Bridging Structured Knowledge and Data: A Unified Framework with Finance Applications

Yi Cao* Zexun Chen† Lin William Cong‡ Heqing Shi§

First draft: Dec 2025; current draft: Feb 2026.

Abstract

We develop Structured-Knowledge-Informed Neural Networks (SKINNs), a unified estimation framework that embeds theoretical, simulated, previously learned, or cross-domain insights as differentiable constraints within flexible neural function approximation. SKINNs jointly estimate neural network parameters and economically meaningful structural parameters in a single optimization problem, enforcing theoretical consistency not only on observed data but over a broader input domain through collocation, and therefore nesting approaches such as functional GMM, Bayesian updating, transfer learning, PINNs, and surrogate modeling. SKINNs define a class of M-estimators consistent and asymptotically normal with $\sqrt{N}$ convergence, sandwich covariance, and recovery of pseudo-true parameters under misspecification. We establish identification of structural parameters under joint flexibility, derive generalization and target-risk bounds under distributional shift in a convex proxy, and provide a restricted-optimal characterization of the weighting parameter that governs the bias–variance tradeoff. In an illustrative financial application to option pricing, SKINNs improve out-of-sample valuation and hedging performance, particularly at longer horizons and during high-volatility regimes, while recovering economically interpretable latent parameters with improved stability relative to conventional calibration. More broadly, SKINNs provide a general econometric framework for combining model-based reasoning with high-dimensional, data-driven estimation.

Keywords: AI, Bayesian, Deep Learning, Derivative Pricing, Econometrics, Machine Learning.
JEL Codes: C45, G13

*Department of Financial and Actuarial Mathematics, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu 215123, PR China. Email: Yi.Cao@xjtlu.edu.cn

†Management Science and Business Economics Group, Business School, University of Edinburgh, 29 Buccleuch Place, Edinburgh EH8 9JS, UK. Email: Zexun.Chen@ed.ac.uk

‡Nanyang Technological University, ABFER, CEPR, & NBER, 91 Nanyang Avenue, Wee Cho Yaw Plaza, Singapore 639956. Email: Will.Cong@ntu.edu.sg

§Management Science and Business Economics Group, Business School, University of Edinburgh, 29 Buccleuch Place, Edinburgh EH8 9JS, UK. Email: Heqing.Shi@ed.ac.uk

1 Introduction

The last decade has witnessed a paradigm shift toward data-driven modelling across scientific and industrial domains, thanks to the rapid advancements in artificial intelligence (AI). From Large Language Models (LLMs) exhibiting human-like fluency to deep learning algorithms achieving expert-level medical diagnoses, the success of these empirical methods has fostered a compelling narrative: sufficient data and computation allow learning complex patterns directly from observations, bypassing or even rendering obsolete theories, traditional econometric/reduced-form models, and knowledge-based conceptual frameworks.[1] Yet many researchers, especially economists, would resist such a notion: Data-driven models are vulnerable to noise and spurious temporal correlations (Harvey et al., 2016), a fragility particularly acute in settings characterized by low signal-to-noise ratios, such as finance (e.g., Gu et al., 2020). The lack of transparent mechanisms, model interpretability, and domain intuition further frustrates social scientists, not to mention the high computational costs associated with large models and AI algorithms.

Traditional theory and econometric approaches indeed help researchers understand a wide range of underlying principles, causal relationships, and economic mechanisms. But their over-simplified assumptions (e.g., market efficiency, perfect rationality, free of arbitrage, etc.), rigid model structures, and over-reliance on human expertise fail to capture the full complexity of real-world phenomena, or the high-dimensionality, non-stationarity, and nonlinearity of modern datasets (Cochrane, 2011; Cong et al., 2019), leading to significant performance degradation and under-adoption by practitioners. This dichotomy, between principled but idealized theory and flexible but fragile empiricism, underscores a critical gap and pressing need in current methodologies in both science and social science inquiries.

To this end, we propose Structured-Knowledge-Informed Neural Networks (SKINNs), a framework that integrates flexible neural function approximation with structured domain knowledge (generically referred to as "theory"). SKINNs jointly estimate the parameters of a deep neural network and a structured theoretical module within a single optimization problem, allowing each component to mitigate the other's misspecification rather than fixing theory parameters ex ante or estimating them sequentially. Estimation proceeds via a composite loss that balances data fidelity and structural consistency, with theoretical restrictions enforced not only at observed data points but over a broader input domain through collocation, thereby preserving economic structure under data sparsity and distribution shift. The framework accommodates high- and infinite-dimensional latent objects (e.g., distributions or state spaces) for which conventional GMM, Bayesian, or transfer-

[1] This viewpoint, famously summarized as the "End of Theory" (Anderson, 2008), suggests that when data can "speak for itself," the scientific method itself may become dispensable.

learning approaches become computationally infeasible. We show that SKINNs define a class of M-estimators: under standard regularity conditions, the joint estimator of neural and structural parameters is consistent and asymptotically normal with $\sqrt{N}$ convergence and sandwich-form covariance, and under misspecification the structural parameters converge to pseudo-true values. Consequently, the learned structural parameters retain economic interpretation and support formal statistical inference, bridging modern machine learning and classical econometrics.

The econometric analysis extends well beyond standard consistency and asymptotic normality. We address three questions that arise naturally from the SKINNs formulation: whether the structural parameters remain identifiable when the neural component is highly flexible, how structured regularization improves generalization under distributional shift, and what governs the optimal balance between data fit and theoretical coherence. To answer them, we establish an identification-via-profiling result, derive stability-based generalization bounds, and characterize the regularization weight in closed form within a GMM interpretation. Together, these results show that the framework provides formal inferential guarantees rather than serving as an ad-hoc regularization device.

To illustrate the empirical relevance of the framework, we apply SKINNs to option pricing, a setting in which theoretical structure is well-developed yet frequently misspecified, and purely data-driven methods often struggle under distributional shifts. Using over two decades of S&P 500 index option data, we show that SKINNs improve out-of-sample pricing accuracy and delta-hedging performance relative to both flexible neural networks and classical structural models, particularly at longer prediction horizons and during high-volatility periods. The gains are modest when market conditions are stable but become economically meaningful when volatility rises. Moreover, the latent parameters embedded in g_ϕ evolve in economically coherent ways and exhibit improved numerical stability relative to conventional calibration. These findings suggest that jointly integrating structured knowledge with flexible function approximation can enhance generalization while preserving economically interpretable latent structure.

Methodological innovations and contributions. SKINNs introduce three methodological innovations that distinguish them from existing hybrid modeling approaches. First, they jointly estimate the parameters of both the data-driven model and the structured theoretical model within a single optimization problem. Unlike approaches that require separate calibration of theoretical parameters (e.g., Raissi et al. (2019)) or sequential pre-training and fine-tuning as in conventional transfer learning (Chen et al. (2023)), SKINNs optimize the neural network parameters and the structured-knowledge parameters simultaneously through a unified objective. The composite loss balances empirical fit with theoretical consistency, and gradients are computed with respect to

both parameter sets and updated concurrently via backpropagation. This co-adaptation allows the neural component to capture rich empirical patterns while remaining disciplined by theory, and permits the theoretical parameters to adjust in light of the data, avoiding both static calibration and purely empirical black-box estimation.

Second, SKINNs accommodate a broad spectrum of structured knowledge representations. The theoretical module may consist of closed-form analytical solutions (e.g., Black–Scholes), surrogate neural approximations to computationally intensive structural models, or abstract constraints defining admissible functions or distributions nonparametrically. The only requirement is differentiability with respect to inputs and learnable parameters, ensuring compatibility with gradient-based optimization. This flexibility enables SKINNs to operate across domains with varying degrees of theoretical structure, from settings with well-established parametric models to those where theory is expressed through equilibrium restrictions, simulation procedures, or distributional constraints.

Third, SKINNs are designed for structural parameter recovery as well as prediction. While most machine learning models focus exclusively on predictive accuracy, SKINNs simultaneously estimate economically interpretable parameters—such as implied volatilities, risk-aversion coefficients, or risk-neutral distributions—that characterize underlying mechanisms. These parameters are learned jointly with the flexible function approximator and inherit formal inferential properties from the M-estimation framework developed in Section C. As a result, SKINNs transform flexible approximation into structurally interpretable estimation, enabling statistical inference on economically meaningful quantities rather than merely improving forecast performance.

SKINNs are closely related to the generalized method of moments (GMM; Hansen, 1982), Bayesian posterior estimation, transfer learning (Chen et al., 2023), and physics-informed neural networks (PINNs; Raissi et al., 2017a,b), but differ in important respects. Like GMM, SKINNs combine empirical fit with theory-implied restrictions; however, they jointly learn both the structural parameters and a flexible function, and enforce theoretical consistency not only at observed data points but over a broader input domain through collocation. From a Bayesian perspective, SKINNs resemble maximum a posteriori estimation with theory-based regularization, but unlike standard Bayesian procedures that impose a fixed ex ante prior distribution over structural parameters, SKINNs determine both the structural parameters and the predictive function jointly through a single optimization problem. The regularization induced by theory is therefore endogenous to the data-fitting objective rather than anchored to a fixed prior specification. Unlike PINNs, which impose differential equations through high-order derivatives and often suffer from gradient pathologies, SKINNs enforce consistency with model solutions—parametric, semi-parametric, or non-parametric—allowing scalable estimation even in the presence of unobservable state variables

and high-dimensional latent structures.

Applications and empirical findings. We illustrate the framework in option pricing, a canonical setting in which (i) strong structural restrictions are available but routinely misspecified, (ii) purely data-driven methods can fit well in-sample yet degrade under distribution shifts, and (iii) "theory" naturally comes in multiple differentiable forms—from closed-form solutions to simulation-based models and nonparametric no-arbitrage restrictions. This makes option pricing a demanding laboratory for evaluating whether structured regularization can improve generalization while preserving interpretability of latent economic objects.

Empirically, we instantiate SKINNs with three classes of structured-knowledge modules g_ϕ. In the parametric class we consider Black–Scholes (one latent parameter), an ad-hoc Black–Scholes specification with strike–maturity dependent volatility (six parameters), Heston' s stochastic volatility model (five parameters), an extension with jumps (nine parameters), and a deliberately high-dimensional SABR-type specification (722 parameters). In the semi-parametric class, we approximate computationally intensive models by differentiable deep surrogates as in Chen et al. (2021), including Heston and a non-affine stochastic volatility model (six parameters). In the nonparametric class, we implement a martingale option pricing approach with a high-dimensional risk-neutral distribution (2,000 parameters) and an autoencoder-based representation (two parameters). We benchmark against plain neural networks (with and without boundary conditions), shape-constrained networks enforcing monotonicity/convexity (via derivative penalties and quadratic programming), forward- and inverse-formulation PINNs (Raissi et al., 2017a,b), and transfer-learning networks (Chen et al., 2023). These alternatives span purely data-driven methods and existing theory-hybrid designs, but either impose only static model-free constraints or become fragile when theoretical structure is high-dimensional or involves unobservable state variables.

Our empirical analysis uses a comprehensive dataset of S&P 500 index options spanning over two decades, on which SKINNs exhibit superior performance across two critical dimensions: out-of-sample option pricing accuracy and delta-hedging effectiveness. We use daily transaction quotes of S&P 500 index options from OptionMetrics, covering the period from January 4, 1996, to December 31, 2022. This extensive 27-year timespan encompasses several major economic crises, including the dot-com bubble, the 2007-2009 global financial crisis, and the COVID-19 pandemic, providing a rigorous stress test for all models under varying market conditions. We utilize raw option data that reflects actual market conditions, including missing values and erroneous prices that challenge neural network models. We focus on call options with maturities between 7 and 365 calendar days to ensure liquidity and information content. Using a forward rolling-window approach, we train

models on three-month panels of option data and evaluate their out-of-sample performance over two consecutive months: a shorter prediction horizon (one month ahead) and a longer prediction horizon (two months ahead). This rolling procedure generates 317 training and testing periods, with the longer horizon presenting significantly greater challenges due to potential shifts in the data-generating process relative to the training sample.

First, we evaluate all models across the 317 rolling periods using the Diebold and Mariano (2002) test to assess the statistical significance of performance differences. For shorter horizons, NN demonstrates competitive performance, statistically outperforming models with gradient pathology issues as well as structural models. This suggests that data-driven learning effectively captures latent option price patterns when the testing data closely resembles the training sample. However, only SKINNs incorporating sophisticated structured-knowledge achieve statistically significant improvements over NN, indicating that marginal gains require more advanced knowledge representations when patterns are familiar. For longer prediction horizons, where the data-generating process may diverge substantially from the training sample, the limitations of pure data-driven approaches become apparent. The NN model underperforms most structural models, as previously learned patterns become less effective. In contrast, almost all SKINN variants significantly outperform both NN and NN with boundary conditions at the 1% significance level, demonstrating substantially enhanced generalizability through the incorporation of structured-knowledge. This improvement is particularly noteworthy given that longer-horizon predictions pose considerably greater challenges due to potential shifts in market dynamics across different economic regimes encountered in our dataset. Overall, SKINNs reduce out-of-sample pricing errors (measured in terms of the root mean square error, RMSE) by roughly 10–15% relative to leading neural network benchmarks and by substantially more in high-volatility regimes.

Furthermore, we assess the delta-hedging capability of the modelsby constructing hedged portfolios and measuring next-day hedging errors across all prediction periods. Delta ratios for neural networks are computed using automatic differentiation, exploiting the homogeneity property of option pricing functions. NN becomes less accurate in this exercise. In both shorter and longer horizons, NN and NN with boundary conditions are significantly outperformed by all SKINN variants. They only manage to outperform those gradient-challenged neural network models, and, interestingly, some sophisticated structural models (e.g., the Heston model and its extension with jumps). This suggests that the static model-free constraints fail to enhance the quality of price predictions as well as the Delta ratio predictions. Additionally, sophisticated structural models tend to underperform simpler ones, such as the Black-Scholes, for the delta-hedging purpose. This is also reflected in SKINNs, as while all SKINN variants consistently outperform the benchmark neural

networks, simpler structured-knowledge representations generally deliver superior delta-hedging performance. These consistent improvements across both pricing and hedging tasks, across different prediction horizons, and through multiple economic crisis periods, underscore the practical value and robustness of incorporating structured knowledge into neural network architectures for financial derivative applications.

We next investigate the economic mechanisms underlying SKINNs' outperformance, focusing on two related properties: countercyclical performance during volatile market conditions and the economic interpretability of the latent parameters embedded in g_ϕ.

Our predictive regressions indicate that the relative pricing accuracy of SKINNs improves when market volatility increases. Across the 317 rolling training periods, a one-unit increase in the average VIX is associated with reductions of approximately 0.09–0.10 in RMSE for SKINNs, compared to roughly 0.03 for boundary-constrained neural networks and statistically insignificant effects for classical structural models. The differential becomes more pronounced in high-volatility regimes: during periods above the 80th percentile of average daily VIX, SKINNs exhibit statistically significant RMSE reductions of 0.14–0.15 at the 5% level, whereas benchmark models do not display comparable improvements. In contrast, during low-volatility periods, when option price surfaces are relatively stable and patterns are easier to learn, performance differences across models are modest. Taken together, these results suggest that the benefits of incorporating structured knowledge through g_ϕ are most apparent when market conditions are unstable and distributional shifts are more likely. In the highest-volatility quintile, SKINNs achieve pricing-error reductions that are approximately three to four times larger than those of boundary-constrained neural networks, while also delivering significantly improved delta-hedging performance.

We further examine the latent economic parameters learned within g_ϕ. Unlike transfer-learning approaches that require separate calibration, SKINNs estimate network parameters and structured-knowledge parameters jointly within a unified optimization. In the one-dimensional Black–Scholes case, the SKINN-learned volatility closely tracks conventional implied volatility estimates. In higher-dimensional settings, the learned state variables display smoother time-series evolution and align with identifiable economic regimes, often exhibiting greater numerical stability than parameters obtained via standalone calibration. Even in the non-parametric martingale option pricing implementation, where g_ϕ contains 2,000 probability parameters, the implied risk-neutral density evolves in economically interpretable ways across market episodes. These patterns indicate that the latent parameters are not merely auxiliary regularization devices, but capture economically meaningful information consistent with market conditions.

Beyond option pricing, the SKINNs framework applies naturally to a broad class of economic

estimation problems. In production function estimation, g_ϕ can encode technological restrictions —such as monotonicity, returns to scale, or equilibrium implications of firm optimization—while f_θ flexibly captures heterogeneous productivity dynamics. In demand estimation, SKINNs can combine flexible demand systems with utility-based or revealed-preference restrictions, allowing simultaneous recovery of demand functions and economically interpretable parameters. In dynamic discrete choice models, Bellman or Euler conditions can be embedded within g_ϕ, enabling joint estimation of value functions and structural parameters without repeatedly solving the dynamic program. In macroeconomic applications, equilibrium conditions from DSGE models—such as Euler equations, resource constraints, or policy rules—can serve as structured knowledge while neural components approximate high-dimensional policy functions or shock processes. In each case, SKINNs provide a unified econometric framework for jointly estimating flexible functional relationships and economically meaningful latent structure under theory-imposed discipline.

The remainder of this paper is organized as follows: Section 2 introduces the SKINNs framework, proves its statistical properties, and discusses the methodological connections to well-established econometric models and existing approaches that integrate data-driven learning with prior knowledge. Section 3 provides various specifications of option pricing structured-knowledge representation for SKINNs. Section 4 and 5 present the empirical findings, comparing the predictive performance of SKINNs against alternative methods and extracting economic insights from the learned theoretical parameters. Section 6 concludes.

2 The SKINNs Framework: Methodology

The SKINNs framework is designed to synergize the predictive power of data-driven models with the explanatory rigor of theory, where the term "theory" should be broadly interpreted as reduced-form representations of prior knowledge. Purely data-driven models struggle with noisy, non-stationary environments where they are prone to overfitting and distributional shifts, while traditional theory-based approaches often rely on rigid, oversimplified assumptions and ad-hoc calibration processes that fall short of capturing complex market dynamics.

SKINNs aim to bridge this gap using a principled mechanism for embedding a theoretical model's structure directly into the learning objective of a neural network. This is achieved through a composite loss function that allows the joint estimation of the neural network's parameters and the latent parameters of the embedded theoretical model. By doing so, SKINNs regularize the learning process, guiding the neural network towards solutions that are not only empirically accurate but also theoretically plausible. This approach generalizes and addresses key limitations of

existing hybrid methods, such as PINNs (Raissi et al., 2017a,b)—which suffer from spectral biases, gradient pathologies, and unobservable differentiation variables (see, e.g., Wang et al., 2020, 2022)—, and transfer learning (e.g., Pratt, 1992), which lacks a mechanism for dynamic latent parameter discovery in its two-stage design.

The architecture of SKINNs is composed of two core components: a data-driven function approximator and a structured-knowledge representative informed by theory. These components are trained in concert to reconcile empirical observations with theoretical principles.

2.1 The Data-Driven Component

Let the primary learning model be a deep neural network, $f(\mathbf{X};\theta) : \mathbb{R}^d \mapsto \mathbb{R}^v$, which maps a set of input features $\mathbf{X} \in \mathbb{R}^d$ to some target outputs $\mathbf{y} \in \mathbb{R}^v$.[2] The vector θ represents the full set of trainable neural network parameters, i.e., the biases and weights. For instance, a standard multi-layer feed-forward neural network is defined by the recursive formulation:

$$f^{(l)} = h\left(\mathbf{b}^{(l-1)} + \mathbf{W}^{(l-1)} f^{(l-1)}\right), \quad l = 1, \ldots, L, \tag{1}$$

where $f^{(0)}$ is the input layer; $f^{(l)}$ is each of the L hidden layers; $f^{(L+1)} = \mathbf{b}^{(L)} + \mathbf{W}^{(L)} f^{(L)}$ is the final output layer; $h(\cdot)$ is a non-linear activation function, e.g., ReLU where $h(x) = max(x, 0)$; and $\theta = \{\mathbf{b}^{(l)}, \mathbf{W}^{(l)}\}_{l=1}^{L}$ contains all the trainable neural network parameters.

The neural network parameters, θ, are traditionally optimized by minimizing a data-centric loss function, $\mathcal{L}_{\text{data}}$, with respect to a set of observations $\mathcal{D}_{\text{obs}} = \{(\mathbf{X}_{\text{obs}}^{[i]}, \mathbf{y}_{\text{obs}}^{[i]})\}_{i=1}^{N}$. A common choice is the mean squared error:

$$\mathcal{L}_{\text{data}}(\theta; \mathcal{D}_{\text{obs}}) = \frac{1}{N} \sum_{i=1}^{N} \left(f_\theta(\mathbf{X}_{\text{obs}}^{[i]}) - \mathbf{y}_{\text{obs}}^{[i]}\right)^2. \tag{2}$$

This loss function, which serves as the data-driven component, anchors the model to the empirical evidence provided by the data. However, with the data-driven component alone, the model will reduce to a plain-vanilla neural network, with the drawbacks of being a black box, prone to overfitting, etc.

Yet the limitations are inherent, plain-vanilla neural networks are still flexible statistical models that help to relax the rigid assumptions and fixed specifications in many disciplines, particularly economics. For example, economists usually describe an economic system using a regression model

[2] In this paper we consider the $v = 1$ case, i.e., the target output y is one-dimensional. It is straightforward to extend SKINNs for multi-dimensional outputs ($v \geq 2$, where $v \subseteq \mathbb{N}^+$), as long as the structure-knowledge representation allows for producing outputs of the same dimension. In the $v \geq 2$ cases, the problem becomes a multi-task learning problem.

of a general form: $\mathbf{y} = \mathbb{E}[\mathbf{y}|\mathbf{X}] + \varepsilon$, where $\mathbf{y}$ is the target economic variable, $\mathbf{X}$ is a set of conditioning variables (covariates) in the conditional expectation $\mathbb{E}[\cdot|\mathbf{X}]$, and ε denotes a vector of residuals. Restrictions such as the linear conditional expectation and the low-dimensional conditioning variables are commonly imposed on the structure of the conditional expectation, which can fail to reconcile the true data-generating process due to model misspecifications and omitted variables. By approximating the conditional expectation with a neural network, $\mathbb{E}[\mathbf{y}|\mathbf{X}] = f(\mathbf{X}; \theta)$, it provides greater flexibility and therefore stronger expressive power. Such applications can be seen in the recent asset pricing studies (Cong et al., 2019; Gu et al., 2020; Chen et al., 2024; Feng et al., 2024).

2.2 The Structured-Knowledge Component

To better tame the over-parameterized neural networks, the second component in SKINNs is a formal representation of structured knowledge, denoted by a function $g(\mathbf{X}^{\text{SK}}; \phi) : \mathbb{R}^{d_{\text{SK}}+d_\phi} \to \mathbb{R}$, where $\mathbf{X}^{\text{SK}}$ is the input features that are observable; ϕ is the latent parameters that are unobservable; and d_{SK}, d_ϕ are their respective dimensions. Typically, $\mathbf{X}^{\text{SK}}$ is a subset of $\mathbf{X}$ (hence, the dimension of $\mathbf{X}^{\text{SK}}$ is less than the dimension of $\mathbf{X}$, i.e., $d_{\text{SK}} \leq d$). $\mathbf{X}^{\text{SK}}$ therefore preserves a model-driven low-dimensional structure, while $\mathbf{X} \in \mathbb{R}^d$, the inputs of the neural network component, can incorporate higher-dimensional empirical features that are possibly omitted. This function g_ϕ encapsulates the structure of theories in a certain domain, mapping the theoretically-relevant inputs $\mathbf{X}^{\text{SK}} \subseteq \mathbf{X}$ to a model-driven estimate. SKINNs embed the structured-knowledge by treating the vector ϕ from the theoretical model g_ϕ as a set of learnable latent parameters additional to the neural network parameters θ.

Such a joint learning mechanism by SKINNs facilitates a co-evolutionary process of both components, which offers improved model flexibility and efficiency over many two-step hybrid approaches (for example, see some recent works in Chen et al., 2023; Zhang et al., 2023). The two sets of learnable parameters—one includes the nuisance neural network parameters and the other includes the interpretable latent economic parameters—make SKINNs also semi-parametric-model alike. However, distinct from the classic semi-parametric econometric models, where both g_ϕ and f_θ contribute to the estimates as seen in the equation (1.1) in Chernozhukov et al. (2018), g_ϕ is only employed in SKINNs to regularize the training of f_θ that ultimately produces estimates. More importantly, the two parameter sets are simultaneously identified by SKINNs instead of sequentially, as classic semi-parametric models do. Hence, the data-driven component compensates for the structured-knowledge component in the same learning process. This naturally mitigates the overfitting impact that the overparameterized f_θ can have on the estimation of the structured-knowledge parameters ϕ, as discussed in Chernozhukov et al. (2018).

The structured-knowledge function g_ϕ admits diverse formats, as long as it remains first-order differentiable with respect to both its inputs $\mathbf{X}^{\text{SK}}$ and the learnable latent parameters ϕ.[3] One advantage of this is that the latent parameters ϕ can be scaled to high- and even infinite-dimensional, thanks to the first-order optimization methods that are suitable for large-scale problems associated with neural networks. These flexibilities allow SKINNs to embed a wide range of theoretical, empirical, and learned structures seamlessly into the learning process, as well as to learn the high-dimensional parameters in theoretical models that are hard to estimate traditionally. We categorize the formats of structured-knowledge into three main types: parametric representations, semi-parametric representations, and non-parametric representations.

2.2.1 Parametric Knowledge Representations

Parametric representations are the most direct methods of embedding established theories, as the prevalence of many established theories stems from their well-posed parametric equations. For parametric representations, the number of latent parameters and the functional form of the equation are expected to be known exactly. Therefore, in this format, we are allowed to conveniently define a structured-knowledge function g_ϕ by a closed-form or semi-closed-form mathematical expression derived directly from theory.[4] A canonical example from finance is the Black and Scholes (1973) option pricing model, where the theoretically-relevant inputs $\mathbf{X}^{\text{SK}}$ would consist of the observable characteristics such as the underlying asset price (S), strike price (K), risk-free rate (r), and time-to-maturity ($T-t$, or τ in short, where t is the current time); the latent parameter vector ϕ would contain the unobservable parameter, namely the asset return volatility, σ. The function g_ϕ is thus explicitly defined as: $g_\phi(\mathbf{X}^{\text{SK}}) \equiv \text{BlackScholes}(\mathbf{X}^{\text{SK}}; \phi = \{\sigma\})$.

In the cases where the theory-driven models admit only semi-closed-form solutions, for example the Heston (1993) model, the function $g_\phi(\mathbf{X}^{\text{SK}}) \equiv \text{Heston}(\mathbf{X}^{\text{SK}}, \mathbf{u}; \phi)$ requires numerical methods such as the fast Fourier transform (FFT) and the Fourier-Cosine series expansions (COS), with $\mathbf{u}$ denotes a vector algorithmic parameters, e.g., the grid points in the frequency domain to evaluate the Fourier transform, and the upper or lower limit of an integral, that are determined in advance.

The central innovation of our framework is that ϕ is not pre-calibrated or fixed, but is treated as a learnable parameter vector. During training, the total loss of a SKINN, which is a function of

[3] Such differentiability ensures the structured-knowledge component can be effectively integrated in the forward-pass and the backward-propagation of the neural network training process, and can be conveniently implemented using modern deep learning libraries, e.g., PyTorch.

[4] We classify the theory-driven models with no perfect closed-form solutions but imperfect semi-closed-form solutions as parametric. Although these solutions are non-trivial and more complicated than the closed-form ones, their functional forms are almost determined except for some algorithmic parameters to ensure numerical precision, and the number of latent parameters is fixed. There are a number of examples in finance, e.g., the stochastic volatility models, and some consumption-based asset pricing models such as Bansal and Yaron (2004).

both the data-driven component $f_\theta(\mathbf{X})$ and the structured-knowledge-driven component $g_\phi(\mathbf{X}^{\text{SK}})$, is minimized by jointly updating the neural network parameters θ and the latent theoretical parameters ϕ via gradient descent. The gradients with respect to the latent theoretical parameters, $\frac{\partial \mathcal{L}}{\partial \phi}$, are computed via the chain rule through the parametric structured-knowledge function g_ϕ. The availability of automatic differentiation in modern deep learning libraries makes this seamless, provided g_ϕ is first-order differentiable with respect to ϕ.[5] This approach allows the SKINN framework to discover the latent theoretical parameters (e.g., implied volatility) that best reconcile the theory-driven model with the empirical data, yielding highly interpretable results with superior predictive power. Figure (1) visualizes the parametric structured-knowledge representations.

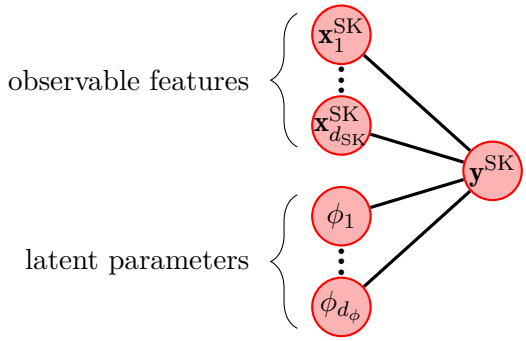


Figure 1: Parametric format structured-knowledge representation. The thick lines connecting the inputs $\{\mathbf{X}^{\text{SK}}, \phi\}$ to the structured-knowledge output $\mathbf{y}^{\text{SK}}$ mean that the mapping $\mathbf{y}^{\text{SK}} = g_\phi(\mathbf{X}^{\text{SK}})$ is known exactly thanks to the theory, possibly with some algorithmic parameters $\mathbf{u}$ if the theory-driven model admits only a semi-closed-form solution. $\mathbf{x}_1^{\text{SK}} \cdots \mathbf{x}_{d_{\text{SK}}}^{\text{SK}}$ denote each of the observable features, and $\phi_1 \cdots \phi_{d_\phi}$ denote each of the latent parameters.

2.2.2 Semi-Parametric Knowledge Representations

In more complex systems, closed-form or even semi-closed-form solutions are intractable or unavailable. Theories may be expressed through mechanistic simulators (e.g., agent-based models), complex stochastic differential equations (SDEs), or high-dimensional partial differential equations (PDEs), for which the system can only be solved via iterated computationally expensive numerical procedures, including, for example, Monte-Carlo simulation, or finite-difference, as there is no exact knowledge about the functional form of $g_\phi(\mathbf{X}^{\text{SK}})$.

SKINNs solve this through building a pre-trained deep surrogate neural network (DSNN) that learns the mapping $\{\mathbf{X}^{\text{SK}}, \phi\} \mapsto \mathbf{y}^{\text{SK}}$ that is unavailable from theory. In this scenario, theory-driven models still take the observable characteristics $\mathbf{X}^{\text{SK}}$ and the fixed number of latent parameters ϕ as the inputs, but the output $\mathbf{y}^{\text{SK}}$ is solved entirely by numerical procedures. In other words, a DSNN

[5] Since the total loss of a SKINN, $\mathcal{L}(\theta, \phi|\cdot)$, is a function of both θ and ϕ, chain rule leads to an efficient computation of the gradients w.r.t. ϕ, as $\partial_\phi \mathcal{L}(\theta, \phi|\cdot) := \partial_g \mathcal{L}(\theta, \phi|\cdot) \times \partial_\phi g(\cdot)$.

is employed to parametrize the unknown function, $g_\phi(\mathbf{X}^{\text{SK}}) := \{\mathbf{X}^{\text{SK}}, \phi\} \mapsto \mathbf{y}^{\text{SK}}$, non-parametrically with a deep neural network, thanks to its universal approximation capability (see, Cybenko, 1989; Hornik et al., 1989; Hanin, 2019).

