term II):

$$\sup_{\theta\in\Theta} |\mathcal{L}_n(\theta) - Q(\theta)| \leq \underbrace{\sup_{\theta\in\Theta} |\mathcal{L}_n(\theta) - \mathbb{E}[\mathcal{L}_n(\theta)]|}_{\text{Term I}} + \underbrace{\sup_{\theta\in\Theta} |\mathbb{E}[\mathcal{L}_n(\theta)] - Q(\theta)|}_{\text{Term II}}.$$

We now show that both terms converge to zero in probability.

Bounding Term I (Stochastic Error): This term requires a Uniform Law of Large Numbers (ULLN) over the class of functions $\mathcal{F}_n = \{\log P_{\theta,\gamma}(\cdot|\cdot) : \theta \in \Theta, \gamma \in \Gamma_n\}$. Such a ULLN holds if the function class is not too complex, a property formally captured by its entropy. Assumption 7 places direct constraints on the entropy and parameter growth rate of our neural network sieve Γ_n. These conditions are sufficient for the class $\mathcal{F}_n$ to be Glivenko-Cantelli [Chen, 2007]. This means the sample average converges to its expectation uniformly over all possible functions in the class:

$$\sup_{\theta\in\Theta,\gamma\in\Gamma_n} |\mathbb{P}_n[\log P_{\theta,\gamma}] - \mathbb{E}[\log P_{\theta,\gamma}]| \xrightarrow{p} 0.$$

Since for any given θ, the estimated parameter $\widehat{\gamma}(\theta)$ is an element of Γ_n, the uniform convergence over the entire class implies convergence along the specific path defined by $\widehat{\gamma}(\theta)$. Thus, Term I converges to zero in probability.

Bounding Term II (Approximation Error): This term captures how the error in estimating

the Value Function V^* affects the objective function. We bound it as follows:

$$\begin{aligned}\sup_{\theta\in\Theta}|\mathbb{E}[\mathcal{L}_n(\theta)] - Q(\theta)| &= \sup_{\theta\in\Theta}|\mathbb{E}[\log P_{\theta,\widehat{\gamma}(\theta)}] - \mathbb{E}[\log P_{\theta,\gamma^*(\theta)}]| \\ &\leq \sup_{\theta\in\Theta} C_\ell \|P_{\theta,\widehat{\gamma}(\theta)} - P_{\theta,\gamma^*(\theta)}\|_{\mathcal{L}_2(\mathbb{P}_X)} \\ &\leq \sup_{\theta\in\Theta} C_\ell L_\Lambda \|V_{\widehat{\gamma}(\theta)} - V^*_\theta\|_{\mathcal{L}_2(\mathbb{P}_X)}\end{aligned}$$

The first inequality holds because the logarithm is Lipschitz continuous on any interval bounded away from zero; our Assumption 4 (strictly interior probabilities) ensures this. The second inequality follows from the Lipschitz continuity of the policy operator Λ_θ, established in Assumption 6.

The final term involves the estimation error of the value function. Proposition 2 shows that the value function estimator converges uniformly over θ:

$$\sup_{\theta\in\Theta}\|V_{\widehat{\gamma}(\theta)} - V^*_\theta\|_{\mathcal{L}_2(\mathbb{P}_X)} = o_p(n^{-1/4}).$$

Therefore, Term II also converges to zero in probability. Since both Term I and Term II converge to zero, we have established the uniform convergence of the objective function. All conditions for consistency are met, and we conclude by the argmax consistency theorem that $\widehat{\theta}$ is a consistent estimator for θ_0.

Step 2: Convergence Rate of Nuisance Functions. With the consistency of $\widehat{\theta}$ established in Step 1, we now state the convergence rates for the nuisance functions required for the subsequent steps.

Rate of the Value Function Estimator ($\widehat{V}$): As formally proven in Proposition 2, the value function estimator, $\widehat{V} = V_{\widehat{\gamma}}$, satisfies:

$$\|\widehat{V} - V^*\|_{L_2} = o_p(n^{-1/4}).$$

Rate of the CCP Estimator ($\widehat{P}$): Assumption 8 posits that the initial non-parametric estimator $\widehat{P}^0$ converges at a rate faster than $n^{-1/4}$:

$$\|\widehat{P} - P^*\|_{L_2} = o_p(n^{-\alpha_P}) \quad \text{with} \quad \alpha_P > 1/4.$$

This ensures that the estimation error from the first-stage CCP estimation is small enough not to create first-order bias in the final estimator for θ.

Step 3: Mean-Value Expansion of the First-Order Condition. The score function is defined as:

$$\psi(W;\theta,V) := \nabla_\theta \log\left((\Lambda_\theta[V_\gamma])(A|X)\right).$$

We perform a mean-value expansion of the moment condition around the true parameter θ_0. This yields:

$$0 = \mathbb{P}_n[\psi(W;\theta_0,\widehat{V})] + \mathbb{P}_n[\nabla_\theta\psi(W;\bar{\theta},\widehat{V})](\widehat{\theta} - \theta_0) \tag{45}$$

where $\bar{\theta}$ is a mean value on the line segment between $\widehat{\theta}$ and θ_0. Rearranging this expression and multiplying by $\sqrt{n}$ gives the basis for the influence function:

$$\sqrt{n}(\widehat{\theta} - \theta_0) = -\left(\mathbb{P}_n[\nabla_\theta\psi(W;\bar{\theta},\widehat{V})]\right)^{-1}\sqrt{n}\mathbb{P}_n[\psi(W;\theta_0,\widehat{V})] \tag{46}$$

The remainder of the proof involves analyzing the two main terms on the right-hand side. We first analyze the Jacobian matrix term.

Convergence of the Jacobian Term. We show that the sample Jacobian matrix converges in probability to its deterministic population counterpart, $G_\theta := \mathbb{E}[\nabla_\theta \psi(W; \theta_0, \gamma^*)]$.

$$\begin{aligned}\mathbb{P}_n[\nabla_\theta \psi(W; \bar{\theta}, \widehat{V})] &= (\mathbb{P}_n - \mathbb{E})[\nabla_\theta \psi(W; \bar{\theta}, \widehat{V})] + \mathbb{E}[\nabla_\theta \psi(W; \bar{\theta}, \widehat{V})] \\ &\xrightarrow{p} \mathbb{E}[\nabla_\theta \psi(W; \theta_0, V^*)].\end{aligned}$$

This convergence relies on three arguments:

(i) **Uniform Law of Large Numbers (ULLN):** The term $(\mathbb{P}_n - \mathbb{E})[\nabla_\theta \psi(W; \bar{\theta}, \widehat{V})]$ converges to zero in probability. This holds because the function class indexed by (θ, V) is sufficiently manageable. Specifically, assumption 7 bounds the complexity of the sieve space Γ_n, which is sufficient for a ULLN to hold over this class [Chen, 2007].

(ii) **Consistency of Estimators:** We have already established that the estimators are consistent: $\widehat{\theta} \xrightarrow{p} \theta_0$, which implies $\bar{\theta} \xrightarrow{p} \theta_0$. From Proposition 2, we have $\|\widehat{V}_{\widehat{\gamma}} - V^*\|_{L_2} = o_p(n^{-1/4})$, which implies consistency of the nuisance function estimator.

By the ULLN and the consistency of the estimators, combined with the Continuous Mapping Theorem, the sample Jacobian converges to the population expectation evaluated at the true parameters. Thus, we have:

$$\mathbb{P}_n[\nabla_\theta \psi(W; \bar{\theta}, \widehat{V})] \xrightarrow{p} G_\theta.$$

Step 4: Decomposition of the Score Term. The core of the asymptotic analysis is to characterize the behavior of the term $\sqrt{n}\mathbb{P}_n[\psi(W; \theta_0, \widehat{V})]$ from the expansion in Step 3. We decompose this term into three distinct components. By adding and subtracting terms, we can write the exact identity:

$$\begin{aligned}\sqrt{n}\mathbb{P}_n[\psi(W; \theta_0, \widehat{V})] = \quad & \underbrace{\sqrt{n}\mathbb{P}_n[\psi(W; \theta_0, V^*)]}_{\text{(A) Asymptotic Distribution Term}} \\ + \quad & \underbrace{\sqrt{n}\mathbb{E}[\psi(W; \theta_0, \widehat{V}) - \psi(W; \theta_0, V^*)]}_{\text{(B) Nuisance Bias Term}} \\ + \quad & \underbrace{\sqrt{n}(\mathbb{P}_n - \mathbb{E})[\psi(W; \theta_0, \widehat{V}) - \psi(W; \theta_0, V^*)]}_{\text{(C) Nuisance Stochastic Term}}.\end{aligned}$$

The three terms have distinct interpretations:

- **Term (A)** is the score evaluated at the true parameters. As a sum of i.i.d. mean-zero random variables, it will converge to a normal distribution by the Central Limit Theorem and will determine the asymptotic variance of our estimator.
- **Term (B)** is a non-stochastic (at a given $\widehat{V}$) term that captures the bias introduced by using an estimated nuisance parameter $\widehat{V}$ instead of the true one V^*. We will show this term is negligible due to the Neyman Orthogonality property of our score.
- **Term (C)** is a stochastic error term that arises because the nuisance parameter $\widehat{V}$ is estimated using the same data. We will show this term is negligible by leveraging the Donsker properties of our neural network sieve.

Step 5: Bounding the Nuisance Bias Term (B). We now show that Term (B) is asymptotically negligible. The term is defined as:

$$\sqrt{n}\mathbb{E}[\psi(W;\theta_0,\widehat{V}) - \psi(W;\theta_0,V^*)].$$

We analyze the term $\mathbb{E}[\psi(W;\theta_0,\widehat{V})]$ using a second-order functional Taylor expansion of $\mathbb{E}[\psi(W;\theta_0,V)]$ around the true parameter V^*. This gives:

$$\mathbb{E}[\psi(W;\theta_0,\widehat{V})] = \mathbb{E}[\psi(W;\theta_0,V^*)] + \mathcal{D}_V\mathbb{E}[\psi(W;\theta_0,V^*)][\widehat{V} - V^*] + R_n,$$

where $\mathcal{D}_V$ denotes the Gateaux derivative with respect to the nuisance parameters V, and R_n is the second-order remainder term.

We analyze each term of this expansion:

(i) **Zeroth-Order Term:** The score evaluated at the true parameters has a population mean of zero, as it is the first-order condition for the population M-estimation problem. Thus,

$$\mathbb{E}[\psi(W;\theta_0,V^*)] = 0.$$

(ii) **First-Order Term:** By Proposition 4, our score function is constructed to be Neyman-orthogonal, meaning:

$$\mathcal{D}_V\mathbb{E}[\psi(W;\theta_0,V^*)] = 0.$$

Therefore, the term $\mathcal{D}_V\mathbb{E}[\psi(W;\theta_0,V^*)][\widehat{V} - V^*]$ has a population mean of zero.

(iii) **Second-Order Remainder Term (R_n):** The remainder term is bounded by the squared norm of the nuisance estimation error:

$$R_n = O_p\left(\|\widehat{V}_{\widehat{\gamma}} - V^*\|^2_{\mathcal{L}_2(\mathbb{P}_X)}\right) = o_p(n^{-1/2}).$$

Combining these results, the entire expression for the bias simplifies to:

$$\mathbb{E}[\psi(W;\theta_0,\widehat{V})] = 0 + 0 + o_p(n^{-1/2}) = o_p(n^{-1/2}).$$

Finally, we scale this bias by $\sqrt{n}$ to evaluate Term (B):

$$\sqrt{n}\mathbb{E}[\psi(W;\theta_0,\widehat{V})] = \sqrt{n}\cdot o_p(n^{-1/2}) = o_p(1).$$

This demonstrates that the bias term is asymptotically negligible and does not affect the first-order asymptotic distribution of our estimator.

Step 6: Bounding the Nuisance Stochastic Term (C). We now show that Term (C) from the score decomposition is asymptotically negligible. This term captures the stochastic error arising from using a data-dependent function $\widehat{V}$:

$$\text{Term (C)} = \sqrt{n}(\mathbb{P}_n - \mathbb{E})[\psi(W;\theta_0,\widehat{V}) - \psi(W;\theta_0,V^*)].$$

For notational convenience, define the centered empirical process

$$\mathbb{G}_n f := \sqrt{n}(\mathbb{P}_n - \mathbb{E})[f].$$

Then Term (C) $= \mathbb{G}_n f_{\widehat{\gamma}}$ where

$$f_{\widehat{\gamma}}(W) := \psi(W;\theta_0,\widehat{V}_{\widehat{\gamma}}) - \psi(W;\theta_0,V^*).$$

We proceed in two steps: (a) define a *deterministic* localized function class on a high-probability event, and (b) apply a maximal inequality to control the empirical process indexed by that class.

(a) Defining the Local Function Class on a High-Probability Event. A key subtlety is that the random estimation error $\|\widehat{V}_{\widehat{\gamma}} - V^*\|_{\mathcal{L}_2(\mathbb{P}_X)(P_X)}$ cannot be used directly as the radius of the indexing class if we want to apply standard maximal inequalities with deterministic entropy bounds. We therefore separate the random error from a deterministic localization radius.

Define the *random* estimation error

$$\widehat{r}_{V,n} := \|\widehat{V}_{\widehat{\gamma}} - V^*\|_{\mathcal{L}_2(\mathbb{P}_X)(P_X)}.$$

By Proposition 2, we have $\widehat{r}_{V,n} = o_p(n^{-\delta})$ for some $1/2 > \delta > 1/4$. Fix the *deterministic* localization sequence

$$r_{V,n} := n^{-\delta},$$

and define the associated high-probability event

$$\mathcal{E}_{V,n} := \{\widehat{r}_{V,n} \le r_{V,n}\}.$$

Since $\widehat{r}_{V,n}/r_{V,n} = o_p(1)$, it follows that $\Pr(\mathcal{E}_{V,n}) \to 1$.

Next define the *deterministic* localized class of score deviations

$$\mathcal{F}_n(r_{V,n}) := \Big\{ f_\gamma(\cdot) := \psi(\cdot;\theta_0, V_\gamma) - \psi(\cdot;\theta_0, V^*) : \gamma \in \Gamma_n,\ \|V_\gamma - V^*\|_{\mathcal{L}_2(\mathbb{P}_X)(P_X)} \le r_{V,n} \Big\}.$$

On the event $\mathcal{E}_{V,n}$ we have $\widehat{r}_{V,n} \le r_{V,n}$, and therefore $f_{\widehat{\gamma}} \in \mathcal{F}_n(r_{V,n})$. Hence

$$|\text{Term (C)}| \le \mathbf{1}\{\mathcal{E}_{V,n}\} \sup_{f \in \mathcal{F}_n(r_{V,n})} |\mathbb{G}_n f| + \mathbf{1}\{\mathcal{E}_{V,n}^c\}\, |\mathbb{G}_n f_{\widehat{\gamma}}|.$$

The second term is negligible because for any $\varepsilon > 0$,

$$\Pr\Big(\mathbf{1}\{\mathcal{E}_{V,n}^c\}\, |\mathbb{G}_n f_{\widehat{\gamma}}| > \varepsilon\Big) \le \Pr\Big(\mathcal{E}_{V,n}^c\Big) \to 0.$$

Therefore, it suffices to bound $\sup_{f \in \mathcal{F}_n(r_{V,n})} |\mathbb{G}_n f|$.

(b) Applying a Maximal Inequality. We apply a standard maximal inequality for centered empirical processes indexed by a class with controlled $\mathcal{L}_2(\mathbb{P}_X)(P)$-entropy (see, e.g., Van der Vaart, 2000).