In engineering, physics, biomedicine, and particularly finance, their underlying complex systems are often prescribed as multivariate SDEs. When the SDE system is in the affine class, in which the drift terms, diffusive variances, and jump intensity are functionally affine in the state vector, there is an analytical treatment that allows the system to be solved with a semi-closed-form solution, and therefore, we are able to construct a parametric structured-knowledge representation as for the Heston model (Duffie et al., 2000; Freire and Vladimirov, 2023). However, in many unfavoured settings, the SDE system is non-affine, and we are only able to approximate the solutions with Monte-Carlo simulation and Euler-Maruyama discretization. For example, in finance, asset prices can be driven by the following multivariate SDEs (Kaeck and Alexander, 2012):

$$dS_t = rS_t dt + \sqrt{v_t} S_t dW_t^{(S)} + (e^{\xi_t^{(S)}} - 1) S_t dN_t^{(S)}, \tag{3}$$

$$dv_t = \kappa_v(m_t - v_t)dt + \sigma_v v_t^{\gamma_v} dW_t^{(v)} + \xi_t^{(v)} dN_t^{(v)}, \tag{4}$$

$$dm_t = \kappa_m(\theta_m - m_t)dt + \sigma_m m_t^{\gamma_m} dW_t^{(m)}, \tag{5}$$

where S_t, v_t, m_t are the stochastic process for the asset price, instantaneous return variance and the long-term average of the instantaneous return variance; $W_t^{(S)}, W_t^{(v)}, W_t^{(m)}$ are the Brownian motions drive the stochasticity of each of the three state variables, possibly with correlations; $N_t^{(S)}$ and $N_t^{(v)}$ are the Poisson jump processes for the price and the instantaneous return variance. There are 10 latent parameters, and even more if the additional parameters in the jump processes are further switched on.[6]

To train a DSNN, one is required to randomly sample a large number of input instances for the complex economic model, $\left(\mathbf{x}_{\text{SK}}^{[i,1]}, \cdots, \mathbf{x}_{\text{SK}}^{[i,d_{\text{SK}}]}, \phi^{[i,1]}, \cdots, \phi^{[i,d_\phi]}\right), 1 \leq i \leq N_{\text{surrogate}}$, from a multivariate uniform distribution, and then query the corresponding numerical solution of the complex economic model, $\tilde{\mathbf{y}}_{\text{SK}}^{[i]}$.[7] The simulated data space $\mathcal{D}_{\text{surrogate}} = \{[\mathbf{X}_{\text{SK}}^{[i]}, \phi^{[i]}], \tilde{\mathbf{y}}_{\text{SK}}^{[i]}\}_{i=1}^{N_{\text{surrogate}}}$ is used

[6] In practice, the risk-free rate r is observable from the market. In the rich SDEs, the latent parameters include: $\kappa_v, \sigma_v, \gamma_v, v_0, \kappa_m, \theta_m, \sigma_m, \gamma_m, m_0$, and the correlation ρ between asset price and the instantaneous return variance. λ_S, λ_v refer to the latent parameters when the asset price jump process $N_t^{(S)}$, or the instance return variance jump process $N_t^{(v)}$, is activated. $\xi_t^{(S)}, \xi_t^{(v)}$ refer to the random variables that decide the size of the jumps, which can even take some further latent parameters. We refer the interested readers to Kaeck and Alexander (2012) for the detailed interpretations of the 10 basic latent parameters.

[7] The number of randomly sampled inputs instances, $N_{\text{surrogate}}$, should be a sufficiently large integer, for example, 50 millions, as long as the computational capacity allows. The notation $\tilde{\mathbf{y}}_{\text{SK}}^{[i]}$ is to address that each of the numerically solved outputs may contain numerical error. $\tilde{\mathbf{y}}_{\text{SK}}^{[i]}$ is solved using the Monte-Carlo method with iterative runs of simulations, and we use 1,000 runs in our implementation. When sampling the input instances, a reasonable proportion should be given to the partial derivatives against ϕ, in order to also well approximate the complex economic model's

for training a DSNN off-line, with the objective of minimizing an approximation loss, as much as possible, without the concern of overfitting. We minimize the mean squared error:

$$\mathcal{L}_{\text{surrogate}}(\theta; \mathcal{D}_{\text{surrogate}}) = \sum_{i=1}^{N_{\text{surrogate}}} \left(f_\theta^{\text{surrogate}} \left(\mathbf{X}_{\text{SK}}^{[i]}, \phi^{[i]} \right) - \tilde{\mathbf{y}}_{\text{SK}}^{[i]} \right)^2 \Big/ N_{\text{surrogate}}. \tag{6}$$

Once the desired approximation accuracy is achieved during the off-line pre-training, we create a neural network shortcut $f_\theta^{\text{surrogate}}$ of the complex economic model that can be used repeatedly.

We then freeze all its biases and weights, but keep the latent parameters ϕ in its input layer as the learnable parameters when integrating it into a SKINN. In this case, the representation function g_ϕ becomes an another deep neural network, i.e., $g_\phi(\mathbf{X}^{\text{SK}}) \equiv f_\theta^{\text{surrogate}}(\mathbf{X}^{\text{SK}}, \phi)$. We visualize such semi-parametric structured-knowledge representations in Figure (2).

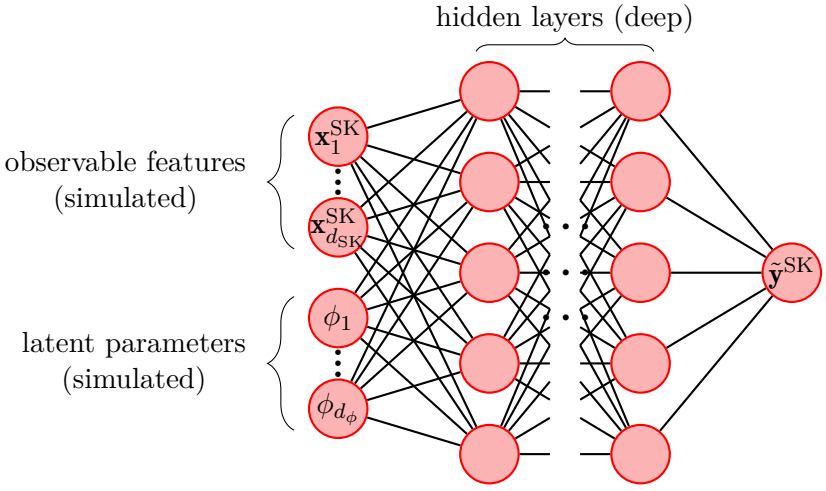


Figure 2: Semi-parametric format structured-knowledge representation. The inputs, including the observable features $\mathbf{X}^{\text{SK}}$ and the latent parameters ϕ, are known and parameterized by economic and finance theory, the same as in parametric format representations. But the representation function $g_\phi(\mathbf{X}^{\text{SK}})$, in this case, is unavailable from the theory and is parameterized by a pre-trained DSNN. The thin lines connecting the neurons within different layers mean that the mapping $\tilde{\mathbf{y}}^{\text{SK}} = g_\phi(\mathbf{X}^{\text{SK}})$ is non-parametrically learned, given the large number of simulated theoretical data.

2.2.3 Non-Parametric Knowledge Representations

Parametric and semi-parametric knowledge representations are characterized by the theory-driven inputs $\mathbf{X}^{\text{SK}}$ and a fixed number of ϕ (but their values are unknown, and hence latent parameters), and more crucially, ϕ is a low-dimensional vector of parameters. This sparsity assumption on the latent knowledge parameters by those established theories is mainly for the sake of tractability,

gradients with respect to these latent parameters.

as the low-dimensional parameters are easier to estimate. However, it can be the case that theories are constructed in a way that they take an unspecified number of latent parameters in ϕ, for greater generality and fewer assumptions. In such high-dimensional settings, the deep surrogate approach becomes infeasible, as the required simulated data space $\mathcal{D}_{\text{surrogate}}$ expands exponentially with the dimension of d_ϕ. We provide two examples below to illustrate how the high-dimensional problems can be tackled by SKINNs.

Non-parametric distribution as the learnable parameters. Theories may provide functional forms that are neither solvable by closed-form nor semi-closed-form expressions, nor by numerical procedures given PDEs or SDEs, but instead require unknown distributions that govern the underlying variables in the theory-driven models, the probabilities from which can be treated as latent parameters. These unknown non-parametric distributions usually appear in theories as the form of expectations for the underlying variables, e.g., $\mathbb{E}[z]$, or of some known functions of them, e.g., $\mathbb{E}[u(z)]$.[8] The underlying variable z determines the output of a theoretical model through the function of its probability distribution.

A typical example in finance is the principle of no-arbitrage, which dictates that an asset's price must equal its discounted expected payoff under a risk-neutral probability measure $\mathbb{Q}$. This principle does not prescribe a specific parametric form for the underlying asset dynamics, but instead requires an expectation from an unknown distribution:

$$\mathbf{y}^{\text{SK}} = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}[u(z)|\mathcal{F}_t], \tag{7}$$

where $u(z)$ is the terminal payoff of the asset at time T (e.g., $u(z) = (S_T - K)^+$ for a European call option, where S_T is the underlying variable), and $\mathbf{y}^{\text{SK}}$ denotes the the theory-driven output, i.e., the no-arbitrage time-t price of the contingent claim.

The expectation of an unknown non-parametric distribution in Equation (7) needs the probabilities $\mathbb{Q}(z)$ of potentially infinite states of nature. The structured-knowledge in this case can be formulated as:

$$\mathbf{y}^{\text{SK}} = g_\phi(\mathbf{X}^{\text{SK}}), \tag{8}$$

$$g_\phi(\mathbf{X}^{\text{SK}}) \equiv \sum_{i=1}^{\infty} u(z_i)\mathbb{Q}(z_i), \tag{9}$$

[8] In essence, the unknown non-parametric distributions appear in theories as integrals of the underlying variables, e.g., $\int_{\mathbb{R}} u(z)\mathbb{Q}(z)dz$, where $u(z)$ is a known function of z and $\mathbb{Q}(z)$ is its distribution function. The integrals represent any moments of the underlying variables required by the theory-driven models.

where $z = \{S_T^{(1)}, S_T^{(2)}, \cdots, S_T^{(\infty)}\}$ refer to the infinite states of the underlying variable S_T, and the learnable latent parameters ϕ are defined as the vector of unknown risk-neutral probabilities associated with each state:

$$\phi = \{\mathbb{Q}(z_1), \mathbb{Q}(z_2), \cdots, \mathbb{Q}(z_\infty)\}. \tag{10}$$

To facilitate the integration of such non-parametric structured-knowledge representations in SKINNs, we use the empirical counterpart of Equation (9):

$$g_\phi(\mathbf{X}^{\text{SK}}) := e^{-r(T-t)} \sum_{i=1}^{N_z} u(z_i)\mathbb{Q}(z_i), \tag{11}$$

where N_z is a sufficiently large integer to approximate the infinite sum. The learnable latent parameters $\phi = \{\mathbb{Q}(z_1), \mathbb{Q}(z_2), \cdots, \mathbb{Q}(z_{N_z})\}$ are constrained such that each of them are non-negative and $\sum_{i=1}^{N_z} \mathbb{Q}(z_i) = 1$, which can be enforced during optimization (e.g., by passing them through a softmax function).

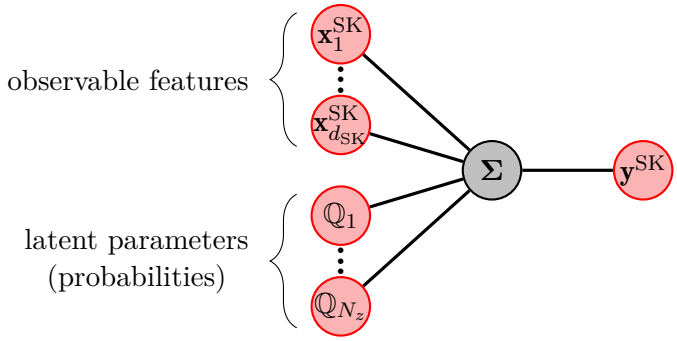


Figure 3: Non-parametric structured-knowledge representations by unknown distributions. The inputs include a high-dimensional vector of probabilities, which serves as the learnable latent parameters when integrated in SKINNs. The probabilities are passed to a softmax function to ensure the non-negativity and the total probability of one unit.

By jointly optimizing for ϕ and the neural network parameters θ, the SKINNs framework can learn a high-dimensional vector of risk-neutral probabilities directly from data, guided solely by the foundational economic principle of no-arbitrage, thus avoiding the potentially misspecified assumptions of a particular parametric or semi-parametric model. We visualize such a non-parametric structured-knowledge representation prescribed by unknown distributions in Figure (3).

Non-parametric latent parameters as the learnable parameters. SKINNs also provide solutions when there is no guiding knowledge from theories. At times, we observe the outputs from a data-generating process, rather than knowing the data-generating process provided by the well-established theories. For example, independent of any assumed asset pricing models, we still

observe stock returns, the cross-section of option prices, and the term-structure of interest rates generated by the market. In these cases, we are uncertain about the data-generating process, but we observe the realizations. We employ an auto-encoder (AE) to learn the structured knowledge non-parametrically, that is, the latent parameters are optimally determined by the machine instead of by theory.

Given a series of observation vectors $\mathcal{D}_{ae} = \{\hat{\mathbf{y}}_i\}_{i=1}^{N_{\text{ae}}}$, probably with noise, which are realizations of an unknown data-generating process, we reconstruct them through training an AE. An AE can be viewed as a non-linear, neural network counterpart of PCA, which is an unsupervised learning method (see, for example, in Gu et al., 2019; Freire and Vladimirov, 2023). We assume that the unknown data-generating process is characterized by a set of arbitrary-dimensional latent parameters ϕ^{ae}.[9] We use the encoder part of an AE to compress the observation vector $\hat{\mathbf{y}}$ into the latent parameter space, and use the decoder part to project the latent parameters back to the input observation vector.[10] Therefore, the objective of training an AE is to minimize a reconstruction loss between the input samples and the AE-generated samples:

$$\mathcal{L}_{\text{ae}}(\theta; \mathcal{D}_{\text{ae}}) = \sum_{i=1}^{N_{\text{ae}}} \left(f_\theta^{\text{ae}}(\hat{\mathbf{y}}^{[i]}) - \hat{\mathbf{y}}^{[i]} \right)^2 \Big/ N_{\text{ae}}. \tag{12}$$

The AE is also pre-trained offline, based on the dataset $\mathcal{D}_{ae}$ where it is believed to contain structured knowledge that determines the data-generating process. The dataset $\mathcal{D}_{\text{ae}}$ can be obtained from expert simulations or collected directly from the real-world. Once the AE is trained, we use the decoder neural network $f_\theta^{\text{dec}}(\mathbf{X}^{\text{SK}}, \phi_{ae})$ to represent the non-parametrically learned structured knowledge. The structured-knowledge representation in this case becomes $g_{\phi_{\text{ae}}}(\mathbf{X}^{\text{SK}}) \equiv_\theta^{\text{dec}} (\mathbf{X}^{\text{SK}}, \phi_{ae})$, and ϕ_{ae} is the vector of learnable latent parameters in the SKINN framework.[11] We illustrate the AE-based non-parametric knowledge representations in Figure (4).

[9] The arbitrary-dimensional vector ϕ^{ae} should be, however, lower-dimensional relative to the potentially very high dimensionality of the input observation vector $\hat{\mathbf{y}}$. The latent parameter ϕ^{ae} learned by an AE can be much higher-dimensional than the latent parameters admitted by the parametric and semi-parametric structured-knowledge representations g_ϕ. For example, in option pricing, the input option price cross-section $\hat{\mathbf{y}}$ can be a 200-dimensional vector (e.g., 200 points $\{(m_i, \tau_i, \hat{y}_i)\}_{i=1}^{200}$ from a price surface). If ϕ^{ae} is set to be a 50-dimensional vector, which is much smaller than 200, but is still a magnitude higher dimensionality than the ϕ provided by classic theories. The SKINNs framework scales the learnable latent parameters to be high-dimensional. Hence, one is allowed to train an AE with an arbitrary latent space dimension.

[10] We assume that the series of economic observation vectors all have the same size, for example, a cross-section of 200 option prices. This is due to the fact that the input layer of an AE, i.e., the dimension of $\hat{\mathbf{y}}$, has a fixed size.

[11] It is also possible to update the latent parameter by feeding the SKINN predictions of the economic outputs, $\hat{\mathbf{y}} = f_\theta(\mathbf{X}^{\text{SK}})$ to the pre-trained encoder neural network $f_\theta^{\text{enc}}(f_\theta(\mathbf{X}^{\text{SK}}))$ that returns the estimates of latent parameters $\hat{\phi}_{\text{ae}}$ for the current epoch. $\mathbf{X}^{\text{SK}}$ are the observable features associated with $\hat{\mathbf{y}}$, though the training of AE does not require these features. By doing this, the latent parameters ϕ_{ae} can be learned without actually creating extra parameters for the gradient-descent optimizer. In addition, the auto-encoder can be replaced with a variational auto-encoder (VAE), which can be more robust to the possibly noisy intermediate ϕ_{ae} values when training SKINNs.

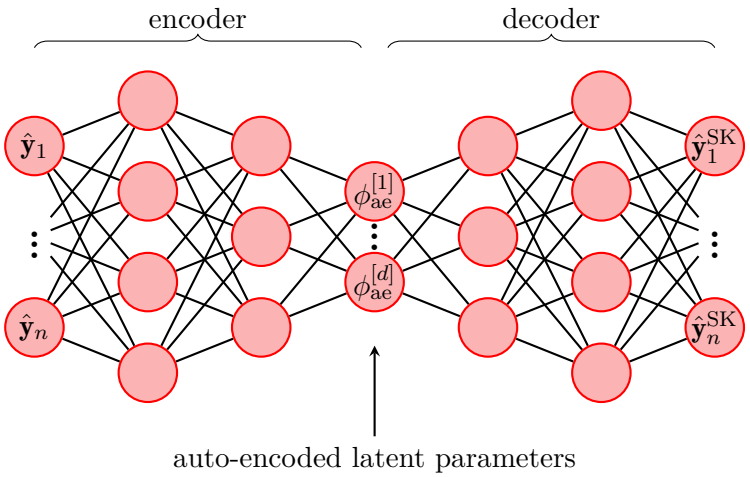


Figure 4: Non-parametric structured-knowledge representations by auto-encoders. The inputs include a vector of economic outputs (i.e., a set of realizations of an unknown data-generating process) of size n. The encoder neural network compresses the economic outputs into a vector of latent parameters of size d, and $d \ll n$. The d-dimensional auto-encoded latent parameters are then passed to a decoder neural network, which aims to returns a n-dimensional vector that reconstructs the input vector as close as possible. The outputs of the decoder are based on the structured knowledge derived by the auto-encoder.

2.3 The Composite Objective

The SKINNs framework unifies the data-driven and the structured-knowledge components through a composite loss function. The data-driven component measures the discrepancy between the neural network's predictions and observed outcomes:

$$\mathcal{L}_{\text{data}}(\theta) = \mathbb{E}\left[\ell(f_\theta(\mathbf{X}), \mathbf{y})\right], \tag{13}$$

where $\ell(\cdot, \cdot)$ is a loss function, typically the squared error $\ell(f, \mathbf{y}) = (f - \mathbf{y})^2$ for regression tasks.

The structured-knowledge component penalizes deviations between the neural network's output f_θ and the structured-knowledge representation's output g_ϕ:

$$\mathcal{L}_{\text{SK}}(\theta, \phi) = \mathbb{E}\left[\ell(f_\theta(\mathbf{X}_{\text{grid}}), g_\phi(\mathbf{X}_{\text{grid}}^{\text{SK}}))\right]. \tag{14}$$

Crucially, this loss is evaluated on a set of collocation points, $\{\mathbf{X}_{\text{grid}}^{[i]}\}_{i=1}^{M}$, which can be randomly sampled from the entire valid input domain of the neural network. These points do not need to correspond to the observed data $\{\mathbf{X}_{\text{obs}}^{[i]}\}_{i=1}^{N}$. This allows the framework to enforce theoretical consistency in regions of the feature space where observations may be sparse or absent, thereby promoting robust generalization. Appendix B.5.1–B.5.2 provides formal generalization and target-

risk bounds in a convex proxy, clarifying how structured regularization and collocation can improve robustness under distributional shifts.

The complete learning objective for a SKINN is the weighted sum of the data and the structured-knowledge loss:

$$\mathcal{L}(\theta, \phi; \mathbf{X}_{\text{obs}} \cup \mathbf{X}_{\text{grid}}) = \lambda_{\text{data}} \mathcal{L}_{\text{data}}(\theta; \mathbf{X}_{\text{obs}}) + \lambda_{\text{SK}} \mathcal{L}_{\text{SK}}(\theta, \phi; \mathbf{X}_{\text{grid}}). \tag{15}$$

The hyperparameters λ_{data} and λ_{SK} control the trade-off between fitting the observed data and adhering to the embedded theoretical structure. Without loss of generality, we can normalize $\lambda_{\text{data}} = 1$ and denote $\lambda = \lambda_{\text{SK}}$, yielding:

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\text{data}}(\theta; \mathbf{X}_{\text{obs}}) + \lambda \mathcal{L}_{\text{SK}}(\theta, \phi; \mathbf{X}_{\text{grid}}). \tag{16}$$

As a special case, consider a regression setting with squared-error loss. The composite objective in Equation (16) can then be written as:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}\left[(f_\theta(\mathbf{X}_{\text{obs}}) - \mathbf{y})^2\right] + \lambda \mathbb{E}\left[(f_\theta(\mathbf{X}_{\text{grid}}) - g_\phi(\mathbf{X}_{\text{grid}}^{\text{SK}}))^2\right]. \tag{17}$$

2.4 The Joint Learning of Parameters

Our SKINNs are trained by jointly minimizing this composite loss with respect to both sets of parameters, e.g., the neural network parameter and the structured-knowledge parameters:

$$(\theta^*, \phi^*) =: \underset{\theta \in \Theta, \phi \in \Phi}{\arg\min}\, \mathcal{L}(\theta, \phi; \mathbf{X}_{\text{obs}} \cup \mathbf{X}_{\text{grid}}). \tag{18}$$

The optimization of Equation (18) is carried out using (stochastic) gradient-based methods at each iteration. While θ can be astronomically-dimensional, ϕ is usually relatively lower-dimensional and domain-interpretable (e.g. material constants in physics, preference or volatility parameters in economics, or control parameters in engineering). The key innovation of SKINNs is the simultaneous, gradient-based discovery of both parameter sets, where gradients of the composite loss with respect to θ and ϕ are computed via automatic differentiation, enabling the theoretical parameters to adapt dynamically to reconcile theory with empirical evidence so that the resulting function approximator respects domain-specific structure while maintaining flexibility. This joint optimisation distinguishes SKINNs from approaches that pre-calibrate ϕ separately or ignore the theoretical structure entirely. Figure 5 illustrates the architecture of a SKINN.

A natural concern is whether the structured-knowledge parameters ϕ can be correctly identified,

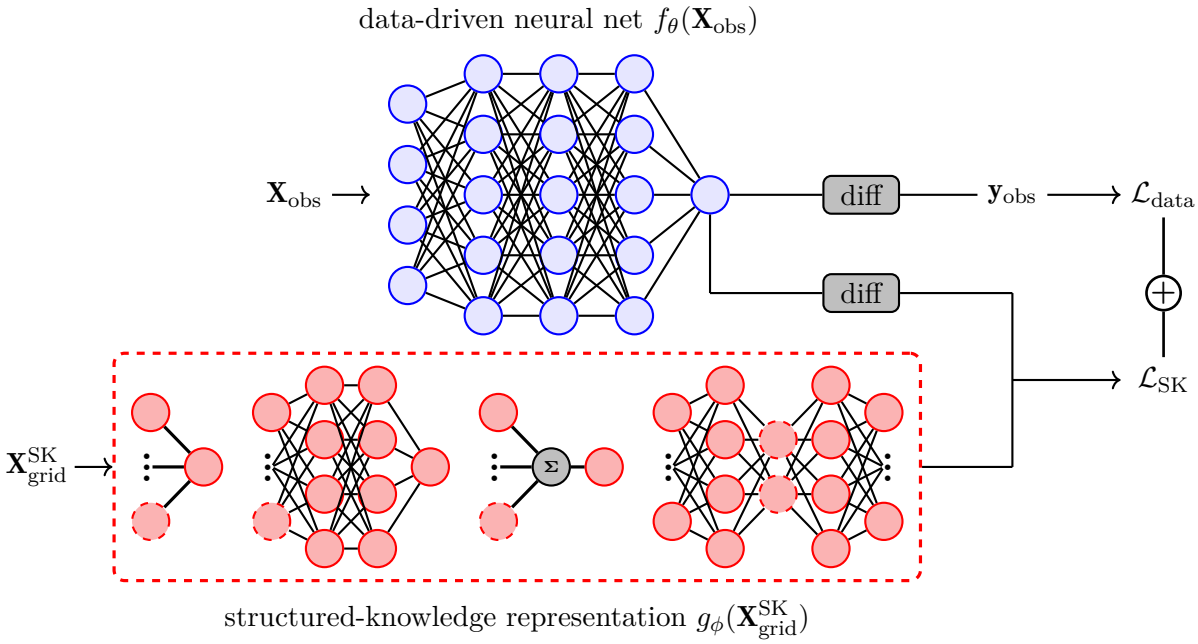


Figure 5: The architecture of a SKINN. The neural network on the top (blue neurons) is a base neural network of a SKINN, which takes some observable features $\mathbf{X} \subseteq \mathbb{R}^d$ as inputs, and outputs predictions. The structures at the bottom (red circles/ neurons) are structured-knowledge representations of different formats. The first and the third are the parametric format and the non-parametric format with unknown distribution, for which the mapping between $\{\mathbf{X}^{\text{SK}}, \phi\}$ and $\mathbf{y}^{\text{SK}}$ is known. The second and the fourth are the semi-parametric format and the non-parametric format with unknown latent parameters, which are parameterized by a separate pre-trained neural network. All the structured-knowledge representations take observable features $\mathbf{X}^{\text{SK}}_{\text{grid}}$ and the current epoch value of the learnable latent parameters as inputs, and output theory-driven estimates. After sufficient epochs of training, the data-driven neural network will be embedded with structured knowledge, and hence becomes a SKINN.

jointly with the neural network parameters. To this end, we provide two complementary results below. The first is a general profiling lemma showing that the two sets of parameters can be jointly identified, and the second provides a sufficient condition under which ϕ can be uniquely identified.

Lemma 1 (Identification through profiling)**.** *Given the composite objective $\mathcal{L}(\theta, \phi)$ in Equation* (17)*, define the profiled criterion*

$$Q(\phi) \equiv \inf_{\theta \in \Theta} \mathcal{L}(\theta, \phi). \tag{19}$$

If $Q(\phi)$ has a unique minimizer ϕ^* over Φ, then ϕ^* is identified as the unique structured-parameter value selected by the SKINNs objective. If, in addition, the minimizer in θ at ϕ^* is unique, then the joint minimizer (θ^*, ϕ^*) is uniquely identified.

Proof. By definition, (θ^*, ϕ^*) minimizes $L(\theta, \phi)$ over $\Theta \times \Phi$ if and only if ϕ^* minimizes the profiled criterion $Q(\phi)$ and $\theta^* \in \arg\min_\theta L(\theta, \phi^*)$. Uniqueness of the minimizer of $Q(\phi)$ yields uniqueness of ϕ^*; uniqueness of the minimizer in θ at ϕ^* yields uniqueness of the joint minimizer. □

Proposition 1 (A sufficient condition)**.** *Consider the squared-error SKINNs objective in Equation* (17)*. Assume: (i) the function class $\{f_\theta\}$ is sufficiently rich so that minimizing over $\theta \in \Theta$ is equivalent to minimizing over all measurable functions $f : \mathcal{X} \to \mathbb{R}$ with finite second moments; (ii) $\mathbf{X}_{obs}$ and $\mathbf{X}_{grid}$ are random inputs with the same distribution, written $\mathbf{X}_{grid} \overset{d}{=} \mathbf{X}_{obs}$, and $\mathbf{X}_{grid}^{SK}$ is a measurable function of $\mathbf{X}_{grid}$ (equivalently, $\mathbf{X}_{obs}^{SK}$ is a measurable function of $\mathbf{X}_{obs}$); and (iii) $\mathbb{E}[\mathbf{y}^2] < \infty$ and $\mathbb{E}\Big[g_\phi(\mathbf{X}_{grid}^{SK})^2\Big] < \infty$ for all $\phi \in \Phi$. Then, for each fixed ϕ, the minimizer over f exists and can be written pointwise in terms of $\mathbf{X}_{obs}$ as*

$$f_\phi^*(\mathbf{X}_{obs}) = \frac{\mathbb{E}[\mathbf{y} \mid \mathbf{X}_{obs}] + \lambda\, g_\phi(\mathbf{X}_{obs}^{SK})}{1 + \lambda}. \tag{20}$$

Moreover, the profiled criterion $Q(\phi) \equiv \inf_{\theta \in \Theta} \mathcal{L}(\theta, \phi)$ satisfies, up to ϕ-independent constants,

$$Q(\phi) = \frac{\lambda}{1+\lambda}\, \mathbb{E}\Big[\big(\mathbb{E}[\mathbf{y} \mid \mathbf{X}_{obs}] - g_\phi(\mathbf{X}_{obs}^{SK})\big)^2\Big] \;+\; \textit{const.}$$

Hence, if the map

$$\phi \mapsto \mathbb{E}\Big[\big(\mathbb{E}[\mathbf{y} \mid \mathbf{X}_{obs}] - g_\phi(\mathbf{X}_{obs}^{SK})\big)^2\Big]$$

has a unique minimizer over Φ, then ϕ is identified [12].

[12] In particular, the substitution of $\mathbf{X}_{\text{obs}}$ for $\mathbf{X}_{\text{grid}}$ in all above formulas is purely notational: it uses $\mathbf{X}_{\text{grid}} \overset{d}{=} \mathbf{X}_{\text{obs}}$ to express both expectations with a common conditioning variable.

Proof. Fix ϕ. Under Assumptions (i)–(iii), the population objective in (17) can be written as

$$\mathbb{E}\Big[\big(f(\mathbf{X}_{\text{obs}}) - \mathbf{y}\big)^2\Big] + \lambda\,\mathbb{E}\Big[\big(f(\mathbf{X}_{\text{grid}}) - g_\phi(\mathbf{X}^{\text{SK}}_{\text{grid}})\big)^2\Big], \tag{21}$$

where f ranges over measurable functions with finite second moments. Since $\mathbf{X}_{\text{grid}} \overset{d}{=} \mathbf{X}_{\text{obs}}$, the second expectation equals

$$\mathbb{E}\Big[\big(f(\mathbf{X}_{\text{grid}}) - g_\phi(\mathbf{X}^{\text{SK}}_{\text{grid}})\big)^2\Big] = \mathbb{E}\Big[\big(f(\mathbf{X}_{\text{obs}}) - g_\phi(\mathbf{X}^{\text{SK}}_{\text{obs}})\big)^2\Big],$$

so both terms can be expressed using the common conditioning variable $\mathbf{X}_{\text{obs}}$. Conditioning on $\mathbf{X}_{\text{obs}}$ and minimizing pointwise gives, for almost every $\mathbf{X}_{\text{obs}}$,

$$f^*_\phi(\mathbf{X}_{\text{obs}}) = \arg\min_{u\in\mathbb{R}}\ \mathbb{E}\big[(u-\mathbf{y})^2 \mid \mathbf{X}_{\text{obs}}\big] + \lambda\big(u - g_\phi(\mathbf{X}^{\text{SK}}_{\text{obs}})\big)^2,$$

which is a strictly convex quadratic in u. The first-order condition is

$$2\big(u - \mathbb{E}[\mathbf{y} \mid \mathbf{X}_{\text{obs}}]\big) + 2\lambda\big(u - g_\phi(\mathbf{X}^{\text{SK}}_{\text{obs}})\big) = 0. \tag{22}$$

Solving (22) yields (20). Substituting (20) back into (21) and completing the square yields

$$Q(\phi) = \frac{\lambda}{1+\lambda}\,\mathbb{E}\Big[\big(\mathbb{E}[\mathbf{y} \mid \mathbf{X}_{\text{obs}}] - g_\phi(\mathbf{X}^{\text{SK}}_{\text{obs}})\big)^2\Big] + \text{const.},$$

where the constant does not depend on ϕ. Uniqueness of the minimizer of the displayed term implies identification of ϕ. □

2.5 Statistical Properties

By correctly identifying the astronomically-dimensional neural network parameters and the structured-knowledge parameters, the proposed SKINNs framework goes beyond a black-box predictive tool. Integrating the discovery process of the latent scientific parameters from structured-knowledge, the SKINNs framework instead functions as a white box that is more interpretable. The SKINNs framework also presents desirable statistical properties, generalizing many classic econometric tools, e.g., GMM and semi-parametric models.

Consistency and asymptotic normality. The SKINNs estimator belongs to the broad class of M-estimators (Huber et al., 1967; Van Der Vaart and Wellner, 1996; Newey and McFadden, 1994). Under standard regularity conditions (identification, compactness, continuity, and uniform

convergence of the empirical loss; see Appendix B.1), the jointly learned parameters converge in probability, $(\hat{\theta}_N, \hat{\phi}_N) \xrightarrow{p} (\theta^*, \phi^*)$, as $N \to \infty$ (Theorem 1). Under additional smoothness and moment conditions, the estimator is asymptotically normal at the parametric rate (Theorem 2):

$$\sqrt{N} \begin{pmatrix} \hat{\theta}_N - \theta^* \\ \hat{\phi}_N - \phi^* \end{pmatrix} \xrightarrow{d} \mathcal{N}(\mathbf{0}, V), \qquad V = H^{-1} \Xi H^{-1}, \tag{23}$$

where H denotes the Hessian of the population loss and Ξ the covariance of the score. The sandwich-form covariance (Huber et al., 1967; White, 1980) is robust to potential misspecification of the structured-knowledge component g_ϕ, so that standard errors and test statistics remain asymptotically valid even if the embedded theory is an imperfect description of reality. These results operate in a double-asymptotic regime where the number of collocation points M_N grows proportionally with the sample size ($M_N/N \to c$ for finite $c > 0$), ensuring that discretization error from $\mathcal{L}_{\text{SK}}$ does not degrade the $O_p(N^{-1/2})$ convergence. In practice, both the Hessian and score covariance can be estimated via automatic differentiation and outer products of gradients, and block-matrix inversion allows extraction of the $q \times q$ covariance submatrix for ϕ without inverting the full $(p+q) \times (p+q)$ Hessian, making confidence intervals and Wald-type hypothesis tests for ϕ computationally feasible even when the neural network contains millions of parameters. The full development, including practical covariance estimation and bootstrap alternatives, appears in Appendix B.2.

A GMM interpretation and the role of λ. For researchers grounded in the econometric tradition, the composite SKINNs objective admits a revealing reinterpretation as a regularized, overidentified GMM problem (Hansen, 1982). The detailed development in Appendix B.3 recasts the two loss components as moment conditions and identifies λ as an econometric weighting parameter: a large λ reflects strong confidence in the embedded theory, while a small λ prioritizes empirical fit. This tradeoff is formalized through a variance-minimizing choice within a restricted diagonal weighting class (Proposition 2) and a closed-form benchmark yielding $\lambda^* = \sigma_\varepsilon^2/\sigma_\eta^2$ in a stylized two-signal setting (Proposition 3).

Semiparametric efficiency. The GMM formulation connects naturally to semiparametric estimation theory. Under orthogonal moment conditions, where f_θ exerts only a second-order influence on the estimation of ϕ, the SKINNs estimator achieves the semiparametric efficiency bound for the given moment model (Theorem 3; Appendix B.4). This places SKINNs within the family of sieve GMM estimators (Ai and Chen, 2003) and deep GMM methods (Bennett et al., 2019; Farrell et al.,

2021), while distinguishing it through the explicit incorporation of a learnable structured-knowledge component.

Generalization and robustness under distributional shift. Out-of-sample behavior is of particular importance in finance, where distributional shifts between training and deployment periods are the norm. The stability-based analysis in Appendix B.5.1 shows that the structured-knowledge penalty tightens the uniform stability bound on the expected generalization gap (Theorem 4). A complementary target-risk decomposition separates the risk under a shifted target distribution into alignment between f_θ and g_ϕ (controlled by $\mathcal{L}_{\text{SK}}$) and portability of g_ϕ to the new environment (Theorem 5). When the collocation distribution matches the target marginal, the alignment term is directly minimized during training (Corollary 6). These results formalize a compelling intuition: if the embedded theoretical model captures structural regularities, such as no-arbitrage conditions or equilibrium relations, that remain stable across regimes, the SKINNs estimator inherits this stability.

Surrogate differentiability and the curse of dimensionality. Joint gradient-based training requires that g_ϕ be first-order differentiable with respect to both $\mathbf{X}^{\text{SK}}$ and ϕ. For parametric closed-form expressions this is immediate, but for semi-parametric deep surrogate models the question is more nuanced. Appendix B.6 establishes that the neural network architecture of the surrogate guarantees end-to-end differentiability through the composite SKINNs objective, and shows that deep surrogates circumvent the curse of dimensionality that afflicts direct PDE/SDE embedding by scaling polynomially rather than exponentially with the input dimension.

SKINNs as a unifying framework. A distinctive strength of the framework is that its composite objective is not tied to a single methodological tradition. As detailed in Appendix B.7, several well-established paradigms emerge as special cases or limiting configurations of the same objective: functional GMM, Bayesian MAP estimation, transfer learning, physics-informed learning, and domain adaptation all reside within the SKINNs framework, with the specific instantiation determined by the choice of g_ϕ, the loss structure, and the regularization strength λ. This multiplicity of valid interpretations reflects a deeper architectural point: SKINNs provide a single composite objective from which diverse methodological traditions can be recovered, while the joint optimization of (θ, ϕ) enables capabilities, such as dynamic latent parameter discovery and bidirectional theory-data reconciliation, that none of the individual paradigms provide in isolation.

3 A SKINNs Application in Finance

Data-driven models for option pricing have been studied since Hutchinson et al. (1994); Aït-Sahalia and Lo (1998). However, these early applications failed to incorporate the necessary domain knowledge. Later on, Garcia and Gençay (2000); Dugas et al. (2000, 2009); Zheng et al. (2021) incorporate the homogeneity hint by using a functional structure similar to the Black-Scholes formula, or the more general shape prior for the price surfaces based on the no static-arbitrage assumptions. More recently, Chen et al. (2023) incorporates the Black-Scholes model via the transfer learning approach; Aboussalah et al. (2024) integrates the dynamic hedging principles, using a partial differential operator as in PINNs.

3.1 Option Pricing Structured Knowledge

All the above hybrid models rely merely on the prior knowledge that is static and model-free. Structured knowledge, on the other hand, captures the dynamic patterns of option prices across time, which contain richer explanatory and predictive information. In this section, we construct several specifications of such knowledge in option pricing, which leads to option pricing SKINNs.

3.1.1 Parametric Specifications

The Black-Scholes-Merton model. The seminal Black-Scholes-Merton model (Black and Scholes, 1973; Merton, 1973) requires only one-dimensional latent parameter, which is the risk-neutral volatility. It assumes the following underlying asset price dynamics:

$$dS_t = rS_t dt + \sigma S_t dW_t, \tag{24}$$

where S_t is the price of the underlying asset at time t; r is the risk-free rate; σ is the instantaneous volatility of the asset return; and W_t is a standard Brownian motion. The model provides a closed-form solution to (24), which gives rise to a parametric format g_ϕ representation:

$$g^{\mathrm{BS}}(\mathbf{X}^{\mathrm{SK}}; \phi \in \{\sigma\}) = S_t \Phi\left(d_1(\phi)\right) - K e^{-r(T-t)} \Phi\left(d_2(\phi)\right), \tag{25}$$

$$d_1(\phi \in \{\sigma\}) = \frac{1}{\sigma\sqrt{T-t}} \left(\log(S_t/K) + (r + \frac{\sigma^2}{2})(T-t) \right), \tag{26}$$

$$d_2(\phi \in \{\sigma\}) = d_1(\phi) - \sigma\sqrt{(T-t)}, \tag{27}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard Gaussian distribution, the asset price S_t, the strike price K, the risk-free rate r, and the time-to-maturity $T-t$ are known in

the input features $\mathbf{X}^{\text{SK}}$. The risk-neutral volatility σ in this case, is the only latent parameter in $g_\phi^{\text{BS}}(\mathbf{X}^{\text{SK}})$. In financial applications, the latent parameters in a structured-knowledge representation usually possess economic meanings. We therefore also refer to them as latent economic parameters.

The Ad-hoc Black-Scholes-Merton model. One can relax the constant volatility assumption in the Black-Scholes-Merton model, which is inconsistent with the stylized fact that the implied volatility varies across strike prices and maturities, by using the so-called ad-hoc Black-Scholes-Merton model (Dumas et al., 1998) that reads:

$$\sigma_{i,t} = \alpha_{0,t} + \alpha_{1,t} m_{i,t} + \alpha_{2,t} m_{i,t}^2 + \alpha_{3,t} \tau_{i,t} + \alpha_{4,t} \tau_{i,t}^2 + \alpha_{5,t} m_{i,t} \tau_{i,t} + \epsilon_{i,t}, \forall i \leq 1 \leq n, \tag{28}$$

where $\{(m_{i,t} = \frac{K_i}{S_t}, \tau_{i,t})\}_{i=1}^n$ represents the moneyness and the maturities observed from the option cross-section of sample size n at time t. The estimated volatilities $\sigma_{i,t}$ are used with the Black-Scholes-Merton formula to estimate option prices.

By viewing the unknown parameters $\{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$ as the learnable latent economic parameters, this gives rise to a parametric format g_ϕ representation:

$$g^{\text{AHBS}}(\mathbf{X}^{\text{SK}}; \phi \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}) = S_t \Phi(d_1(\phi)) - K e^{-r\tau} \Phi(d_2(\phi)), \tag{29}$$

$$\begin{aligned} d_1(\phi \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}) = & \frac{1}{(\alpha_0 + \alpha_1 m + \alpha_2 m^2 + \alpha_3 \tau + \alpha_4 \tau^2 + \alpha_5 m\tau)\sqrt{\tau}} \\ & \times \left(\log(S_t/K) + (r + \frac{(\alpha_0 + \alpha_1 m + \alpha_2 m^2 + \alpha_3 \tau + \alpha_4 \tau^2 + \alpha_5 m\tau)^2}{2})\tau \right), \end{aligned} \tag{30}$$

$$d_2(\phi \in \{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}) = d_1(\phi) - (\alpha_0 + \alpha_1 m + \alpha_2 m^2 + \alpha_3 \tau + \alpha_4 \tau^2 + \alpha_5 m\tau)\sqrt{\tau}. \tag{31}$$

The SABR Black-Scholes-Merton model. Another way of extending the Black-Scholes-Merton model is to use the stochastic alpha-beta-rho (SABR) volatility model. As introduced in Hagan et al. (2002), the SABR model assumes the following dynamics for the volatility:

$$dF_t = \alpha_t F_t^\beta dW_t^1, \qquad F_0 = \hat{f}, \tag{32}$$

$$d\alpha_t = v_t \alpha_t dW_t^2, \qquad \alpha_0 = \alpha, \tag{33}$$

$$\langle dW^1, dW^2 \rangle_t = \rho_t dt, \tag{34}$$

where $F_t = S_t e^{r(T-t)}$ denotes the forward price of the underlying asset, α_t denotes the volatility process for the asset return, v_t is a time-dependent function of the volatility-of-volatility, and dW_t^1, dW_t^2 are two correlated Brownian motions, where the correlation coefficient ρ_t is another time-dependent function. Since the time-variant nature of the diffusion coefficient v_t (the volatility-

of-volatility) and the correlation ρ_t, the SABR model described by equation (32)-(34) is actually a dynamic SABR model. Osajima (2007) provides an approximation to the implied volatility:

$$\sigma_{\text{SABR}}(K,T,\hat{f}) \approx \frac{1}{w}\left(1 + A_1(T)\log\left(\frac{K}{\hat{f}}\right) + A_2(T)\log^2\left(\frac{K}{\hat{f}}\right) + B(T)T\right), \tag{35}$$

where $w := \frac{\hat{f}^{1-\beta}}{\alpha}$, $A_1(T)$, $A_2(T)$, $B(T)$ are functions of the dynamic SABR latent economic parameters.[13] There are some special parametric specifications for the time-dependent functions v_t, ρ_t, such that $v_t > 0, -1 \leq \rho_t \leq 1, \forall t \in [0,T]$, leading to a different number of learnable latent economic parameters for SKINNs.[14] However, to showcase the capability of SKINNs in dealing with high-dimensional estimation problems, we treat v_t and ρ_t themselves as learnable parameters.In our implementation, we set $v_t = \{v_{t_i}\}_{i=1}^{360}$ and $\rho_t = \{\rho_{t_i}\}_{i=1}^{360}$ as 720 latent parameters in total, where $t_i = \frac{i}{360}, 1 \leq i \leq 360$. Given the values of v_t, ρ_t on the fine grids $t_i, 1 \leq i \leq 360$, the integrals in the equation (39)-(42) can be calculated by discrete summations.

By viewing $\{v_{t_i}\}_{i=1}^{360}$, $\{\rho_{t_i}\}_{i=1}^{360}$, α and β as the learnable latent economic parameters, this gives rise to a parametric format g_ϕ representation, with the use of the Black-Scholes-Merton formula:

$$\begin{aligned} g^{\text{SABR}}(\mathbf{X}^{\text{SK}}; \phi \in \{v_{t_i}\}_{i=1}^{360} \cup \{\rho_{t_i}\}_{i=1}^{360} \cup \{\alpha,\beta\}) = S_t\Phi\left(d_1\left(\sigma_{\text{SABR}}(\phi)\right)\right) \\ - Ke^{-r\tau}\Phi\left(d_2\left(\sigma_{\text{SABR}}(\phi)\right)\right), \end{aligned} \tag{43}$$

in which specification, the learnable latent economic parameter vector ϕ becomes 722-dimensional.

[13] These functions are fully-determined by, and differentiable w.r.t. the latent parameters:

$$A_1(T) = \frac{\beta-1}{2} + \frac{\eta_1(T)w}{2}, \tag{36}$$

$$A_2(T) = \frac{(1-\beta)^2}{12} + \frac{1-\beta-\eta_1(T)w}{4} + \frac{4v_1^2(T) + 3\left(\eta_2^2(T) + 3\eta_1^2(T)\right)}{24}w^2, \tag{37}$$

$$B(T) = \frac{1}{w^2}\left(\frac{(1-\beta)^2}{24} + \frac{w\beta\eta_1(T)}{4} + \frac{2v_2^2(T) - 3\eta_2^2(T)w^2}{24}\right), \text{where,} \tag{38}$$

$$v_1^2(T) = \frac{3}{T^3}\int_0^T (T-t)^2 v_t^2 dt, \tag{39}$$

$$v_2^2(T) = \frac{6}{T^3}\int_0^T (T-t) t v_t^2 dt, \tag{40}$$

$$\eta_1(T) = \frac{2}{T^2}\int_0^T (T-t) v_t \rho_t dt, \text{and,} \tag{41}$$

$$\eta_2(T) = \frac{12}{T^4}\int_0^T \int_0^t \left(\int_0^s v_u \rho_u du\right)^2 ds dt. \tag{42}$$

[14] For example, v_t, ρ_t can be specified as two constants, $v_t = v_0, \rho_t = \rho_0$ (two parameters, v_0, ρ_0); as classical formulations, $v_t = v_0 e^{-bt}, \rho_t = \rho_0 e^{-at}$ (four parameters, v_0, ρ_0, a, b); as piece-wise functions, $v_t = v_0, \rho_t = \rho_0, \forall t \leq T^*$, and $v_t = v_1, \rho_t = \rho_1, \forall t > T^*$ (five parameters, $v_0, \rho_0, v_1, \rho_1, T^*$); or as more general formulations, $v_t = (v_0 + q_v t)e^{-bt} + d_v$, and $\rho_t = (\rho_0 + q_\rho t)e^{-at} + d_\rho$ (eight parameters, $v_0, \rho_0, q_v, q_\rho, a, b, d_v, d_\rho$). See the details of the dynamic SABR volatility models in Hagan et al. (2002); Gatheral (2011).

The Heston's stochastic volatility model. The capability of SKINNs in dealing with highly non-linear structured-knowledge representations and high-dimensional latent parameters enables us to incorporate more sophisticated g_ϕ, for example, the Heston (1993) model below:

$$dS_t = rS_t\,dt + \sqrt{v_t}S_t\,dW_t^1, \tag{44}$$

$$dv_t = \kappa\left(v_\theta - v_t\right)\,dt + \sigma_v\sqrt{v_t}\,dW_t^2, \tag{45}$$

$$\langle dW^1, dW^2\rangle_t = \rho dt, \tag{46}$$

where v_t is the instantaneous variance that is driven by a mean-reverting process. κ is the mean-reverting speed of the variance process, v_θ is the long-term average of the variance, and σ_v is the volatility of the variance. ρ is the correlation of the Brownian motions from the two processes.

For such sophisticated theory-driven models, convenient closed-form solutions are unavailable, and numerical methods are needed. The Heston model belongs to the affine class of SDEs that are solvable with characteristic functions (Carr and Madan, 1999; Duffie et al., 2000).[15] Knowing the characteristic function $\psi_\tau(u;\phi)$, European option prices can be evaluated as a semi-closed-form expression using the Fourier transform (Carr and Madan, 1999).

We employ the COS method developed by Fang and Oosterlee (2009) to make the Fourier inversion with the log-return characteristic function $\psi_\tau(u;\phi)$ more efficient, which gives rise to a parametric format g_ϕ representation:

$$g^{\mathrm{HSV}}(\mathbf{X}^{\mathrm{SK}};\phi\in\{v_\theta,v_0,\sigma_v,\rho,\kappa\}) = \frac{1}{2}e^{-r\tau}\mathfrak{Re}\left\{\psi_\tau\left(0;S,r,\phi\right)\right\}V_0(K) \tag{52}$$

$$+\,e^{-r\tau}\sum_{w=1}^{N-1}\mathfrak{Re}\left\{\psi_\tau\left(\frac{w\pi}{b-a};S,r,\phi\right)e^{-iw\pi\frac{a}{b-a}}\right\}V_w\left(K\right), \tag{53}$$

[15] The log-return characteristic function of the Heston model is given as:

$$\psi_\tau(u;\phi) = \exp\left[C_\tau\left(u;\phi\right)v_\theta + D_\tau\left(u;\phi\right)v_0 + iu\log\left(Se^{r\tau}\right)\right], \tag{47}$$

where v_0 is the initial instantaneous variance, and C_τ, D_τ are functions of the Heston model latent economic parameters, known as:

$$C_\tau = \kappa\sigma_v^{-2}\left(\left(\kappa-\rho\sigma_v ui-d\right)\tau - 2\log\left(\frac{1-ge^{-d\tau}}{1-g}\right)\right), \tag{48}$$

$$D_\tau = \sigma_v^{-2}\left(\kappa-\rho\sigma_v iu-d\right)\left(1-e^{-d\tau}\right)\Big/\left(1-ge^{-d\tau}\right), \tag{49}$$

where the function d and g are defined as:

$$d = \left(\left(\rho\sigma_v ui-\kappa\right)^2 - \sigma_v^2\left(-iu-u^2\right)\right)^{1/2}, \tag{50}$$

$$g = \left(\kappa-\rho\sigma_v ui-d\right)\Big/\left(\kappa-\rho\sigma_v ui+d\right). \tag{51}$$

where a, b, N are the algorithmic parameters of the COS method that control the numerical precision, and $V_w(K)$ denotes the option payoff series coefficients, which can be known analytically for European options (see Fang and Oosterlee, 2009).

The Heston model with double-exponential jumps. The Heston (1993) model can be extended to incorporate a jump process in the asset price dynamics. We consider an extension of the Heston (1993) model with a double-exponential jump process (Kou, 2002):

$$dS_t = rS_t\,dt + \sqrt{v_t}S_t\,dW_t^1 + S_t(1 - e^{J_t})dN_t, \tag{54}$$

$$dv_t = \kappa\left(v_\theta - v_t\right)\,dt + \sigma_v\sqrt{v_t}\,dW_t^2, \tag{55}$$

$$\langle dW^1, dW^2\rangle_t = \rho dt, \tag{56}$$

where the process J_t represents the size of relative price jumps, which has a double-exponential probability density defined by:

$$f_J(y) = p\eta_1 e^{-\eta_1 y}\mathbb{1}_{\{y\geq 0\}} + (1-p)\eta_2 e^{-2\eta_2 y}\mathbb{1}_{\{y<0\}}, \tag{57}$$

where $p \in [0,1]$ is the probability of a positive jump in the asset return ($1-p$ is the probability of a negative jump), η_1 is the size of a positive jump, and η_2 is the size of a negative jump. N_t is a Poisson process of rate λ that controls the arrival time of jumps. Such an extension further adds four extra parameters to the Heston model.

The Heston model with Kou's double exponential jumps still lies within the affine class, and hence the log-return characteristic function is the product of the Heston model characteristic function, the equation (47), and the characteristic function for the jump component:

$$\psi_\tau^J(u; \phi \in \{p, \eta_1, \eta_2, \lambda\}) = \exp\left[\lambda\tau\left(\frac{p\eta_1}{\eta_1 - iu} + \frac{(1-p)\eta_2}{\eta_2 + iu} - 1\right)\right]. \tag{58}$$

The complete log-return characteristic function for this model becomes $\psi_\tau(u; \phi \in \{v_\theta, v_0, \sigma_v, \rho, \kappa\}) \times \psi_\tau^J(u; \phi \in \{p, \eta_1, \eta_2, \lambda\})$, and we still rely on the COS method to evaluate the option prices given the characteristic function. This gives rise to a parametric format g_ϕ representation:

$$g^{\text{HSV}}(\mathbf{X}^{\text{SK}}; \phi \in \{v_\theta, v_0, \sigma_v, \rho, \kappa, p, \eta_1, \eta_2, \lambda\}) = \frac{1}{2}e^{-r\tau} \tag{59}$$

$$\times\, \mathfrak{Re}\left\{\psi_\tau\left(0; S, r, v_\theta, v_0, \sigma_v, \rho, \kappa\right) \times \psi_\tau^J(0; p, \eta_1, \eta_2, \lambda)\right\} V_0(K) \tag{60}$$

$$+e^{-r\tau}\sum_{w=1}^{N-1}\mathfrak{Re}\left\{\psi_\tau\left(\frac{w\pi}{b-a}; S, r, v_\theta, v_0, \sigma_v, \rho, \kappa\right)\psi_\tau^J\left(\frac{w\pi}{b-a}; p, \eta_1, \eta_2, \lambda\right)e^{-iw\pi\frac{a}{b-a}}\right\} V_w\left(K\right). \tag{61}$$

3.2 Semi-parametric Specifications

Semi-parametric format structured-knowledge representations parameterize the intractable theory-driven models using DSNNs. The DSNNs then serve as the differentiable function g_ϕ for SKINNs.

The DSNN Heston's stochastic volatility model. We draw 5 million instances of Heston's model inputs independently from a multivariate uniform distribution. For each of the drawn instances, we query the European call option price using the Heson's SDEs, as described in Equation (44)-(46) with a Monte-Carlo simulation of 1,000 runs. A DSNN, $f^{\text{surrogate}}_{\tilde{\theta}}$, is pre-trained based on the drawn inputs and the queried option prices. $\tilde{\theta}$ denotes the learned DSNN parameters. This gives rise to a semi-parametric format g_ϕ representation:

$$g^{\text{DSNN-HSV}}(\mathbf{X}^{\text{SK}}, \tilde{\theta}; \phi \in \{v_\theta, v_0, \sigma_v, \rho, \kappa\}) = f^{\text{surrogate}}_{\tilde{\theta}}(\mathbf{X}^{\text{SK}}, \phi), \tag{62}$$

where the inputs $v_\theta, v_0, \sigma_v, \rho, \kappa$ of the DSNN become the learnable latent economic parameters in this SKINN specification.

The DSNN non-affine stochastic volatility model. We also consider a non-affine stochastic volatility SDE system, where there is no known characteristic function. This system is nested in the non-affine jump-diffusion described in Equation (3)-(5):

$$dS_t = rS_t dt + \sqrt{v_t} S_t dW^1_t, \tag{63}$$

$$dv_t = \kappa(v_\theta - v_t)dt + \sigma_v v_t^{\gamma_v} dW^2_t, \tag{64}$$

$$\langle dW^1, dW^2 \rangle_t = \rho dt. \tag{65}$$

We draw 50 million instances of $\{m, r, \tau, v_\theta, v_0, \sigma_v, \rho, \kappa, \gamma_v\}$ in this case from a multivariate uniform distribution. For each of the drawn instances, we again query the European call option prices using the SDEs, Equations (63)-(64), with a Monte-Carlo simulation of 1,000 runs. Since there are more inputs for the non-affine stochastic volatility model than the Heston model, it requires a larger number of instances to span the input space. Once the DSNN for this model is pre-trained, it gives rise to a semi-parametric format g_ϕ representation:

$$g^{\text{DSNN-NASV}}(\mathbf{X}^{\text{SK}}, \tilde{\theta}; \phi \in \{v_\theta, v_0, \sigma_v, \rho, \kappa, \gamma_v\}) = f^{\text{surrogate}}_{\tilde{\theta}}(\mathbf{X}^{\text{SK}}, \phi), \tag{66}$$

where the inputs $v_\theta, v_0, \sigma_v, \rho, \kappa, \gamma_v$ of the DSNN become the learnable latent economic parameters in this SKINN specification.

3.3 Non-parametric Specifications

We proceed to consider two scenarios where parametric and semi-parametric formats become infeasible, both of which are characterized by an unspecified number (potentially infinite) of latent parameters. In these scenarios, the structured knowledge provided by theory is generally at a high level of abstraction, or is even abstracted by a machine, such as auto-encoders. SKINNs can still effectively solve the estimation problems in these potentially very high-dimensional settings.

The martingale option pricing approach. The non-parametric martingale option pricing approach (MOPA) to construct structured knowledge is to utilize Equation (11). The probabilities $\mathbb{Q}(z_1), \mathbb{Q}(z_2), \cdots, \mathbb{Q}(z_{N_z})$ govern the conditional expectation of the underlying asset terminal price at one specific terminal time T. In practice, an option pricing model is estimated with multiple maturities $T_h, 1 \leq h \leq s, \tau_h = T_h - t$. Therefore, we design a risk-neutral probability matrix:

$$\mathbb{Q}_{s\times q} = \begin{bmatrix} \mathbb{Q}_{\tau_1,1} & \mathbb{Q}_{\tau_1,2} & \cdots & \mathbb{Q}_{\tau_1,q} \\ \mathbb{Q}_{\tau_2,1} & \mathbb{Q}_{\tau_2,2} & \cdots & \mathbb{Q}_{\tau_2,q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{Q}_{\tau_s,1} & \mathbb{Q}_{\tau_s,2} & \cdots & \mathbb{Q}_{\tau_s,q} \end{bmatrix}, \tag{67}$$

where $S^{(i)}_{\tau_h}, 1 \leq i \leq q, 1 \leq h \leq s$ is the i-th state of nature for the asset price at time T_h. In our implementation, we assume that different maturities share a common possible states of terminal asset price $\{0.5 \times S_t, \cdots, 1.5 \times S_t\}$ which is an array with 200 equally-spaced elements, ranging from the half, to the one and a half, of the current asset price S_t. We assume 10 time-to-maturities $\{0.1, 0.2, \cdots, 1.0\}$.[16] The risk-neutral probability matrix $\mathbb{Q}_{s\times q}$ thus contains 2,000 learnable latent parameters.

For maturity T_h, there are N_h options with different strike prices $K_1, K_2, \cdots, K_{N_h}$, in ascending order. The terminal payoffs, considering all the states of terminal asset prices, are known analytically as:

$$\mathbf{V}_{T_h} = \begin{bmatrix} \left(S^{(1)}_{T_h} - K_1\right)^+ & \left(S^{(2)}_{T_h} - K_1\right)^+ & \cdots & \left(S^{(q)}_{T_h} - K_1\right)^+ \\ \left(S^{(1)}_{T_h} - K_2\right)^+ & \left(S^{(2)}_{T_h} - K_2\right)^+ & \cdots & \left(S^{(q)}_{T_h} - K_2\right)^+ \\ \vdots & \vdots & \ddots & \vdots \\ \left(S^{(1)}_{T_h} - K_{N_h}\right)^+ & \left(S^{(2)}_{T_h} - K_{N_h}\right)^+ & \cdots & \left(S^{(q)}_{T_h} - K_{N_h}\right)^+ \end{bmatrix}, \tag{68}$$

[16] In practice, 10 unknown distributions over the 200 equally-spaced terminal asset prices are sufficient to embed the MOPA-type of structured knowledge. In any chance that the time-to-maturities in $\mathbf{X}^{\text{SK}}_{\text{grid}}$ falls between the elements in the array of 10 time-to-maturities, we find their closest counterparts and apply the corresponding distributions.

for example, for European call options. We stack the terminal payoff matrix for each maturity as a block-diagonal matrix:

$$\mathbf{V}_{N\times sq} = \begin{bmatrix} \mathbf{V}_{T_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{T_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{T_s} \end{bmatrix}. \tag{69}$$

This leads to a non-parametric format g_ϕ representation with unknown underlying distributions:

$$g^{\text{MOPA}}(\mathbf{X}^{\text{SK}}; \phi \in \{\mathbb{Q}_{\tau_h,j}\}_{1\leq h\leq s, 1\leq j\leq q}) = D_t(T) \times \mathbf{V} \times \text{flatten}(\mathbb{Q}), \tag{70}$$

where the probabilities in the vector ϕ are learnable and determine the distributions non-parametrically, $D_t(T)$ is a vector of discount factors. We flatten the matrix $\mathbb{Q}_{s\times q}$ to match the shape for the multiplication. The matrix $\mathbb{Q}_{s\times q}$ is passed to a row-wise softmax operation at each training iteration of a SKINN, to ensure the regularity conditions of probability distributions.