Step 1: Variance radius. By Assumption 6 (smoothness/Lipschitz stability of the policy map and the score construction), the score map is Lipschitz in the value function in $\mathcal{L}_2(\mathbb{P}_X)$, because it is a composition of the map $\gamma \mapsto V_\gamma$, the policy operator Λ_{θ_0}, and the logarithm, all of which are smooth or Lipschitz. Hence, there exists a constant $L_\psi < \infty$ such that for all $V_1, V_2 \in \mathcal{S}_{V,n}$,

$$\big\|\psi(\cdot;\theta_0, V_1) - \psi(\cdot;\theta_0, V_2)\big\|_{\mathcal{L}_2(\mathbb{P}_X)(P)} \le L_\psi\, \|V_1 - V_2\|_{\mathcal{L}_2(\mathbb{P}_X)(P_X)}.$$

In particular, for any $f_\gamma \in \mathcal{F}_n(r_{V,n})$,

$$\|f_\gamma\|_{\mathcal{L}_2(\mathbb{P}_X)(P)} \le L_\psi\, \|V_\gamma - V^*\|_{\mathcal{L}_2(\mathbb{P}_X)(P_X)} \le L_\psi\, r_{V,n}.$$

Define the variance radius

$$\sigma_n^2 := \sup_{f \in \mathcal{F}_n(r_{V,n})} \mathbb{E}[f(W)^2], \qquad \text{so that} \qquad \sigma_n \le L_\psi r_{V,n}.$$

Step 2: Entropy bound for the localized score class. The Lipschitz property also implies that the $\mathcal{L}_2(\mathbb{P}_X)(P)$-covering numbers of $\mathcal{F}_n(r_{V,n})$ are controlled by those of the underlying value-function sieve. Specifically, for every $\varepsilon > 0$,

$$N\big(\varepsilon, \mathcal{F}_n(r_{V,n}), \mathcal{L}_2(\mathbb{P}_X)(P)\big) \leq N\big(\varepsilon/L_\psi, \mathcal{S}_{V,n}, \mathcal{L}_2(\mathbb{P}_X)(P_X)\big).$$

By Assumption 7, there exist constants $A_0, B_0 < \infty$ such that for all $\varepsilon \in (0, B_0]$,

$$\log N\big(\varepsilon, \mathcal{F}_n(r_{V,n}), \mathcal{L}_2(\mathbb{P}_X)(P)\big) \leq A_0 p_n \log(B_0/\varepsilon),$$

where constants (including L_ψ) are absorbed into B_0.

Step 3: Dudley integral bound and stochastic order. A maximal inequality (Dudley-type entropy integral) yields

$$\begin{aligned}
\mathbb{E}\Big[\sup_{f \in \mathcal{F}_n(r_{V,n})} |\mathbb{G}_n f|\Big] &\lesssim \int_0^{\sigma_n} \sqrt{\log N\big(\varepsilon, \mathcal{F}_n(r_{V,n}), \mathcal{L}_2(\mathbb{P}_X)(P)\big)}\, d\varepsilon \\
&\lesssim \int_0^{L_\psi r_{V,n}} \sqrt{p_n \log(B_0/\varepsilon)}\, d\varepsilon \\
&\lesssim r_{V,n}\sqrt{p_n \log(1/r_{V,n})}.
\end{aligned}$$

Since the right-hand side is deterministic, Markov's inequality implies

$$\sup_{f \in \mathcal{F}_n(r_{V,n})} |\mathbb{G}_n f| = O_p\Big(r_{V,n}\sqrt{p_n \log(1/r_{V,n})}\Big). \tag{47}$$

Combining (47) with part (a) yields

$$\text{Term (C)} = O_p\Big(r_{V,n}\sqrt{p_n \log(1/r_{V,n})}\Big) + o_p(1).$$

Finally, under $r_{V,n} = n^{-\delta}$ with $\delta > 1/4$ and the sieve-growth condition $p_n = o\big(n^{2\delta}/\log n\big)$ (so that $r_{V,n}\sqrt{p_n \log n} = o(1)$), we conclude that Term (C) $= o_p(1)$.

Step 7: Asymptotic Normality via Influence Function Representation. Our starting point is the mean-value expansion from Step 3:

$$\sqrt{n}(\widehat{\theta} - \theta_0) = -\Big(\mathbb{P}_n[\nabla_\theta \psi(W; \bar{\theta}, \widehat{V})]\Big)^{-1} \sqrt{n}\mathbb{P}_n[\psi(W; \theta_0, \widehat{V})]$$

In Steps 4, 5 and 6, we analyzed the score term $\sqrt{n}\mathbb{P}_n[\psi(W; \theta_0, \widehat{V})]$ by decomposing it into three parts (A), (B), and (C). We proved that the scaled bias term (B) and the scaled stochastic term (C) are both $o_p(1)$. This leaves only the leading term (A), allowing us to simplify the score:

$$\begin{aligned}
\sqrt{n}\mathbb{P}_n[\psi(W; \theta_0, \widehat{V})] &= \sqrt{n}\mathbb{P}_n[\psi(W; \theta_0, V^*)] + \text{Term (B)} + \text{Term (C)} \\
&= \frac{1}{\sqrt{n}} \sum_{i=1}^n \psi(W_i; \theta_0, V^*) + o_p(1).
\end{aligned}$$

Substituting this back into the expansion and recalling from Step 3 that $\mathbb{P}_n[\nabla_\theta \psi(W; \bar{\theta}, \widehat{V})] \xrightarrow{p} G_\theta$, we obtain:

$$\sqrt{n}(\widehat{\theta} - \theta_0) = -G_\theta^{-1}\left(\frac{1}{\sqrt{n}} \sum_{i=1}^n \psi(W_i; \theta_0, V^*)\right) + o_p(1).$$

The term $\psi(W;\theta_0,V^*)$ is a sum of i.i.d. mean-zero random vectors. By the Lindeberg-Feller Central Limit Theorem:

$$\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\psi(W_i;\theta_0,\gamma^*) \xrightarrow{d} \mathcal{N}(0,\Omega).$$

By Slutsky's Theorem , the product converges in distribution to the product of the limits:

$$\sqrt{n}(\widehat{\theta}-\theta_0) \xrightarrow{d} -G_\theta^{-1}\cdot\mathcal{N}(0,\Omega) = \mathcal{N}\left(0, G_\theta^{-1}\Omega(G_\theta^{-1})'\right).$$

Step 8: Simplification to the Semiparametric Efficiency Bound. The final step is to show that the asymptotic sandwich variance, $G_\theta^{-1}\Omega(G_\theta^{-1})'$, simplifies to the semiparametric efficiency bound, $\mathcal{I}_{\theta\theta}^{-1}$, which is the inverse of the Fisher information matrix. This simplification relies on the Information Matrix Equality, which holds because the score function $\psi(W;\theta_0,V^*)$ is the *efficient score* of the model.

First, we must establish that our score function, when evaluated at the true parameters, is identical to the efficient score for the structural parameter θ. The NNES score function is defined as $\psi(W;\theta,V) = \nabla_\theta \log\left((\Lambda_\theta[V_\gamma])(A|X)\right)$. At the true parameters (θ_0,V^*), this becomes:

$$\psi(W;\theta_0,V^*) = \nabla_\theta \log\left((\Lambda_\theta[V_{\gamma^*}])(A|X)\right)|_{\theta=\theta_0}.$$

Furthermore, the true Conditional Choice Probability function, P^*, is the fixed point of the policy-iteration operator at θ_0, meaning $P^* = \Psi_{\theta_0}[P^*] = \Lambda_{\theta_0}[\varphi_{\theta_0}[P^*]]$. Since $V^* = \varphi_{\theta_0}[P^*]$, we have the crucial identity:

$$P^*(A|X) = (\Lambda_{\theta_0}[V^*])(A|X).$$

The efficient score for the model is therefore identical to the score of a correctly specified parametric likelihood where the true nuisance function is known. The efficient score of our DDC model is thus $\nabla_\theta \log(P^*(A|X;\theta_0))$. For any correctly specified likelihood model, the Information Matrix Equality states that the expected outer product of the efficient score is equal to the negative of the expected Hessian of the log-likelihood. Let $\mathcal{I}_{\theta\theta}$ be the Fisher information matrix.

- The covariance of the efficient score, Ω, is by definition the Fisher information matrix:

$$\Omega := \mathbb{E}[\psi(W;\theta_0,V^*)\psi(W;\theta_0,V^*)'] = \mathcal{I}_{\theta\theta}.$$

- The expected Jacobian of the efficient score, G_θ, is the negative of the Fisher information matrix:

$$G_\theta := \mathbb{E}[\nabla_\theta\psi(W;\theta_0,V^*)] = -\mathcal{I}_{\theta\theta}.$$

Substituting these identities into the sandwich variance formula from Step 7 gives the final result:

$$\begin{aligned} G_\theta^{-1}\Omega(G_\theta^{-1})' &= (-\mathcal{I}_{\theta\theta})^{-1}(\mathcal{I}_{\theta\theta})((-\mathcal{I}_{\theta\theta})^{-1})' \\ &= (-\mathcal{I}_{\theta\theta}^{-1})(\mathcal{I}_{\theta\theta})(-\mathcal{I}_{\theta\theta}^{-1})' \\ &= \mathcal{I}_{\theta\theta}^{-1}\mathcal{I}_{\theta\theta}(\mathcal{I}_{\theta\theta}^{-1}) \quad \text{(since } \mathcal{I}_{\theta\theta} \text{ is symmetric)} \\ &= \mathcal{I}_{\theta\theta}^{-1}. \end{aligned}$$

This completes the proof. We have shown that the NNES estimator $\widehat{\theta}$ is asymptotically normal and that its asymptotic variance is equal to the inverse of the Fisher information matrix, thereby achieving the semiparametric efficiency bound. □

C Additional Simulation Results

C.1 Sensitivity to the Initial CCP Neural Network

To assess how sensitive NNES is to the first–stage CCP estimator, we re–run the estimation on data generated from the *same* DGP as in section 6: the two buses replacement problem with i.i.d. T1EV shocks, exponential mileage increments, $N = 50$ buses, $T = 20$ kept periods ($n = 1000$ observations), and discount $\beta = 0.9$. For the Initial CCP estimators, we consider six feed-forward softmax classifiers for the initial CCPs P^0: widths $w \in \{4, 8, 16\}$ crossed with depths $d \in \{1, 2\}$. These networks are trained by cross-entropy to predict A from X as in Section 4.4 using 200 epochs, and are then passed to NNES as the starting policy. The remainder steps of the algorithm are identical across specifications.

Results. Table 3 reports the error metrics for the *initial* CCPs themselves before any NNES iteration. Table 4 summarizes the NNES estimates obtained when the initial CCPs come from each architecture; All parameters are initialized at 0.

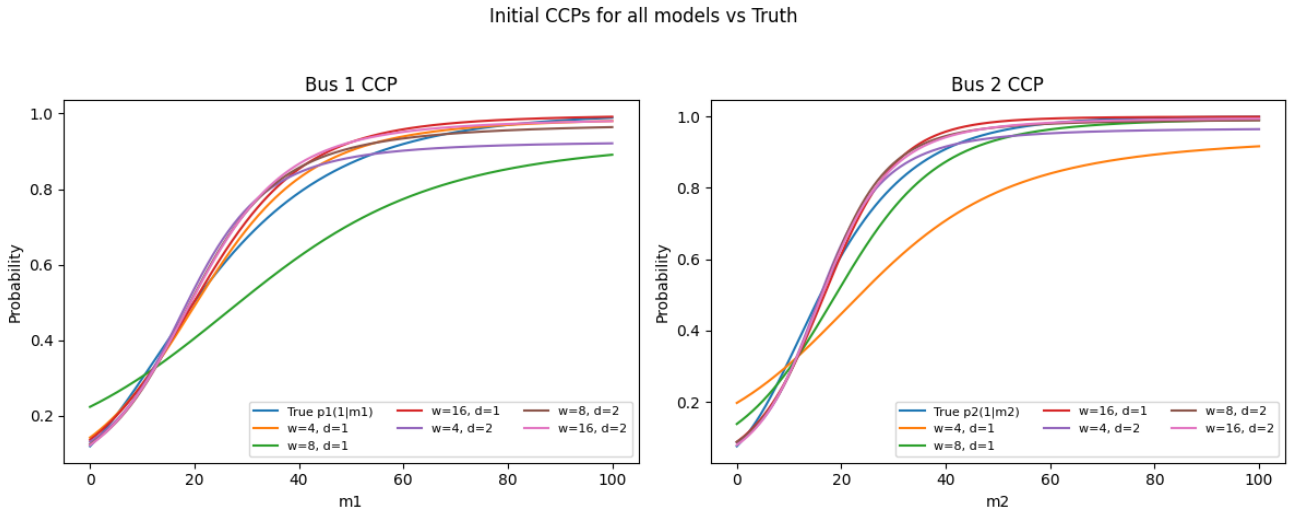


Figure 3: **Initial CCPs (all architectures) vs. truth.** Left: bus 1; right: bus 2. The blue curve is the true CCPs; Other curves are the initial estimated CCPs.

Table 3: Error metrics for the initial CCPs by architecture

width	depth	Bus 1			Bus 2		
		MSE	RMSE	MAE	MSE	RMSE	MAE
16	1	1.164×10^{-3}	0.0341	0.0273	8.258×10^{-4}	0.0287	0.0212
8	1	1.599×10^{-2}	0.1264	0.1195	1.564×10^{-3}	0.0396	0.0314
4	1	4.119×10^{-4}	0.0203	0.0163	2.035×10^{-2}	0.1426	0.1328
16	2	1.612×10^{-3}	0.0401	0.0304	5.919×10^{-4}	0.0243	0.0177
8	2	1.320×10^{-3}	0.0363	0.0289	7.927×10^{-4}	0.0282	0.0210
4	2	2.165×10^{-3}	0.0465	0.0403	8.965×10^{-4}	0.0299	0.0276

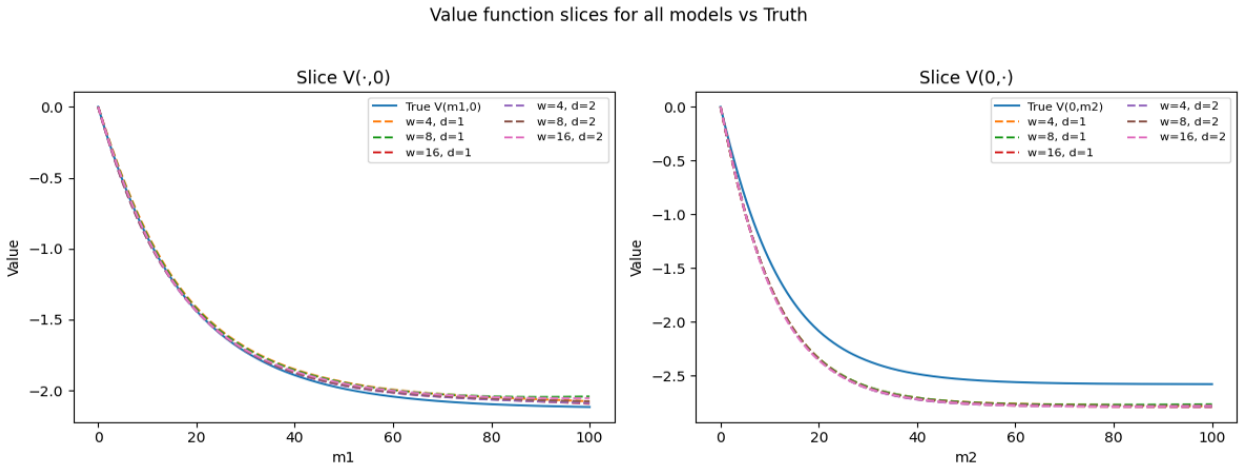


Figure 4: **Value function slices for all specifications vs. truth** (left: $V(m_1, 0)$; right: $V(0, m_2)$). Dashed lines are the NNES approximated value functions; the solid line is the truth.

Table 4: NNES estimates under alternative initial-CCP architectures

Method	$\hat{c}_{\text{rep},1}$	$\hat{c}_{\text{rep},2}$	$\hat{c}_1$	$\hat{c}_2$
True value	2.0000	2.5000	0.0500	0.0800
NNES (k = 10)	1.9419	2.6051	0.0522	0.0880
Ini CCP Net (w=4, d=1)	(0.1755)	(0.1905)	(0.0104)	(0.0141)
NNES (k = 10)	1.9421	2.5825	0.0515	0.0878
Ini CCP Net (w=8, d=1)	(0.1755)	(0.1903)	(0.0103)	(0.0141)
NNES (k = 10)	1.9431	2.5823	0.0516	0.0876
Ini CCP Net (w=16, d=1)	(0.1754)	(0.1902)	(0.0103)	(0.0140)
NNES (k = 10)	1.9433	2.5824	0.0512	0.0877
Ini CCP Net (w=4, d=2)	(0.1757)	(0.1902)	(0.0102)	(0.0140)
NNES (k = 10)	1.9425	2.5830	0.0510	0.0874
Ini CCP Net (w=8, d=2)	(0.1755)	(0.1903)	(0.0102)	(0.0141)
NNES (k = 10)	1.9424	2.5828	0.0511	0.0871
Ini CCP Net (w=16, d=2)	(0.1755)	(0.1903)	(0.0103)	(0.0140)

Discussion and implications. Two patterns stand out.

1. **NNES estimates of θ and their SEs are stable.** Across all six initial CCPs, the structural estimates are numerically indistinguishable at the reported precision, and the reported standard errors are close up to four decimals. This stability is *expected*: the likelihood score is Neyman-orthogonal to first-stage CCP errors, so first-order sensitivity of $\hat{\theta}$ to the initial CCP vanishes; only second-order terms remain. Hence, as long as the initial CCP learns at a rate faster than $n^{-1/4}$, which is equivalent to the $RMSE \approx 0.178$ in this experiment, NNES is $\sqrt{n}$–consistent and semiparametrically efficient.