The auto-encoder-derived model. SKINNs can also be embedded with the structured knowledge that is entirely derived non-parametrically from data by an AE. In this case, economic models with presumed underlying latent parameters, e.g., the Black-Scholes-Merton and the Heston model, are not necessarily needed. Instead, the unknown latent parameters ϕ_{AE}, and the unknown parameterization for $g_{\phi_{\text{AE}}}(\mathbf{X}^{\text{SK}})$, are learned from only the target data, $\mathcal{D}_{\text{AE}} = \hat{\mathbf{y}}^{\text{SK}}$, with AEs.

The target data $\mathcal{D}_{\text{AE}}$ can be directly from the real-world observations, or from simulated data based on expert knowledge. We choose to simulate using a basket of theoretical models, e.g., the Black-Scholes-Merton model, the Heston model, and its extension with double-exponential jumps, plus some random noise. The simulated $\mathcal{D}_{\text{AE}}$ then cannot be represented by any existing model, but is rather based on a mixture of models, whose data-generating process is unknown. One can design other methods to form $\mathcal{D}_{\text{AE}}$, or simply use the market observations, as long as the target data can be regarded as the outcomes of an unknown data-generating process. There are 200 options on $\{(m_i, \tau_i)\}_{i=1}^{200}$, from each cross-section we simulate.[17] The prices become the 200-dimensional inputs of an AE. We simulate 300 thousand such inputs, 10 thousand from each model in the basket. The middle layer of the AE, a.k.a. the bottleneck, can be of an arbitrary dimension, but much lower-dimensional than the inputs, with the assumption that the unknown latent parameters should have a sparse structure.

The observable features $\mathbf{X}^{\text{SK}}$ are not required by an AE. The decoder reconstructs $\hat{\mathbf{y}}^{\text{SK}}$ in the

[17] We simulate option prices on a 2D mesh-grid with 20 moneyness from an equally-spaced array $[0, \cdots, 2.0]$, and 10 time-to-maturities from another equally-spaced array $[0, \cdots, 1]$.

same order as the input, given some values for the unknown latent parameters ϕ_{AE}.[18] One can use the entire AE as the function g_ϕ, in which case the encoder receives the option price predictions by the neural network component of a SKINN on the mesh grids, $\{m_i, \tau_i\}_{i=1}^{200}$, and compresses them to ϕ_{AE}. The decoder then returns the structured-knowledge outputs. This actually unrolls the learning of ϕ_{AE} to the forward- and backward-pass of the AE.[19] Alternatively, one can use only the decoder as the structured-knowledge representation, by treating ϕ_{AE} as the learnable latent economic parameters. An AE therefore provides a flexible non-parametric format g_ϕ representation:

$$g^{\text{AE-MIX}}(\mathbf{X}^{\text{SK}}) = (f_\theta^{\text{enc}} \circ f_\theta^{\text{dec}})(\hat{\mathbf{y}}^{\text{SK}}), \tag{71}$$

or, alternatively, by using only the decoder part:

$$g^{\text{AE-MIX}}(\mathbf{X}^{\text{SK}}; \phi \in \{\phi_{\text{AE}}^{[1]}, \phi_{\text{AE}}^{[2]}, \cdots, \phi_{\text{AE}}^{[\phi_d]}\}) = f_\theta^{\text{dec}}(\mathbf{X}^{\text{SK}}, \phi_{\text{AE}}^{[1]}, \phi_{\text{AE}}^{[2]}, \cdots, \phi_{\text{AE}}^{[\phi_d]}).$$

3.4 Benchmark Neural Network Models

While SKINNs allow to embed structured-knowledge in a variety of representations, the existing hybrid models in the literature are less flexible. They only manage to incorporate static, model-free constraints, and are prohibitive to sophisticated, higher-dimensional structures. In this section, we revise some popular hybrid models. These models serve as the benchmark models for the evaluation of SKINNs in our later empirical findings section.

3.4.1 Neural Networks with Model-free Constraints

The model-free shape prior, such as the monotonicity and convexity of option prices surfaces with respect to strike prices (or, equivalently, moneyness) and maturities, can be enforced to a neural network using derivative regularizations (see, for example, Dugas et al., 2000, 2009; Ackerer et al., 2020; Chataigner et al., 2020). Specifically, for European call options, the following penalizations are used: (i) strike price monotonicity, i.e., $\partial_K f_\theta(\mathbf{X}) \leq 0 \rightarrow \|\max(\partial_K f_\theta(\mathbf{X}), 0)\|_2$; (ii) strike price convexity, i.e., $\partial_K^2 f_\theta(\mathbf{X}) \leq 0 \rightarrow \|\max(\partial_K^2 f_\theta(\mathbf{X}), 0)\|_2$; (iii) maturity monotonicity, i.e., $\partial_\tau f_\theta(\mathbf{X}) \leq 0 \rightarrow \|\max(\partial_\tau f_\theta(\mathbf{X}), 0)\|_2$, which are summed together as a shape-derivative regularization term:

$$\mathcal{L}_{\text{ShapeDer}}(\theta; \mathbf{X}) := \|\max(\partial_K f_\theta(\mathbf{X}), 0)\|_2 + \|\min(\partial_K^2 f_\theta(\mathbf{X}), 0)\|_2 + \|\min(\partial_\tau f_\theta(\mathbf{X}), 0)\|_2. \tag{72}$$

We refer to the neural networks trained with the regularization (72) as the ShapeDerNN. The

[18] One can also choose to incorporate $\mathbf{X}^{\text{SK}}$ into the bottleneck, and switch the grid-based evaluation of the decoder to the point-wise evaluation (see, for example, in Bergeron et al., 2021).

[19] By unrolling the learning of ϕ_{AE}, there is no need to set up the learnable latent parameters in the SKINN.

partial differential terms are effectively computed using automatic differentiation, similar to PINNs. Hence, ShapeDerNN inherently suffers from pathological gradient dynamics. Additionally, the static and model-free shape constraints offer no structured economic knowledge that is important for improving the model's generalizability. The gradient pathologies can be addressed by casting the partial differentiations to a quadratic programming operation (see, for example, Cohen et al., 2020), for which we refer the neural networks to the ShapeQPNN. The technical details of ShapeQPNN can be found in the appendix.

3.4.2 Neural Networks with PDE Constraints

PINNs incorporate the economic knowledge into neural networks through a differentiable PDE regularization term, which is a non-linear partial differential operation of the neural networks:

$$r(\theta; t, \mathbf{X}, \phi_{\text{PDE}}) = \partial_t f_\theta(t, \mathbf{X}) + \mathfrak{N}_{\mathbf{X}}[f_\theta(t, \mathbf{X}); \phi_{\text{PDE}}], \tag{73}$$

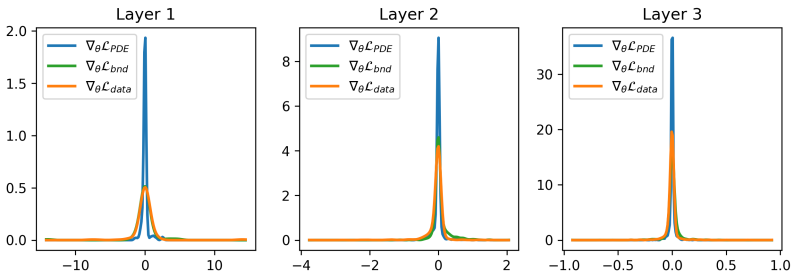
$$\mathcal{L}_{\text{PDE}}(\theta; t, \mathbf{X}, \phi_{\text{PDE}}) = \|r(\theta; t, \mathbf{X}, \phi_{\text{PDE}})\|_2, \tag{74}$$

where the first term in $r(\theta; t, \mathbf{X}, \phi_{\text{PDE}})$ is the neural network derivative against the time coordinate t, and the second term is the neural network derivatives against the space coordinate $\mathbf{X}$. The non-linear partial differential operation is specified and parameterized by a PDE provided by theories, with ϕ_{PDE} being the PDE parameters. A PINN typically aims to find the data-driven solution to the PDE by minimizing the PDE loss $\mathcal{L}_{\text{PDE}}$, given that the PDE parameters ϕ_{PDE} are known (see, Raissi et al., 2017a). This approach is also known as the forward-problem PINN, as ϕ_{PDE} is determined. In the financial context, it can be costly to obtain a reliable estimation of ϕ_{PDE} a priori, and one might have to learn it together with market data by using the inverse-problem PINN (see, Raissi et al., 2017b).

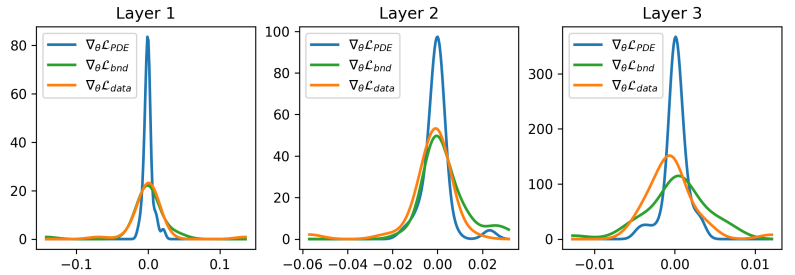
Option pricing models can be formulated as PDEs. Consider the Black-Scholes-Merton model, which has a PDE formulation:

$$-\partial_\tau V + rS\partial_S V + \frac{1}{2}\sigma^2 S^2 \partial_{SS} V - rV = 0, \tag{75}$$

A PINN $f_\theta(t, \mathbf{X})$ is trained to approximate the Black-Scholes PDE solution $V(t, \mathbf{X})$. This approach, however, faces several challenges in option pricing. The first and foremost challenge is the vanishing gradient pathologies due to the deep chain-rule computation when back-propagating through the PDE loss, which leads to problematic gradient dynamics as documented in Wang et al. (2021, 2022). We illustrate these pathologies in Figure (6), with some simulated data. Meanwhile, the gradient dynamics of a SKINN do not suffer from this problem, as shown in Figure (7).

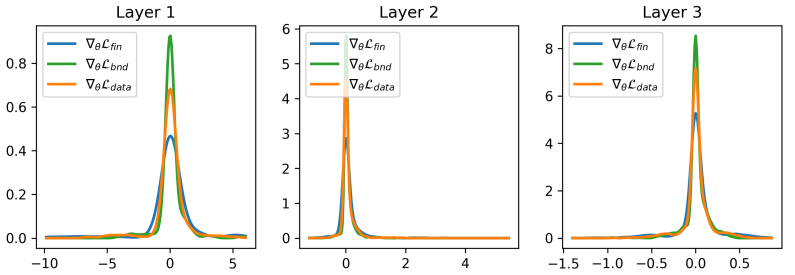


(a) Loss gradients with respect to the neural network weights.

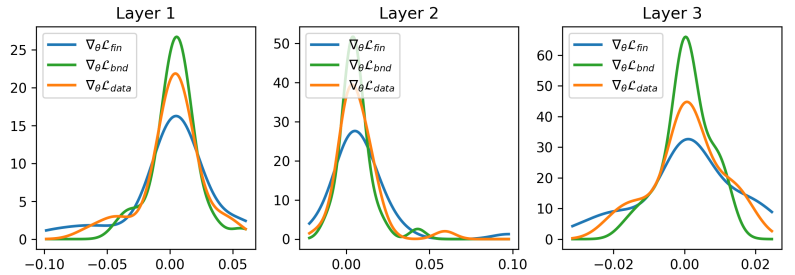


(b) Loss gradients with respect to the neural network biases.

Figure 6: The probability density of the loss gradients with respect to the neural network weights and biases, for each loss component in a PINN (with inverse problem approach), including a boundary loss. The PINN is trained, for 5,000 iterations, using a cross-section of simulated European call option prices with noise, according to the Black-Scholes-Merton model.



(a) Gradients with respect to the neural network weights.



(b) Gradients with respect to the neural network biases.

Figure 7: The probability density of the loss gradients with respect to the neural network weights and biases, for each loss component in a SKINN, including a boundary loss. The SKINN is trained, for 5,000 iterations, using a cross-section of simulated European call option prices with noise, with the structured-knowledge representation of the Black-Scholes-Merton model.

One other challenge to PINNs for financial tasks is that PINNs can not handle the PDEs where there are unobservable differentiation variables. A practical example is the Heston PDE, which is formulated as:

$$
\begin{aligned}
&-\partial_\tau V + \frac{1}{2} v S^2 \partial_{SS} V + \rho \sigma v S \partial_{Sv} V \\
&\quad + \frac{1}{2} \sigma^2 v \partial_{vv} V + r S \partial_S V + \left[\kappa \left(\theta - v \left(t \right) \right) - \lambda \left(S, v, t \right) \right] \partial_v V - r V = 0.
\end{aligned} \tag{76}
$$

The instantaneous variance v is not observable, and it is impossible to evaluate the partial differential terms $\partial_{Sv} f_\theta(t, \mathbf{X}), \partial_{vv} f_\theta(t, \mathbf{X}), \partial_v f_\theta(t, \mathbf{X})$ from the Heston's PDE with market data. There are many more such PDEs in finance that are prohibitive for PINNs to learn, especially when the system is multivariate with high-dimensional latent parameters.

3.4.3 Transfer-Learning Neural Networks

Transfer learning models attempt to incorporate the knowledge learned from a source domain into a related task with a different dataset, also called a target domain. The knowledge from the source domain is expected to improve the generalizability of the model in the task domain. Chen et al. (2023) apply this method to the option pricing problem. Given a structural option pricing model $g(\mathbf{X}; \phi)$, transfer learning pre-trains a deep surrogate model that can decently approximate $g(\mathbf{X}; \phi)$ in a first step. This step is the same as the construction of the semi-parametric format structured-knowledge representations for SKINNs, by pre-training a deep neural network on a large simulated source domain dataset. In a second step, the knowledge of the pre-trained model is transferred to learn from the target domain dataset by slightly updating the parameters of the pre-trained surrogate neural network.

The transfer learning method faces some limitations. Since the transfer-learning neural networks need both the observable features $\mathbf{X}^{\text{SK}}$ and the latent parameters ϕ to calculate an output, a separate estimation of ϕ_{obs} for the target domain is needed before target domain learning. This makes it infeasible when the surrogate model requires high-dimensional latent parameters. In addition, one has the flexibility to choose how many parameters to update, e.g., updating the parameters from all layers, which nevertheless may result in catastrophic forgetting; or updating the parameters from some ending layers before the final output layer, but which can reduce the expressivity.

[20] "No" in the column structured knowledge means that there is no prior knowledge; "Bnd" means that the boundary conditions are incorporated; "MF" means that the model-free constraints are incorporated; "MF+Bnd" means that both the boundary and the model-free constraints are incorporated; "BS" means the Black-Scholes-Merton model; "HSV" means the Heston model; "AHBS" means the ad-hoc Black-Scholes-Merton model; "HSVKDEJ" means the Heston model with Kou's double exponential jump; "DSNN-HSV" means the deep surrogate neural network for HSV; "DSNN-NASV" means the deep surrogate neural network for the non-affine stochastic volatility model; "MOPA"

Table 1: The full list of the neural network models we consider in this paper. Only the PINNs, the TLNNs, and the SKINNs introduce the structural model as prior knowledge from option pricing to the neural networks. Others introduce either the boundary constraints, the model-free shape constraints, or both to the neural networks.

Model	**Structured Knowledge**[20]	**Latent parameters dimension (learnable/total)**	**Activation function**
Benchmark models:			
NN	No	0/0	ReLU
NN+Bnd	Bnd	0/0	ReLU
ShapeDerNN	MF	0/0	SiLU
ShapeQPNN	MF	0/0	ReLU
ShapeQPNN+Bnd	MF+Bnd	0/0	ReLU
inv-PINN+BS	BS	1/1	SiLU
fwd-PINN+BS	BS	0/1	SiLU
fwd-TLNN+HSV	HSV	0/5	ReLU
Parametric SKINNs:			
SKINN+BS	BS	1/1	ReLU
SKINN+HSV	HSV	5/5	ReLU
SKINN+AHBS	AHBS	6/6	ReLU
SKINN+HSVKDEJ	HSVKDEJ	9/9	ReLU
SKINN+SABR	SABR	722/722	ReLU
Semi-parametric SKINNs:			
SKINN+DSNN-HSV	DSNN-HSV	5/5	ReLU
SKINN+DSNN-NASV	DSNN-NASV	6/6	ReLU
Non-parametric SKINNs:			
SKINN+MOPA	MOPA	2,000/2,000	ReLU
SKINN+AE-BS	AE-BS	2/2	ReLU

Table (1) lists all the neural network models for option pricing that we consider in this paper. The upper panel includes all the benchmark models that we compare our SKINNs with. The lower panel includes 8 variants of SKINN with different specifications of structured-knowledge representations, for which we elaborated their constructions in Section (3.1).

4 Empirical Findings

We implement our proposed SKINNs framework to learn from the S&P 500 index options with structured-knowledge representations, over a long time period, from 1996 to 2022, using daily transaction quotes. This allows SKINNs to be tested through several major recessions, which include the dot-com bubble, the 2007-2009 global financial crisis, and the COVID-19 crisis. We document statistically significant improvements of SKINNs in both the pricing and hedging capability, compared with benchmark models, especially in the out-of-sample data.

4.1 Data

We use the daily transaction quotes of the S&P 500 index options from OptionMetrics. The sample period, to be specific, starts from 04 January 1996 and ends on 31 December 2022. Instead of using the standardized option dataset from OptionMetrics, where options are recorded on a regular meshgrid of strike prices and maturities, and are smoothed by a proprietary kernel smoothing algorithm, we use the raw option dataset. Though there are missing values and erroneous prices in the raw option dataset, which challenge data-driven models, it reflects the true market and also stress-tests the models.

We consider only European call options in this paper for simplicity, but all models can be conveniently adapted to European put options. Additionally, for each transaction day, we include only the options that expire after 7 calendar days but before 365 calendar days, as these options are more liquid and are expected to contain more useful information than others.

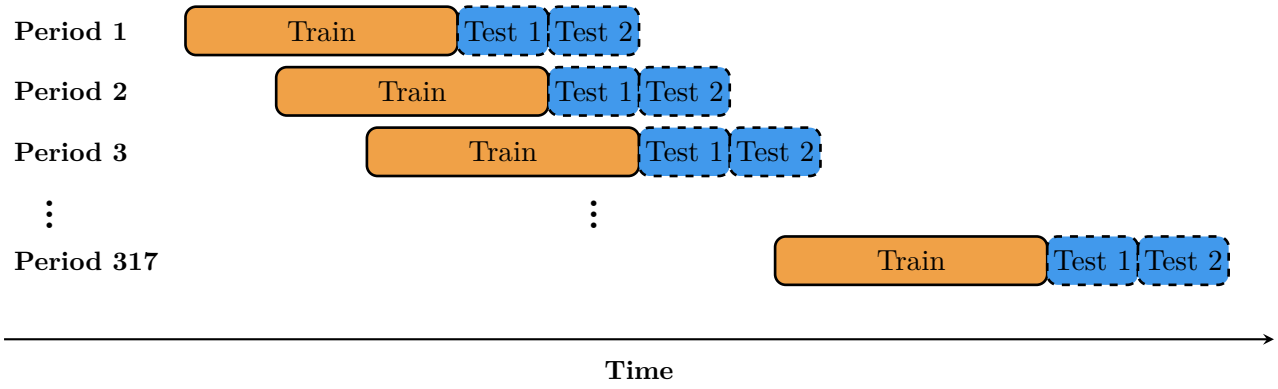
4.2 Configurations, Training and Testing Schedule

To ensure a fair comparison across different models, we apply the same configuration for the neural networks. All neural networks have a fully-connected, feed-forward architecture, with 3 hidden layers and 32 neurons in each hidden layer, except for the neural networks of g_ϕ. To ensure that all neural networks start with the same parameters before training, we initialize them with a

means the martigale option pricing approach; and "AE-BS" means the autoencoder trained with the noisy option price cross-sections based on the Black-Scholes-Merton model.

fixed random seed. For the models that do not require partial differentiation operations, we use the ReLU as the activation function; and for others, e.g., ShapeDerNN, PINN, we use the SiLU, i.e., the sigmoid linear unit, $h(x) = \frac{x}{1+\exp(-x)}$, as the activation function, which, according to Chen et al. (2023), helps to alleviate the pathological gradient problem.

Figure 8: The forward rolling model training schedule. The first training period covers the daily option panels from 04 Jan 1996 to 29 Mar 1996. The corresponding shorter test horizon (Test 1) covers options from 01 Apr 1996 to 30 Apr 1996, and the longer test horizon (Test 2) covers options from 01 May 1996 to 31 May 1996. For each of the periods from period 2 to period 317, all the Train, Test 1, and Test 2 are rolled one month forward based on the preceding period.



Starting from 04 January 1996, we train models using a panel of three-month options data, on a forward rolling basis with a one-month rolling window. That is, the starting date of a training period is always one month later than the starting date of its preceding training period. Once the models are trained using a three-month panel of European call options of the S&P 500 index, we test their out-of-sample pricing and hedging performance using the options in the following two consecutive months. We refer to the first testing months as the shorter prediction horizons, and the second testing months as the longer prediction horizons. Options within the longer prediction horizons are more difficult to price, as the data-generating process in these periods can be changed drastically relative to the corresponding training periods. In total, there are 317 model training and testing periods (317 iterations of shorter and longer prediction horizons) from our option dataset. Figure (8) illustrates the training and testing schedule with the forward rolling basis.

4.3 Out-of-Sample Model Performance

We evaluate the out-of-sample option pricing and hedging capability using the Diebold and Mariano (2002) test. This test is a statistical test to evaluate the out-of-sample predictive accuracy between two forecasting models, and is adopted by Gu et al. (2020) to evaluate the machine learning

based asset pricing models. We denote e_j^1 and e_j^2 the out-of-sample pricing or hedging error of two candidate models, model 1 and model 2, respectively, for the period j, $1 \leq j \leq 317$. The Diebold-Mariano test statistically compares the out-of-sample model error differences $d_j = \{e_j^1 - e_j^2\}_{j=1}^{317}$ with a zero series. Since we aim to test whether model 1 provides a statistically smaller out-of-sample error than model 2, we use the one-sided test with the null hypothesis that d_j is distributed with an expectation greater than zero. For robustness check, we also perform the non-parametric Wilcoxon (1945) signed-rank test, for which we report the results in Appendix (A).

4.3.1 Option Pricing Performance

While there is a wide range of error metrics to measure the option pricing accuracy of a model, e.g., mean squared error, median absolute error, mean absolute percentage error, median absolute percentage error (see Ruf and Wang, 2019), we use the root mean squared error (RMSE) throughout the paper. The choice of different error metrics makes an insignificant difference.

Shorter prediction horizons. We first consider the option pricing performance in the shorter prediction horizons. Table (2) reports Diebold-Mariano test statistics for pairwise option pricing error comparisons of a column model versus a row model. A negative Diebold-Mariano test statistic indicates that the column model outperforms (has a lower option pricing error than) the row model, and vice versa.

Without any prior domain knowledge from the option pricing theory, NN in fact provides decent out-of-sample pricing performance, in the shorter prediction horizons. From Table (2), it outperforms NN+MF and inv-PINN+BS, two models that require non-linear differentiation operations and hence suffer from the gradient pathologies, and also outperforms the classical structural option pricing models, including BS, AHBS, and HSV.

It is not surprising that a completely data-driven NN statistically outperforms completely theory-driven structural models. This is actually the advantage of neural networks, as the over-parameterization and the high non-linear nature allow for capturing the predictive option data patterns that are not easily incorporated by rigid structural models, especially within the shorter prediction horizons in which data matters more. Adding the boundary conditions further improves the out-of-sample pricing performance of NN.

However, both NN and NN+Bnd are outperformed by SKINN+HSV, SKINN+HSVKDEJ, SKINN+MOPA, and SKINN+AE-BS. This implies that SKINNs with more sophisticated g_ϕ tend to benefit the out-of-sample option pricing performance, when there is a slight shift between the

training and the evaluation dataset.[21] One explanation is that, in this case, the patterns uncovered by data-driven models already overlap the patterns that are prescribed in the relatively simple structured-knowledge representations g_ϕ, and hence only more sophisticated g_ϕ can provide marginal improvements.

Longer prediction horizons. We then consider the option pricing performance in the longer prediction horizons. In this case, the performance of the data-driven models decay significantly, as it becomes difficult for them to learn the generalizable predictive information, as the patterns in the training dataset can be shifted significantly in the evaluation dataset. Table (3) reports pairwise test statistics from Diebold-Mariano tests. Again, NN+MF and inv-PINN+BS are outperformed by other models due to their gradient pathologies. In the longer horizons, NN fails to outperform structural models. Particularly, it statistically underperforms BS and the AHBS. This confirms that the option price patterns uncovered by data-driven models become less effective for the out-of-sample options in the longer prediction horizons. Adding the boundary conditions still helps to improve the performance of NN, but it can not outperform structural models at this time. In this longer prediction horizon case, all variants of SKINNs statistically outperform NN and NN+Bnd, at the 5% significance level at least. This indicates that, by embedding the structured knowledge from theories, SKINNs offer significantly improved generalizability, especially when there is a considerable shift in the data patterns.

4.3.2 Option Hedging Performance

Another economic purpose of option pricing models is hedging. For structural models, hedging is straightforward, as the option Greeks, such as Delta, can be effectively calculated using a closed-form or semi-closed-form solution. We use all the considered models to delta-hedge the short positions in the S&P 500 call options, as an evaluation of the option hedging performance. While it is straightforward to derive the option Delta for structural models, we have to use the automatic differentiation technique to compute the option Delta for the neural network models. Since the neural networks take moneyness, $m = K/S$, as one of the inputs, according to the homogeneous-of-degree-one property of an option pricing model with respect to S and K, the option Delta for a neural network can be defined as:

$$\Delta_{\text{NN}} := \frac{\partial f_\theta(\mathbf{X})}{\partial m}\frac{K}{S^2}, \tag{77}$$

where $\frac{\partial f_\theta(\mathbf{X})}{\partial m}$ is computed using automatic differentiation, and $\frac{K}{S^2}$ are known from the inputs.

[21] The option price patterns in the shorter horizons are more similar to the patterns in the training samples than the patterns in the longer horizons.

Table 2: This table reports pairwise Diebold-Mariano test statistics comparing the out-of-sample option pricing performance, measured in terms of RMSE, of different models. We compare 8 variants of SKINNs with different specifications of structured-knowledge representations, with 4 classical structural models, and 4 benchmark neural networks. The out-of-sample option pricing performance is evaluated in the 317 shorter prediction horizons. We square all the pricing errors to penalize large errors.

Model	**Structural models**				**Benchmark neural networks**			
Panel (A)	BS	AHBS	HSV	HSVKDEJ	NN	NN +Bnd	NN +MF	inv-PINN +BS
BS	–	1.99**	-2.16**	-9.24***	-1.72**	-2.42***	0.16	0.21
AHBS	–	–	-2.89***	-8.91***	-1.88**	-2.58***	-0.07	-0.02
HSV	–	–	–	-12.95***	-1.49*	-2.22**	0.59	0.64
HSVKDEJ	–	–	–	–	-0.61	-1.35*	1.81**	1.85**
NN	–	–	–	–	–	-2.41***	3.01***	2.79***
NN+Bnd	–	–	–	–	–	–	4.49***	4.26***
NN+MF	–	–	–	–	–	–	–	0.13
inv-PINN+BS	–	–	–	–	–	–	–	–

Model	**Parametric SK**				**Semi-parametric SK**		**Non-parametric SK**	
Panel (B)	SKINN +BS	SKINN +AHBS	SKINN +HSV	SKINN +HSVKDEJ	SKINN +DSNN-HSV	SKINN +DSNN-NASV	SKINN +MOPA	SKINN +AE-BS
BS	-4.45***	-4.79***	-4.58***	-5.27***	-3.47***	-4.20***	-4.18***	-2.98***
AHBS	-4.67***	-5.03***	-4.80***	-5.51***	-3.68***	-4.43***	-4.38***	-3.17***
HSV	-4.40***	-4.76***	-4.64***	-5.44***	-3.40***	-4.20***	-4.18***	-2.84***
HSVKDEJ	-2.80***	-2.94***	-3.09***	-3.66***	-1.98**	-2.54***	-2.97***	-1.73**
NN	-1.27	-1.16	-1.65**	-1.68**	-0.88	-0.99	-2.42***	-1.79**
NN+Bnd	-0.24	-0.11	-0.57	-0.63	0.15	0.01	-1.16	-0.06
NN+MF	-3.92***	-4.10***	-4.53***	-4.62***	-3.30***	-3.66***	-4.70***	-4.34***
inv-PINN+BS	-3.76***	-3.88***	-4.26***	-4.37***	-3.16***	-3.48***	-4.46***	-4.03***

Table 3: This table reports pairwise Diebold-Mariano test statistics comparing the out-of-sample option pricing performance, measured in terms of RMSE, of different models. We compare 8 variants of SKINNs with different specifications of structured-knowledge representations, with 4 classical structural models, and 4 benchmark neural networks. The out-of-sample option pricing performance is evaluated in the 317 longer prediction horizons. We square all the pricing errors to penalize large errors.

Model	**Structural models**				**Benchmark neural networks**			
Panel (A)	BS	AHBS	HSV	HSVKDEJ	NN	NN +Bnd	NN +MF	inv-PINN +BS
BS	–	2.49***	0.62	-5.23***	2.67***	1.36*	2.51***	2.54***
AHBS	–	–	-0.57	-5.83***	2.56***	1.26	2.40***	2.42***
HSV	–	–	–	-12.95***	2.63***	1.33*	2.45***	2.47***
HSVKDEJ	–	–	–	–	3.09***	1.74**	2.94***	2.97***
NN	–	–	–	–	–	-2.51***	-0.44	-0.38
NN+Bnd	–	–	–	–	–	–	1.30*	1.14
NN+MF	–	–	–	–	–	–	–	0.01
inv-PINN+BS	–	–	–	–	–	–	–	–

Model	**Parametric SK**				**Semi-parametric SK**		**Non-parametric SK**	
Panel (B)	SKINN +BS	SKINN +AHBS	SKINN +HSV	SKINN +HSVKDEJ	SKINN +DSNN-HSV	SKINN +DSNN-NASV	SKINN +MOPA	SKINN +AE-BS
BS	-3.61***	-3.61***	-3.06***	-3.56***	-2.09**	-2.78***	-2.53***	0.04
AHBS	-3.85***	-3.87***	-3.30***	-3.80***	-2.31**	-3.01***	-2.74***	-0.10
HSV	-3.89***	-3.91***	-3.37***	-3.89***	-2.33**	-3.08***	-2.74***	-0.03
HSVKDEJ	-2.69***	-2.63***	-2.23**	-2.69***	-1.26	-1.88**	-1.76**	0.56
NN	-3.98***	-3.95***	-3.83***	-4.02***	-3.51***	-3.70***	-3.83***	-3.61***
NN+Bnd	-2.72***	-2.69***	-2.63***	-2.77***	-2.28**	-2.44***	-2.61***	-2.54***
NN+MF	-3.55***	-3.57***	-3.44***	-3.66***	-3.06***	-3.27***	-3.43***	-3.05***
inv-PINN+BS	-3.50***	-3.53***	-3.42***	-3.62***	-3.02***	-3.23***	-3.41***	-2.82***

To evaluate the out-of-sample delta-hedging performance, we initialize a delta-hedged portfolio on the trading date t_j within the evaluation horizon, by entering the following positions:

$$\Pi^{(i)}_{\text{Stock}}(t_j) = S_{t_j}\Delta^{(i)}(t_j), \tag{78}$$

$$\Pi^{(i)}_{\text{Call}}(t_j) = -C^{(i)}_{t_j}, \tag{79}$$

$$\Pi^{(i)}_{\text{Bond}}(t_j) = -\left(\Pi^{(i)}_{\text{Stock}}(t_j) + \Pi^{(i)}_{\text{Call}}(t_j)\right), \tag{80}$$

where $1 \leq j \leq m$, and m is the total number of trading days in the prediction horizon; $1 \leq i \leq n_j$, and n_j is the total number of options from day t_j; $\Delta^{(i)}(t_j)$ is the Delta, estimated by a model, for the option i at day t_j; $\Pi^{(i)}_{\text{Stock}}(t_j)$ is the value of the underlying asset position; $\Pi^{(i)}_{\text{Bond}}(t_j)$ is the value of the zero-coupon bond position; $\Pi^{(i)}_{\text{Call}}(t_j)$ is the value of the short call option position that we aim to hedge. For all $t_j, 1 \leq j \leq m$, the initial delta-hedged portfolio perfectly hedges, as $\sum_{i=1}^{n_j} \Pi^{(i)}_{\text{Stock}}(t_j) + \Pi^{(i)}_{\text{Call}}(t_j) + \Pi^{(i)}_{\text{Bond}}(t_j) = 0$ by construction. We then evaluate the delta-hedging performance on the next day of the hedged portfolio construction, $t_j + 1 = t_{j+1}, 1 \leq j \leq m$. We calculate the next-day delta-hedged portfolio value by:

$$\Pi^{(i)}_{\text{Stock}}(t_j + 1) = S_{t_{j+1}}\Delta^{(i)}(t_j), \tag{81}$$

$$\Pi^{(i)}_{\text{Call}}(t_j + 1) = -C^{(i)}_{t_{j+1}}, \tag{82}$$

$$\Pi^{(i)}_{\text{Bond}}(t_j + 1) = \Pi^{(i)}_{\text{Bond}}(t_j)e^{r \times \frac{1}{252}}. \tag{83}$$

Considering all the options we hedge on t_j, the overall delta-hedge portfolio has the value:

$$\Pi(t_j + 1) = \frac{1}{n_j}\sum_{i=1}^{n_j} \Pi^{(i)}_{\text{Stock}}(t_j + 1) + \Pi^{(i)}_{\text{Call}}(t_{j+1}) + \Pi^{(i)}_{\text{Bond}}(t_j + 1), \tag{84}$$

on all the hedging performance evaluation date $t_j + 1, 1 \leq j \leq m$. For different models, we plug in the corresponding Delta hedge ratio $\Delta_{\text{Model}}(t_j)$ to construct and assess the hedged portfolios.

Ideally, a delta-hedged portfolio should hedge the risk of the underlying asset price fluctuation, and hence the next-day delta-hedged portfolio is expected to have a zero value, i.e., $\Pi(t_j + 1) = 0$. In reality, however, option pricing models have residual risk, and $\Pi(t_j + 1)$ can deviate from zero. To compare the out-of-sample delta-hedging performance of each model, we define the following hedging error metric:

$$\text{HE}^{\text{Model}} = \frac{1}{m}\sum_{j=1}^{m} \left|\Pi(t_j + 1)\right|. \tag{85}$$

We compare the out-of-sample hedging accuracy of models using Diebold-Mariano test with a series

of hedging error differences $d_j = \{e_j^1 - e_j^2; e_j^1 = \text{HE}_j^1, e_j^2 = \text{HE}_j^2\}_{j=1}^{317}$.

Shorter prediction horizons. Table (4) reports the pairwise Diebold-Mariano test statistics for the out-of-sample hedging performance comparisons, considering the shorter prediction horizons. Interestingly, in this test, NN is generally outperformed by classical structural models significantly. At this time, NN+Bnd fails to provide any improvements by adding the boundary conditions. All SKINNs significantly outperform all the benchmark neural network models in terms of hedging purpose, thanks to the embedding of the structured-knowledge representations. We observe that classical structural models are robust to the change of the evaluation objective, i.e., estimating them by minimizing the pricing errors insignificantly influences their hedging capability. However, the over-parameterization nature of neural networks makes them disadvantaged when changing the evaluation objective, as the weights and biases are optimized by minimizing the empirical option pricing errors, which can be distant from the optimal weights and biases by minimizing the empirical hedging errors. Though disadvantaged, unlike the benchmark neural networks, SKINNs do not underperform the structural models. For SKINN+BS and SKINN+AHBS, their hedging performance can even statistically outperform the structural models in the shorter prediction horizons.

Longer prediction horizons. We continue to test whether the hedging performance improvements by SKINNs are robust to longer prediction horizons. Table (5) reports the pairwise Diebold-Mariano test statistics in this case. Similar to the result for the shorter prediction horizons, all SKINNs continue to statistically outperform all the benchmark neural networks. The benchmark neural networks statistically underperform the classical structural models, as is expected for the longer prediction horizons, where the data-driven patterns are less effective. Again, in the longer prediction horizons, SKINN+BS and SKINN+AHBS not only outperform the benchmark neural networks but also outperform the structural models significantly.

4.4 SKINNs Versus Transfer Learning Models

In this section, we compare our SKINNs with the transfer-learning models to incorporate the option pricing knowledge in more detail. We choose Heston's model as the source of knowledge in this comparison because of its high dimensionality in the latent economic parameters and its model complexity. Chen et al. (2023), however, employs a very deep neural network to transfer from the one-parameter Black-Scholes model. To facilitate a fair comparison, we shrink the neural network size by using 6 hidden layers with 64 neurons each layer, for the transfer-learning models. By not going so deep, the neural network architecture we use does not suffer from the vanishing

Table 4: This table reports pairwise Diebold-Mariano test statistics comparing the out-of-sample option hedging performance, measured in terms of MHE, of different models. We compare 8 variants of SKINNs with different specifications of structured-knowledge representations, with 4 classical structural models, and 4 benchmark neural networks. The out-of-sample option hedging performance is evaluated in the 317 shorter prediction horizons. We square all the hedging errors to penalize large errors.

Model	**Structural models**				**Benchmark neural networks**			
Panel (A)	BS	AHBS	HSV	HSVKDEJ	NN	NN +Bnd	NN +MF	inv-PINN +BS
BS	–	-3.62***	6.63***	6.56***	2.64***	2.82***	2.99***	2.48***
AHBS	–	–	6.92***	6.83***	2.90***	3.19***	3.17***	2.66***
HSV	–	–	–	-1.31*	-1.65**	-2.59***	-0.12	-0.31
HSVKDEJ	–	–	–	–	-1.55*	-2.49***	0.01	-0.21
NN	–	–	–	–	–	-1.15	2.25**	1.11
NN+Bnd	–	–	–	–	–	–	2.50***	1.57*
NN+MF	–	–	–	–	–	–	–	-0.31
inv-PINN+BS	–	–	–	–	–	–	–	–
Model	**Parametric SK**				**Semi-parametric SK**		**Non-parametric SK**	
Panel (B)	SKINN +BS	SKINN +AHBS	SKINN +HSV	SKINN +HSVKDEJ	SKINN +DSNN-HSV	SKINN +DSNN-NASV	SKINN +MOPA	SKINN +AE-BS
BS	-6.58***	-6.34***	-0.16	-0.71	0.70	0.79	0.07	0.68
AHBS	-5.59***	-5.57***	1.11	0.63	1.75**	1.80**	1.44*	1.48*
HSV	-7.01***	-6.90***	-7.32***	-6.78***	-6.26***	-6.51***	-6.69***	-5.68***
HSVKDEJ	-6.88***	-6.77***	-7.11***	-6.56***	-6.06***	-6.43***	-6.66***	-5.61***
NN	-3.94***	-4.02***	-2.74***	-2.78***	-2.25**	-2.34***	-2.80***	-3.14***
NN+Bnd	-4.44***	-4.58***	-2.87***	-2.94***	-2.33**	-2.47***	-2.96***	-3.59***
NN+MF	-3.86***	-3.89***	-3.11***	-3.10***	-2.70***	-2.77***	-3.03***	-3.34***
inv-PINN+BS	-3.23***	-3.29***	-2.41***	-2.41***	-2.22**	-2.34***	-2.53***	-2.46***

Table 5: This table reports pairwise Diebold-Mariano test statistics comparing the out-of-sample option hedging performance, measured in terms of MHE, of different models. We compare 8 variants of SKINNs with different specifications of structured-knowledge representations, with 4 classical structural models, and 4 benchmark neural networks. The out-of-sample option hedging performance is evaluated in the 317 longer prediction horizons. We square all the hedging errors to penalize large errors.

Model	**Structural models**				**Benchmark neural networks**			
Panel (A)	BS	AHBS	HSV	HSVKDEJ	NN	NN +Bnd	NN +MF	inv-PINN +BS
BS	–	-4.30***	7.69***	7.67***	3.44***	3.48***	3.52***	3.20***
AHBS	–	–	7.83***	7.81***	3.68***	3.83***	3.72***	3.43***
HSV	–	–	–	-1.42*	0.25	-1.15	1.11	0.20
HSVKDEJ	–	–	–	–	0.39	-0.99	1.21	0.32
NN	–	–	–	–	–	-2.12**	1.40*	-0.05
NN+Bnd	–	–	–	–	–	–	2.76***	1.47*
NN+MF	–	–	–	–	–	–	–	-1.94**
inv-PINN+BS	–	–	–	–	–	–	–	–

Model	**Parametric SK**				**Semi-parametric SK**		**Non-parametric SK**	
Panel (B)	SKINN +BS	SKINN +AHBS	SKINN +HSV	SKINN +HSVKDEJ	SKINN +DSNN-HSV	SKINN +DSNN-NASV	SKINN +MOPA	SKINN +AE-BS
BS	-5.07***	-6.33***	0.84	0.35	0.72	1.45*	0.26	0.68
AHBS	-3.66***	-4.96***	1.78**	1.55*	1.45*	2.01**	1.25	1.35*
HSV	-8.32***	-8.12***	-8.42***	-8.11***	-6.37***	-5.63***	-6.69***	-5.64***
HSVKDEJ	-8.26***	-8.03***	-8.34***	-8.07***	-6.18***	-5.58***	-6.47***	-5.57***
NN	-4.71***	-4.59***	-3.44***	-3.63***	-3.41***	-2.89***	-3.58***	-3.83***
NN+Bnd	-5.07***	-5.12***	-3.40***	-3.71***	-3.14***	-2.56***	-3.50***	-4.07***
NN+MF	-4.41***	-4.46***	-3.49***	-3.67***	-3.31***	-3.03***	-3.60***	-3.88***
inv-PINN+BS	-4.32***	-4.29***	-3.01***	-3.23***	-2.94***	-2.45***	-3.29***	-3.34***

gradient problem as in Chen et al. (2023). A sufficiently deep neural network is inevitable in transfer learning, as it requires sufficient capacity to learn from the source domain accurately.

We first train a deep surrogate model to learn completely from a theoretical Heston model by simulation, without any noise. We then transfer the deep surrogate model to the market data. We refer to the transfer-learning models as TLNN+HSV. In Chen et al. (2023), they only fine-tune the deep surrogate model on the market data with a very small learning rate and very few epochs, based on the assumption that the market data has a data-generating process close to the theoretical model. We find this is, however, not the case empirically. Data patterns can contradict the theoretical structures of the Heston model significantly. We therefore train a series of TLNN+HSV with different numbers of frozen (or trainable) layers and epochs. Table (6) lists all the TLNN+HSV with different frozen layers and epochs, along with the deep surrogate model of Heston's model (DSNN+HSV) and our SKINN+HSV, and their respective training configurations. Since TLNN+HSV models require the calibrated latent parameters as their inputs, and the calibrated parameters from Heston's model are numerically unstable, as we observed in Figure (11). This is the limitation of the transfer-learning approach when being applied to high-dimensional theoretical models, and can seriously deteriorate the model performance. Although it is unfair to SKINN+HSV, we instead feed the SKINN-learned latent parameters as the TLNN+HSV inputs, as they present better numerical stability.

Model	Frozen/Trainable layers	Learning rate	Epochs	Inputs	ϕ
DSNN+HSV	6/0	2E-4	500	$\mathbf{X}, \phi$	Simulated
TLNN+HSV1	0/6	6E-6	6	$\mathbf{X}, \phi$	SKINN-learned
TLNN+HSV2	4/2	6E-6	6	$\mathbf{X}, \phi$	SKINN-learned
TLNN+HSV3	4/2	6E-6	100	$\mathbf{X}, \phi$	SKINN-learned
TLNN+HSV4	4/2	6E-6	500	$\mathbf{X}, \phi$	SKINN-learned
SKINN+HSV	0/3	1E-3	500	$\mathbf{X}$	

Table 6: This table summarizes the neural network configurations applied to train TLNNs and SKINN+HSV. We employ a smaller network size in order to allow TLNNs (6 hidden layers, 64 neurons each) to be comparable to our SKINNs (3 hidden layers, 32 neurons each). The different number of frozen layers for TLNNs corresponds to the different compliance with the theory.

Table (7) reports the pairwise Deibold-Mariano test statistics, which compare the out-of-sample option pricing error of the different configuration TLNN+HSV models with SKINN+HSV. For the TLNN+HSV models, allowing more trainable layers, or sufficiently training the model on the market data with more epochs, generally can improve the performance. This implies that there exists a considerable gap between the source domain, i.e., a theoretical option pricing model, and the target domain, i.e., the market option prices. SKINN+HSV statistically outperforms all the TLNN+HSV models with different configurations. The out-of-sample option pricing performance

of each model is evaluated using the options in the longer prediction horizons.

Model	DSNN +HSV	TLNN +HSV1	TLNN +HSV2	TLNN +HSV3	TLNN +HSV4	SKINN +HSV
DSNN+HSV	–	-10.48***	-10.88***	-10.48***	-11.39***	-13.87****
TLNN+HSV1	–	–	9.12***	-10.37***	-13.71***	-11.30***
TLNN+HSV2	–	–	–	-10.59***	-13.70***	-11.60***
TLNN+HSV3	–	–	–	–	-22.46***	-6.63***
TLNN+HSV4	–	–	–	–	–	-2.80***
SKINN+HSV	–	–	–	–	–	–

Table 7: The pair-wise Diebold-Mariano test statistics, which compare the out-of-sample option pricing errors (measured by RMSE) across all TLNNs and SKINN+HSV. The out-of-sample option pricing errors are evaluated using the options within the 317 periods of longer prediction horizons. Negative number indicates the column model outperforms the row model, and vice versa.

5 Economic Interpretations of SKINNs

Our empirical study has shown that, by embedding the structured-knowledge representations from the option pricing domain into the neural networks, SKINNs can provide statistically superior option pricing capability, compared with both the existing neural network pricing models and the classical structural models, especially in the longer prediction horizons. Although we do not train SKINNs to hedge options, their hedging capability is as good as the structural models, and is significantly better than the benchmark neural networks. In this section, we discuss the reasons for the superior performance of SKINNs.

5.1 Model Performances and Market Volatilities

It is important to understand under what market conditions SKINNs are able to provide significant marginal improvements for option pricing. As there is no one model that is suitable for all market conditions. We treat NN as the baseline, and for other models, we calculate the difference of their out-of-sample option pricing error from that of NN:

$$\Delta\text{RMSE}^{\text{Model}}(t_j) := \text{RMSE}^{\text{Model}}(t_j) - \text{RMSE}^{\text{NN}}(t_j). \tag{86}$$

The smaller the $\Delta\text{RMSE}^{\text{Model}}(t_j)$ is, the model outperforms more in out-of-sample compared with NN, and vice versa. We investigate the $\Delta\text{RMSE}^{\text{Model}}(t_j)$ of each model in both the shorter and the longer prediction horizons, but our main focus is the latter, as pricing options accurately in these horizons is tough for data-driven models.

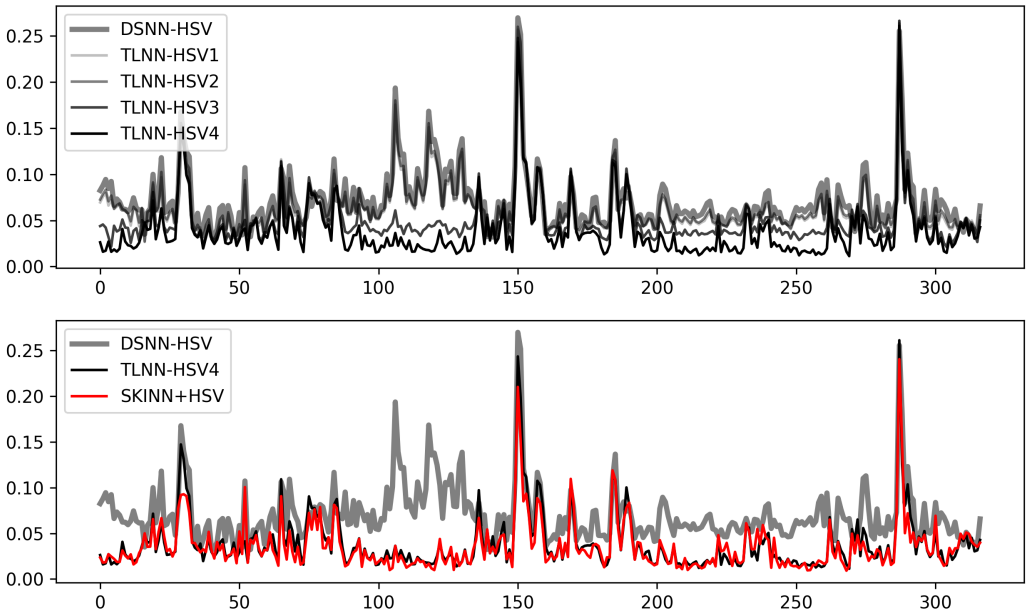


Figure 9: The out-of-sample option pricing performance of TLNNs and SKINN+HSV, over the 317 periods of longer prediction horizons. Among all the TLNNs, TLNN+HSV4 performs the best.

We use the averaged VIX index over a testing period as the proxy of the market conditions, which is defined as:

$$\text{AvgVIX}(t_j) := \frac{1}{N_{t_j}} \sum_{i=1}^{N_{t_j}} \frac{\text{VIX}_i}{100}, \tag{87}$$

where N_{t_j} is the number of trading days in the testing period t_j, $1 \leq j \leq 317$, and VIX_i is the daily close price of the VIX index. $\text{AvgVIX}(t_j)$ measures the volatility of the market during the period t_j. A higher $\text{AvgVIX}(t_j)$ is usually associated with the feared sentiment, high market uncertainty, and the tendency of mispricing, which means more noise in the test options. Therefore, it is also challenging for data-driven models to survive in these conditions. We regress the out-of-sample option pricing error differences from the longer prediction horizons against the averaged VIX index over the testing period:

$$\Delta\text{RMSE}^{\text{Model}} = \beta_0 + \beta_1 \text{AvgVIX} + \varepsilon. \tag{88}$$

Since the marginal improvements of a model against NN, $\Delta\text{RMSE}^{\text{Model}}$, is measured from the same testing periods as the market condition indicator, AvgVIX, the regression in Equation (88) tests how the market volatility impacts the option pricing performance of a model.

Table (8) reports the regression results for 3 structural models, NN+Bnd, and 8 SKINN variants.

Table 8: We run the regression (88) for each model. A statistically significant negative β_1 implies that the model performs better compared with a plain vanilla neural network when the market is more volatile. The results are based on all 317 test periods of the shorter prediction horizons.

Model	BS	AHBS	HSV	HSVKDEJ	NN	–	–	–
Panel (A)					+Bnd			
AvgVIX	0.0843***	0.0891***	0.0563***	0.0720***	-0.0116*	–	–	–
	(0.0141)	(0.0140)	(0.0146)	(0.0143)	(0.0069)	–	–	–
Constant	-0.0134***	-0.0141***	-0.0082**	-0.0151***	0.0005	–	–	–
	(0.0031)	(0.0031)	(0.0032)	(0.0032)	(0.0015)	–	–	–
Obs	317	317	317	317	317	–	–	–
Adj. R^2	0.0994	0.1111	0.0423	0.0712	0.0058	–	–	–
Model	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN
Panel (B)	+BS	+AHBS	+HSV	+HSVKDEJ	+DSNN-HSV	DSNN-NASV	+MOPA	+AE-BS
AvgVIX	0.0133	0.0264**	0.0118	0.0107	-0.0372***	0.0021	-0.0083	-0.0195**
	(0.0128)	(0.0128)	(0.0130)	(0.0133)	(0.0137)	(0.0131)	(0.0116)	(0.0094)
Constant	-0.0039	-0.0069**	-0.0051*	-0.0051*	0.0087***	-0.0014	-0.0013	0.0041*
	(0.0028)	(0.0028)	(0.0029)	(0.0029)	(0.0030)	(0.0029)	(0.0025)	(0.0021)
Obs	317	317	317	317	317	317	317	317
Adj. R^2	0.0003	0.0102	-0.0005	-0.0011	0.0199	-0.0031	-0.0015	0.0104

We do not consider NN+MF and inv-PINN+BS in this test due to their gradient pathologies. When considering the shorter prediction horizons as the testing periods, NN+Bnd and SKINNs perform closely, as it is hard for all of them to provide large and significant option pricing improvements against NN. NN+Bnd provides a tiny marginal improvement when the market is volatile, which is significant at only the 10% significance level. SKINN+BS-AE, in this case, provides a slightly larger improvement, which is significant at the 5% significance level. In these shorter horizons, structural models, however, present significantly larger errors than NN when the market in the testing period has a higher volatility.

Table (9) reports the results for the longer prediction horizons. At this time, all the coefficients of AvgVIX for NN+Bnd, and SKINNs are significantly negative at 1% level. This aligns with our finding from Table (3) that SKINNs, including NN+Bnd, are capable of improving the out-of-sample pricing accuracy, compared with the plain data-driven NN. The negative coefficients indicate that NN+Bnd and SKINNs provide marginal improvements against NN when the market is more volatile. Such countercyclical option pricing performance is economically desirable, as volatile market conditions, associated with noisier option price patterns, usually deteriorate plain data-driven as well as structural option pricing models. BS, AHBS can not provide significantly lower out-of-sample pricing errors, and HSV provide significantly higher errors instead. Though statistically significant, the magnitude of the marginal improvement during high volatility market conditions provided by NN+Bnd is small, which is around 0.03. Meanwhile, the marginal improvement magnitude of all SKINN variants is 3–4 times larger than that of NN+Bnd. Interestingly, for

Table 9: We run the regression (88) for each model. A statistically significant negative β_1 implies that the model performs better compared with a plain vanilla neural network when the market is more volatile. The results are based on all 317 test periods of the longer prediction horizons.

Model	BS	AHBS	HSV	HSVKDEJ	NN	–	–	–
Panel (A)					+Bnd			
AvgVIX	-0.0317	-0.0274	-0.0483**	-0.0340	-0.0303***	–	–	–
	(0.0214)	(0.0214)	(0.0215)	(0.0216)	(0.0096)	–	–	–
Constant	0.0019	0.0013	0.0053	-0.0008	0.0027	–	–	–
	(0.0047)	(0.0047)	(0.0047)	(0.0048)	(0.0021)	–	–	–
Obs	317	317	317	317	317	–	–	–
Adj. R^2	0.0038	0.0020	0.0127	0.0047	0.0273	–	–	–
Model	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN
Panel (B)	+BS	+AHBS	+HSV	+HSVKDEJ	+DSNN-HSV	+DSNN-NASV	+MOPA	+AE-BS
AvgVIX	-0.1000***	-0.0868***	-0.0944***	-0.0948***	-0.1369***	-0.1031***	-0.0923***	-0.0944***
	(0.0205)	(0.0201)	(0.0199)	(0.0196)	(0.0195)	(0.0202)	(0.0177)	(0.0132)
Constant	0.0119***	0.0091**	0.0102**	0.0101**	0.0227***	0.0135***	0.0106***	0.0155***
	(0.0045)	(0.0044)	(0.0044)	(0.0043)	(0.0043)	(0.0045)	(0.0039)	(0.0029)
Obs	317	317	317	317	317	317	317	317
Adj. R^2	0.0671	0.0528	0.0634	0.0661	0.1329	0.0733	0.0764	0.1374

SKINN+AE-BS, it does not statistically outperform the structural models in the longer prediction horizons, according to the Diebold-Mariano test; however, from the regression, it offers significantly larger marginal improvement than the structural models.

We then perform zoom-in analysis for the marginal improvements of models against NN in the longer prediction horizons, where the option price patterns shift significantly from the patterns in the model estimation periods. We inspect whether the magnitude of the marginal option pricing improvement of SKINNs in volatile market conditions differs by the level of volatility itself. We divide the 317 longer horizon testing periods into three groups: low volatility periods, medium volatility periods, and high volatility periods.[22]

Table (10) reports the regression results for the low volatility periods of the longer prediction horizons. In these periods, all models can not be statistically differentiated from NN, as all coefficients of AvgVIX are statistically insignificant. Except that SKINN+DSNN-HSV displays a significantly large improvement, which is driven by outliers in this small sample. This is expected, since the option price patterns from low volatility market conditions are relatively easier to learn, due to less mispricing and less noise in prices.

Table (11) reports the regression results for the median volatility market conditions. SKINNs, together with NN+Bnd, provide a significant marginal option pricing improvement against NN when the AvgVIX increases, across the median volatility testing periods. Though statistically

[22]Low volatility periods, medium volatility periods, and high volatility periods contain the periods for which the AvgVIX is below the 20% percentile, from the 20% to the 80% percentile, and greater than the 80% percentile of the AvgVIX over all periods, respectively.

Table 10: We run the regression (88) for each model. A statistically significant negative β_1 implies that the model performs better compared with a plain vanilla neural network when the market is more volatile. The results are based on the 64 low volatility testing periods in the longer prediction horizons, where AvgVIX(t) is below the 20% quantile of AvgVIX across all the longer horizon testing periods.

Model	BS	AHBS	HSV	HSVKDEJ	NN	–	–	–
Panel (A)					+Bnd			
AvgVIX	-0.1979	-0.1172	-0.1354	-0.1288	-0.0346	–	–	–
	(0.1657)	(0.1644)	(0.1646)	(0.1671)	(0.1192)	–	–	–
Constant	0.0230	0.0125	0.0167	0.0119	0.0024	–	–	–
	(0.0205)	(0.0204)	(0.0204)	(0.0207)	(0.0148)	–	–	–
Obs	64	64	64	64	64	–	–	–
Adj. R^2	0.0067	-0.0079	-0.0052	-0.0065	-0.0148	–	–	–
Model	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN
Panel (B)	+BS	+AHBS	+HSV	+HSVKDEJ	+DSNN-HSV	+DSNN-NASV	+MOPA	+AE-BS
AvgVIX	-0.2303	-0.1690	-0.0628	0.0021	-0.6827***	-0.1875	-0.1741	-0.1343
	(0.1504)	(0.1528)	(0.1614)	(0.1836)	(0.1828)	(0.1524)	(0.1270)	(0.1275)
Constant	0.0252	0.0162	0.0037	-0.0035	0.0912***	0.0221	0.0175	0.0167
	(0.0186)	(0.0189)	(0.0200)	(0.0227)	(0.0226)	(0.0189)	(0.0157)	(0.0158)
Obs	64	64	64	64	64	64	64	64
Adj. R^2	0.0209	0.0035	-0.0137	-0.0161	0.1705	0.0081	0.0138	0.0018

significant, the marginal improvement magnitude of NN+Bnd is much smaller than that of SKINNs. SKINN+DSNN-NASV, in this case, provides the largest marginal improvement in magnitude.

Table (12) reports the regression results for the high volatility market conditions. Within these most volatile market periods, only SKINNs still survive to provide statistically significant marginal option pricing improvements against NN when the AvgVIX increases, at 10% level (SKINN+BS, SKINN+AHBS, and SKINN+DSNN-NASV) and 5% level (SKINN+HSV, SKINN+HSVKDEJ, SKINN+DSNN-HSV, SKINN+MOPA, and SKINN+AE-BS). More importantly, the marginal improvements offered by SKINNs in this case are the largest among all volatility groups. SKINN+AE-BS provides the greatest marginal improvement for the option pricing performance when the market volatility is even higher in an already volatile condition.

5.2 The Estimator Side of SKINNs

The SKINNs framework not only regularizes the neural networks, but also functions as an estimator for sophisticated and high-dimensional economic models that converges to a GMM estimator under certain conditions. We proceed to examine the SKINN framework as an estimator for latent economic parameters. We randomly initialize the parameters ϕ in the structured-knowledge representations g_ϕ, before training a SKINN. Different from the sequential approach adopted by Chen et al. (2023) in transfer-learning models, which requires a separate non-convex and non-linear esti-

Table 11: We run the regression (88) for each model. A statistically significant negative β_1 implies that the model performs better compared with a plain vanilla neural network when the market is more volatile. The results are based on the 189 medium volatility testing periods in the longer prediction horizons, where AvgVIX(t) is between the 20% quantile and the 80% quantile of AvgVIX across all the longer horizon testing periods.

Model	BS	AHBS	HSV	HSVKDEJ	NN	–	–	–
Panel (A)					+Bnd			
AvgVIX	-0.0708	-0.0630	-0.1299**	-0.0974*	-0.0686**	–	–	–
	(0.0525)	(0.0525)	(0.0524)	(0.0532)	(0.0275)	–	–	–
Constant	0.0086	0.0075	0.0202**	0.0104	0.0098*	–	–	–
	(0.0102)	(0.0103)	(0.0102)	(0.0104)	(0.0054)	–	–	–
Obs	189	189	189	189	189	–	–	–
Adj. R^2	0.0043	0.0023	0.0266	0.0124	0.0270	–	–	–
Model	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN
Panel (B)	+BS	+AHBS	+HSV	+HSVKDEJ	+DSNN-HSV	+DSNN-NASV	+MOPA	+AE-BS
AvgVIX	-0.1027**	-0.0992**	-0.1062**	-0.1114**	-0.1295***	-0.1303***	-0.0758*	-0.1298***
	(0.0471)	(0.0464)	(0.0473)	(0.0458)	(0.0449)	(0.0466)	(0.0420)	(0.0323)
Constant	0.0130	0.0120	0.0126	0.0131	0.0202**	0.0188**	0.0080	0.0222***
	(0.0092)	(0.0090)	(0.0092)	(0.0089)	(0.0088)	(0.0091)	(0.0082)	(0.0063)
Obs	189	189	189	189	189	189	189	189
Adj. R^2	0.0195	0.0187	0.0211	0.0255	0.0374	0.0350	0.0119	0.0747

Table 12: We run the regression (88) for each model. A statistically significant negative β_1 implies that the model performs better compared with a plain vanilla neural network when the market is more volatile. The results are based on the 64 high volatility testing periods in the longer prediction horizons, where AvgVIX(t) is above the 80% quantile of AvgVIX across all the longer horizon testing periods.