2. **Approximated Value functions are close in level and shape.** Figure 4 shows that the approximated value functions from all models are close to the true value function, even with different estimated initial CCPs. Since NNES iterates policy improvement, the outer loop quickly projects these initial policies toward the fixed point, further insulating $\hat{\theta}$ from first-stage variation.

C.2 Numerical versus Analytical Derivatives of the Value Function with respect to θ

We validate that the numerical derivative of the value function with respect to θ is a reliable approximation to the analytical derivative when the policy P and transition kernel $F(\cdot)$ are held fixed—exactly the regime relevant for the inner step with P^{k-1} treated as given. Throughout we use the model and operators from Section 2: for any P, the evaluation and improvement operators are $\varphi_\theta[P]$ and $\Lambda_\theta[V]$, and the implied CCPs satisfy $P = \Lambda_\theta[V]$.

Setting. We adopt the canonical bus-engine replacement model with state $x \in \{0, \dots, M-1\}$ (mileage since last replacement) and actions $a \in \{0, 1\}$, where $a = 0$ denotes "replace" and $a = 1$ denotes "keep." Per-period utilities are

$$u_0(x; \theta) = -\theta_R, \qquad u_1(x; \theta) = -\theta_S x,$$

with discount factor $\beta = 0.9$, $\theta_R = 2, \theta_S = 1$ and Type-I extreme value shocks. For fixed P, the Bellman representation is

$$V = \sum_{a \in \{0,1\}} P(a) \odot \{u_a + e_a(P)\} + \beta\, F_U(P) V, \qquad F_U(P) := \sum_a P(a) \odot F(a),$$

so the *analytical* derivative at fixed P is

$$\frac{\partial V}{\partial \theta_j} = \big(I - \beta F_U(P)\big)^{-1} \sum_a P(a) \odot \frac{\partial u_a}{\partial \theta_j}\,. \tag{48}$$

With the above utility, this yields $\big(I - \beta F_U(P)\big)\, \partial_{\theta_R} V = -P(0)$ and $\big(I - \beta F_U(P)\big)\, \partial_{\theta_S} V = -P(1) \odot x$.

Numerical derivative protocol (finite differences at fixed P). Let V_γ be a feed-forward network approximating V. We (i) fit a *baseline* value network at (θ, P) by minimizing the mean-squared Bellman residual of $\varphi_\theta[P]$ in equation 10; (ii) for each coordinate θ_j, warm-start two refits at $\theta_\pm = \theta \pm h e_j$ with the same P and training settings until the residual RMSE matches the baseline tolerance; (iii) form the symmetric central difference

$$\partial_{\theta_j} V^{\mathrm{FD}}(x) = \frac{V_{\gamma^+}(x) - V_{\gamma^-}(x)}{2h},$$

and (iv) choose h on a descending grid using a blind stability rule (both branches must meet the residual tolerance and successive finite-difference estimates must stabilize).

Results: `tanh` networks. Using two hidden layers with 32 nodes and `tanh` activation, the baseline value fit achieves a Bellman-residual MSE of 5.885×10^{-8}, and the network reproduces the true V almost exactly. Relative to the analytical derivative (48) at the same fixed P, the finite-difference errors are

$$\mathrm{MSE}\Big(\partial_{\theta_R} V^{\mathrm{FD}} - \partial_{\theta_R} V^{\mathrm{ana}}\Big) = 2.143 \times 10^{-8}, \quad \mathrm{MSE}\Big(\partial_{\theta_S} V^{\mathrm{FD}} - \partial_{\theta_S} V^{\mathrm{ana}}\Big) = 2.962 \times 10^{-6},$$

corresponding to RMS errors 1.46×10^{-4} and 1.72×10^{-3}, respectively. Here, V^{ana} is the analytical value function. The numerical and analytical curves are visually indistinguishable across the state grid, including the high-curvature region near $x = 0$; see Figure 5.

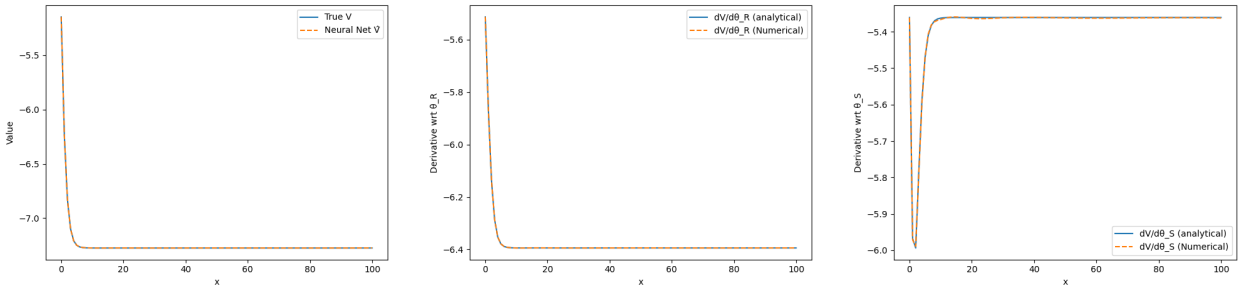


Figure 5: **tanh activation.** (a) True vs. approximated V. (b) $\partial_{\theta_R} V$: analytical (solid) vs. numerical (dashed). (c) $\partial_{\theta_S} V$: analytical (solid) vs. numerical (dashed).

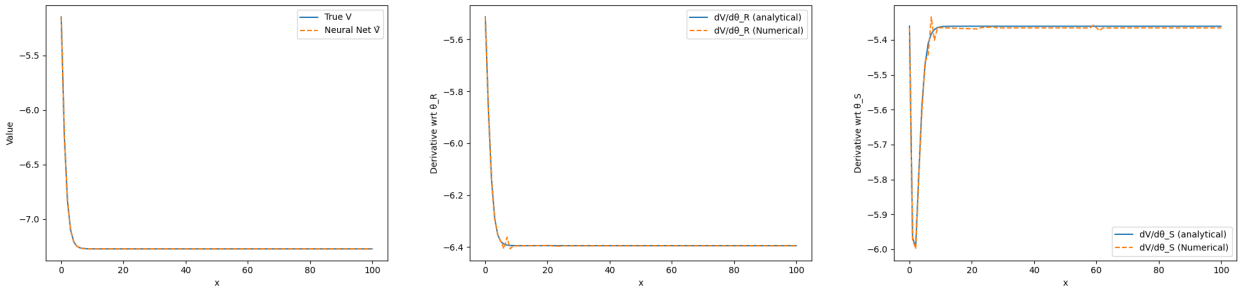


Figure 6: **ReLU activation.** (a) True vs. approximated V. (b) $\partial_{\theta_R} V$: analytical (solid) vs. numerical (dashed). (c) $\partial_{\theta_S} V$: analytical (solid) vs. numerical (dashed).

Results: `ReLU` networks. With the same architecture but `ReLU` activation, the residual fit is likewise tight (3.829×10^{-8}). The derivative errors are

$$\mathrm{MSE}\Big(\partial_{\theta_R} V^{\mathrm{FD}} - \partial_{\theta_R} V^{\mathrm{ana}}\Big) = 1.419 \times 10^{-5}, \quad \mathrm{MSE}\Big(\partial_{\theta_S} V^{\mathrm{FD}} - \partial_{\theta_S} V^{\mathrm{ana}}\Big) = 6.956 \times 10^{-5},$$

i.e., RMS 3.77×10^{-3} and 8.34×10^{-3}. These are below 0.2% of the derivative magnitudes over the grid. Plots again overlay almost perfectly; small ripples for very small x reflect the piecewise-linear `ReLU` shape and finite-difference truncation (Figure 6).

Takeaway. Holding P fixed, the numerical value-derivative computed by the finite-difference protocol matches the analytical target (48) to numerical precision for both `tanh` and `ReLU`. This validates using numerical $\partial_\theta V$ to evaluate cross-terms in the profiled likelihood gradient within our algorithm.

C.3 Simulation results with different initial guess of $\hat{\theta}$

In this section, we estimate the structural parameters in the 2-Dimensional bus engine replacement model as in section 6, but with different initial guess of $\hat{\theta}$. As part of the comparison, we also implemented the joint sieve-based efficient estimators (SEES) as in Luo and Sang [2025] to estimate the structural parameters θ. We approximate the Value Function using anchored Neural Networks with the same specifications in 6.4. Throughout this section, $\hat{\theta}$ are initialized at $(1.5, 2.0, 0.02, 0.04)$ for all estimators.

Table 5 averages 100 replications; CPU times are per replication; Discount factor $\beta = 0.9$; NNES and SEES attains the same precision as an oracle estimator that knows the separable structure

Table 5: Monte-Carlo estimates (100 replications, $n = 1000$)

Method	$\hat{c}_{\text{rep},1}$	$\hat{c}_{\text{rep},2}$	$\hat{c}_1$	$\hat{c}_2$	Avg. sec
True value	2.000	2.500	0.0500	0.0800	—
NNES (k = 2)	1.993	2.496	0.0508	0.0835	243
	(0.1693)	(0.1707)	(0.0113)	(0.0121)	
Oracle NFXP	1.999	2.502	0.0513	0.0846	22
	(0.1692)	(0.1708)	(0.0113)	(0.0123)	
SEES	1.995	2.493	0.0510	0.0830	569
	(0.1692)	(0.1710)	(0.0115)	(0.0122)	

ex ante, while solving a 33-parameter optimisation and avoiding any dynamic-programming grid search. NNES's CPU cost is about ten times higher than the oracle because the network must learn separability from data—but remains negligible for empirical sample sizes. SEES's CPU cost is high due to the requirement of inverting the full Hessian matrix.

Table 6 averages 100 replications; CPU times are per replication; Discount factor $\beta = 0.9999$. The Value function is not anchored.

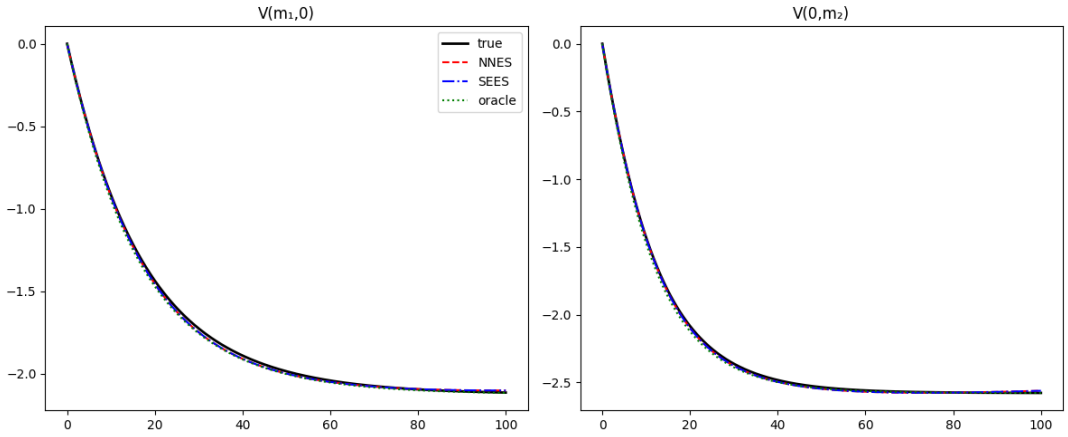
Table 6: Monte-Carlo estimates (100 replications, $n = 1000$)

Method	$\hat{c}_{\text{rep},1}$	$\hat{c}_{\text{rep},2}$	$\hat{c}_1$	$\hat{c}_2$	Avg. sec
True value	2.000	2.500	0.0500	0.0800	—
NNES (k = 2)	2.2161	2.6166	0.0456	0.0726	533
	(0.1834)	(0.1936)	(0.0189)	(0.0234)	
Oracle NFXP	1.9993	2.5007	0.0512	0.0793	80
	(0.1681)	(0.1704)	(0.0117)	(0.0122)	
SEES	2.2358	2.7235	0.0466	0.0736	733
	(0.1837)	(0.1926)	(0.0192)	(0.0254)	

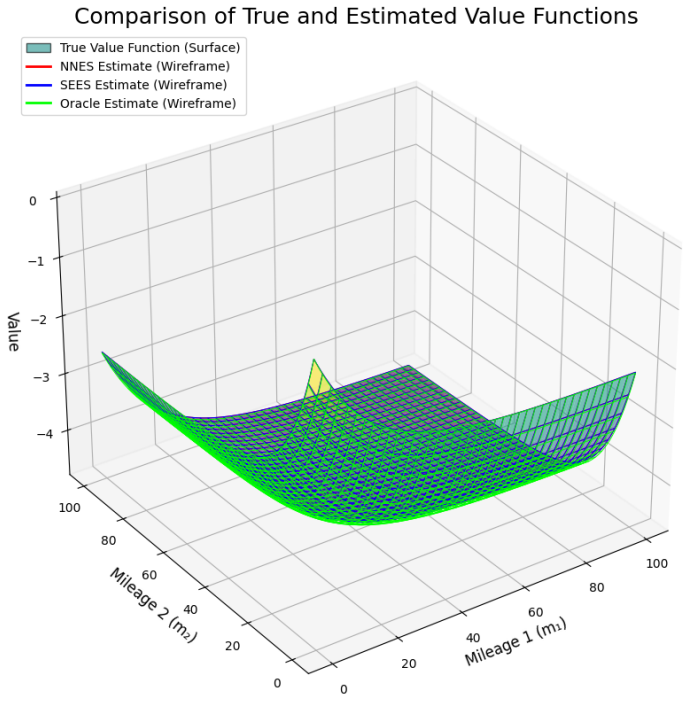
Table 7 averages 100 replications; CPU times are per replication; Discount factor $\beta = 0.9999$; The Value function is anchored.

Table 7: Monte-Carlo estimates (100 replications, $n = 1000$)

Method	$\hat{c}_{\text{rep},1}$	$\hat{c}_{\text{rep},2}$	$\hat{c}_1$	$\hat{c}_2$	Avg. sec
True value	2.000	2.500	0.0500	0.0800	—
NNES (k=2)	1.9899	2.4996	0.0489	0.0808	533
	(0.1683)	(0.1681)	(0.0116)	(0.0122)	
Oracle NFXP	1.9993	2.5007	0.0512	0.0793	80
	(0.1681)	(0.1682)	(0.0117)	(0.0122)	
SEES	1.9991	2.4994	0.0487	0.0812	733
	(0.1682)	(0.1684)	(0.0116)	(0.0122)	



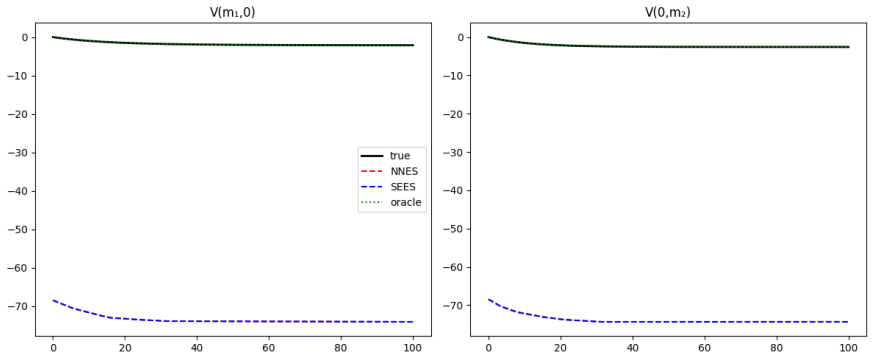
(a) Cross-sectional slices of the value function at $m_2 = 0$ (left) and $m_1 = 0$ (right).



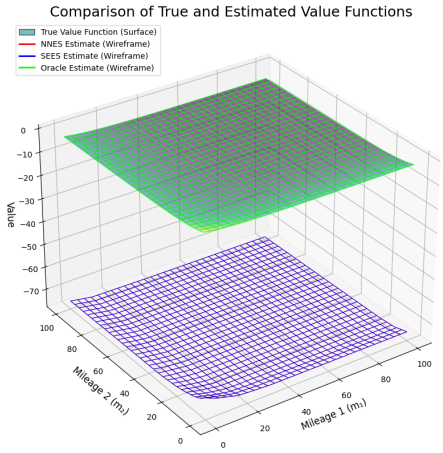
(b) 3-D value function surface and wireframe estimates.

Figure 7: Comparison of true and estimated value functions. The NNES and SEES-estimated functions almost perfectly recover the true function, matching the performance of the Oracle estimator. NNES and SEES achieve this without prior knowledge of the model's additive structure.

The Monte-Carlo study highlights the importance of anchoring for the numerical stability of the estimator when the discount factor is high. Table 6 and figure 8 present the results for an unanchored NNES estimator with $\beta = 0.9999$. The numerical instability, theoretically outlined in Section D, is immediately apparent. Figure 8 shows that the estimated value function experienced a level shift, drifting to large negative values and failing to approximate the true function. This



(a) Cross-sectional slices of the value function at $m_2 = 0$ (left) and $m_1 = 0$ (right).