Model	BS	AHBS	HSV	HSVKDEJ	NN	–	–	–
Panel (A)					+Bnd			
AvgVIX	-0.0364	-0.0416	-0.0345	-0.0272	-0.0466	–	–	–
	(0.0783)	(0.0787)	(0.0788)	(0.0791)	(0.0287)	–	–	–
Constant	0.0051	0.0077	0.0025	-0.0013	0.0094	–	–	–
	(0.0262)	(0.0264)	(0.0264)	(0.0265)	(0.0096)	–	–	–
Obs	64	64	64	64	64	–	–	–
Adj. R^2	-0.0126	-0.0116	-0.0130	-0.0142	0.0255	–	–	–
Model	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN	SKINN
Panel (B)	+BS	+AHBS	+HSV	+HSVKDEJ	+DSNN-HSV	+DSNN-NASV	+MOPA	+AE-BS
AvgVIX	-0.1464*	-0.1372*	-0.1516**	-0.1474**	-0.1555**	-0.1373*	-0.1526**	-0.1825***
	(0.0780)	(0.0761)	(0.0739)	(0.0724)	(0.0720)	(0.0767)	(0.0660)	(0.0453)
Constant	0.0282	0.0270	0.0308	0.0295	0.0307	0.0263	0.0315	0.0478***
	(0.0261)	(0.0255)	(0.0248)	(0.0243)	(0.0241)	(0.0257)	(0.0221)	(0.0152)
Obs	64	64	64	64	64	64	64	64
Adj. R^2	0.0385	0.0344	0.0484	0.0475	0.0549	0.0337	0.0645	0.1946

mation, SKINNs learn ϕ simultaneously with the neural network, during the same online training process. We show that the learned ϕ by SKINNs possesses strong economic interpretations, rather than just some uninterpretable nuisance parameters, particularly in the high-dimensional setting.

5.2.1 The Accuracy of the Learned Latent Economic Parameters

Figure (11) illustrates the time-series of the learned latent economic parameters from the SKINNs embedded with the parametric format g_ϕ, over the 317 model training periods. We compare the ϕ learned by SKINNs with those calibrated by using the classic non-linear least-squares method. The dynamics of most of the SKINN-learned latent economic parameters evolve with obvious economic regimes, signalling that those learned parameters load economic information. For the one-dimensional Black-Scholes case, the SKINN-learned σ closely tracks the calibrated implied volatility. For more sophisticated higher-dimensional cases, the SKINN-learned latent parameters can diverge from those obtained by the classic calibration. In general, the SKINN-learned latent economic parameters show smoother evolution over time.

The latent economic parameter vector learned from SKINN+MOPA, which uses a non-parametric structured-knowledge representation of an unknown distribution, is very high-dimensional (2,000-dimensional in our implementation). It is inconvenient to visualize the evolution of such a very high-dimensional vector. However, since these parameters can be interpreted as the risk-neutral probabilities, we visualize the probability densities that consist of these learned probabilities. Figure (12) is an example of the learned densities from a selected SKINN+MOPA training period.

According to Chen et al. (2021), if one option pricing model is well-specified, then we should expect small variations in the model parameters across time. We calculate the l_2-norm of the difference between the latent parameters ϕ_{t+1} and ϕ_t, $1 \leq t \leq 316$, from two consecutive periods. The latent parameters are from SKINN+BS, SKINN+AHBS, SKINN+HSV, SKINN+HSVKDEJ, and the respective structural models. Figure (13) shows the distribution of the periodic parameter variations (the l_2 norms) across our training periods. For the sophisticated structured-knowledge representations, e.g., g_ϕ^{HSV} and g_ϕ^{HSVKDEJ}, the learned ϕ presents smaller variations than the counterparts estimated by calibrating the structural models to the market data. This indicates that the SKINN-learned latent parameters have improved stability.

5.2.2 The Economic Inference of the Learned Economic Parameters

The latent economic parameters learned with SKINNs are expected to carry some economic information if they are not nuisance parameters. We examine this by regressing the AvgVIX over the shorter prediction horizons against the calibrated ϕ and the SKINN-learned ϕ separately, from

all training periods. For SKINN+BS, SKINN+AHBS, SKINN+HSV, and the respective structural models, we run the following regression:

$$\text{AvgVIX}_{t+1} = \beta_0 + \beta_1 \phi_t^{(d)} + \varepsilon, \quad 1 \le t \le 317,\ d \in \{1, 2, 3, \cdots d_\phi\}, \tag{89}$$

We only include one parameter at a time to examine its individual predictive power for the market volatility in the subsequent months.

Table (13) reports the results of such univariate regressions by using the calibrated ϕ from structural models. Table (14) reports the results for the learned ϕ from SKINNs. Most of the SKINN-learned latent parameters carry significant predictive information for the market volatility in the subsequent month, as most of their regression coefficients are significantly non-zero. Additionally, for more than half of the parameters, the SKINN-learned one has a relatively higher adjusted R^2 than the calibrated one.

Table 13: The predictive information contained in each of the parameters calibrated according to the standard procedure, for the lagged averaged VIX over one month. $\{\sigma\}$ is from the Black-Scholes model, $\{a_1, a_2, a_3, a_4, a_5, a_6\}$ is from the ad-hoc Black-Scholes model, $\{\bar{v}, v_0, \sigma_v, \rho, \kappa\}$ is from the Heston's model, and $\{\bar{v}, v_0, \sigma_v, \rho, \kappa, \lambda, \eta_0, \eta_1, p_{up}\}$ is from the Heston's model with Kou's double exponential jump.

Parameter (BS)	σ								
β_1	1.1181***								
	(18.781)								
β_0	0.0255**								
	(2.543)								
Obs	317								
Adj. R^2	52.7%								
Parameter (AHBS)	a_1	a_2	a_3	a_4	a_5	a_6			
β_1	0.7555***	-0.1997	-0.2349	-1.4114***	-2.4120***	-3.1940***			
	(0.0521)	(0.2302)	(0.1733)	(0.3630)	(0.9176)	(0.9057)			
β_0	0.0944***	0.2053***	0.2061***	0.2159***	0.2084***	0.2090***			
	(0.0084)	(0.0046)	(0.0046)	(0.0053)	(0.0047)	(0.0046)			
Obs	317	317	317	317	317	317			
Adj. R^2	39.85%	-0.08%	0.26%	4.28%	1.84%	3.49%			
Parameter (HSV)	$\bar{v}$	v_0	σ_v	ρ	κ				
β_1	-0.0106	1.3240***	-0.0024***	-0.0412***	0.0022				
	(-0.250)	(13.915)	(-3.321)	(-3.038)	(1.314)				
β_0	0.2063***	0.1502***	0.2156***	0.1826***	0.2000***				
	(26.263)	(28.394)	(39.002)	(21.413)	(34.945)				
Obs	317	317	317	317	317				
Adj. R^2	-0.3%	37.9%	3.1%	2.5%	0.2%				
Parameter (HSVKDEJ)	$\bar{v}$	v_0	σ_v	ρ	κ	λ	η_0	η_1	p_{up}
β_1	1.1575***	1.7827***	0.0546***	0.1118***	-0.0022**	0.0257***	-0.0001	-0.0000	-0.0099
	(0.2342)	(0.1088)	(0.0077)	(0.0224)	(0.0009)	(0.0044)	(0.0001)	(0.0003)	(0.0119)
β_0	0.1737***	0.1502***	0.1784***	0.2933***	0.2183***	0.1967***	0.2053***	0.2049***	0.2076***
	(0.0076)	(0.0047)	(0.0056)	(0.0183)	(0.0069)	(0.0045)	(0.0046)	(0.0049)	(0.0057)
Obs	317	317	317	317	317	317	317	317	317
Adj. R^2	6.90%	45.83%	13.38%	7.03%	1.75%	9.51%	-0.16%	-0.31%	-0.10%

Table 14: The predictive information contained in each of the parameters learned from the SKINNs, for the lagged averaged VIX over one month. $\{\sigma^*\}$ is from g_ϕ^{BS}, $\{a_1^*, a_2^*, a_3^*, a_4^*, a_5^*, a_6^*\}$ is from g_ϕ^{AHBS}, $\{\bar{v}^*, v_0^*, \sigma_v^*, \rho^*, \kappa^*\}$ is from g_ϕ^{HSV}, and $\{\bar{v}^*, v_0^*, \sigma_v^*, \rho^*, \kappa^*, \lambda^*, \eta_0^*, \eta_1^*, p_{up}^*\}$ is from g_ϕ^{HSVKDEJ}.

Parameter (BS)	σ^*								
β_1	1.0896***								
	(17.329)								
β_0	0.0323***								
	(3.087)								
Obs	317								
Adj. R^2	48.6%								
Parameter (AHBS)	a_1^*	a_2^*	a_3^*	a_4^*	a_5^*	a_6^*			
β_1	1.0456***	-13.3473*	-25.9413*	-1.8508***	-3.0981	-32.0793**			
	(17.481)	(-1.656)	(-1.916)	(-5.932)	(-1.453)	(-2.572)			
β_0	0.0458***	0.2109***	0.2096***	0.2256***	0.2071***	0.2108***			
	(4.750)	(36.071)	(40.597)	(40.700)	(43.138)	(41.685)			
Obs	317	317	317	317	317	317			
Adj. R^2	49.1%	0.5%	0.8%	9.8%	0.4%	1.7%			
Parameter (HSV)	$\bar{v}^*$	v_0^*	σ_v^*	ρ^*	κ^*				
β_1	0.1047	1.8552***	0.1744***	0.0424*	0.0441***				
	(1.578)	(15.838)	(10.453)	(1.881)	(3.878)				
β_0	0.1938***	0.1478***	0.1510***	0.2348***	0.1889***				
	(23.568)	(30.009)	(23.446)	(14.112)	(31.475)				
Obs	317	317	317	317	317				
Adj. R^2	0.5%	44.2%	25.5%	0.8%	4.3%				
Parameter (HSVKDEJ)	$\bar{v}^*$	v_0^*	σ_v^*	ρ^*	κ^*	λ^*	η_0^*	η_1^*	p_{up}^*
β_1	0.1999***	1.8239***	0.1990***	0.0323*	0.0356**	0.4998	-0.0701	0.0274***	-0.1316***
	(0.0585)	(0.1151)	(0.0144)	(0.0183)	(0.0151)	(0.4753)	(0.0469)	(0.0067)	(0.0234)
β_0	0.1793***	0.1491***	0.1452***	0.2261***	0.1963***	0.2022***	0.2195***	0.1803***	0.2374***
	(0.0086)	(0.0049)	(0.0056)	(0.0130)	(0.0057)	(0.0051)	(0.0109)	(0.0074)	(0.0072)
Obs	317	317	317	317	317	317	317	317	317
Adj. R^2	3.27%	44.19%	37.57%	0.66%	1.43%	0.03%	0.39%	4.77%	8.83%

For the case of SKINN-MOPA, we translate the 2,000-dimensional learned risk-neutral probabilities to the variances over 10 tenors, then we examine the economic information of the variances over the 10 tenors, via the following regression:

$$\text{AvgVIX}_{t+1} = \beta_0 + \beta_1 \mathbb{V}\text{ar}[\phi_t^{(\tau_k)}] + \varepsilon, \quad 1 \leq t \leq 317, \; k \in \{1, 2, 3, \cdots 10\}, \tag{90}$$

where τ_k denotes the tenor k, and $\phi_t^{(\tau_k)}$ represents the 200-dimensional learned risk-neutral probabilities for this tenor, for each model training period t.

Figure (10) shows the translated risk-neutral variances over the 317 model training periods, and the corresponding AvgVIX over the subsequent months. Table (15) reports the regression results. Even in this very high-dimensional setting, the learned latent parameters still carry substantial predictive economic information.

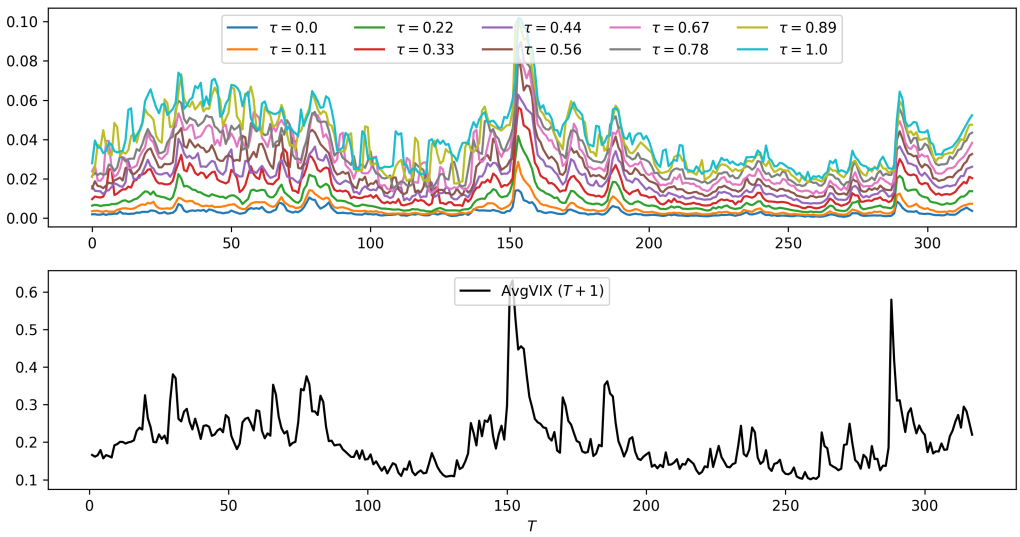
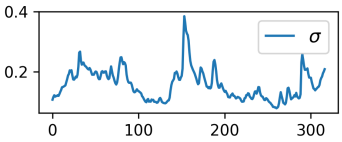
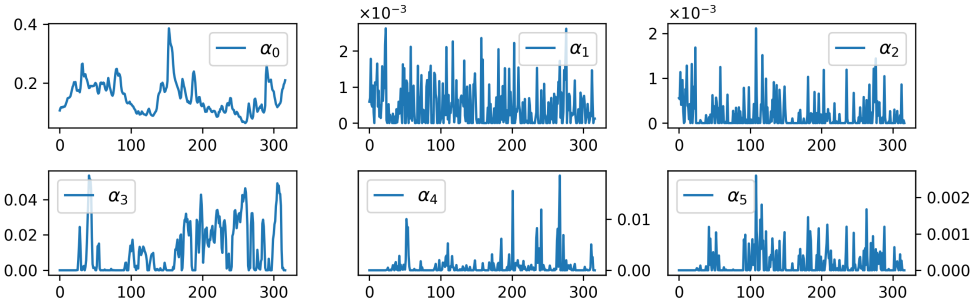


Figure 10: The risk-neutral variances over 10 different tenors. Each risk-neutral variance $\mathbb{V}\text{ar}[\phi_t^{(\tau_k)}]$ is translated from the learned risk-neutral probabilities by SKINN+MOPA for the k-th tenor, $1 \leq k \leq 10$. The upper panel illustrates the learned risk-neutral variances, and the lower panel illustrates the AvgVIX over the subsequent months after the SKINN+MOPA training periods.

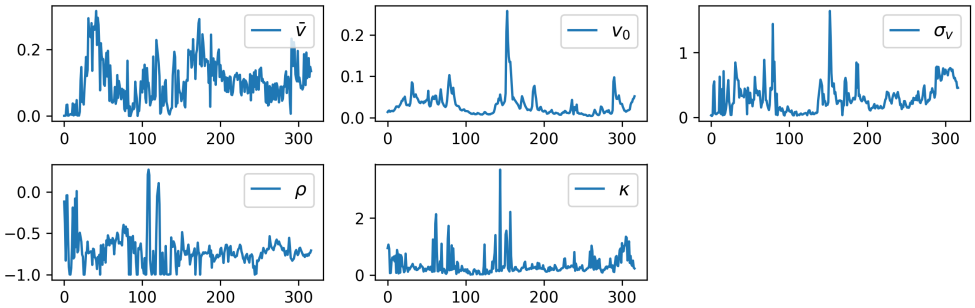
Lastly, we examine the learned latent parameters from the SKINNs with the semi-parametric and the auto-encoder type of non-parametric format structured-knowledge representations. For SKINN+DSNN-HSV and SKINN+DSNN-NASV, the learned latent parameters are based on the pre-trained deep surrogate neural networks that themselves my not exactly resemble the true economic models. For SKINN+AE-BS, the learned latent parameters are based on the pre-trained auto-encoder, given the simulated noisy option cross-sections. Figure (14) illustrates the evolution



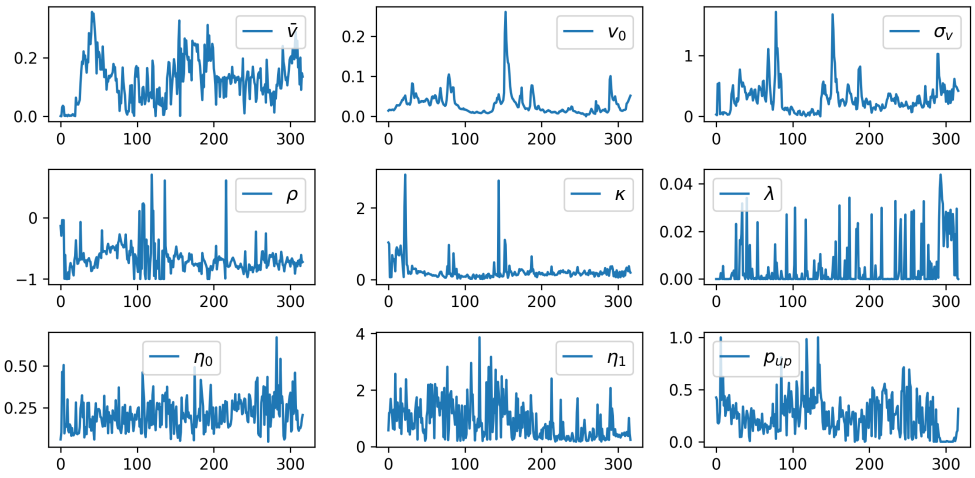
(a) SKINN+BS



(b) SKINN+AHBS



(c) SKINN+HSV

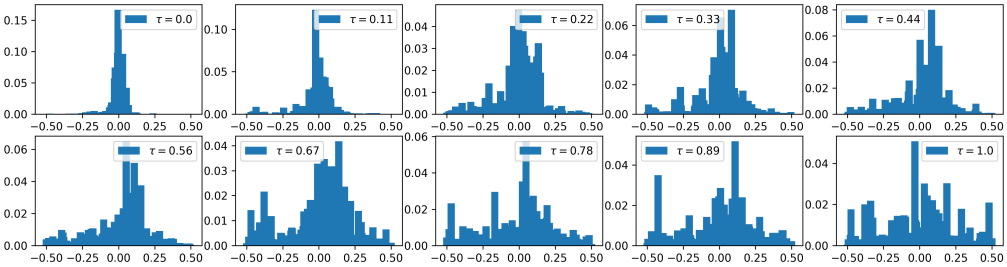


(d) SKINN+HSVKDEJ

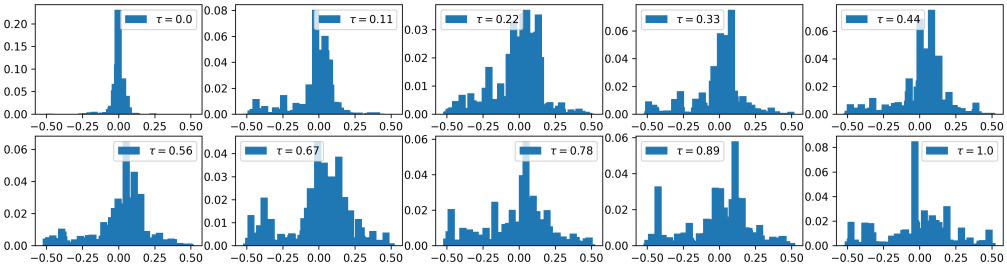
Figure 11: The latent economic parameters learned by SKINNs with parametric g_ϕ.

Table 15: The predictive information contained in the risk-neural variances converted from the latent parameters (i.e., the 2,000 risk-neutral probabilities over 10 tenors) learned from SKINN+MOPA, for the lagged averaged VIX over one month. $\mathbb{V}\text{ar}[\phi_t^{(\tau_k)}]$ is the risk-neutral variance converted according to the learned risk-neutral probabilities for the k-th tenor.

Parameter (MOPA)	$\mathbb{V}\text{ar}[\phi^{(\tau_1)}]$	$\mathbb{V}\text{ar}[\phi^{(\tau_2)}]$	$\mathbb{V}\text{ar}[\phi^{(\tau_3)}]$	$\mathbb{V}\text{ar}[\phi^{(\tau_4)}]$	$\mathbb{V}\text{ar}[\phi^{(\tau_5)}]$
β_1	26.2949***	16.3890***	10.2331***	6.6412***	5.9800***
	(1.6483)	(0.9353)	(0.5712)	(0.4056)	(0.3548)
β_0	0.1292***	0.1217***	0.1098***	0.1041***	0.0884***
	(0.0058)	(0.0057)	(0.0062)	(0.0070)	(0.0076)
Obs	317	317	317	317	317
Adj. R^2	44.5%	49.2%	50.3%	45.8%	47.3%
Parameter (MOPA)	$\mathbb{V}\text{ar}[\phi^{(\tau_6)}]$	$\mathbb{V}\text{ar}[\phi^{(\tau_7)}]$	$\mathbb{V}\text{ar}[\phi^{(\tau_8)}]$	$\mathbb{V}\text{ar}[\phi^{(\tau_9)}]$	$\mathbb{V}\text{ar}[\phi^{(\tau_{10})}]$
β_1	4.7931***	4.0436***	4.1399***	3.4236***	3.3714***
	(0.2970)	(0.2599)	(0.2573)	(0.2226)	(0.2269)
β_0	0.0889***	0.0849***	0.0694***	0.0720***	0.0629***
	(0.0079)	(0.0084)	(0.0090)	(0.0093)	(0.0101)
Obs	317	317	317	317	317
Adj. R^2	45.1%	43.3%	44.9%	42.7%	41.0%



(a) Train period id: 152 (01 Aug 2008 to 31 Oct 2008).

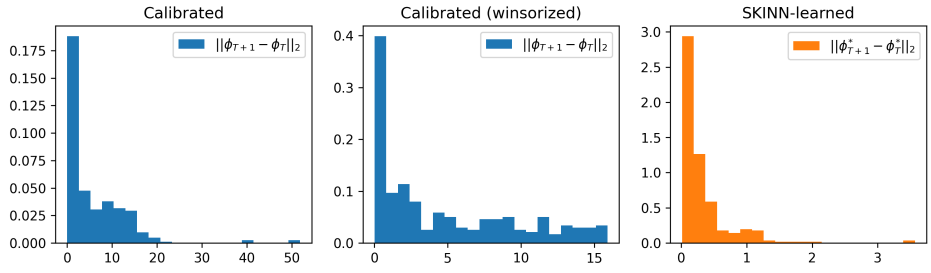


(b) Train period id: 290 (03 Feb 2020 to 30 Apr 2020).

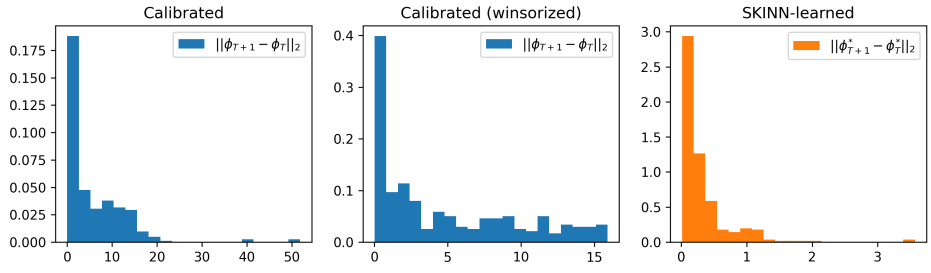
Figure 12: The latent economic parameters (risk-neutral probabilities) learned by SKINN+MOPA, which is a non-parametric format structured-knowledge representation with an unknown distribution. We select two training periods to demonstrate the learned probabilities, from the 2007-2009 global financial crisis and the COVID-19 crisis, respectively.

(a) SKINN+BS

(b) SKINN+AHBS



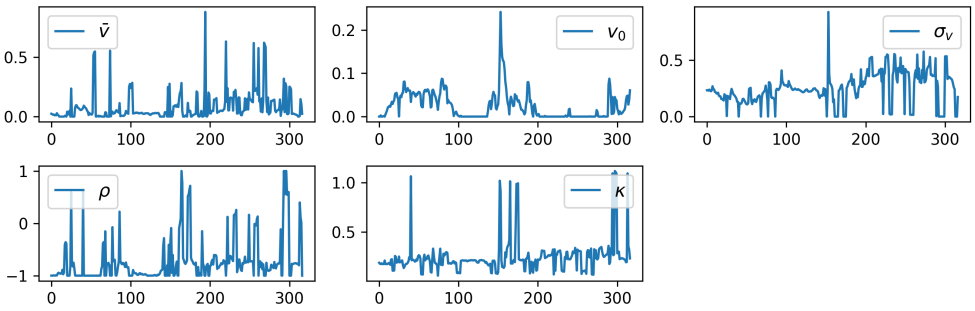
(c) SKINN+HSV



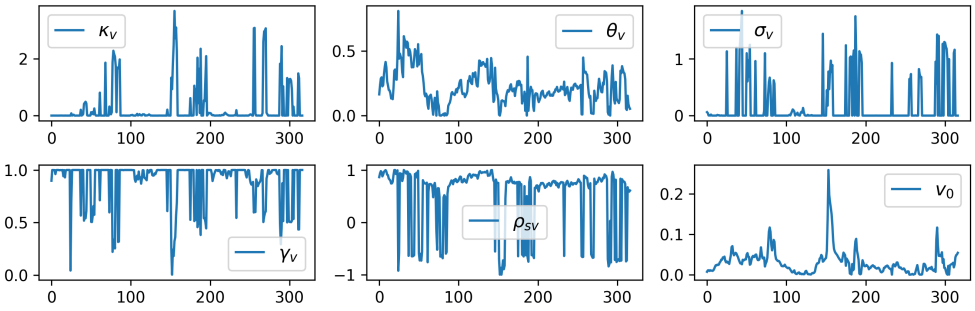
(d) SKINN+HSVKDEJ

Figure 13: The distributions of the variation of the SKINN-learned (orange) and the calibrated (blue) latent economic parameters, between two consecutive training periods over time. The SKINN-learned latent economic parameters exhibit a higher density around zero, indicating higher numerical stability compared to those from the conventional calibration procedure.

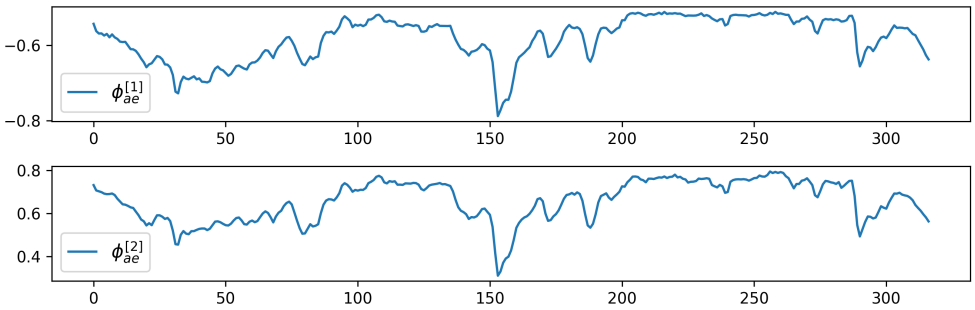
of their learned latent parameters over time. Table (16) reports the regression results for these learned parameters. Though the structured-knowledge in this case is represented by neural networks, instead of parametric expressions, our SKINNs framework can still identify the unknown parameters correctly. Interestingly, though the deep surrogates, as well as the auto-encoder, are imperfect approximations for the option price data-generating process, all the latent parameters are statistically significant and economically informative, with considerably high adjusted R^2, and even surpass the parameters from parametric representations.



(a) SKINN+DSNN-HSV



(b) SKINN+DSNN-NASV



(c) SKINN+AE-BS

Figure 14: The latent economic parameters learned from the SKINNs with semi- and non-parametric format structured-knowledge representations, which construct g_ϕ using deep surrogate neural networks or autoencoders (SKINN+DSNN-HSV, SKINN+DSNN-NASV and SKINN+AE-BS).

Table 16: The predictive information contained in the latent parameters learned from SKINN+DSNN-HSV, SKINN+DSNN-NASV and SKINN+AE-BS, for the lagged averaged VIX over one month. $\{\bar{v}, v_0, \sigma_v, \rho, \kappa\}$ is from $g_\phi^{\text{DSNN-HSV}}$, $\{\kappa_v, \theta_v, \sigma_v, \gamma_v, \rho_{sv}, v_0\}$ is from $g_\phi^{\text{DSNN-NASV}}$, and $\{\phi_{\text{AE}}^{(1)}, \phi_{\text{AE}}^{(2)}\}$ is from $g_\phi^{\text{AE-BS}}$.

Parameter (DSNN-HSV)	$\bar{v}$	v_0	σ_v	ρ	κ	
β_1	-7.8261**	167.5138***	-12.7938***	1.9625*	6.7324***	
	(3.6186)	(10.7461)	(3.1635)	(1.0485)	(2.5814)	
β_0	21.1113***	16.3179***	23.6611***	21.8788***	18.8241***	
	(0.5378)	(0.4312)	(0.9039)	(0.8771)	(0.7726)	
Obs	317	317	317	317	317	
Adj. R^2	1.15%	43.37%	4.63%	0.79%	1.80%	
Parameter (DSNN-NASV)	κ_v	θ_v	σ_v	γ_v	ρ_{sv}	v_0
β_1	3.4702***	-8.2771**	4.1704***	-12.8520***	-4.3081***	165.5858***
	(0.6080)	(3.3347)	(1.0834)	(1.9883)	(0.7073)	(11.3337)
β_0	19.4509***	22.2949***	19.6744***	31.8214***	22.6236***	15.4147***
	(0.4653)	(0.8610)	(0.4870)	(1.8069)	(0.5546)	(0.4911)
Obs	317	317	317	317	317	317
Adj. R^2	9.08%	1.61%	4.19%	11.43%	10.25%	40.20%
Parameter (AE-BS)	$\phi_{\text{AE}}^{(1)}$	$\phi_{\text{AE}}^{(2)}$				
β_1	-89.4417***	-59.3616***				
	(5.9282)	(3.4113)				
β_0	-31.5612***	59.4865***				
	(3.4657)	(2.2653)				
Obs	317	317				
Adj. R^2	41.77%	48.85%				

6 Conclusion

Despite recent successes of AI and machine learning algorithms in economics, they often lack the transparency, interpretability, and clarity on economic intuition and mechanism that economists often require of conventional econometric and theory models. We introduce a Structured-Knowledge-Informed Neural Networks (SKINNs) framework for embedding existing domain knowledge, in flexible formats (analytical models, deep surrogate models, and general implicit/non-parametric models), into deep-learning-based data-driven analyses. Improving upon and going beyond existing methods of encoding prior knowledge into neural networks, including transfer learning and PINNs, SKINNs serve as a general tool for scientific machine learning, or AI for science or social science, especially when the latent parameters for structured-knowledge are high-dimensional and hard to estimate or interpret using conventional methods.

On the theoretical side, we provide a unified econometric foundation for SKINNs. Beyond consistency and asymptotic normality, we establish identification of structural parameters under joint flexibility, derive generalization and target-risk bounds under distributional shift in a convex proxy, and characterize the regularization parameter as a restricted-optimal weighting choice within a GMM interpretation. Under orthogonal moment conditions, the structural parameter estimator achieves the optimal asymptotic variance relative to the imposed moment restrictions. These results demonstrate that integrating structured knowledge into neural networks is not merely a heuristic regularization device, but defines a statistically well-behaved estimator with formal inferential guarantees.

For illustration, we apply SKINNs to option pricing, a setting featuring structured theoretical, conceptual, or evidence-based knowledge in a wide variety of formats. SKINNs statistically improve out-of-sample pricing performance and hedging capability for the S&P 500 index options, compared with the plain-vanilla neural network, neural network with the model-free constraints, standard structural models, and even transfer learning informed by the Heston model. The outperformance of SKINNs is greater when pricing options in periods more distant from the training sample period and in highly volatile markets, where a pure data-driven approach tends to overfit or ill-adjust for distributional shifts. We also find that the learned latent parameters for structured-knowledge representations are not merely nuisance parameters; they carry economic meaning with improved stability over time, over the estimations from standard calibration procedures. Moreover, the dimensionality of the latent parameters can efficiently scale under SKINNs, something prohibitive in other approaches, such as transfer learning.

Appendix A The Wilcoxon Signed-Rank Test

Besides the Diebold and Mariano (2002) test results we report in Section (4.3), we also perform the Wilcoxon (1945) signed-rank test to compare the performance difference in pricing and hedging among the considered models, for the robustness check. While the Diebold-Mariano test adjusts for the possible autocorrelation in the series of error differentials $d_j = \{e_j^1 - e_j^2\}_{j=1}^{317}$, the non-parametric Wilcoxon signed-rank test does not account for this. We report the results of the Wilcoxon signed-rank test in this section. The results of using two tests are close. Since the test statistic of the Wilcoxon signed-rank test is always a positive integer, we place the asterisks on the right-top of a number to indicate that the column model outperforms the row model, and on the left-top of a number to indicate that the row model outperforms the column model. Table (17) and Table (18) report the results for the out-of-sample pricing performance comparisons. Table (19) and Table (20) report the results for the out-of-sample hedging performance comparisons.

Table 17: The Wilcoxon (1945) test results for the out-of-sample option pricing accuracy comparisons in shorter prediction horizons. We square all the pricing errors to penalize large errors.

Model	**Structural models**				**Benchmark neural networks**			
Panel (A)	BS	AHBS	HSV	HSVKDEJ	NN	NN +Bnd	NN +MF	inv-PINN +BS
BS	–	22.48**	23.43	5.68***	17.71***	14.13***	25.18	25.34
AHBS	–	–	22.58*	5.20***	17.15***	13.82***	24.53	25.12
HSV	–	–	–	2.94***	19.13***	16.19***	25.89	26.53
HSVKDEJ	–	–	–	–	25.74	23.54	35.30***	35.12***
NN	–	–	–	–	–	20.75***	35.23***	35.50***
NN+Bnd	–	–	–	–	–	–	39.29***	38.64***
NN+MF	–	–	–	–	–	–	–	26.41
inv-PINN+BS	–	–	–	–	–	–	–	–

Model	**Parametric SK**				**Semi-parametric SK**		**Non-parametric SK**	
Panel (B)	SKINN +BS	SKINN +AHBS	SKINN +HSV	SKINN +HSVKDEJ	SKINN +DS-HSV	SKINN +DS-NASV	SKINN +MOPA	SKINN +AE-BS
BS	8.60***	7.35***	8.40***	7.25***	18.79***	11.51***	11.18***	16.19***
AHBS	9.21***	7.53***	8.26***	7.21***	18.72***	11.91***	10.92***	16.16***
HSV	13.13***	12.08***	10.07***	8.68***	19.84***	14.02***	12.35***	17.76***
HSVKDEJ	24.71	23.36	20.17***	18.58***	30.37***	25.80	21.24***	29.03***
NN	25.90	24.46	22.80*	21.48**	28.91**	26.28	21.60**	28.38**
NN+Bnd	28.98**	27.45*	23.50	22.81*	31.38***	28.40**	22.96*	32.26***
NN+MF	14.64***	13.50***	10.88***	10.56***	19.76***	14.02***	10.96***	15.74***
inv-PINN+BS	14.40***	12.80***	11.27***	10.15***	19.13***	14.32***	9.91***	15.99***

Table 18: The Wilcoxon (1945) test results for the out-of-sample option pricing accuracy comparisons in longer prediction horizons. We square all the pricing errors to penalize large errors.

Model	**Structural models**				**Benchmark neural networks**			
Panel (A)	BS	AHBS	HSV	HSVKDEJ	NN	NN +Bnd	NN +MF	inv-PINN +BS
BS	–	25.23	25.75	9.68***	27.71*	22.11**	27.52*	28.20**
AHBS	–	–	24.24	8.48***	27.44*	21.81**	27.11	27.78*
HSV	–	–	–	3.79***	27.75*	22.40**	27.20	28.15**
HSVKDEJ	–	–	–	–	32.41***	27.46*	33.16***	34.08***
NN	–	–	–	–	–	19.16***	26.05	26.34
NN+Bnd	–	–	–	–	–	–	31.56***	31.65***
NN+MF	–	–	–	–	–	–	–	27.27
inv-PINN+BS	–	–	–	–	–	–	–	–
Model	**Parametric SK**				**Semi-parametric SK**		**Non-parametric SK**	
Panel (B)	SKINN +BS	SKINN +AHBS	SKINN +HSV	SKINN +HSVKDEJ	SKINN +DS-HSV	SKINN +DS-NASV	SKINN +MOPA	SKINN +AE-BS
BS	15.94***	16.10***	15.62***	15.30***	23.95	19.26***	18.37***	25.17
AHBS	16.02***	15.78***	15.32***	14.97***	23.22	19.05***	17.93***	24.55
HSV	16.26***	16.23***	15.72***	15.54***	23.85	18.93***	18.27***	25.39
HSVKDEJ	25.28	25.00	23.89	23.23	30.64***	27.94**	24.84	32.38***
NN	19.65***	18.28***	17.07***	16.21***	21.68**	19.72***	15.96***	22.69*
NN+Bnd	22.23**	22.03**	19.91***	18.16***	25.81	22.69*	19.03***	27.66*
NN+MF	15.96***	15.66***	13.88***	13.90***	19.95***	16.67***	13.87***	19.55***
inv-PINN+BS	14.43***	14.56***	13.48***	12.91***	19.24***	16.02***	12.88***	19.35***

Table 19: The Wilcoxon (1945) test results for the out-of-sample option hedging accuracy comparisons in shorter prediction horizons. We square all the hedging errors to penalize large errors.

Model	**Structural models**				**Benchmark neural networks**			
Panel (A)	BS	AHBS	HSV	HSVKDEJ	NN	NN +Bnd	NN +MF	inv-PINN +BS
BS	–	5.16***	46.50***	47.13***	22.27**	21.40***	25.08	24.21
AHBS	–	–	46.79***	47.31***	23.79	23.39	27.21	25.96
HSV	–	–	–	20.19**	9.80***	7.81***	12.32***	10.56***
HSVKDEJ	–	–	–	–	9.59***	8.12***	12.60***	10.81***
NN	–	–	–	–	–	24.41	30.24***	27.59*
NN+Bnd	–	–	–	–	–	–	32.07***	28.85**
NN+MF	–	–	–	–	–	–	–	18.98***
inv-PINN+BS	–	–	–	–	–	–	–	–
Model	**Parametric SK**				**Semi-parametric SK**		**Non-parametric SK**	
Panel (B)	SKINN +BS	SKINN +AHBS	SKINN +HSV	SKINN +HSVKDEJ	SKINN +DS-HSV	SKINN +DS-NASV	SKINN +MOPA	SKINN +AE-BS
BS	4.84***	4.66***	15.47***	14.85***	16.36***	16.69***	17.03***	14.50***
AHBS	7.49***	6.30***	18.35***	18.52***	19.29***	20.05***	19.88***	17.28***
HSV	1.41***	1.41***	2.42***	2.35***	3.63***	3.62***	2.90***	3.78***
HSVKDEJ	1.35***	1.33***	2.42***	2.14***	3.98***	3.81***	3.06***	3.67***
NN	11.61***	11.50***	20.50***	19.68***	20.98***	19.82***	21.25***	18.91***
NN+Bnd	9.61***	9.27***	20.46***	19.87***	20.21***	21.12***	21.07***	17.88***
NN+MF	6.92***	7.12***	15.35***	14.76***	16.16***	16.99***	15.39***	14.01***
inv-PINN+BS	9.39***	9.29***	17.70***	17.80***	18.38***	19.21***	16.76***	16.61***

Table 20: The Wilcoxon (1945) test results for the out-of-sample option hedging accuracy comparisons in longer prediction horizons. We square all the hedging errors to penalize large errors.

Model	**Structural models**				**Benchmark neural networks**			
Panel (A)	BS	AHBS	HSV	HSVKDEJ	NN	NN +Bnd	NN +MF	inv-PINN +BS
BS	–	6.19***	46.34***	46.83***	22.52*	21.78**	24.95	24.75
AHBS	–	–	46.59***	47.12***	23.76	23.64	27.32*	26.56
HSV	–	–	–	20.55**	12.36***	10.91***	14.20***	12.84***
HSVKDEJ	–	–	–	–	12.51***	11.17***	14.49***	13.24***
NN	–	–	–	–	–	23.63	29.38***	28.98**
NN+Bnd	–	–	–	–	–	–	32.50***	31.19***
NN+MF	–	–	–	–	–	–	–	20.48***
inv-PINN+BS	–	–	–	–	–	–	–	–
Model	**Parametric SK**				**Semi-parametric SK**		**Non-parametric SK**	
Panel (B)	SKINN +BS	SKINN +AHBS	SKINN +HSV	SKINN +HSVKDEJ	SKINN +DS-HSV	SKINN +DS-NASV	SKINN +MOPA	SKINN +AE-BS
BS	7.08***	6.07***	16.40***	16.85***	18.39***	17.97***	18.96***	17.60***
AHBS	9.79***	8.27***	19.07***	19.67***	20.70***	20.71***	21.65**	20.35***
HSV	2.23***	2.19***	3.46***	3.12***	4.77***	4.77***	5.70***	5.62***
HSVKDEJ	2.25***	2.18***	3.78***	3.12***	5.02***	5.00***	5.98***	5.93***
NN	12.25***	12.10***	20.85***	20.02***	21.87**	21.18***	20.05***	20.37***
NN+Bnd	9.94***	9.11***	21.32***	20.13***	22.27**	22.03**	20.45***	19.93***
NN+MF	7.82***	7.62***	15.41***	15.19***	16.14***	16.59***	15.83***	15.74***
inv-PINN+BS	8.84***	9.29***	17.80***	17.35***	18.23***	17.83***	16.86***	16.30***

Appendix B Theoretical Foundations of SKINNs

We begin by formally stating the SKINNs optimization problem. Consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ on which all random variables are defined. Let $(\mathbf{X}, \mathbf{y})$ denote a random vector where $\mathbf{X} \in \mathcal{X} \subseteq \mathbb{R}^d$ represents observable input features and $\mathbf{y} \in \mathbb{R}$ represents the target variable. The data-driven component is a deep neural network $f_\theta : \mathcal{X} \to \mathbb{R}$, parameterized by $\theta \in \Theta \subset \mathbb{R}^p$, where Θ is a compact parameter space and p represents the potentially high-dimensional network architecture (all weights and biases across layers). The structured-knowledge component is a function $g_\phi : \mathcal{X}^{\text{SK}} \to \mathbb{R}$, where $\mathcal{X}^{\text{SK}} \subseteq \mathcal{X}$ denotes the subspace of theoretically relevant inputs and $\phi \in \Phi \subset \mathbb{R}^q$ represents the latent parameters of the embedded economic model, with Φ being a compact parameter space.

B.1 Regularity Assumptions

The following assumptions underpin the consistency and asymptotic normality results established in Sections B.2–B.2.3. They are standard in the M-estimation literature (Newey and McFadden, 1994; Van Der Vaart and Wellner, 1996) and are satisfied in most practical implementations of the SKINNs framework. When the structured-knowledge component g_ϕ is misspecified, the minimizer ϕ^* should be interpreted as a pseudo-true parameter, that is, the value that minimizes the population objective $\mathcal{L}(\theta, \phi)$; in this case, the asymptotic results below characterize convergence to this pseudo-true value, as is standard in M-estimation under misspecification (White, 1982).

Assumption 1 (Identification)**.** *The composite objective $\mathcal{L}(\theta, \phi)$ has a unique minimizer (θ^*, ϕ^*) in the interior of $\Theta \times \Phi$. Moreover, the minimum is well-separated: for any $\epsilon > 0$, there exists $\delta > 0$ such that*

$$\inf_{\|(\theta,\phi)-(\theta^*,\phi^*)\| \geq \epsilon} \mathcal{L}(\theta, \phi) > \mathcal{L}(\theta^*, \phi^*) + \delta.$$

Lemma 1 and Proposition 1 in Section 2.4 provide verifiable primitive conditions under which this assumption holds. In particular, Proposition 1 shows that, under squared-error loss with a sufficiently rich function class, identification of ϕ reduces to uniqueness of the minimizer of $\mathbb{E}\Big[\big(\mathbb{E}[\mathbf{y} \mid \mathbf{X}_{\text{obs}}] - g_\phi(\mathbf{X}_{\text{obs}}^{\text{SK}})\big)^2\Big]$ over Φ, a condition that can be checked on a case-by-case basis for specific structured-knowledge representations g_ϕ.

Assumption 2 (Compactness)**.** *The parameter spaces $\Theta \subset \mathbb{R}^p$ and $\Phi \subset \mathbb{R}^q$ are compact.*

Compactness is a technical condition that ensures the existence of minimizers and facilitates uniform convergence arguments. In practice, it can be enforced through bounded parameter constraints or weight clipping. We note that the population identification analysis in Proposition 1

works with an unrestricted function class to characterize the profiled objective, whereas the finite-sample asymptotic theory here requires compactness to control the complexity of the parameter space. This two-stage reasoning—population identification under richness, then finite-sample convergence under compactness—is standard in the sieve estimation literature (Chen, 2007).

Assumption 3 (Continuity)**.** *The loss function $\ell(f, \mathbf{y})$ is continuous in f for all $\mathbf{y}$, and the neural network $f_\theta(\mathbf{X})$ and structured-knowledge function $g_\phi(\mathbf{X}^{SK})$ are continuous in their parameters for all $\mathbf{X}$ and $\mathbf{X}^{SK}$.*

This ensures that small perturbations in (θ, ϕ) induce only small changes in the objective, as required by standard extremum-estimator convergence arguments (Newey and McFadden, 1994). Neural networks with continuous activation functions (e.g., $\text{ReLU}, \text{sigmoid}, \tanh$) satisfy this condition automatically. Throughout the paper, we use squared-error loss unless stated otherwise.

Assumption 4 (Collocation Point Growth)**.** *The number of collocation points M_N is a deterministic sequence satisfying $M_N \to \infty$ as $N \to \infty$. For the asymptotic normality result (Theorem 2), we further require proportional growth:*

$$M_N/N \to c \quad \textit{for some constant } c \in (0, \infty).$$

The first condition ensures that the empirical approximation of the structured-knowledge loss $\mathcal{L}_{\text{SK}}$ converges to its population counterpart, which suffices for consistency (Theorem 1). The proportional growth condition is stronger and ensures that discretization error from the finite collocation grid diminishes at the same rate as sampling error, preserving the parametric $N^{-1/2}$ convergence rate. If M_N remained fixed as N grows, persistent discretization bias would degrade the convergence rate below $N^{-1/2}$. Conversely, if M_N grew too rapidly relative to N (for example, $M_N = N^2$), computational costs would become prohibitive without improving the statistical rate. The proportional growth condition strikes this balance. In the sieve estimation literature (Chen, 2007), analogous conditions govern the growth of basis functions with sample size. In practice, researchers typically set $M_N = N$ (reusing observed inputs as collocation points) or $M_N = cN$ for a small constant $c \in [1, 5]$.

Assumption 5 (Uniform Convergence)**.** *As $N \to \infty$ with M_N satisfying Assumption 4, the empirical loss $\hat{\mathcal{L}}_{N,M_N}(\theta, \phi)$ converges uniformly to the population loss $\mathcal{L}(\theta, \phi)$ over $\Theta \times \Phi$:*

$$\sup_{(\theta,\phi)\in\Theta\times\Phi} \left|\hat{\mathcal{L}}_{N,M_N}(\theta, \phi) - \mathcal{L}(\theta, \phi)\right| \xrightarrow{p} 0.$$

Here $\hat{\mathcal{L}}_{N,M_N}$ denotes the sample analog of the composite objective in Equation (16), with the data loss averaged over N observations and the structured-knowledge loss averaged over M_N collocation points. This uniform law of large numbers requires that the function class $\{(\theta, \phi) \mapsto \ell(f_\theta(\mathbf{X}), \mathbf{y}) + \lambda\, \ell(f_\theta(\mathbf{X}_{\text{grid}}), g_\phi(\mathbf{X}^{\text{SK}}_{\text{grid}}))\}$ is not too complex. For neural networks with a fixed architecture (finite p), bounded activation functions, and compact $\Theta \times \Phi$, this condition is satisfied by standard empirical process theory (Van Der Vaart and Wellner, 1996; Vapnik and Chervonenkis, 1971).

Assumption 6 (Asymptotic Normality of the Score)**.** *The score function*

$$s(\mathbf{X}, \mathbf{y}; \theta, \phi) = \nabla_{(\theta,\phi)} \Big[\ell(f_\theta(\mathbf{X}), \mathbf{y}) + \lambda\, \ell(f_\theta(\tilde{\mathbf{X}}), g_\phi(\tilde{\mathbf{X}}^{SK}))\Big]$$

satisfies a central limit theorem at the true parameter values:

$$\frac{1}{\sqrt{N}} \sum_{i=1}^{N} s(\mathbf{X}_i, \mathbf{y}_i; \theta^*, \phi^*) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \Xi)\,,$$

where $\Xi = \mathbb{E}\big[s(\mathbf{X}, \mathbf{y}; \theta^*, \phi^*)\, s(\mathbf{X}, \mathbf{y}; \theta^*, \phi^*)^\top\big]$ *is the score covariance matrix, assumed to be finite and positive definite.*

This is a standard condition for M-estimator inference (Newey and McFadden, 1994; Van Der Vaart and Wellner, 1996). It holds when the score has finite second moments and the observations are independent and identically distributed, or more generally when the score forms a Donsker class.

Assumption 7 (Second-Order Differentiability)**.** *The objective* $\mathcal{L}(\theta, \phi)$ *is twice continuously differentiable in a neighborhood of* (θ^*, ϕ^*)*, and the Hessian matrix*

$$H = \nabla^2_{(\theta,\phi)}\, \mathcal{L}(\theta^*, \phi^*)$$

exists and is positive definite.

This assumption requires, in particular, that the structured-knowledge function $g_\phi(\mathbf{X}^{\text{SK}})$ is twice differentiable with respect to ϕ. For parametric representations with smooth closed-form expressions (e.g., the Black–Scholes formula), this is immediate. For semi-parametric representations based on deep surrogate neural networks, twice differentiability is inherited from the smoothness of the surrogate's activation functions, as discussed in Section B.6. Positive definiteness of H ensures that (θ^*, ϕ^*) is a strict local minimum and that the sandwich covariance $V = H^{-1}\Xi H^{-1}$ is well-defined.

B.2 Statistical Properties

The SKINNs estimator $(\hat{\theta}_N, \hat{\phi}_N)$ belongs to the class of M-estimators, a broad family of estimators defined as minimizers of an empirical criterion function (Huber et al., 1967; Van Der Vaart and Wellner, 1996). This class encompasses maximum likelihood, least squares, and method of moments estimators as special cases, and admits a mature asymptotic toolkit that we exploit here. Our presentation follows the standard approach in the econometric and statistical literature (Newey and McFadden, 1994; Van Der Vaart and Wellner, 1996), establishing consistency and asymptotic normality under the regularity conditions stated in Appendix B.1. Importantly, these results are domain-agnostic: they apply whether SKINNs are used to learn physical laws from experimental data, infer economic parameters from market observations, or discover structural regularities in other scientific domains.

B.2.1 Consistency

Consistency ensures that, as the sample size grows, the jointly estimated parameters converge in probability to the population minimizers of the composite objective in Equation (16).

Theorem 1 (Consistency)**.** *Suppose Assumptions 1–5 hold, and $M_N \to \infty$ as $N \to \infty$. Then the SKINNs estimator is consistent:*

$$(\hat{\theta}_N, \hat{\phi}_N) \xrightarrow{p} (\theta^*, \phi^*) \quad \text{as } N \to \infty.$$

Proof. The argument follows the standard route for extremum estimators (Newey and McFadden, 1994, Theorem 2.1). By Assumption 5, the empirical loss surface $\hat{\mathcal{L}}_{N,M_N}$ converges uniformly in probability to the population loss $\mathcal{L}$ over the compact set $\Theta \times \Phi$ (Assumption 2). By continuity of $\mathcal{L}$ in (θ, ϕ) (Assumption 3), for any $\epsilon > 0$,

$$\sup_{(\theta,\phi)\in\Theta\times\Phi} \left| \hat{\mathcal{L}}_{N,M_N}(\theta, \phi) - \mathcal{L}(\theta, \phi) \right| \xrightarrow{p} 0$$

implies that any sequence of approximate minimizers of $\hat{\mathcal{L}}_{N,M_N}$ must eventually enter every neighborhood of the set of minimizers of $\mathcal{L}$. The well-separation condition in Assumption 1 guarantees that the minimizer (θ^*, ϕ^*) is unique, so every convergent subsequence of $(\hat{\theta}_N, \hat{\phi}_N)$ converges to (θ^*, ϕ^*). Since $\Theta \times \Phi$ is compact, every subsequence has a further convergent subsequence, and uniqueness of the limit yields convergence of the full sequence. The growth condition $M_N \to \infty$ in Assumption 4 ensures that the collocation-based approximation of the structured-knowledge loss $\mathcal{L}_{\text{SK}}$ contributes to the uniform convergence of the composite objective. □

Under misspecification of the structured-knowledge component g_ϕ, the limiting parameters should be interpreted as pseudo-true values in the sense of White (1982): the parameters that minimize the population objective $\mathcal{L}(\theta, \phi)$ even though g_ϕ may not coincide with the true data-generating process. The consistency guarantee then ensures convergence to these pseudo-true values. This applies universally across domains. In physics, ϕ might represent material constants; in economics, ϕ could be risk preferences or volatility parameters. In either case, Theorem 1 guarantees that given sufficient data, the learned parameters approach their population-optimal values, despite the high-dimensional and potentially nonconvex nature of the neural network optimization landscape.

B.2.2 Asymptotic Normality and Convergence Rates

We now characterize the sampling distribution of the estimation error, enabling the construction of confidence intervals and hypothesis tests for the learned parameters. Define the score function as in Assumption 6:

$$s(\mathbf{X}, \mathbf{y}; \theta, \phi) = \nabla_{(\theta,\phi)} \left[\ell(f_\theta(\mathbf{X}), \mathbf{y}) + \lambda \, \ell(f_\theta(\tilde{\mathbf{X}}), g_\phi(\tilde{\mathbf{X}}^{\mathrm{SK}})) \right], \tag{91}$$

and let

$$\Xi = \mathbb{E}\Big[s(\mathbf{X}, \mathbf{y}; \theta^*, \phi^*) \, s(\mathbf{X}, \mathbf{y}; \theta^*, \phi^*)^\top \Big], \qquad H = \nabla^2_{(\theta,\phi)} \, \mathcal{L}(\theta^*, \phi^*) \tag{92}$$

denote the score covariance and the Hessian of the population loss, respectively.

Theorem 2 (Asymptotic Normality)**.** *Suppose Assumptions 1–7 hold and $M_N/N \to c$ for some $c \in (0, \infty)$. Then the SKINNs estimator is asymptotically normal at the parametric rate:*

$$\sqrt{N} \begin{pmatrix} \hat{\theta}_N - \theta^* \\ \hat{\phi}_N - \phi^* \end{pmatrix} \xrightarrow{d} \mathcal{N}(\mathbf{0}, V), \qquad V = H^{-1} \Xi \, H^{-1}. \tag{93}$$

Proof. Since $(\hat{\theta}_N, \hat{\phi}_N)$ minimizes $\hat{\mathcal{L}}_{N,M_N}$, the first-order condition gives

$$\nabla_{(\theta,\phi)} \hat{\mathcal{L}}_{N,M_N}(\hat{\theta}_N, \hat{\phi}_N) = \mathbf{0}.$$

A mean-value expansion of this condition around the true parameter (θ^*, ϕ^*) yields

$$\mathbf{0} = \frac{1}{N} \sum_{i=1}^{N} s(\mathbf{X}_i, \mathbf{y}_i; \theta^*, \phi^*) + \nabla^2_{(\theta,\phi)} \hat{\mathcal{L}}_{N,M_N}(\bar{\theta}_N, \bar{\phi}_N) \begin{pmatrix} \hat{\theta}_N - \theta^* \\ \hat{\phi}_N - \phi^* \end{pmatrix}, \tag{94}$$

where $(\bar{\theta}_N, \bar{\phi}_N)$ lies on the line segment between $(\hat{\theta}_N, \hat{\phi}_N)$ and (θ^*, ϕ^*). By consistency (Theorem 1), $(\bar{\theta}_N, \bar{\phi}_N) \xrightarrow{p} (\theta^*, \phi^*)$. Combined with the uniform convergence in Assumption 5 and the smoothness in Assumption 7, the empirical Hessian converges in probability:

$$\nabla^2_{(\theta,\phi)} \hat{\mathcal{L}}_{N,M_N}(\bar{\theta}_N, \bar{\phi}_N) \xrightarrow{p} H.$$

Rearranging Equation (94) gives

$$\sqrt{N} \begin{pmatrix} \hat{\theta}_N - \theta^* \\ \hat{\phi}_N - \phi^* \end{pmatrix} = - \left[\nabla^2_{(\theta,\phi)} \hat{\mathcal{L}}_{N,M_N}(\bar{\theta}_N, \bar{\phi}_N) \right]^{-1} \frac{1}{\sqrt{N}} \sum_{i=1}^{N} s(\mathbf{X}_i, \mathbf{y}_i; \theta^*, \phi^*).$$

By Assumption 6, the normalized score converges in distribution to $\mathcal{N}(\mathbf{0}, \Xi)$. Slutsky's theorem, together with the positive definiteness of H (Assumption 7) guaranteeing invertibility, yields

$$\sqrt{N} \begin{pmatrix} \hat{\theta}_N - \theta^* \\ \hat{\phi}_N - \phi^* \end{pmatrix} \xrightarrow{d} \mathcal{N}(\mathbf{0},\, H^{-1} \Xi\, H^{-1}).$$

The proportional growth condition $M_N/N \to c \in (0, \infty)$ in Assumption 4 ensures that the collocation-based approximation of the structured-knowledge loss introduces no additional bias that would degrade the $O_p(N^{-1/2})$ convergence rate: both the data-loss component and the structured-knowledge-loss component of the score contribute to Ξ at the same order. □

Several features of this result merit emphasis. First, the asymptotic covariance $V = H^{-1} \Xi\, H^{-1}$ takes the sandwich form of Huber et al. (1967) and White (1980). This structure is robust to potential misspecification of the structured-knowledge component g_ϕ: even if the embedded theory is an imperfect description of the true data-generating process, the sandwich covariance provides asymptotically valid inference. Second, the convergence rate is $O_p(N^{-1/2})$, the standard parametric rate. This holds despite the potentially very high dimensionality of the neural network parameters θ. The structured-knowledge component acts as implicit regularization, effectively reducing the complexity of the estimation problem. Third, asymptotic normality enables the construction of confidence intervals and hypothesis tests. For a scalar component ϕ_j of the structural parameters, an approximate $(1 - \alpha)$ confidence interval takes the form

$$\hat{\phi}_{j,N} \pm z_{\alpha/2} \sqrt{\frac{V_{jj}}{N}}, \tag{95}$$

where V_{jj} is the (j, j)-th element of V and $z_{\alpha/2}$ is the standard normal critical value.

In Section B.4, we refine these results for the structural parameters under orthogonal moment

conditions, establishing semiparametric efficiency within the given moment model. We note that the formal results above are derived under a fixed neural network architecture, where the dimension p of θ remains constant as the sample size N increases. This aligns with standard M-estimator theory and reflects common empirical practice, where a network architecture is selected and held fixed during training. Extending these results to the growing-width or growing-depth regime, in which the network architecture scales with N as in sieve estimation theory (Chen, 2007), would require additional regularity conditions on the rate of growth and is left for future work. Even under a fixed architecture, modern neural networks with millions of parameters provide substantial approximation capacity for practical applications.

B.2.3 Inference and Practical Considerations

Implementation of the asymptotic results in Theorems 1–2 requires consistent estimators of the Hessian H and the score covariance Ξ. The Hessian can be estimated via the empirical analog

$$\hat{H}_N = \frac{1}{N} \sum_{i=1}^{N} \nabla^2_{(\theta,\phi)} \hat{\mathcal{L}}_{N,M_N}(\hat{\theta}_N, \hat{\phi}_N; \mathbf{X}_i, \mathbf{y}_i), \tag{96}$$

computed efficiently through automatic differentiation. The score covariance is estimated by the outer product of gradients:

$$\hat{\Xi}_N = \frac{1}{N} \sum_{i=1}^{N} s(\mathbf{X}_i, \mathbf{y}_i; \hat{\theta}_N, \hat{\phi}_N)\, s(\mathbf{X}_i, \mathbf{y}_i; \hat{\theta}_N, \hat{\phi}_N)^\top. \tag{97}$$

The sandwich covariance estimator $\hat{V}_N = \hat{H}_N^{-1} \hat{\Xi}_N \hat{H}_N^{-1}$ is then consistent for V under the conditions of Theorem 2. Because scientific conclusions typically center on the low-dimensional structural parameters ϕ rather than the high-dimensional nuisance weights θ, block-matrix inversion can be used to extract the $q \times q$ covariance submatrix for ϕ without computing or inverting the full $(p+q) \times (p+q)$ Hessian. This makes confidence intervals as in Equation (95) and standard Wald-type hypothesis tests for ϕ computationally feasible even when the neural network contains millions of parameters. As a finite-sample alternative, nonparametric bootstrap resampling of $\mathcal{D}_N$ with re-optimization of the SKINNs objective can be used to estimate the sampling distribution of $(\hat{\theta}_N, \hat{\phi}_N)$ without relying on asymptotic approximations.

B.3 The GMM Interpretation for Economic Applications

The statistical properties established in Section B.2 apply universally across scientific domains. For economic and financial applications, however, the SKINNs framework admits a natural rein-

terpretation through the lens of the generalized method of moments (GMM; Hansen, 1982). This perspective clarifies the economic meaning of the composite objective, connects SKINNs to a rich body of established econometric theory, and provides practical guidance for selecting the regularization parameter λ. We emphasize that GMM is one of several valid theoretical lenses on SKINNs; Bayesian, variational, and machine learning perspectives are developed in Section B.7.