(b) 3-D value function surface and wireframe estimates.

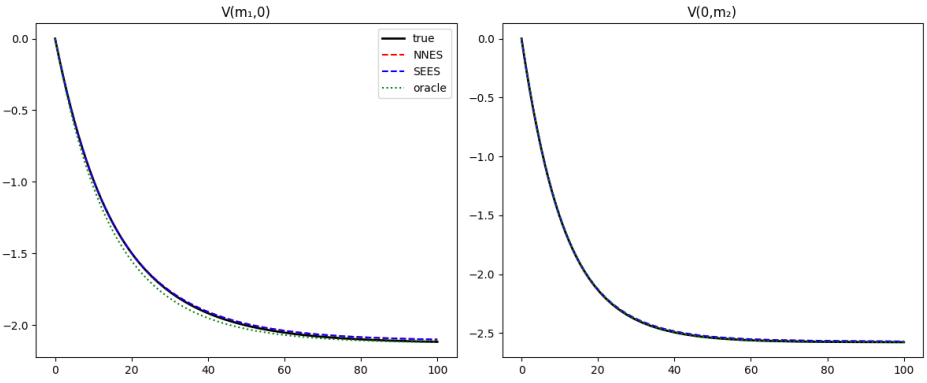
Figure 8: Comparison of true and estimated value functions.

failure to correctly estimate the nuisance component directly contaminates the structural parameter estimates, which, as shown in Table 6, are significantly biased compared to their true values.

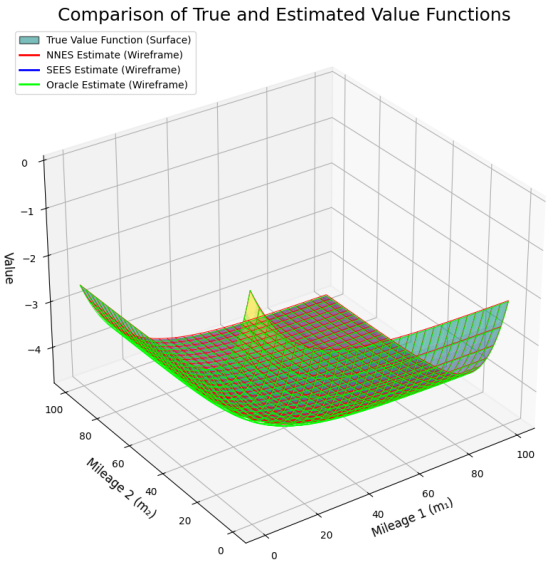
In contrast, the results for the anchored NNES estimator, presented in Table 7 and Figure 9, demonstrate the efficacy of the solution. By enforcing the normalization $V_\gamma(x_0) = 0$, the estimator avoids numerical drift and almost perfectly recovers the true value function in both level and shape, even with the high discount factor. Consequently, the NNES estimator attains the same precision as an oracle estimator in Table 7. This comparison provides an evidence that anchoring is an essential procedure for obtaining reliable estimates in models with patient, forward-looking agents.

D Value Function Anchoring for Numerical Stability

The Identification Problem of the Value Function Level. A key property of dynamic discrete choice models with Type-I extreme value shocks is that the agent's behavior is driven by differences in value, not the absolute level of the value function. The conditional choice probability



(a) Cross-sectional slices of the value function at $m_2 = 0$ (left) and $m_1 = 0$ (right).



(b) 3-D value function surface and wireframe estimates.

Figure 9: Comparison of true and estimated value functions. The anchored NNES and SEES estimated functions almost perfectly recover the true function when $\beta = 0.9999$.

(CCP) for an action a is:

$$P(a \mid x) = \frac{\exp\big(u_a(x;\theta) + \beta \mathbb{E}[V(X') \mid x, a]\big)}{\sum_{j \in \mathcal{A}} \exp\big(u_j(x;\theta) + \beta \mathbb{E}[V(X') \mid x, j]\big)}. \tag{49}$$

If we replace the value function $V(x)$ with an alternative, $V'(x) = V(x) + C$ for any constant C, the choice-specific value functions are all shifted by βC. This constant term $\exp(\beta C)$ factors out of the numerator and denominator of the CCP expression and cancels, leaving the choice probabilities unchanged. Consequently, the likelihood function is invariant to the level of the value function, meaning the level itself is not identified from choice data.

Numerical Instability of an Unanchored Estimator. To understand the necessity of anchoring, we first consider a hypothetical estimator that uses the un-anchored neural network output $G(x;\gamma)$ directly. The corresponding Bellman penalty would be

$$\tilde{\rho}_n(\gamma,\theta,P) = \left\|G(\cdot;\gamma) - \varphi_\theta[P_{G(\cdot;\gamma)}]\right\|^2_{L^2(\widehat{\pi})}, \tag{50}$$

where $P_{G(\cdot;\gamma)} = \Lambda_\theta[G(\cdot;\gamma)]$. While the likelihood is invariant to the level of $G(x;\gamma)$, this penalty term is not. Consider a perturbation to the network that only shifts its level, represented by $G(x;\gamma) + C$. As established, the CCPs are invariant, so $P_{G+C} = P_G$. However, the evaluation operator φ_θ is not level-invariant. A level shift in the continuation value function by C induces a shift of βC in the expected value function: $\varphi_\theta[P_{G+C}] = \varphi_\theta[P_G] + \beta C$. The Bellman residual for $G + C$ is thus:

$$(G(\cdot;\gamma) + C) - \varphi_\theta[P_{G+C}] = \big(G(\cdot;\gamma) - \varphi_\theta[P_G]\big) + (1-\beta)C. \tag{51}$$

The penalty term becomes $\left\|G(\cdot;\gamma) - \varphi_\theta[P_G] + (1-\beta)C\right\|^2_{L^2(\widehat{\pi})}$. As $\beta \to 1$, the term $(1-\beta)C \to 0$. This means that the penalty function becomes nearly flat with respect to changes in the level C of the value function. The gradient of $\tilde{\rho}_n$ with respect to any network parameter changes that simply shift the level of $G(x;\gamma)$ becomes vanishingly small. This ill-conditioning makes the inner-loop optimization in (12) extremely slow and numerically unstable, as the optimizer struggles to find a unique minimum.

Anchoring as a Solution. The NNES estimator defined in this paper avoids this problem by construction. By employing the anchored value function $V_\gamma(x) = G(x;\gamma) - G(x_0;\gamma)$, we enforce the normalization $V_\gamma(x_0) = 0$. This removes the redundant degree of freedom corresponding to the function's level, making the optimization landscape well-conditioned. To see this formally, we analyze how the penalty term responds to a perturbation of the network weights that would, for an unanchored function, correspond to a pure level shift.

Let $\gamma \in \Gamma_n$ be a set of network weights. Consider an alternative set of weights $\gamma' \in \Gamma_n$ that only shifts the level of the unanchored output, such that $G(x;\gamma') = G(x;\gamma) + C$ for some constant C and for all $x \in \mathcal{X}$. We now analyze each component of the penalty term $\rho_n(\gamma',\theta,P) = \left\|V_{\gamma'} - \varphi_\theta[P_{\gamma'}]\right\|^2_{L^2(\widehat{\pi})}$ under this perturbation.

First, consider the anchored value function itself, $V_{\gamma'}(x)$. By definition:

$$\begin{aligned} V_{\gamma'}(x) &= G(x;\gamma') - G(x_0;\gamma') && (52)\\ &= \big(G(x;\gamma) + C\big) - \big(G(x_0;\gamma) + C\big) && (53)\\ &= G(x;\gamma) - G(x_0;\gamma) = V_\gamma(x). && (54) \end{aligned}$$

The anchored value function V_γ is therefore invariant to level-shifting perturbations of the underlying unanchored network G. Second, consider the policy evaluation term, $\varphi_\theta[P_{\gamma'}]$. The policy $P_{\gamma'}$ is generated from the anchored value function $V_{\gamma'}$:

$$P_{\gamma'} = \Lambda_\theta[V_{\gamma'}].$$

Since we have just shown that $V_{\gamma'} = V_\gamma$, it follows immediately that the generated policies are identical, $P_{\gamma'} = P_\gamma$. Consequently, the output of the evaluation operator is also unchanged:

$$\varphi_\theta[P_{\gamma'}] = \varphi_\theta[P_\gamma].$$

Both terms in the Bellman residual, V_γ and $\varphi_\theta[P_\gamma]$, are invariant to the perturbation from γ to γ'. Therefore, the entire penalty term is invariant:

$$\rho_n(\gamma', \theta, P) = \left\|V_{\gamma'} - \varphi_\theta[P_{\gamma'}]\right\|^2_{L^2(\widehat{\pi})} = \left\|V_\gamma - \varphi_\theta[P_\gamma]\right\|^2_{L^2(\widehat{\pi})} = \rho_n(\gamma, \theta, P). \tag{55}$$

With anchoring, the penalty term is only zero if the shape of the value function is correct, regardless of the level shift C. This provides the optimizer with a well-posed objective, forcing it to find a solution that is not just parallel to the true value function, but is correctly aligned with it relative to the anchor point x_0.

Invariance of the Asymptotic Distribution of $\widehat{\theta}$. We now demonstrate that this anchoring normalization does not alter the asymptotic properties of the structural parameter estimator $\widehat{\theta}$. The proof rests on showing that the score function for θ is invariant to the anchoring.

Let $G(x;\gamma)$ denote the un-anchored neural network output and $V_\gamma(x) = G(x;\gamma) - G(x_0;\gamma)$ be the anchored value function used in the estimator. The conditional choice probabilities (CCPs) are generated by applying the soft-max operator, Λ_θ, to the value function. We first show that the CCPs are invariant to this anchoring. The choice-specific value function for an action a using the anchored value function V_γ is

$$\begin{aligned} Q_a(x;\theta,V_\gamma) &= u_a(x;\theta) + \beta\mathbb{E}[V_\gamma(X') \mid x,a] && (56)\\ &= u_a(x;\theta) + \beta\mathbb{E}[G(X';\gamma) - G(x_0;\gamma) \mid x,a] && (57)\\ &= (u_a(x;\theta) + \beta\mathbb{E}[G(X';\gamma) \mid x,a]) - \beta G(x_0;\gamma) && (58)\\ &= Q_a(x;\theta,G) - \beta G(x_0;\gamma). && (59) \end{aligned}$$

The term $\beta G(x_0;\gamma)$ is a constant with respect to the action a. Applying the soft-max operator Λ_θ yields

$$\begin{aligned} P_\gamma(a \mid x;\theta) = \Lambda_\theta[V_\gamma] &= \frac{\exp(Q_a(x;\theta,V_\gamma))}{\sum_{j\in\mathcal{A}}\exp(Q_j(x;\theta,V_\gamma))} && (60)\\ &= \frac{\exp(Q_a(x;\theta,G) - \beta G(x_0;\gamma))}{\sum_{j\in\mathcal{A}}\exp(Q_j(x;\theta,G) - \beta G(x_0;\gamma))} && (61)\\ &= \frac{\exp(Q_a(x;\theta,G))\cdot\exp(-\beta G(x_0;\gamma))}{\left(\sum_{j\in\mathcal{A}}\exp(Q_j(x;\theta,G))\right)\cdot\exp(-\beta G(x_0;\gamma))} = \Lambda_\theta[G]. && (62) \end{aligned}$$

This confirms that the CCPs, and thus the likelihood and score functions, are mathematically identical regardless of whether they are generated from the un-anchored network output $G(x;\gamma)$ or its anchored counterpart $V_\gamma(x)$. Consequently, the asymptotic distribution of $\widehat{\theta}$ is unchanged. The anchoring is a normalization of the nuisance component γ that is essential for numerical stability but has no effect on the statistical properties of the structural parameters of interest θ.

Recovering the Level of the Value Function from an Anchored Shape Recall the anchored representation used in the estimator,

$$V_\gamma(x) \;:=\; G(x;\gamma) - G(x_0;\gamma) \quad \text{so that} \quad V_\gamma(x_0) = 0,$$

and the induced policy $P_\gamma := \Lambda_\theta[V_\gamma]$. Define the choice–specific value under the anchored shape,

$$Q_a(x;\theta,V_\gamma) \;:=\; u_a(x;\theta) + \beta\,\mathbb{E}\big[\,V_\gamma(X') \,\big|\, x,a\big]\,.$$

By substituting $P_\gamma = \Lambda_\theta[V_\gamma]$ into the definition of the evaluation operator and using the entropy identity for the soft–max,

$$(\varphi_\theta[P_\gamma])(x) \;=\; \log \sum_{a\in\mathcal{A}} \exp\Big\{\, Q_a(x;\theta,V_\gamma)\Big\}.$$

At the anchor state x_0, the unanchored value satisfies the log–sum–exp Bellman equation with the decomposition $G(\cdot;\gamma) = V_\gamma(\cdot) + G(x_0;\gamma)$:

$$G(x_0;\gamma) = \log \sum_{a\in\mathcal{A}} \exp\Big\{\, u_a(x_0;\theta) + \beta\, \mathbb{E}\big[V_\gamma(X') + G(x_0;\gamma) \mid x_0, a\big] \,\Big\} = \beta\, G(x_0;\gamma) + (\varphi_\theta[P_\gamma])(x_0).$$

Hence the level is pinned down by a single evaluation at the anchor:

$$(1-\beta)\, G(x_0;\gamma) \;=\; (\varphi_\theta[P_\gamma])(x_0) \qquad\Longrightarrow\qquad G(x_0;\gamma) \;=\; \frac{(\varphi_\theta[P_\gamma])(x_0)}{1-\beta}.$$

Finally, recover the unanchored (and thus true, at the optimum) value function by adding back the level:

$$G(x;\gamma) \;=\; V_\gamma(x) \;+\; G(x_0;\gamma).$$

Correctness. If $V_\gamma(x) = V^*(x) - V^*(x_0)$, then the identity above gives $(\varphi_\theta[P_\gamma])(x) = \log\sum_a \exp\{u_a(x;\theta) + \beta\, \mathbb{E}[V_\gamma(X') \mid x, a]\} = V^*(x) - \beta V^*(x_0)$, so at x_0 we have $(1-\beta)V^*(x_0) = (\varphi_\theta[P_\gamma])(x_0)$. Therefore the recovered level equals the truth, $G(x_0;\gamma) = V^*(x_0)$, and $G(x;\gamma) = V_\gamma(x) + G(x_0;\gamma) = V^*(x)$ for all x.

D.1 Regularizing the Value Network

Deep networks are typically over-parameterized, and without constraints they can interpolate noise and produce unstable solutions. *Explicit* regularization (e.g., weight decay, sparsity, spectral/Jacobian penalties) contracts the effective hypothesis space and improves generalization Bühlmann and Van De Geer [2011], Bartlett et al. [2017], Neyshabur et al. [2017]. *Implicit* regularization (early stopping, SGD noise, dropout) biases the optimization trajectory toward simpler solutions and can be interpreted as data-dependent Tikhonov-type shrinkage Hardt et al. [2016], Wager et al. [2013], Soudry et al. [2018], Gunasekar et al. [2018]. In our setting, regularization is useful to stabilize the policy-evaluation network V_γ and to preserve the $o_p(n^{-1/4})$ nuisance rate needed for root-n inference on θ (cf. Proposition 2 and Theorem 4.3). This section aim to show how to add a statistical regularizer to the *inner* NNES step and why it does not alter first–order inference for θ.

Inner problems and notation. Fix (θ, P). Write

$$F_n(\gamma;\theta,P) \;:=\; -\,\ell_n(\theta,\gamma) \;+\; \omega_n\, \rho_n(\gamma,\theta,P),$$

where ℓ_n and ρ_n are as in Sections 3.1, and ω_n follows Assumption 9. The *unregularized* inner solution is

$$\widehat{\gamma}(\theta;P) \;\in\; \arg\min_{\gamma\in\Gamma_n}\; F_n(\gamma;\theta,P).$$

Given a Fréchet–differentiable penalty $\mathcal{R}_n : \Gamma_n \to \mathbb{R}_+$, we incorporate a statistical regularizer by augmenting the inner problem 11 as

$$\widetilde{\gamma}(\theta;P,\lambda_n) \;\in\; \arg\min_{\gamma\in\Gamma_n}\; \Big\{F_n(\gamma;\theta,P) \;+\; \lambda_n\, \mathcal{R}_n(\gamma)\Big\},$$

with tuning $\lambda_n \geq 0$. The corresponding value functions are $V_{\widehat{\gamma}}$ and $V_{\widetilde{\gamma}}$, and the profiled objective is $L_n(\theta) = \ell_n(\theta, \widetilde{\gamma}(\theta;P,\lambda_n))$.

Assumption 12 (Penalty sequence)**.** $\lambda_n \to 0$ *and* $\omega_n \to \infty$ *as* $n \to \infty$*, with the tuning requirement*

$$\frac{\lambda_n}{\omega_n} \;=\; o\Big(n^{-1/2}\Big)\,.$$

A concrete choice is $\omega_n = n^{\delta}$ *for some* $1/4 < \delta < 1/2$ *in Assumption 9 and* $\lambda_n = \omega_n\, n^{-1/2-\varepsilon}$ *for any* $\varepsilon > 0$*.*

Assumption 13 (Local curvature and smoothness)**.** *There exist constants* c_0, B_R, L_V *and a (random) neighborhood* $\mathcal{N} \subset \Gamma_n$ *containing* $\widehat{\gamma}$ *and* $\widetilde{\gamma}$ *w.p.a.1 such that:*

(i) ***Quadratic growth in value space:*** *for all* $\gamma \in \mathcal{N}$*,*

$$F_n(\gamma;\theta,P) - F_n(\widehat{\gamma};\theta,P) \;\geq\; c_0\,\omega_n\,\|V_\gamma - V_{\widehat{\gamma}}\|^2_{L^2(\pi_{M_n})}.$$

(ii) ***Regularizer regularity:*** $\mathcal{R}_n$ *is* C^1 *on* $\mathcal{N}$*, and* $\sup_{\gamma\in\mathcal{N}} \|\nabla_\gamma \mathcal{R}_n(\gamma)\| \leq B_R$*.*

(i) *Quadratic growth in value space.* The inner objective is $F_n(\gamma;\theta,P) = -\ell_n(\theta,\gamma) + \omega_n\,\rho_n(\gamma,\theta,P)$, where ρ_n is the *squared* Bellman residual measured in the grid $L^2(\pi_{M_n})$ norm. Holding P fixed, ρ_n is locally quadratic in the value V_γ, and the anchoring $V_\gamma(x_0) = 0$ removes the otherwise flat "level" direction of the value function. By contrast, the likelihood term is locally flat in the value argument near a policy fixed point because of the zero-Jacobian/orthogonality property (Proposition 4), so it does not offset this curvature. Consequently, in a small neighborhood of the inner minimizer $\hat{\gamma}$, the penalty dominates the local geometry and the objective grows at least quadratically with the $L^2(\pi_{M_n})$ distance between values:

$$F_n(\gamma;\theta,P) - F_n(\hat{\gamma};\theta,P) \;\geq\; c_0\,\omega_n\,\|V_\gamma - V_{\hat{\gamma}}\|^2_{L^2(\pi_{M_n})},$$

for some $c_0 > 0$.

(ii) *Regularizer regularity.* Standard choices (e.g., weight decay, spectral or Jacobian/Sobolev penalties) are C^1. On any compact neighborhood selected by early stopping, norm constraints, or projection (the set $\mathcal{N}$), the gradient is uniformly bounded, giving $\sup_{\gamma\in\mathcal{N}} \|\nabla_\gamma \mathcal{R}_n(\gamma)\| \leq B_R$.

Proposition 6 (Size of regularization bias)**.** *Let the unregularized and regularized inner solutions be*

$$\widehat{\gamma} \in \arg\min_{\gamma\in\Gamma_n} \Big\{ -\ell_n(\theta,\gamma) + \omega_n\,\rho_n(\gamma,\theta,P) \Big\}, \qquad \widetilde{\gamma} \in \arg\min_{\gamma\in\Gamma_n} \Big\{ -\ell_n(\theta,\gamma) + \omega_n\,\rho_n(\gamma,\theta,P) + \lambda_n R_n(\gamma) \Big\},$$

and write $\Delta\gamma := \widetilde{\gamma} - \widehat{\gamma}$ *and* $\Delta V := V_{\widetilde{\gamma}} - V_{\widehat{\gamma}}$*. Under Assumptions 12–13, there exists a neighborhood* $\mathcal{N} \subset \Gamma_n$ *that contains* $\widehat{\gamma}$ *and* $\widetilde{\gamma}$ *w.p.a.1 and has radius* $R_{\mathcal{N}} := \sup\{\|\gamma' - \gamma\| : \gamma, \gamma' \in \mathcal{N}\} = O(1)$ *such that*

$$\|\Delta V\|_{L^2(\pi_{M_n})} \;\leq\; \Big(\tfrac{B_R\, R_{\mathcal{N}}}{c_0}\Big)^{1/2} \Big(\tfrac{\lambda_n}{\omega_n}\Big)^{1/2}.$$

In particular, under Assumption 12, the bound in Proposition 6 implies $\|\Delta V\|_{L^2(\pi_{M_n})} = o_p(n^{-1/4})$*.*

Proof. The Proof is provided in Appendix E. □

Taking $R_N = O(1)$ is an operational choice rather than an assumption: in the inner problem one can enforce a bounded neighborhood $\mathcal{N} \subset \Gamma_n$ by (i) hard constraints (projection onto an ℓ_2 ball, weight clipping, spectral normalization), (ii) explicit regularization (weight decay, Jacobian/Sobolev or

spectral penalties), and (iii) implicit regularization (early stopping, SGD noise, dropout). These practices are grounded in the statistical learning literature (Bühlmann and Van De Geer [2011], Bartlett et al. [2017], Neyshabur et al. [2017], Hardt et al. [2016], Gunasekar et al. [2018]). In our setting, Assumption 13(ii) explicitly contemplates selecting a compact $\mathcal{N}$ by early stopping, norm constraints, or projection.

Corollary 1 (Nuisance rate is preserved)**.** *If the unregularized inner step delivers* $\|V_{\widehat{\gamma}} - V^*\|_{L^2(\pi_{M_n})} = o_p(n^{-1/4})$*, then by triangle inequality*

$$\|V_{\widetilde{\gamma}} - V^*\|_{L^2(\pi_{M_n})} \;\le\; \|V_{\widehat{\gamma}} - V^*\|_{L^2(\pi_{M_n})} \;+\; \|V_{\widetilde{\gamma}} - V_{\widehat{\gamma}}\|_{L^2(\pi_{M_n})} \;=\; o_p(n^{-1/4}).$$

Hence $\|V_{\widetilde{\gamma}} - V^*\|_{L^2(\pi_{M_n})} = o_p(n^{-1/4})$ *whenever* $\lambda_n/\omega_n = o(n^{-1/2})$*. Under Assumption 12, this is satisfied.*

The value-function error with regularization decomposes into the original estimation error plus the extra drift introduced by the penalty: $\|V_{\widetilde{\gamma}} - V^*\|_{L^2(\pi_{M_n})} \le \|V_{\widehat{\gamma}} - V^*\|_{L^2(\pi_{M_n})} + \|V_{\widetilde{\gamma}} - V_{\widehat{\gamma}}\|_{L^2(\pi_{M_n})}$. By the proposition 6, the second term is $o_p(n^{-1/4})$, and so the nuisance rate is preserved. Let $\psi(W;\theta,V) := \partial_\theta \log\big(\Lambda_\theta[V](A \mid X)\big)$ denote the likelihood score. As shown in Proposition 4, the score is Neyman–orthogonal at (θ_0, V^*), and Corollary 1 shows that $\|V_{\widetilde{\gamma}} - V^*\|_{L^2(\pi_{M_n})} = o_p(n^{-1/4})$.

Corollary 2 (NNES with value-net regularization)**.** *Suppose Assumptions 1–10 hold. Let* $\mathcal{R}_n$ *and* λ_n *satisfy Assumptions 12 and 13, respectively, and define the regularized inner estimator*

$$\widetilde{\gamma}(\theta; P, \lambda_n) \;\in\; \arg\min_{\gamma\in\Gamma_n} \Big\{ -\ell_n(\theta,\gamma) \;+\; \omega_n\, \rho_n(\gamma,\theta,P) \;+\; \lambda_n\, \mathcal{R}_n(\gamma) \Big\},$$

with profiled objective $L_n(\theta) := \ell_n(\theta, \widetilde{\gamma}(\theta; P, \lambda_n))$*. Let* $\widehat{\theta}$ *maximize* $L_n(\theta)$ *after a fixed number of outer iterations* $k \ge 1$*. Then*

$$\sqrt{n}\,(\widehat{\theta} - \theta_0) \;\xrightarrow{d}\; \mathcal{N}(0,\, I_{\theta\theta}^{-1}).$$

Proof. The proof is provided in Appendix F. □

The inner problem augments the usual policy–evaluation objective $F_0(\gamma;\theta,P)$ with a regularizer $\lambda_n\mathcal{R}_n(\gamma)$. Because the policy–iteration step drives the Bellman residual to zero and $\omega_n \to \infty$, the gradient $\partial_\gamma\rho_n$ vanishes, so the regularizer can only perturb the inner solution by an amount of order $(\lambda_n/\omega_n)^{1/2}$. Consequently, the induced value–function error remains $o_p(n^{-1/4})$ under the stated tuning (Assumptions 12—13). Two structural features ensure that the small regularization bias does not affect first–order inference. First, the *Envelope* argument makes the cross term involving $\partial_\theta\widetilde{\gamma}(\theta)$ asymptotically negligible because the inner FOC holds. Second, the *zero–Jacobian* property of the policy–iteration/evaluation operators at the fixed point implies that the Gâteaux derivative of the expected score with respect to V is zero. Hence the score is Neyman–orthogonal: perturbations of V—including those induced by the regularizer—enter only at second order. Therefore the profiled likelihood shares the same linear expansion as if V^* were known, and, after any fixed number $k \ge 1$ of policy–iteration steps, the estimator $\widehat{\theta}$ is $\sqrt{n}$–consistent and attains the semiparametric efficiency bound.

E Proof of proposition 6

Proof. Write $F_0(\gamma) := -\ell_n(\theta, \gamma) + \omega_n \rho_n(\gamma, \theta, P)$ and $F_\lambda(\gamma) := F_0(\gamma) + \lambda_n R_n(\gamma)$. By optimality of $\widetilde{\gamma}$ for F_λ and of $\widehat{\gamma}$ for F_0,

$$F_0(\widetilde{\gamma}) - F_0(\widehat{\gamma}) \;\le\; \lambda_n \{R_n(\widehat{\gamma}) - R_n(\widetilde{\gamma})\}. \tag{63}$$

Assumption 13(i) (quadratic growth in value space) yields the lower bound

$$F_0(\widetilde{\gamma}) - F_0(\widehat{\gamma}) \;\ge\; c_0\, \omega_n \, \|V_{\widetilde{\gamma}} - V_{\widehat{\gamma}}\|^2_{L^2(\pi_{M_n})} \;=\; c_0\, \omega_n \, \|\Delta V\|^2_{L^2(\pi_{M_n})}. \tag{64}$$

To bound the right-hand side of (63), apply the mean-value formula to R_n on the line segment $\gamma_t := \widehat{\gamma} + t\,\Delta\gamma$ ($t \in [0,1]$), which lies in $\mathcal{N}$ w.p.a.1:

$$R_n(\widehat{\gamma}) - R_n(\widetilde{\gamma}) = \int_0^1 \nabla_\gamma R_n(\gamma_t)^\top (\widehat{\gamma} - \widetilde{\gamma})\, dt \;\le\; \Big(\sup_{\gamma \in \mathcal{N}} \|\nabla_\gamma R_n(\gamma)\| \Big) \, \|\Delta\gamma\| \;\le\; B_R \, \|\Delta\gamma\|,$$

by Assumption 13(ii). Since $\widehat{\gamma}, \widetilde{\gamma} \in \mathcal{N}$ w.p.a.1, we also have $\|\Delta\gamma\| \le R_{\mathcal{N}}$. Combining with (63)–(64) gives

$$c_0\, \omega_n \, \|\Delta V\|^2_{L^2(\pi_{M_n})} \;\le\; \lambda_n \, B_R \, \|\Delta\gamma\| \;\le\; \lambda_n \, B_R \, R_{\mathcal{N}}.$$

Rearranging yields $\|\Delta V\|_{L^2(\pi_{M_n})} \le (B_R R_{\mathcal{N}}/c_0)^{1/2} \, (\lambda_n/\omega_n)^{1/2}$. Finally, since $R_{\mathcal{N}} = O(1)$, and under Assumption 12, $\lambda_n/\omega_n = o(n^{-1/2})$, so $(\lambda_n/\omega_n)^{1/2} = o(n^{-1/4})$, and the concluding statement follows. □

F Proof of Corollary 2

Proof. We verify that the profile-score analysis used in Theorem 4.3 carries over verbatim once we show the nuisance value error remains $o_p(n^{-1/4})$.

Step 1 (Profile score and cross term). Write the profiled gradient

$$\frac{d}{d\theta} L_n(\theta) \;=\; \partial_\theta \ell_n(\theta, \widetilde{\gamma}) \;+\; \nabla_\gamma \ell_n(\theta, \widetilde{\gamma}) \cdot \partial_\theta \widetilde{\gamma}, \tag{65}$$

and recall the KKT condition at $\widetilde{\gamma}$,

$$\nabla_\gamma \ell_n(\theta, \widetilde{\gamma}) \;=\; \omega_n \, \nabla_\gamma \rho_n(\widetilde{\gamma}, \theta, P) \;+\; \lambda_n \, \nabla_\gamma \mathcal{R}_n(\widetilde{\gamma}). \tag{66}$$

Step 2 (Regularization bias and nuisance rate). By Corollary 1, we obtain

$$\|V_{\widetilde{\gamma}} - V^*\|_{L^2} \;=\; o_p(n^{-1/4}) \qquad \text{uniformly in } \theta \in \Theta. \tag{67}$$

Step 3 (Orthogonality remainder is second order). By Proposition 4 (Neyman orthogonality),

$$\Big\| \mathbb{E}[\psi(W; \theta_0, V_{\widetilde{\gamma}})] - \mathbb{E}[\psi(W; \theta_0, V^*)] \Big\| \;\lesssim\; \|V_{\widetilde{\gamma}} - V^*\|^2_{L^2} \;=\; o_p(n^{-1/2}),$$

using (67). Thus the population bias in the score due to using $V_{\widetilde{\gamma}}$ is $o_p(n^{-1/2})$.

Step 4 (Asymptotics for $\widehat{\theta}$). With the orthogonality remainder $o_p(n^{-1/2})$ (Step 3), the *profile score identity and mean-value expansion* used in Appendix B go through verbatim, now with $V_{\widetilde{\gamma}}$ in

place of $V_{\widehat{\gamma}}$. The empirical process control is unchanged (Lemma 4.2), and the Jacobian limit and information identity are the same as in Steps 7–8 of Appendix B. Therefore the influence function and variance coincide with those in Theorem 4.3, yielding

$$\sqrt{n}\,(\widehat{\theta} - \theta_0) \xrightarrow{d} \mathcal{N}(0, I_{\theta\theta}^{-1}).$$

□

G Analytical expressions of the Lipschitz constants L_φ and L_Λ

We will use two uniform bounds that hold on a compact state space by Assumptions 4 and 6:

$$\bar{u} := \sup_{x,a} |u_a(x;\theta)| < \infty, \qquad B := \sup_x |V(x)| < \infty,$$

G.1 A bound for the policy–improvement operator Λ_θ

Fix θ and two value functions V_1, V_2 and set $\Delta V := V_1 - V_2$. Write the choice–specific continuation values

$$Q_a(x;V) := u_a(x;\theta) + \beta\,\mathbb{E}[V(X') \mid X = x, a], \quad \text{so that} \quad \Lambda_\theta[V](a|x) = \frac{e^{Q_a(x;V)}}{\sum_j e^{Q_j(x;V)}}.$$

Let $\Delta Q(x) \in \mathbb{R}^J$ collect the differences $\Delta Q_a(x) := \beta\,\mathbb{E}[\Delta V(X') \mid X = x, a]$ and let $\Delta P(x) := \Lambda_\theta[V_1](\cdot|x) - \Lambda_\theta[V_2](\cdot|x)$, where J is the number of possible actions.

Step 1: Softmax is 1/2–Lipschitz in ℓ_2. The Jacobian of softmax(q) at $q \in \mathbb{R}^J$ equals $H(q) = \mathrm{diag}(p) - pp^\top$ with $p = \mathrm{softmax}(q)$. For row i, the absolute row sum is

$$\sum_{j=1}^{J} |H(q)_{ij}| = p_i(1-p_i) + \sum_{j\neq i} p_i p_j = 2\,p_i(1-p_i) \le \tfrac{1}{2},$$

since $p_i(1-p_i) \le 1/4$. Hence $\|H(q)\|_\infty \le \frac{1}{2}$ and, by the standard inequality $\|M\|_2 \le \sqrt{\|M\|_1 \|M\|_\infty}$, also $\|H(q)\|_2 \le \frac{1}{2}$. By the mean–value theorem in $\mathbb{R}^J$,

$$\|\Delta P(x)\|_2 \;\le\; \sup_{t\in[0,1]} \|H(Q(x;V_2) + t\,\Delta Q(x))\|_2\, \|\Delta Q(x)\|_2 \;\le\; \frac{1}{2}\,\|\Delta Q(x)\|_2. \tag{68}$$

Step 2: Propagating ΔV through the transition. Define the linear map $S : L_2(P_X) \to L_2(P_X; \mathbb{R}^J)$ by

$$(S\Delta V)(x) := \big(\mathbb{E}[\Delta V(X') \mid X = x, a]\big)_{a=1}^{J}.$$

Introduce its operator norm

$$C_{\mathrm{tr}} := \sup_{\|\Delta V\|_2 = 1} \big\|S\,\Delta V\big\|_2 = \sup_{\|\Delta V\|_2=1} \Big(\mathbb{E}\Big[\sum_{a=1}^{J} \big(\mathbb{E}[\Delta V(X') \mid X, a]\big)^2\Big]\Big)^{1/2}.$$

Using (68) and $\Delta Q = \beta\, S\Delta V$ we obtain, pointwise in x, $\|\Delta P(x)\|_2 \le (\beta/2)\,\|(S\Delta V)(x)\|_2$; squaring and integrating over X yields

$$\|\Lambda_\theta[V_1] - \Lambda_\theta[V_2]\|_2 = \|\Delta P\|_2 \;\le\; \frac{\beta}{2}\,\|S\Delta V\|_2 \;\le\; \frac{\beta}{2}\,C_{\mathrm{tr}}\,\|\Delta V\|_2.$$

Step 3: A simple primitive bound for C_{tr}. For each action a, let

$$\mu_a(dx') := \int f_{\kappa_0}(x' \mid x, a)\, P_X(dx), \qquad r_a(x') := \frac{d\mu_a}{dP_X}(x')$$

Writing $(T_a h)(x) := \mathbb{E}[h(X') \mid X = x, a]$, Jensen's inequality gives, for any $h \in L^2(P_X)$,

$$\|T_a h\|_2^2 = \mathbb{E}[\mathbb{E}[h(X') \mid X, a]^2] \le \int h(x')^2\, \mu_a(dx') = \int h(x')^2 r_a(x')\, P_X(dx') \le \|r_a\|_{L^\infty(P_X)} \|h\|_2^2.$$

Let $h(X') = \Delta V(X')$, then $C_{\mathrm{tr}}^2 = \sup_{\|h\|_2=1} \sum_{a=1}^J \|T_a h\|_2^2$. We obtain

$$\boxed{C_{\mathrm{tr}}^2 \le \sum_{a=1}^J \|r_a\|_{L^\infty(P_X)}} \quad \Rightarrow \quad L_\Lambda = \frac{\beta}{2} C_{\mathrm{tr}} \;\le\; \frac{\beta}{2} \Big(\textstyle\sum_{a=1}^J \|r_a\|_{L^\infty(P_X)} \Big)^{1/2}.$$

G.2 A bound for the evaluation operator φ_θ

Fix θ and V, and consider two policy functions P_1, P_2 with $\Delta P := P_1 - P_2$. Using the decomposition $\varphi_\theta[P, V] = c_{\theta,P} + \beta\, T_P V$ in equation 6,

$$\begin{aligned}(\varphi_\theta[P_1, V] - \varphi_\theta[P_2, V])(x) = &\underbrace{\sum_a \Delta P(a|x)\, u_a(x;\theta)}_{\text{(i)}} \;-\; \underbrace{\sum_a \left[P_1(a|x) \log P_1(a|x) - P_2(a|x) \log P_2(a|x)\right]}_{\text{(ii)}} \\ &+ \underbrace{\beta \sum_a \Delta P(a|x)\, \mathbb{E}[V(X')|x, a]}_{\text{(iii)}}.\end{aligned}$$

We bound the three terms at any given x and then integrate.

Term (i) (utility part). By Cauchy–Schwarz,

$$\Big| \sum_a \Delta P(a|x)\, u_a(x;\theta) \Big| \le \|\Delta P(x)\|_2 \Big(\sum_a u_a(x;\theta)^2 \Big)^{1/2} \le \sqrt{J}\, \bar{u}\, \|\Delta P(x)\|_2.$$

Term (ii) (entropy part). For $g(p) := p \log p$ one has $g'(p) = 1 + \log p$. By the mean–value theorem applied to each coordinate a and interiority $P_i(a|x) \ge p_{\min}$ (holds by Assumption 4),

$$\big|\, P_1 \log P_1 - P_2 \log P_2 \,\big| \le (1 + |\log p_{\min}|)\, |P_1 - P_2|,$$

hence

$$\Big| \sum_a P_1(a|x) \log P_1(a|x) - P_2(a|x) \log P_2(a|x) \Big| \le (1 + |\log p_{\min}|) \sum_a |\,\Delta P(a|x)\,| \le \sqrt{J}\, (1 + |\log p_{\min}|)\, \|\Delta P(x)\|_2.$$

Term (iii) (continuation part). Since $|\mathbb{E}[V(X')|x, a]| \le \sup_{x'} |V(x')| =: B$,

$$\Big| \beta \sum_a \Delta P(a|x)\, \mathbb{E}[V(X')|x, a] \Big| \le \beta\, B \sum_a |\Delta P(a|x)| \le \beta\, B\, \sqrt{J}\, \|\Delta P(x)\|_2.$$

Combine and integrate. Adding (i)–(iii) and applying the triangle inequality,

$$|\varphi_\theta[P_1, V](x) - \varphi_\theta[P_2, V](x)| \le \sqrt{J}\, (\bar{u} + 1 + |\log p_{\min}| + \beta B)\, \|\Delta P(x)\|_2.$$

Squaring and taking expectations (over X) yields

$$\|\varphi_\theta[P_1, V] - \varphi_\theta[P_2, V]\|_2 \;\le\; L_\varphi\, \|P_1 - P_2\|_2, \qquad L_\varphi := \sqrt{J}\, (\bar{u} + 1 + |\log p_{\min}| + \beta B).$$

H Asymptotic Theory under Hölder–Sparse Smoothness and K–Fold Cross-Fitting

In this section, we relax the Sieve complexity assumption 7 on the value function and CCP, and instead allows the sieve to be over-parameterized. We demonstrate that by imposing sparsity assumption and incorporating cross-fitting into the NNES algorithm, we can accommodate over-parameterized Neural Networks and still achieve $\sqrt{n}$-consistency and semiparametric efficiency for the structural parameter θ without imposing stronger (e.g., Donsker) conditions on the score function classes.

H.1 Cross-Fitted NNES Algorithm

The Cross-Fitted NNES algorithm is a policy-iteration procedure where the structural parameter estimation at each step is made robust via K-fold cross-fitting. This structure prevents overfitting in the nuisance function estimation from biasing the structural parameter estimates. The algorithm proceeds as follows.

1. Initialization (Iteration $m = 0$). First, partition the observation indices $\mathcal{I}_n = \{1, \ldots, n\}$ randomly into K disjoint folds $\{\mathcal{I}_k\}_{k=1}^K$. For each fold $k \in \{1, \ldots, K\}$, obtain an initial non-parametric estimate of the conditional choice probabilities, $P^{(0,-k)}$, using only the training dataset $\mathcal{I}_{-k} := \mathcal{I}_n \setminus \mathcal{I}_k$. This can be done using a flexible classifier (e.g., a neural network). This yields a set of K initial policies $\{P^{(0,-k)}\}_{k=1}^K$ to start the policy iteration.

2. Policy Iteration Loop (for $m = 1, 2, \ldots, M$). For each iteration m, we perform two main steps: (a) estimation of the structural parameter $\theta^{(m)}$ and (b) a fold-specific update of the policies to $\{P^{(m,-k)}\}_{k=1}^K$.

(a) M-Estimation of $\theta^{(m)}$ via Cross-Fitting. The structural parameter estimate at iteration m, denoted $\theta^{(m)}$, is obtained by maximizing a cross-fitted pseudo-likelihood. This objective function uses the set of *fold-specific* CCP estimates from the previous iteration, $\{P^{(m-1,-k)}\}_{k=1}^K$, as fixed inputs:

$$\theta^{(m)} := \arg\max_{\theta \in \Theta} \ell_{\mathrm{CF}}(\theta; \{P^{(m-1,-k)}\}_{k=1}^K)$$

where the evaluation of the objective function for any given candidate θ requires the following three-step procedure:

i. **Fold-Specific Value Function Estimation (Training):** For each fold $k \in \{1, \ldots, K\}$, use the training data $\mathcal{I}_{-k}$ to estimate the value function parameters. This is done by solving a minimization problem that depends on θ and the corresponding *fold-specific* policy $P^{(m-1,-k)}$:

$$\widehat{\gamma}^{(-k)}(\theta, P^{(m-1,-k)}) := \arg\min_{\gamma \in \Gamma_n} \frac{1}{|\mathcal{I}_{-k}|} \sum_{i \in \mathcal{I}_{-k}} \mathcal{L}(W_i; \theta, \gamma, P^{(m-1,-k)}), \tag{69}$$

where the loss function $\mathcal{L}$ is

$$\mathcal{L}(\cdot) = -\log\left((\Lambda_\theta[V_\gamma])(A_i|X_i)\right) + \omega_n \left(V_\gamma(X_i) - (\varphi_\theta[P^{(m-1,-k)}])(X_i)\right)^2.$$

Let the resulting value function estimator for this fold be $V_\theta^{(-k)} := V_{\widehat{\gamma}^{(-k)}(\theta, P^{(m-1,-k)})}$.

ii. **Fold-Specific Policy Construction:** Construct the implied CCP function for fold k, which represents the *one-step-improved* policy based on the value function from the previous step:

$$\eta_\theta^{(-k)} := \Lambda_\theta[V_\theta^{(-k)}].$$

By construction, this nuisance function $\eta_\theta^{(-k)}$ is a function only of data in $\mathcal{I}_{-k}$.

iii. **Cross-Fitted Likelihood Calculation:** The final objective function is constructed by averaging the log-likelihoods evaluated on the hold-out folds:

$$\ell_{\mathrm{CF}}(\theta; \{P^{(m-1,-k)}\}) := \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{n_k}\sum_{i\in\mathcal{I}_k}\log\left((\eta_\theta^{(-k)})(A_i|X_i)\right)\right).$$

(b) Fold-Specific Policy Update (Policy Improvement). After obtaining the parameter estimate $\theta^{(m)}$ from step (a), the policy functions are updated for the next iteration. This is done **for each fold** $k \in \{1, \ldots, K\}$ **separately**, using only the training data $\mathcal{I}_{-k}$:

i. Estimate a value function $\widehat{V}^{(m,-k)}$ using the new parameter estimate $\theta^{(m)}$ and the previous policy for that fold, $P^{(m-1,-k)}$, on the training data $\mathcal{I}_{-k}$:

$$\widehat{V}^{(m,-k)} := V_{\arg\min_V \frac{1}{|\mathcal{I}_{-k}|}\sum_{i\in\mathcal{I}_{-k}}\left[\left(V(X_i)-(\varphi_{\theta^{(m)}}[P^{(m-1,-k)}])(X_i)\right)^2\right]}.$$

ii. The updated CCP for the next iteration's training on $\mathcal{I}_{-k}$ is then:

$$P^{(m,-k)} := \Lambda_{\theta^{(m)}}[\widehat{V}^{(m,-k)}].$$

This step produces the set of policies $\{P^{(m,-k)}\}_{k=1}^K$ needed for iteration $m+1$.

3. Convergence and Final Estimators. Repeat the policy iteration loop (Step 2) for a fixed number of iterations M, or until convergence. The final estimator for the structural parameter is the output of the last iteration:

$$\widehat{\theta} = \theta^{(M)}.$$

If a single final policy function $\widehat{P}$ is required (e.g., for simulation), it can be estimated using the final parameter $\widehat{\theta}$ and the full dataset $\mathcal{I}_n$ as a post-processing step. For theoretical purposes, a small, fixed $M \geq 1$ is sufficient for efficiency.

H.2 Assumptions for the Cross-Fitting Estimator

The asymptotic theory for the cross-fitted estimator relies on a set of assumptions governing the data generating process, the structure of the nuisance functions, the performance of the first-stage estimators, and the properties of the final score function. All random variables are defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and $\|\cdot\|_2$ denotes the $L^2(\pi)$ norm.

Assumption 14 (Model Regularity and Primitives)**.** *1.* ***Sampling:*** *The data $\{W_i = (A_i, X_i)\}_{i=1}^n$ are independent and identically distributed (i.i.d.) draws from a stationary data generating process P_W. The structural parameter space $\Theta \subset \mathbb{R}^{d_\theta}$ is compact, and the true parameter θ_0 lies in its interior.*

2. ***Primitives Smoothness:*** *The per-period utility function $u_a(x;\theta)$ and the conditional transition density $f_\theta(x'|x,a)$ are twice continuously differentiable in $\theta \in \Theta$. The mapping $\gamma \mapsto V_\gamma(x)$ defined by the neural network architecture is twice continuously differentiable with respect to the network parameters $\gamma \in \Gamma_n$ for all $x \in \mathcal{X}$. These conditions ensure that the operators $\Lambda_\theta[V_\gamma]$ and $\varphi_\theta[P]$ are sufficiently smooth in their arguments.*

3. ***Bounded Probabilities:*** *The true conditional choice probabilities $P^*(a|x)$ are uniformly bounded away from 0 and 1. That is, there exists a constant $\underline{p} \in (0, 1/2]$ such that for all (a,x), $\underline{p} \leq P^*(a|x) \leq 1 - \underline{p}$.*

Assumption 15 (Nuisance Target Function Structure)**.** 1. ***Hölder-Sparsity of the True Value Function:*** *The true value function $V^*(x) := V^*_{\theta_0}(x)$ is sparse, depending only on a subset of state variables x_S where $S \subset \{1,\ldots,d\}$ with $|S| = s \ll d$. Formally, $V^*(x) = g(x_S)$ for a function g belonging to a Hölder space $\mathcal{H}^p(L)$ with smoothness $p > s/2$ and finite Hölder norm L. This structural property of the economic model's solution ensures that the minimax nonparametric estimation rate, $\alpha = p/(2p+s)$, exceeds the $1/4$ threshold.*

2. ***Stability of the Value Function Map:*** *For each $\theta \in \Theta$, let V^*_θ be the unique fixed point of the Bellman operator. The mapping $\theta \mapsto V^*_\theta$ is Lipschitz continuous in the $L^2(P_X)$ norm in a neighborhood of θ_0. That is, there exists a constant $C_V < \infty$ such that $\|V^*_\theta - V^*_{\theta_0}\|_{L^2} \leq C_V\|\theta - \theta_0\|$. This can be derived from Lipschitz continuity of the model primitives in θ and the contraction property of the Bellman operator.*

Assumption 16 (First-Stage Estimator Performance)**.** *For each iteration m, let $\{P^{(m-1,-k)}\}_{k=1}^K$ be a set of fold-specific policy estimators consistent for the true policy P^*. The neural network value function estimators $V_{\widehat{\gamma}^{(-k)}(\theta)}$, where $\widehat{\gamma}^{(-k)}(\theta)$ solves the regularized minimization problem on fold $\mathcal{I}_{-k}$, satisfy:*

1. ***Regularized Estimation:*** *The loss function in (69) is amended with an explicit regularizer $\lambda_n\mathcal{R}(\gamma)$, such as a weight decay or group-lasso penalty, necessary for controlling model complexity.*

2. ***Uniform Convergence Rate:*** *With an appropriately chosen tuning parameter sequence λ_n, the first-stage estimators converge uniformly over θ and all folds to the true value function V^*_θ at a rate faster than $n^{-1/4}$:*

$$\max_{k\leq K}\sup_{\theta\in\Theta}\|V_{\widehat{\gamma}^{(-k)}(\theta)} - V^*_\theta\|_{L^2} = O_p(n^{-\alpha}), \quad \text{where } \alpha = \frac{p}{2p+s} > \frac{1}{4}.$$

This is a high-level condition justified by results on high-dimensional non-parametric estimation with neural networks under the structure imposed by Assumption 15 Schmidt-Hieber [2020].

Assumption 17 (Score Function Properties and Identification)**.** *Let the score function be defined with the neural network weights γ as the nuisance parameter:*

$$\psi(W;\theta,\gamma) := \nabla_\theta \log\left((\Lambda_\theta[V_\gamma])(A|X)\right).$$

Let $\gamma^*(\theta)$ denote the population minimizer of the expected squared Bellman error for a given θ, representing the best-in-class neural network approximation to V^*_θ. The score function satisfies:

1. ***Moment and Lipschitz Conditions:*** *The score function satisfies the following regularity conditions, which are sufficient for controlling remainder terms in the cross-fitting expansion without requiring Donsker-class assumptions:*

i. (Uniform Moment Bound) $\sup_{\theta\in\Theta,\gamma\in\Gamma_n} \mathbb{E}\left[\|\psi(W;\theta,\gamma)\|^2\right] < \infty$.

ii. (Lipschitz Continuity) The map $\gamma \mapsto \psi(W;\theta,\gamma)$ *is Lipschitz continuous in the* $L^2(P_W)$ *norm, uniformly in* θ*. There exists a constant* $L_\psi < \infty$ *such that for all* $\gamma_1, \gamma_2 \in \Gamma_n$*:*

$$\sup_{\theta\in\Theta} \left(\mathbb{E}\left[\|\psi(W;\theta,\gamma_1) - \psi(W;\theta,\gamma_2)\|^2\right]\right)^{1/2} \leq L_\psi \|\gamma_1 - \gamma_2\|_2.$$

2. ***Identification:*** *The model parameters are identified. The equation* $\mathbb{E}[\psi(W;\theta,\gamma^*(\theta))] = 0$ *has a unique solution at* $\theta = \theta_0$*. This requires that for any* $\theta \neq \theta_0$*, the implied policy* $\Lambda_\theta[V_{\gamma^*(\theta)}]$ *differs from the true policy* P^* *on a set of positive measure.*

3. ***Information Matrix:*** *The Fisher information matrix, defined with respect to the true nuisance value* $\gamma_0^* := \gamma^*(\theta_0)$*, is finite and positive definite:*

$$\mathcal{I}_{\theta\theta} := \mathbb{E}\left[\psi(W;\theta_0,\gamma_0^*)\psi(W;\theta_0,\gamma_0^*)'\right] \text{ is positive definite.}$$

H.3 Neyman Orthogonality with Respect to Network Parameters

Lemma H.1 (Neyman Orthogonality with Respect to Nuisance Parameters)**.** *Let* γ_0 *be the population value of the network parameters such that* V_{γ_0} *is the best approximation to the true value function* V^* *within the specified network class. Let the score function be defined through the one-step policy iteration map as* $\psi(W;\theta,\gamma) := \nabla_\theta \log\left((\Psi_\theta[\Lambda_\theta[V_\gamma]])(A|X)\right)$*.*

Under the smoothness and regularity conditions specified in Assumptions 14 and 17, the gradient of the expected score with respect to the network parameters γ *is zero at the true parameters* (θ_0,γ_0)*:*

$$\nabla_\gamma \mathbb{E}\left[\psi(W;\theta_0,\gamma)\right]\Big|_{\gamma=\gamma_0} = \mathbf{0}_{d_\theta \times d_\gamma}. \tag{70}$$

Proof. The proof proceeds by applying the chain rule for derivatives in a composite function structure, leveraging the Zero-Jacobian property of the policy-iteration operator Ψ.

Step 1: Interchange of Differentiation and Expectation.

Under the dominance and regularity conditions of Assumption 17, which ensure that ψ and its derivatives are suitably bounded, we can interchange the gradient operator with respect to the finite-dimensional parameter γ and the expectation operator:

$$\nabla_\gamma \mathbb{E}[\psi(W;\theta_0,\gamma)] = \mathbb{E}[\nabla_\gamma \psi(W;\theta_0,\gamma)].$$

Furthermore, due to the smoothness of the model primitives (Assumption 14), we can interchange the order of differentiation with respect to θ and γ:

$$\mathbb{E}[\nabla_\gamma \psi(W;\theta_0,\gamma)] = \mathbb{E}\left[\nabla_\theta\left(\nabla_\gamma \log\left((\Psi_\theta[\Lambda_\theta[V_\gamma]])(A|X)\right)\right)\right]_{\theta=\theta_0}.$$

To prove the lemma, we will show that the inner term, $\nabla_\gamma \log(\ldots)$, is a zero vector when evaluated at the true parameters, which implies its derivative with respect to θ is also zero.

Step 2: Chain Rule for the Derivative with Respect to γ.

Let's compute the derivative of the log-policy term with respect to γ. This requires a careful application of the chain rule. For notational clarity, let $\eta_\gamma(\theta) := \Lambda_\theta[V_\gamma]$. The expression is $\log\left((\Psi_\theta[\eta_\gamma(\theta)])(A|X)\right)$. The derivative is:

$$\begin{aligned}\nabla_\gamma \log\left((\Psi_\theta[\eta_\gamma(\theta)])(A|X)\right) &= \frac{1}{(\Psi_\theta[\eta_\gamma(\theta)])(A|X)} \nabla_\gamma \left((\Psi_\theta[\eta_\gamma(\theta)])(A|X)\right) \\ &= \frac{1}{(\Psi_\theta[\eta_\gamma(\theta)])(A|X)} \left(D_\eta \Psi_\theta(\eta_\gamma(\theta)) \circ \nabla_\gamma \eta_\gamma(\theta)\right)(A|X).\end{aligned}$$

The term $D_\eta \Psi_\theta(\eta_\gamma(\theta))$ is the Fréchet derivative of the operator Ψ_θ with respect to the function η, evaluated at $\eta_\gamma(\theta)$. The term $\nabla_\gamma \eta_\gamma(\theta)$ represents the derivative of the CCP map with respect to the weights γ, which is itself a chain rule:

$$\nabla_\gamma \eta_\gamma(\theta) = \nabla_\gamma(\Lambda_\theta[V_\gamma]) = D_V \Lambda_\theta(V_\gamma) \circ \nabla_\gamma V_\gamma.$$

Here, $D_V \Lambda_\theta$ is the Fréchet derivative of the improvement operator Λ_θ with respect to the value function V, and $\nabla_\gamma V_\gamma$ is the gradient of the network output with respect to its weights.

Step 3: Evaluation at the True Parameters and Application of the Zero-Jacobian Property.

We now evaluate this derivative at the true parameters (θ_0, γ_0). At this point, $V_{\gamma_0} \approx V^*$ and thus $\eta_{\gamma_0}(\theta_0) = \Lambda_{\theta_0}[V_{\gamma_0}] \approx P^*$. The crucial term in the chain rule is the Fréchet derivative $D_\eta \Psi_{\theta_0}(\eta_{\gamma_0}(\theta_0))$.

The Zero-Jacobian property (Assumption 14(d)) states that at the fixed point of the policy-iteration operator, its derivative is the zero operator:

$$D_\eta \Psi_{\theta_0}(P^*) = \mathbf{0}.$$

By continuity, evaluating at $\eta_{\gamma_0}(\theta_0) \approx P^*$ for a well-specified network, we have $D_\eta \Psi_{\theta_0}(\eta_{\gamma_0}(\theta_0)) = \mathbf{0}$. Therefore, the entire chain of derivatives for $\nabla_\gamma \log(\ldots)$ at θ_0 collapses to zero:

$$\nabla_\gamma \log\left((\Psi_{\theta_0}[\eta_\gamma(\theta_0)])(A|X)\right)\Big|_{\gamma=\gamma_0} = \frac{1}{P^*(A|X)} \cdot (\mathbf{0} \circ D_V \Lambda_{\theta_0}(V^*) \circ \nabla_\gamma V_{\gamma_0}) = \mathbf{0}.$$

Let $F(\theta, \gamma) := \nabla_\gamma \log\left((\Psi_\theta[\eta_\gamma(\theta)])(A|X)\right)$. We have just shown that $F(\theta_0, \gamma_0) = \mathbf{0}_{1 \times d_\gamma}$.

Step 4: Final Calculation of the Expected Score Gradient.

Returning to the expression from Step 1, we need to evaluate $\mathbb{E}\left[\nabla_\theta F(\theta, \gamma)\right]_{\theta=\theta_0, \gamma=\gamma_0}$. We have a function $F(\theta, \gamma)$ that is identically the zero vector at the point (θ_0, γ_0). Its derivative with respect to θ at this point, $\nabla_\theta F(\theta_0, \gamma_0)$, must therefore also be a zero matrix. Taking the expectation of a zero matrix results in a zero matrix.

A more formal argument is to note that the identity in Step 3 holds for every realization of $W = (A, X)$. Let $f(W, \theta, \gamma) := \nabla_\gamma \log\left((\Psi_\theta[\eta_\gamma(\theta)])(A|X)\right)$. We have shown $f(W, \theta_0, \gamma_0) = \mathbf{0}$ for all W. Taking the partial derivative with respect to θ of a function that is identically zero at (θ_0, γ_0) over the entire sample space results in a function that is also identically zero at that point:

$$\nabla_\theta f(W, \theta_0, \gamma_0) = \mathbf{0}_{d_\theta \times d_\gamma} \quad \text{for all } W.$$

The expectation of this zero matrix is itself the zero matrix. Therefore:

$$\nabla_\gamma \mathbb{E}\left[\psi(W; \theta_0, \gamma)\right]\Big|_{\gamma=\gamma_0} = \mathbb{E}[\nabla_\theta f(W, \theta_0, \gamma_0)] = \mathbb{E}[\mathbf{0}] = \mathbf{0}.$$

This confirms that the score is orthogonal to the first-stage neural network parameters, which is the key property enabling robust and efficient estimation of θ_0. □

H.4 Uniform Convergence Rate of the First-Stage Estimator

Before proving the main theorem, we must establish a key technical result: the uniform convergence rate of our first-stage nuisance estimator, $V_\theta^{(-k)}$. This is more complex than a standard pointwise result because our estimator depends on the structural parameter θ, and we need the convergence to hold uniformly over all possible values of θ in the compact set Θ.

Lemma H.2 (Uniform First-Stage Estimation Rate)**.** *Under the high-level conditions specified in Assumption 16(a), which are derived from primitive Assumptions 14-15, the fold-specific neural network value function estimator $V_\theta^{(-k)}$ converges to its corresponding population target V_θ^* uniformly over $\theta \in \Theta$ and over all folds $k \in \{1, \dots, K\}$:*

$$\sup_{k \le K} \sup_{\theta \in \Theta} \|V_\theta^{(-k)} - V_\theta^*\|_2 = O_p(n^{-\alpha}),$$

where $\alpha = \frac{p}{2p+s} > \frac{1}{4}$ is the rate determined by the Hölder-Sparsity of the value function.

Proof. The proof proceeds in three stages. First, we establish a pointwise convergence rate for a fixed θ. Second, we extend this to a uniform result over a finite grid of θ values. Third, we use a Lipschitz condition to extend the result from the grid to the entire compact set Θ.

Step 1: Pointwise Convergence Rate (for a fixed θ).

For any fixed $\theta \in \Theta$ and fold k, $V_\theta^{(-k)}$ is a penalized M-estimator of V_θ^*. The standard proof strategy for such estimators is to derive an oracle inequality that bounds the estimation error by the sum of an approximation error and a stochastic complexity term.

1. **Approximation Error:** By Assumption 15(a), the target function V_θ^* (which inherits the Hölder-Sparsity property) can be approximated by a neural network. Results such as Proposition 8 in Schmidt-Hieber [2020] show that for a network with m_n neurons, there exists a best-in-class approximator $\tilde{V}_{\theta,n}$ from the network class such that the approximation error is bounded:
$$\|\tilde{V}_{\theta,n} - V_\theta^*\|_2 \le C_1 m_n^{-p/s}.$$

2. **Oracle Inequality:** The estimator $V_\theta^{(-k)}$ minimizes the objective in (69). By analyzing the basic inequality that the loss at the minimum is no greater than the loss at the best-in-class approximator $\tilde{V}_{\theta,n}$, and leveraging concentration inequalities for empirical processes indexed by the network class, one can derive an oracle inequality of the form (with high probability):
$$\|V_\theta^{(-k)} - V_\theta^*\|_2^2 \le C_2 \inf_{V \in \mathcal{G}_n} \|V - V_\theta^*\|_2^2 + \text{StochasticTerm}(\lambda_n, s, n_k),$$
where $\mathcal{G}_n$ is the class of network functions and the stochastic term is controlled by the regularization parameter λ_n (e.g., a group-Lasso or weight decay penalty). This is a standard argument in high-dimensional M-estimation; see, for example, Chapter 14 of Bühlmann and Van De Geer [2011].

3. **Choosing Tuning Parameters:** By setting the number of neurons $m_n \asymp n_k^{s/(2p+s)}$ and the penalty parameter $\lambda_n \asymp \sqrt{\log(d)/n_k}$, the approximation and stochastic terms are balanced to optimize the overall rate. This yields the minimax optimal rate for the given function class

and gives the pointwise rate: for fixed θ, k,

$$\|V_\theta^{(-k)} - V_\theta^*\|_2 = O_p(n_k^{-\alpha}) = O_p(n^{-\alpha}), \quad \text{where } \alpha = \frac{p}{2p+s}.$$

Step 2: Uniformity over a Finite Grid.

Now we establish uniformity over Θ. Since Θ is a compact subset of $\mathbb{R}^{d_\theta}$, we can cover it with a finite grid $\Theta_N = \{\theta_j\}_{j=1}^N$ where the number of points N can depend on n. By applying a union bound over the K folds and the N grid points:

$$\mathbb{P}\left(\max_{k\le K}\max_{\theta_j\in\Theta_N} \|V_{\theta_j}^{(-k)} - V_{\theta_j}^*\|_2 > Cn^{-\alpha}\right) \le K\cdot N\cdot \max_{k,j}\mathbb{P}\left(\|V_{\theta_j}^{(-k)} - V_{\theta_j}^*\|_2 > Cn^{-\alpha}\right).$$

The probability on the right-hand side can be shown to decay polynomially in n, e.g., $O(n_k^{-c})$ for a large constant c, by using sharper concentration inequalities. If we choose the grid size N to grow at a polynomial rate with n (e.g., $N \asymp n^{d_\theta}$), the total probability still vanishes as $n\to\infty$. This establishes the result uniformly over the grid:

$$\max_{k\le K}\max_{\theta_j\in\Theta_N} \|V_{\theta_j}^{(-k)} - V_{\theta_j}^*\|_2 = O_p(n^{-\alpha}).$$

Step 3: Extension to the Full Parameter Space Θ.

To extend the result from the grid Θ_N to the continuous set Θ, we rely on a Lipschitz-continuity property of the relevant maps. For any $\theta\in\Theta$, let θ_j be the closest grid point in Θ_N. The total error can be decomposed using the triangle inequality:

$$\|V_\theta^{(-k)} - V_\theta^*\|_2 \le \underbrace{\|V_\theta^{(-k)} - V_{\theta_j}^{(-k)}\|_2}_{\text{(I)}} + \underbrace{\|V_{\theta_j}^{(-k)} - V_{\theta_j}^*\|_2}_{\text{(II)}} + \underbrace{\|V_{\theta_j}^* - V_\theta^*\|_2}_{\text{(III)}}.$$

- Term (II) is $O_p(n^{-\alpha})$ uniformly over the grid from Step 2.
- Term (III) is bounded by $C_V\|\theta-\theta_j\|$ due to Assumption 15(b). By choosing the grid density such that $\max_\theta \min_j \|\theta-\theta_j\|$ vanishes faster than $n^{-\alpha}$, this term is negligible.
- Term (I) represents the stability of the M-estimator with respect to θ. Under smoothness conditions on the objective function (69) with respect to θ, it can be shown that this term is also bounded by a constant times $\|\theta-\theta_j\|$. Such stability results are common in the analysis of M-estimators (e.g., Newey and McFadden [1994]).

Combining these bounds, the additional error incurred by moving off the grid is controlled by the grid spacing. By choosing the grid fine enough, this additional error is of a smaller order than the primary $O_p(n^{-\alpha})$ rate. Therefore, taking the supremum over $\theta\in\Theta$ preserves the rate. This completes the proof of uniform convergence. □

H.5 Decomposition of the Cross-Fitted Score

The key to the asymptotic normality result is a careful analysis of the cross-fitted score evaluated at the true parameter θ_0. The following lemma shows that it can be decomposed into the sum of the efficient influence function and a remainder term that vanishes asymptotically. This proof is structured following the modern literature on debiased machine learning and cross-fitting **?**.

Lemma H.3 (Asymptotic Expansion of the Cross-Fitted Score)**.** *Let the score function be* $\psi(W;\theta,\gamma) = \nabla_\theta \log\left((\Lambda_\theta[V_\gamma])(A|X)\right)$*. Let* $\widehat{\gamma}_{\theta_0}^{(-k)}$ *be the nuisance parameter estimator obtained on the training fold* $\mathcal{I}_{-k}$ *for a fixed* $\theta = \theta_0$*. Let* γ_0^* *be the population value of the network parameters corresponding to the best-in-class neural network approximation of the true value function* $V_{\theta_0}^*$*.*

Define the cross-fitted score at θ_0 *as* $\widehat{\Psi}(\theta_0) = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in\mathcal{I}_k}\psi(W_i;\theta_0,\widehat{\gamma}_{\theta_0}^{(-k)})$*. Let the efficient score be* $U_i := \psi(W_i;\theta_0,\gamma_0^*)$*.*

Under Assumptions 14–17, the following expansion holds:

$$\sqrt{n}\widehat{\Psi}(\theta_0) = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} U_i + o_p(1).$$

Proof. We define the remainder term for an observation $i \in \mathcal{I}_k$ with respect to the network parameters γ as:

$$R_i^{(-k)} := \psi(W_i;\theta_0,\widehat{\gamma}_{\theta_0}^{(-k)}) - \psi(W_i;\theta_0,\gamma_0^*).$$

The scaled cross-fitted score can be decomposed as:

$$\sqrt{n}\widehat{\Psi}(\theta_0) = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} U_i + \Delta_n, \quad \text{where} \quad \Delta_n := \frac{1}{\sqrt{n}}\sum_{k=1}^{K}\sum_{i\in\mathcal{I}_k} R_i^{(-k)}.$$

The proof proceeds by showing that the remainder term Δ_n converges to zero in probability. We achieve this by demonstrating that its second moment, $\mathbb{E}[\|\Delta_n\|^2]$, converges to zero.

The key insight from the cross-fitting literature is that the second moment of the remainder can be bounded without assuming independence between the nuisance estimators of different folds. For K-fold cross-fitting, the second moment is bounded by the sum of a bias and a variance component:

$$\mathbb{E}[\|\Delta_n\|^2] \lesssim \mathbb{E}\left[\left\|\mu_R^{(-k)}\right\|^2\right] + \frac{1}{n_k}\mathbb{E}\left[\mathrm{Var}(R_i^{(-k)} \mid \mathcal{D}_{-k})\right],$$

where $\mu_R^{(-k)} := \mathbb{E}[R_i^{(-k)} \mid \mathcal{D}_{-k}]$ is the conditional mean of the remainder, $n_k \approx n/K$ is the size of fold k, and $\mathcal{D}_{-k} = \{W_j\}_{j\in\mathcal{I}_{-k}}$ is the training data for fold k. We now bound each term.

Step 1: Bounding the Conditional Mean (Bias Term). The conditional mean of the remainder for $i \in \mathcal{I}_k$ is

$$\mu_R^{(-k)} = \mathbb{E}\left[R_i^{(-k)} \mid \mathcal{D}_{-k}\right] = \mathbb{E}_W\left[\psi(W;\theta_0,\widehat{\gamma}_{\theta_0}^{(-k)}) - \psi(W;\theta_0,\gamma_0^*)\right],$$

where the expectation $\mathbb{E}_W$ is over the population distribution of W, conditional on the training data $\mathcal{D}_{-k}$. A preceding lemma (e.g., Lemma H.1) must establish the Neyman Orthogonality property, which states that the Gâteaux derivative of $\mathbb{E}_W[\psi(W;\theta_0,\gamma)]$ with respect to γ is zero at γ_0^*. This implies that the conditional mean is of second order in the nuisance estimation error. Combined with a suitable second-order differentiability condition, we have:

$$\|\mu_R^{(-k)}\|_2 \lesssim \left\|V_{\widehat{\gamma}_{\theta_0}^{(-k)}} - V_{\theta_0}^*\right\|_{L^2}^2.$$

By the uniform convergence rate from Assumption 16(b), $\|V_{\widehat{\gamma}_{\theta_0}^{(-k)}} - V_{\theta_0}^*\|_{L^2} = O_p(n^{-\alpha})$. Therefore, the conditional mean satisfies:

$$\|\mu_R^{(-k)}\|_2 = O_p\left((n^{-\alpha})^2\right) = O_p(n^{-2\alpha}).$$

The contribution of the squared bias term to the second moment is $\mathbb{E}[\|\mu_R^{(-k)}\|^2] = O(n^{-4\alpha})$.

Step 2: Bounding the Conditional Variance. The variance term is controlled by the expected conditional second moment of the remainder. Using the Stochastic Lipschitz Continuity of the score function from Assumption 17(a)(ii):

$$\begin{aligned}\mathbb{E}\left[\|R_i^{(-k)}\|^2 \mid \mathcal{D}_{-k}\right] &= \mathbb{E}_W\left[\|\psi(W;\theta_0,\widehat{\gamma}_{\theta_0}^{(-k)}) - \psi(W;\theta_0,\gamma_0^*)\|^2\right] \\ &\lesssim \left\|V_{\widehat{\gamma}_{\theta_0}^{(-k)}} - V_{\theta_0}^*\right\|_{L^2}^2 \\ &= O_p\left((n^{-\alpha})^2\right) = O_p(n^{-2\alpha}).\end{aligned}$$

Taking the unconditional expectation, we find $\mathbb{E}[\|R_i^{(-k)}\|^2] = O(n^{-2\alpha})$. The total contribution of the variance component to $\mathbb{E}[\|\Delta_n\|^2]$ is approximately $\frac{1}{n} \cdot n \cdot O(n^{-2\alpha}) = O(n^{-2\alpha})$.

Step 3: Final Rate of the Remainder. Combining the bounds for the bias and variance components, the second moment of the scaled remainder is:

$$\mathbb{E}[\|\Delta_n\|^2] = O(n^{-4\alpha}) + O(n^{-2\alpha}) = O(n^{-2\alpha}).$$

By Assumption 15(a), the Hölder-sparsity structure ensures that the first-stage estimation rate satisfies $\alpha > 1/4$. This implies that the exponent of the remainder's second moment is $2\alpha > 1/2$. Consequently,

$$\lim_{n\to\infty} \mathbb{E}[\|\Delta_n\|^2] = \lim_{n\to\infty} O(n^{-2\alpha}) = 0.$$

Since the second moment of Δ_n converges to zero, by Markov's inequality, Δ_n converges in probability to zero. This completes the proof that the remainder term is asymptotically negligible. □

H.6 Asymptotic Normality and Efficiency

Theorem H.4 (Asymptotic Normality and Efficiency)**.** *Let $\widehat{\theta}$ be the cross-fitted NNES estimator. Under Assumptions 14–17,*

$$\sqrt{n}(\widehat{\theta} - \theta_0) \xrightarrow{d} \mathcal{N}\left(0, \mathcal{I}_{\theta\theta}^{-1}\right),$$

where $\mathcal{I}_{\theta\theta}$ is the Fisher information matrix. The estimator achieves the semiparametric efficiency bound.

Proof. The proof proceeds in six main steps. We first use a mean-value expansion of the estimator's first-order condition. Then, we analyze the convergence of the Jacobian and the asymptotic behavior of the score term separately. Finally, we assemble these results to derive the asymptotic distribution and establish efficiency.

Step 1: First-Order Condition and Mean-Value Expansion. The cross-fitted NNES estimator $\widehat{\theta}$ is an M-estimator that, by definition, solves the first-order condition $\nabla_\theta \ell_{\mathrm{CF}}(\widehat{\theta}; \{\widehat{\gamma}_{\widehat{\theta}}^{(-k)}\}) = 0$. Let the score for the full sample be denoted by:

$$\widehat{\Psi}(\theta) := \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in\mathcal{I}_k} \psi(W_i;\theta,\widehat{\gamma}_\theta^{(-k)}),$$

where $\psi(W;\theta,\gamma) = \nabla_\theta \log((\Lambda_\theta[V_\gamma])(A|X))$. Consistency of $\widehat{\theta}$ for θ_0 can be established under the given assumptions by showing uniform convergence of the objective function. Given $\widehat{\theta} \xrightarrow{p} \theta_0$, we perform a mean-value expansion of the first-order condition $\widehat{\Psi}(\widehat{\theta}) = 0$ around the true parameter θ_0:

$$0 = \widehat{\Psi}(\theta_0) + \nabla_\theta \widehat{\Psi}(\bar{\theta})(\widehat{\theta} - \theta_0),$$

where $\bar{\theta}$ is a mean-value point on the line segment between $\widehat{\theta}$ and θ_0, which implies $\bar{\theta} \xrightarrow{p} \theta_0$. Rearranging this expression yields the basis for our analysis:

$$\sqrt{n}(\widehat{\theta} - \theta_0) = -\left(\nabla_\theta \widehat{\Psi}(\bar{\theta})\right)^{-1} \sqrt{n}\widehat{\Psi}(\theta_0). \tag{71}$$

Step 2: Convergence of the Sample Jacobian Matrix. We show that the sample Jacobian, $J(\bar{\theta}) := \nabla_\theta \widehat{\Psi}(\bar{\theta})$, converges in probability to the negative of the Fisher Information matrix, $-\mathcal{I}_{\theta\theta}$. We decompose the difference $\|J(\bar{\theta}) + \mathcal{I}_{\theta\theta}\|$ using the triangle inequality:

$$\|J(\bar{\theta}) + \mathcal{I}_{\theta\theta}\| \leq \underbrace{\|J(\bar{\theta}) - J(\theta_0)\|}_{\text{(I): Error from consistency}} + \underbrace{\|J(\theta_0) - \mathbb{E}[\nabla_\theta \psi(W;\theta_0,\gamma_0^*)]\|}_{\text{(II): Error from estimation}} + \underbrace{\|\mathbb{E}[\nabla_\theta \psi(W;\theta_0,\gamma_0^*)] + \mathcal{I}_{\theta\theta}\|}_{\text{(III): Population Identity}}$$

- **Term (I):** The map $\theta \mapsto J(\theta)$ is continuous in a neighborhood of θ_0 under the smoothness conditions in Assumption 14. Since $\bar{\theta} \xrightarrow{p} \theta_0$, Term (I) converges to zero in probability by the Continuous Mapping Theorem.
- **Term (III):** This term is zero by the Information Matrix Equality. This equality holds because the score $\psi(W;\theta_0,\gamma_0^*)$ is the *efficient score* for the model. This is because γ_0^* yields the best approximation $V_{\gamma_0^*}$ to the true value function $V_{\theta_0}^*$, and the score of the correctly specified likelihood with the true nuisance function is efficient.
- **Term (II):** This term captures the error from using estimated nuisance parameters and from finite-sample variation. We show it converges to zero in probability by applying a Law of Large Numbers. First, by the triangle inequality:

$$\begin{aligned}
&\left\| \frac{1}{n} \sum_{k,i} \nabla_\theta \psi(W_i;\theta_0,\widehat{\gamma}_{\theta_0}^{(-k)}) - \mathbb{E}[\nabla_\theta \psi(W;\theta_0,\gamma_0^*)] \right\| \\
&\leq \underbrace{\left\| \frac{1}{n} \sum_{k,i} \left(\nabla_\theta \psi(W_i;\theta_0,\widehat{\gamma}_{\theta_0}^{(-k)}) - \nabla_\theta \psi(W_i;\theta_0,\gamma_0^*) \right) \right\|}_{\text{Term (IIa)}} + \underbrace{\left\| \frac{1}{n} \sum_{i} \nabla_\theta \psi(W_i;\theta_0,\gamma_0^*) - \mathbb{E}[\nabla_\theta \psi(W;\theta_0,\gamma_0^*)] \right\|}_{\text{Term (IIb)}}.
\end{aligned}$$

  Term (IIb) converges to zero in probability by the Weak Law of Large Numbers for i.i.d. data. For Term (IIa), we use the Lipschitz continuity of $\nabla_\theta \psi$ in γ (implied by primitive smoothness) and the uniform convergence rate of the nuisance estimators from Assumption 16(b):

$$\text{Term (IIa)} \leq \frac{1}{n} \sum_{k,i} L_{\nabla\psi} \|\widehat{\gamma}_{\theta_0}^{(-k)} - \gamma_0^*\|_2 \leq L_{\nabla\psi} \max_k \|\widehat{\gamma}_{\theta_0}^{(-k)} - \gamma_0^*\|_2 = O_p(n^{-\alpha}).$$

  Since $\alpha > 0$, Term (IIa) is $o_p(1)$. Thus, Term (II) converges to zero in probability.

Combining these results, $J(\bar{\theta}) \xrightarrow{p} -\mathcal{I}_{\theta\theta}$. Since matrix inversion is a continuous function, the Con-

tinuous Mapping Theorem implies $(J(\bar{\theta}))^{-1} \xrightarrow{p} -\mathcal{I}_{\theta\theta}^{-1}$.

Step 3: Asymptotic Representation of the Scaled Score. We invoke Lemma H.3, which provides the asymptotic expansion of the cross-fitted score at θ_0. The lemma's proof, based on Neyman orthogonality and cross-fitting, shows that the estimation error in the nuisance parameters $\widehat{\gamma}_{\theta_0}^{(-k)}$ is asymptotically negligible:

$$\sqrt{n}\widehat{\Psi}(\theta_0) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \psi(W_i; \theta_0, \gamma_0^*) + o_p(1).$$

Let $U_i := \psi(W_i; \theta_0, \gamma_0^*)$ denote the efficient score contribution from observation i.

Step 4: Assembling the Asymptotic Distribution. We now substitute the results from Step 2 (Jacobian convergence) and Step 3 (score expansion) into our mean-value expansion from Equation (71):

$$\sqrt{n}(\widehat{\theta} - \theta_0) = -\left(-\mathcal{I}_{\theta\theta}^{-1} + o_p(1)\right) \left(\frac{1}{\sqrt{n}} \sum_{i=1}^{n} U_i + o_p(1)\right).$$

We expand this product term by term. Let $S_n = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} U_i$, which is $O_p(1)$ by the Central Limit Theorem.

$$\begin{aligned}
\sqrt{n}(\widehat{\theta} - \theta_0) &= \left(\mathcal{I}_{\theta\theta}^{-1} - o_p(1)\right)(S_n + o_p(1)) \\
&= \mathcal{I}_{\theta\theta}^{-1} S_n + \mathcal{I}_{\theta\theta}^{-1} o_p(1) - o_p(1) S_n - o_p(1) o_p(1) \\
&= \mathcal{I}_{\theta\theta}^{-1} S_n + o_p(1),
\end{aligned}$$

since $o_p(1) S_n = o_p(1) O_p(1) = o_p(1)$.

Step 5: Applying the Central Limit Theorem. The efficient scores $\{U_i\}_{i=1}^n$ are i.i.d. random vectors.

- **Mean:** By the first-order condition for the population problem at the true parameters, $\mathbb{E}[U_i] = \mathbb{E}[\psi(W; \theta_0, \gamma_0^*)] = 0$.
- **Variance:** By definition in Assumption 17(c), the variance-covariance matrix is the Fisher Information matrix: $\mathrm{Var}(U_i) = \mathbb{E}[U_i U_i'] = \mathcal{I}_{\theta\theta}$.

By the multivariate Lindeberg-Lévy Central Limit Theorem, the normalized sum of the scores converges in distribution to a normal distribution:

$$S_n = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} U_i \xrightarrow{d} \mathcal{N}(0, \mathcal{I}_{\theta\theta}).$$

Step 6: Final Result and Semiparametric Efficiency. Finally, we apply Slutsky's Theorem to the expression derived in Step 4. Since $\mathcal{I}_{\theta\theta}^{-1} - o_p(1) \xrightarrow{p} \mathcal{I}_{\theta\theta}^{-1}$ and $S_n \xrightarrow{d} \mathcal{N}(0, \mathcal{I}_{\theta\theta})$, their product converges in distribution:

$$\sqrt{n}(\widehat{\theta} - \theta_0) = \mathcal{I}_{\theta\theta}^{-1} S_n + o_p(1) \xrightarrow{d} \mathcal{I}_{\theta\theta}^{-1} \cdot \mathcal{N}(0, \mathcal{I}_{\theta\theta}).$$

The variance of the resulting distribution is $\mathcal{I}_{\theta\theta}^{-1}\text{Var}(\mathcal{N}(0,\mathcal{I}_{\theta\theta}))(\mathcal{I}_{\theta\theta}^{-1})' = \mathcal{I}_{\theta\theta}^{-1}\mathcal{I}_{\theta\theta}(\mathcal{I}_{\theta\theta}^{-1})' = \mathcal{I}_{\theta\theta}^{-1}$. Thus,

$$\sqrt{n}(\widehat{\theta} - \theta_0) \xrightarrow{d} \mathcal{N}\left(0, \mathcal{I}_{\theta\theta}^{-1}\right).$$

The influence function of the estimator $\widehat{\theta}$ is $\mathcal{I}_{\theta\theta}^{-1}U_i = \mathcal{I}_{\theta\theta}^{-1}\psi(W_i;\theta_0,\gamma_0^*)$. As established, $\psi(W_i;\theta_0,\gamma_0^*)$ is the efficient score for the structural parameter θ. Therefore, the asymptotic variance of our estimator, $\mathcal{I}_{\theta\theta}^{-1}$, achieves the semiparametric efficiency bound. This concludes the proof. □