B.3.1 SKINNs as Regularized Overidentified GMM

Define the vector-valued moment function

$$\mathbf{m}(\mathbf{X}, \mathbf{y}; \theta, \phi) = \begin{bmatrix} f_\theta(\mathbf{X}) - \mathbf{y} \\ f_\theta(\tilde{\mathbf{X}}) - g_\phi(\tilde{\mathbf{X}}^{\mathrm{SK}}) \end{bmatrix}, \tag{98}$$

where the first component is the prediction error on observed data and the second is the deviation between the neural network and the structured-knowledge function evaluated at collocation points. Under correct specification, both moment conditions are satisfied in expectation:

$$\mathbb{E}[\mathbf{m}(\mathbf{X}, \mathbf{y}; \theta^*, \phi^*)] = \mathbf{0}. \tag{99}$$

The SKINNs objective can then be written as a quadratic form in these moments:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}\Big[\mathbf{m}(\mathbf{X}, \mathbf{y}; \theta, \phi)^\top \mathbf{W}\, \mathbf{m}(\mathbf{X}, \mathbf{y}; \theta, \phi)\Big], \tag{100}$$

with the diagonal weighting matrix $\mathbf{W} = \mathrm{diag}(1, \lambda)$. The empirical counterpart is

$$\hat{\mathcal{L}}_{N,M_N}(\theta, \phi) = \mathbf{m}_N(\theta, \phi)^\top \mathbf{W}\, \mathbf{m}_N(\theta, \phi), \tag{101}$$

where $\mathbf{m}_N(\theta, \phi)$ is the sample analog of the moment vector, and the estimator $(\hat{\theta}_N, \hat{\phi}_N)$ minimizes this weighted quadratic form, exactly as in standard GMM.

The system is overidentified in the sense that two conceptually distinct sources of information, observed outcomes and theoretical structure, each supply a moment condition for the joint estimation of (θ, ϕ). SKINNs differ from classical GMM in three respects. First, θ is high-dimensional and approximates an infinite-dimensional function, whereas classical GMM typically involves finite-dimensional parameters. Second, $\mathbf{W}$ is not the efficiency-optimal weight (the inverse of the moment covariance) but a user-specified diagonal matrix encoding a regularization choice. Third, the second moment condition enforces alignment between f_θ and g_ϕ over a potentially broader input domain than the observed data, acting as a soft structural constraint that promotes generalization to regions

where observations may be sparse.

This formulation clarifies what SKINNs achieves from an economic standpoint. The first moment condition ensures empirical accuracy; the second ensures theoretical plausibility. The weighting parameter λ governs the relative importance of these two objectives: a large λ reflects strong confidence in the embedded theory, while a small λ prioritizes empirical fit.

B.3.2 The Role of λ as Econometric Weighting

In classical GMM, efficiency is achieved by setting the weighting matrix to the inverse of the moment covariance:

$$\mathbf{W}_{\text{optimal}} = \left(\mathbb{E}\Big[\mathbf{m}(\mathbf{X}, \mathbf{y}; \theta^*, \phi^*)\, \mathbf{m}(\mathbf{X}, \mathbf{y}; \theta^*, \phi^*)^\top\Big]\right)^{-1}. \tag{102}$$

SKINNs instead use $\mathbf{W} = \text{diag}(1, \lambda)$, which is generally not optimal in the Hansen-Singleton sense. This deviation is deliberate: when the model is complex (high-dimensional θ) and data are noisy, imposing structure through regularization can improve finite-sample performance even at the cost of asymptotic efficiency, a principle well established in the high-dimensional econometrics literature (Belloni et al., 2014). The parameter λ governs a bias-variance tradeoff: increasing λ reduces variance by constraining the function space but may introduce bias if g_ϕ is misspecified.

Even within the restricted diagonal class $\mathbf{W}(\lambda) = \text{diag}(I, \lambda I)$, one can choose λ to optimize a well-defined criterion.

Proposition 2 (Restricted-optimal λ)**.** *Let $V_\phi(\lambda)$ denote the asymptotic covariance matrix of $\hat{\phi}$ implied by the sandwich formula under weighting $\mathbf{W}(\lambda) = \text{diag}(I, \lambda I)$, assuming the regularity conditions for asymptotic normality hold. Define*

$$\lambda^* \in \arg\min_{\lambda > 0} \ \text{tr}(V_\phi(\lambda))\,. \tag{103}$$

Then λ^* is the variance-minimizing weight within the restricted diagonal class.

Proof. This follows directly from the definition of λ^* as the minimizer of the stated criterion over the admissible set $\lambda > 0$. □

The next result provides a transparent closed-form expression for λ in a stylized setting, clarifying its interpretation as a relative precision ratio.

Proposition 3 (Closed-form λ under two noisy signals)**.** *Assume the target function $f_0(\mathbf{X})$ is*

observed through two conditionally unbiased noisy signals:

$$\mathbf{y} = f_0(\mathbf{X}) + \varepsilon, \qquad g_{\phi_0}(\mathbf{X}^{SK}) = f_0(\mathbf{X}) + \eta, \tag{104}$$

where $\mathbb{E}[\varepsilon \mid \mathbf{X}] = 0$, $\mathbb{E}[\eta \mid \mathbf{X}] = 0$, $\varepsilon \perp\!\!\!\perp \eta \mid \mathbf{X}$, *and* $\mathrm{Var}(\varepsilon \mid \mathbf{X}) = \sigma_\varepsilon^2$, $\mathrm{Var}(\eta \mid \mathbf{X}) = \sigma_\eta^2$ *are constants. Consider estimating* f_0 *by minimizing the SKINNs objective over all measurable* f *with finite second moment:*

$$\mathbb{E}\big[(f(\mathbf{X}) - \mathbf{y})^2\big] + \lambda\, \mathbb{E}\Big[(f(\mathbf{X}) - g_{\phi_0}(\mathbf{X}^{SK}))^2\Big]. \tag{105}$$

Then the minimizer is

$$f^*(\mathbf{X}) = \frac{\mathbf{y} + \lambda\, g_{\phi_0}(\mathbf{X}^{SK})}{1 + \lambda}, \tag{106}$$

and the mean squared error $\mathbb{E}\big[(f^*(\mathbf{X}) - f_0(\mathbf{X}))^2\big]$ *is minimized at*

$$\lambda^* = \frac{\sigma_\varepsilon^2}{\sigma_\eta^2}. \tag{107}$$

Proof. Given ϕ_0, the pointwise minimizer under squared loss is $f^*(\mathbf{X}) = (\mathbf{y} + \lambda\, g_{\phi_0}(\mathbf{X}^{\mathrm{SK}}))/(1 + \lambda)$. Substituting the signal model yields

$$f^*(\mathbf{X}) - f_0(\mathbf{X}) = \frac{\varepsilon + \lambda\, \eta}{1 + \lambda}.$$

Using conditional independence and constant conditional variances,

$$\mathbb{E}\big[(f^*(\mathbf{X}) - f_0(\mathbf{X}))^2 \mid \mathbf{X}\big] = \frac{\sigma_\varepsilon^2 + \lambda^2 \sigma_\eta^2}{(1 + \lambda)^2}.$$

Differentiating with respect to λ and setting the result to zero gives

$$\frac{d}{d\lambda}\left(\frac{\sigma_\varepsilon^2 + \lambda^2 \sigma_\eta^2}{(1 + \lambda)^2}\right) = 0 \quad \Longleftrightarrow \quad \lambda^* = \frac{\sigma_\varepsilon^2}{\sigma_\eta^2},$$

and the second-order condition is satisfied since the objective is strictly convex in λ for $\sigma_\eta^2 > 0$. □

Proposition 3 formalizes the interpretation of λ as a relative precision (inverse-noise-variance) weight: the structured-knowledge signal receives more weight when the data are noisy (σ_ε^2 large) and less weight when the theory is imprecise (σ_η^2 large). In general applications, Proposition 2 motivates choosing λ to optimize a variance or mean squared error criterion within the restricted diagonal class.

B.4 Semiparametric Efficiency

The GMM formulation in Section B.3 connects SKINNs to the literature on semiparametric estimation and penalized moment-based methods. In the semiparametric framework (Bickel et al., 1993), the structural parameters ϕ are finite-dimensional and economically interpretable, while the neural network parameters θ approximate an infinite-dimensional nuisance function. SKINNs can thus be viewed as a neural network implementation of sieve GMM (Ai and Chen, 2003), where the universal approximation properties of deep networks (Cybenko, 1989; Hornik et al., 1989) ensure that f_θ can approximate any continuous function as the network capacity grows. The structured-knowledge component g_ϕ plays a role analogous to an instrument in classical GMM: it provides additional identifying information that refines the estimation of ϕ beyond what pure data fitting would achieve. Recent work on deep GMM (Bennett et al., 2019; Farrell et al., 2021) has shown that neural networks can learn efficient instruments and conditional expectations in moment-based frameworks. SKINNs extend these ideas by incorporating an explicit structured-knowledge component with learnable parameters, enabling joint discovery of both the flexible function and the economically meaningful structural parameters.

The use of λ to balance moment conditions also connects to the literature on regularized GMM. Carrasco et al. (2007) and Antoine and Renault (2009) study GMM with regularization penalties to stabilize estimation under weak instruments or ill-conditioned moment covariance matrices. While their focus is on numerical stability, the underlying principle is shared: judiciously penalizing certain dimensions of the objective can improve finite-sample performance. SKINNs extend this idea to infinite-dimensional function approximation with neural networks.

A natural question is whether the SKINNs estimator achieves any efficiency optimality for the structural parameters ϕ. We show below that under orthogonal moment conditions, in which the infinite-dimensional nuisance parameter exerts only a second-order influence on the estimation of ϕ, the SKINNs estimator attains the semiparametric efficiency bound for the given moment model.

B.4.1 Setup and Assumptions

Let $\mathbf{Z} = (\mathbf{X}, \mathbf{y})$ denote the generic observation and let the finite-dimensional parameter of interest be $\phi \in \mathbb{R}^q$.[23] Let f denote an infinite-dimensional nuisance object taking values in a function space $\mathcal{F}$, with true value f_0. In the SKINNs context, f corresponds to the neural network approximator f_θ and f_0 to its population limit. Suppose the model is characterized by a vector of moment

[23] We use $\mathbf{Z}$ for the generic observation in this subsection to avoid conflict with the GMM weighting matrix $\mathbf{W}$ in Section B.3.

restrictions

$$\mathbb{E}[m(\mathbf{Z};\, \phi^*,\, f_0)] = \mathbf{0}, \qquad m(\cdot;\, \phi,\, f) \in \mathbb{R}^k, \quad k \geq q, \tag{108}$$

where ϕ^* denotes the true (or pseudo-true) parameter value.[24] Let $\hat{f}$ be a first-stage estimator of f_0 and define the feasible optimal-weight GMM estimator

$$\hat{\phi} \in \arg\min_{\phi \in \Phi}\ \hat{g}(\phi)^\top \hat{\Omega}^{-1} \hat{g}(\phi), \qquad \hat{g}(\phi) \equiv \frac{1}{N}\sum_{i=1}^{N} m(\mathbf{Z}_i;\, \phi,\, \hat{f}), \tag{109}$$

where $\hat{\Omega}$ is a consistent estimator of $\Omega \equiv \mathbb{E}\big[m(\mathbf{Z};\, \phi^*,\, f_0)\, m(\mathbf{Z};\, \phi^*,\, f_0)^\top\big]$.

Assumption 8. (i) ***(Smoothness)*** *$m(\mathbf{Z};\, \phi,\, f)$ is continuously differentiable in ϕ in a neighborhood of (ϕ^*, f_0), and the Jacobian*

$$G \equiv \frac{\partial}{\partial \phi^\top}\mathbb{E}[m(\mathbf{Z};\, \phi,\, f_0)]\,\Big|_{\phi=\phi^*} \tag{110}$$

exists and has full column rank q.

(ii) ***(Orthogonality)*** *For every direction h in the tangent set of $\mathcal{F}$ at f_0,*

$$\frac{d}{dt}\,\mathbb{E}[m(\mathbf{Z};\, \phi^*,\, f_0 + t\,h)]\Big|_{t=0} = \mathbf{0}.$$

(iii) ***(Rates)*** *$\|\hat{f} - f_0\| = o_p(N^{-1/4})$ under a norm such that the remainder bounds in* (iv) *hold.*

(iv) ***(Remainder control)*** *Uniformly over ϕ in a neighborhood of ϕ^*,*

$$\left\| \frac{1}{N}\sum_{i=1}^{N} \Big(m(\mathbf{Z}_i;\, \phi,\, \hat{f}) - m(\mathbf{Z}_i;\, \phi,\, f_0) \Big) - \mathbb{E}\Big[m(\mathbf{Z};\, \phi,\, \hat{f}) - m(\mathbf{Z};\, \phi,\, f_0) \Big] \right\| = o_p(N^{-1/2}),$$

and the population bias admits a second-order bound:

$$\Big\| \mathbb{E}\Big[m(\mathbf{Z};\, \phi^*,\, \hat{f}) - m(\mathbf{Z};\, \phi^*,\, f_0) \Big] \Big\| \leq C\, \|\hat{f} - f_0\|^2$$

for some constant C with probability approaching one.

Assumption 8(ii) is the key condition. Orthogonality ensures that estimation error in the first-stage nuisance parameter $\hat{f}$ has only a second-order effect on the estimation of ϕ, so that the infinite-dimensional estimation problem does not degrade the $N^{-1/2}$ convergence rate for the finite-dimensional parameter of interest. This condition is the semiparametric analog of the Neyman orthogonality condition used in the debiased machine learning literature (Chernozhukov et al.,

[24] We use Ω for the moment covariance matrix in this subsection, following the convention in the semiparametric efficiency literature. This is distinct from the score covariance Ξ defined in Section B.2. Under correct specification of the moment model, Ω coincides with the relevant block of Ξ.

2018). Assumption 8(iii) requires that the first-stage estimator converges at a rate faster than $N^{-1/4}$, a condition satisfied by neural network sieves under standard smoothness conditions on the target function (Chen, 2007; Farrell et al., 2021).

B.4.2 Efficiency Result

Theorem 3 (Asymptotic linearity and semiparametric efficiency)**.** *Under Assumption 8 and consistency of* $\hat{\Omega}$*,*

$$\sqrt{N}(\hat{\phi} - \phi^*) \;\Rightarrow\; \mathcal{N}(\mathbf{0},\, V^*)\,, \qquad V^* \equiv (G^\top \Omega^{-1} G)^{-1}. \tag{111}$$

Moreover, among regular asymptotically linear estimators based on the moment restriction $\mathbb{E}[m(\mathbf{Z};\, \phi,\, f_0)] = \mathbf{0}$*, the covariance* V^* *is the minimal achievable asymptotic variance (the optimal-GMM bound), so* $\hat{\phi}$ *is semiparametrically efficient for* ϕ *relative to this moment model.*

Proof. Step 1 (Expansion of sample moments). Write

$$\hat{g}(\phi) = \frac{1}{N}\sum_{i=1}^{N} m(\mathbf{Z}_i;\, \phi,\, f_0) + A_N(\phi) + B_N(\phi), \tag{112}$$

where

$$A_N(\phi) \equiv \frac{1}{N}\sum_{i=1}^{N}\Big(m(\mathbf{Z}_i;\, \phi,\, \hat{f}) - m(\mathbf{Z}_i;\, \phi,\, f_0)\Big) - \mathbb{E}\Big[m(\mathbf{Z};\, \phi,\, \hat{f}) - m(\mathbf{Z};\, \phi,\, f_0)\Big]$$

is the stochastic equicontinuity remainder, and

$$B_N(\phi) \equiv \mathbb{E}\Big[m(\mathbf{Z};\, \phi,\, \hat{f}) - m(\mathbf{Z};\, \phi,\, f_0)\Big]$$

is the population bias. By Assumption 8(iv), uniformly near ϕ^*, $A_N(\phi) = o_p(N^{-1/2})$.

Step 2 (Orthogonality eliminates the first-stage bias). Applying Assumption 8(ii) and the second-order remainder bound in Assumption 8(iv) at $\phi = \phi^*$:

$$\|B_N(\phi^*)\| = \Big\|\mathbb{E}\Big[m(\mathbf{Z};\, \phi^*,\, \hat{f}) - m(\mathbf{Z};\, \phi^*,\, f_0)\Big]\Big\| \le C\,\|\hat{f} - f_0\|^2 = o_p(N^{-1/2}),$$

where the final equality uses $\|\hat{f} - f_0\| = o_p(N^{-1/4})$ from Assumption 8(iii).

Step 3 (Taylor expansion in ϕ*).* Expand $\hat{g}(\phi)$ around ϕ^*:

$$\hat{g}(\phi) = \hat{g}(\phi^*) + \hat{G}(\bar{\phi})\,(\phi - \phi^*),$$

where $\bar{\phi}$ lies between ϕ and ϕ^*, and $\hat{G}(\phi) \equiv \frac{\partial}{\partial \phi^\top}\hat{g}(\phi)$. By Assumption 8(i) and standard law-of-large-numbers arguments, $\hat{G}(\bar{\phi}) \xrightarrow{p} G$ uniformly in a neighborhood of ϕ^*.

Step 4 (First-order condition). The GMM objective is $J_N(\phi) = \hat{g}(\phi)^\top \hat{\Omega}^{-1} \hat{g}(\phi)$. The first-order condition at $\hat{\phi}$ is

$$\mathbf{0} = \hat{G}(\hat{\phi})^\top \hat{\Omega}^{-1} \hat{g}(\hat{\phi}).$$

Substituting the Taylor expansion from Step 3 and rearranging:

$$\sqrt{N}(\hat{\phi} - \phi^*) = -\left(\hat{G}(\hat{\phi})^\top \hat{\Omega}^{-1} \hat{G}(\bar{\phi})\right)^{-1} \hat{G}(\hat{\phi})^\top \hat{\Omega}^{-1} \sqrt{N}\, \hat{g}(\phi^*). \tag{113}$$

By consistency, $\hat{\Omega} \xrightarrow{p} \Omega$ and $\hat{G}(\cdot) \xrightarrow{p} G$, so the matrix prefactor converges to $(G^\top \Omega^{-1} G)^{-1} G^\top \Omega^{-1}$, which is well-defined by the full-rank condition in Assumption 8(i).

Step 5 (CLT for the leading term). Combining Steps 1 and 2,

$$\sqrt{N}\, \hat{g}(\phi^*) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} m(\mathbf{Z}_i;\, \phi^*,\, f_0) + o_p(1).$$

By the multivariate central limit theorem, the leading term converges in distribution to $\mathcal{N}(\mathbf{0}, \Omega)$. Applying the continuous mapping theorem to Equation (113) yields

$$\sqrt{N}(\hat{\phi} - \phi^*) \;\Rightarrow\; \mathcal{N}\Big(\mathbf{0},\, (G^\top \Omega^{-1} G)^{-1}\Big).$$

Step 6 (Efficiency). For the moment condition $\mathbb{E}[m(\mathbf{Z};\, \phi,\, f_0)] = \mathbf{0}$ with Jacobian G and moment covariance Ω, the class of regular GMM estimators indexed by positive definite weighting matrices $\mathbf{W}$ has asymptotic variance

$$V(\mathbf{W}) = (G^\top \mathbf{W} G)^{-1} G^\top \mathbf{W}\, \Omega\, \mathbf{W} G\, (G^\top \mathbf{W} G)^{-1}.$$

A standard positive-semidefinite ordering argument shows that $V(\mathbf{W}) - V^*$ is positive semidefinite for all positive definite $\mathbf{W}$, with equality attained at $\mathbf{W} = \Omega^{-1}$. Hence $V^* = (G^\top \Omega^{-1} G)^{-1}$ is the minimal asymptotic variance achievable by regular GMM estimators based on these moments, establishing semiparametric efficiency. □

Theorem 3 places SKINNs within the family of sieve GMM estimators (Ai and Chen, 2003) and deep GMM methods (Bennett et al., 2019; Farrell et al., 2021), while distinguishing it through the explicit incorporation of a learnable structured-knowledge component. The orthogonality condition ensures that the flexibility of the neural network approximator does not inflate the asymptotic variance of the structural parameter estimates, a property that is particularly valuable in high-dimensional settings where the nuisance parameter space is vast relative to the sample size.

B.5 Generalization Bounds and Distributional Robustness

Out-of-sample performance is a central concern in finance, where distributional shifts between training and deployment periods are common. This subsection provides two complementary formal results. The first (Theorem 4) shows that structured regularization tightens the uniform stability bound on the expected generalization gap, formalized in a tractable convex proxy. The second (Theorem 5) decomposes the risk under a shifted target distribution into an alignment term controlled by the structured-knowledge loss and a portability term measuring how well g_ϕ transfers to the new environment. Together, these results formalize a compelling intuition: if the embedded theoretical model captures structural regularities that remain stable across regimes, the SKINNs estimator inherits this stability.

B.5.1 Generalization via Uniform Stability

We analyze the generalization properties of structured regularization in a convex proxy that admits sharp stability bounds. Consider a linear-in-parameters approximation $f_\theta(\mathbf{X}) = \theta^\top \mathbf{X}$ with $\theta \in \mathbb{R}^p$, and assume bounded features $\|\mathbf{X}\| \le B_x$ almost surely.[25] Let $S = \{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^N$ denote the observed sample and let $\tilde{S} = \{\tilde{\mathbf{X}}_j\}_{j=1}^M$ be a set of collocation points drawn independently from a (possibly different) distribution on the input domain. Fix ϕ and write $g_j \equiv g_\phi(\tilde{\mathbf{X}}_j^{\mathrm{SK}})$.

Define the empirical SKINNs objective with an explicit ridge term ($\rho > 0$):

$$\hat{\mathcal{L}}_S(\theta) \equiv \frac{1}{N}\sum_{i=1}^{N}(\theta^\top \mathbf{X}_i - \mathbf{y}_i)^2 + \lambda \frac{1}{M}\sum_{j=1}^{M}(\theta^\top \tilde{\mathbf{X}}_j - g_j)^2 + \rho\,\|\theta\|^2, \tag{114}$$

and let $\hat{\theta}(S)$ denote its unique minimizer. Define the population risk and the empirical risk as

$$R(\theta) \equiv \mathbb{E}\Big[(\theta^\top \mathbf{X} - \mathbf{y})^2\Big], \qquad \hat{R}_S(\theta) \equiv \frac{1}{N}\sum_{i=1}^{N}(\theta^\top \mathbf{X}_i - \mathbf{y}_i)^2.$$

Theorem 4 (Uniform stability and generalization bound)**.** *Assume $\|\mathbf{X}\| \le B_x$ and $\|\tilde{\mathbf{X}}\| \le B_x$ almost surely, and $|\mathbf{y}| \le B_y$ almost surely. Let $\mu \equiv \rho$, so that $\hat{\mathcal{L}}_S$ is at least μ-strongly convex in θ. Then $\hat{\theta}(\cdot)$ is uniformly stable: for any two samples S, S' that differ in exactly one observation, and for any $(\mathbf{X}, \mathbf{y})$ with $\|\mathbf{X}\| \le B_x$ and $|\mathbf{y}| \le B_y$,*

$$\Big|\ell(\hat{\theta}(S);\, \mathbf{X}, \mathbf{y}) - \ell(\hat{\theta}(S');\, \mathbf{X}, \mathbf{y})\Big| \le \frac{4\,B_x^2\,(B_x\|\hat{\theta}(S)\| + B_y)}{\mu\, N}, \tag{115}$$

[25] The restriction to a linear model is made to obtain an explicit stability bound. The qualitative insight, that structured regularization tightens the generalization gap by increasing the strong-convexity parameter of the objective, extends to nonlinear models under appropriate conditions (see, e.g., **?**).

where $\ell(\theta;\, \mathbf{X}, \mathbf{y}) \equiv (\theta^\top \mathbf{X} - \mathbf{y})^2$. *Consequently, the expected generalization gap satisfies*

$$\mathbb{E}\Big[R(\hat{\theta}(S)) - \hat{R}_S(\hat{\theta}(S))\Big] \leq \frac{4\,B_x^2}{\mu\,N}\,\mathbb{E}\Big[B_x\|\hat{\theta}(S)\| + B_y\Big]. \tag{116}$$

In particular, for fixed data and bounded $\mathbb{E}\|\hat{\theta}(S)\|$, *increasing* μ *(equivalently, increasing* ρ *or* λ*) tightens the bound.*

Proof. Step 1 (Strong convexity). The ridge term $\rho\,\|\theta\|^2$ makes $\hat{\mathcal{L}}_S$ at least ρ-strongly convex. The structured-knowledge term with $\lambda > 0$ adds further convexity, so $\mu = \rho$ is a conservative lower bound.

Step 2 (Parameter sensitivity). Let S and S' differ only in the N-th observation. Define

$$\Delta(\theta) \equiv \hat{\mathcal{L}}_S(\theta) - \hat{\mathcal{L}}_{S'}(\theta) = \frac{1}{N}\left((\theta^\top \mathbf{X}_N - \mathbf{y}_N)^2 - (\theta^\top \mathbf{X}'_N - \mathbf{y}'_N)^2\right).$$

By the optimality conditions for strongly convex objectives,

$$\|\hat{\theta}(S) - \hat{\theta}(S')\| \leq \frac{\|\nabla\Delta(\hat{\theta}(S))\|}{\mu}.$$

Computing the gradient,

$$\nabla\Delta(\theta) = \frac{2}{N}\left((\theta^\top \mathbf{X}_N - \mathbf{y}_N)\,\mathbf{X}_N - (\theta^\top \mathbf{X}'_N - \mathbf{y}'_N)\,\mathbf{X}'_N\right),$$

and applying the triangle inequality with the bounds $\|\mathbf{X}_N\|, \|\mathbf{X}'_N\| \leq B_x$ and $|\mathbf{y}_N|, |\mathbf{y}'_N| \leq B_y$ yields

$$\|\hat{\theta}(S) - \hat{\theta}(S')\| \leq \frac{4\,B_x\,(B_x\|\hat{\theta}(S)\| + B_y)}{\mu\,N}.$$

Step 3 (Loss sensitivity). For any $(\mathbf{X}, \mathbf{y})$ and any θ, θ',

$$|\ell(\theta;\, \mathbf{X}, \mathbf{y}) - \ell(\theta';\, \mathbf{X}, \mathbf{y})| \leq \big(B_x(\|\theta\| + \|\theta'\|) + 2\,B_y\big) \cdot B_x\,\|\theta - \theta'\|.$$

Substituting $\theta = \hat{\theta}(S)$, $\theta' = \hat{\theta}(S')$, and the bound from Step 2, and absorbing $\|\theta'\|$ into a conservative bound, yields Equation (115).

Step 4 (Expected generalization gap). Uniform stability with modulus β_N implies $\mathbb{E}[R(\hat{\theta}(S)) - \hat{R}_S(\hat{\theta}(S))] \leq \beta_N$ by the standard argument of **?**. Applying the bound from Step 3 yields Equation (116). □

The bound in Equation (116) makes precise the regularization benefit of the structured-knowledge

component: both ρ (explicit ridge) and λ (structured-knowledge penalty) contribute to the strong-convexity parameter μ, tightening the generalization gap. In the absence of structured regularization ($\lambda = 0$), only the ridge term controls stability, and a larger ridge penalty is needed to achieve the same bound, at the cost of greater shrinkage bias. The structured-knowledge penalty achieves a similar stabilizing effect while directing the regularization toward theoretically meaningful regions of the parameter space.

B.5.2 Target-Risk Decomposition Under Distribution Shift

We now consider the setting where the deployment (target) distribution differs from the training (source) distribution, as is typical in financial applications subject to regime changes. Let P denote the training distribution of $(\mathbf{X}, \mathbf{y})$ and Q a target distribution. For any measurable predictor f and any ϕ, define the squared risk under Q:

$$R_Q(f) \equiv \mathbb{E}_Q\left[(f(\mathbf{X}) - \mathbf{y})^2\right]. \tag{117}$$

Theorem 5 (Target-risk decomposition)**.** *For any measurable f and any ϕ,*

$$R_Q(f) \leq 2\,\mathbb{E}_Q\Big[(f(\mathbf{X}) - g_\phi(\mathbf{X}^{SK}))^2\Big] + 2\,\mathbb{E}_Q\Big[(g_\phi(\mathbf{X}^{SK}) - \mathbf{y})^2\Big]. \tag{118}$$

The first term measures the alignment between f and g_ϕ under the target distribution, and the second term measures the portability of g_ϕ to the target environment. If g_ϕ is portable (the second term is small), then controlling the alignment term controls target risk.

Proof. Write $f(\mathbf{X}) - \mathbf{y} = (f(\mathbf{X}) - g_\phi(\mathbf{X}^{\mathrm{SK}})) + (g_\phi(\mathbf{X}^{\mathrm{SK}}) - \mathbf{y})$. Applying the elementary inequality $(a+b)^2 \leq 2a^2 + 2b^2$ and taking expectations under Q yields Equation (118). □

Corollary 6 (Collocation design controls the alignment term)**.** *Suppose the collocation distribution used for $\tilde{\mathbf{X}}$ equals the target marginal distribution of $\mathbf{X}$ under Q. Then the first term in Theorem 5 coincides with the structured-knowledge loss:*

$$\mathbb{E}_Q\Big[(f(\mathbf{X}) - g_\phi(\mathbf{X}^{SK}))^2\Big] = \mathbb{E}_{\tilde{\mathbf{X}}}\Big[(f(\tilde{\mathbf{X}}) - g_\phi(\tilde{\mathbf{X}}^{SK}))^2\Big],$$

which is directly minimized during SKINNs training via $\mathcal{L}_{SK}$.

Proof. Under the stated condition, the marginal distribution of $\mathbf{X}$ under Q equals the distribution of $\tilde{\mathbf{X}}$, so the two expectations coincide. □

Theorem 5 and Corollary 6 together formalize a key practical insight for financial applications. If the embedded theoretical model captures structural regularities, such as no-arbitrage conditions or equilibrium relations, that remain stable across market regimes, then g_ϕ is portable and the second term in Equation (118) remains small under distributional shift. The first term is directly controlled by the SKINNs training objective when the collocation distribution is chosen to reflect the anticipated target environment. This provides a principled mechanism for robustness: rather than relying solely on the i.i.d. assumption underlying standard generalization theory, SKINNs anchor predictions to theoretical structures whose validity transcends any particular data-generating regime.

B.6 Differentiability and Curse of Dimensionality

The semi-parametric structured-knowledge representations introduced in Section 2.2 require that the surrogate model $g_\phi(\mathbf{X}^{\mathrm{SK}}) \equiv f_{\theta_s}^{\mathrm{surrogate}}(\mathbf{X}^{\mathrm{SK}}, \phi)$ be first-order differentiable with respect to both its observable inputs $\mathbf{X}^{\mathrm{SK}}$ and the learnable latent parameters ϕ. This subsection establishes that the neural network architecture of the surrogate guarantees this differentiability, derives the explicit gradient flow, and clarifies how deep surrogates circumvent the curse of dimensionality that afflicts direct PDE/SDE-based knowledge embedding.

B.6.1 End-to-End Differentiability

Once trained offline (as described in Section 2.2), the surrogate weights θ_s are frozen. The input to the surrogate is the concatenation of observable features and latent parameters:

$$\mathbf{z}_0 = \begin{bmatrix} \mathbf{X}^{\mathrm{SK}} \\ \phi \end{bmatrix}. \tag{119}$$

The surrogate is a composition of differentiable affine transformations and smooth activation functions $\sigma(\cdot)$:

$$f_{\theta_s}^{\mathrm{surrogate}}(\mathbf{z}_0) = \mathbf{W}_L\, \sigma(\cdots \sigma(\mathbf{W}_1 \mathbf{z}_0 + \mathbf{b}_1) \cdots) + \mathbf{b}_L. \tag{120}$$

Since each layer is differentiable with respect to its input, the composite function is differentiable with respect to ϕ by the chain rule. The Jacobian of the surrogate output with respect to ϕ is the product of layer-wise Jacobians:

$$\mathbf{J}_s(\phi) \equiv \frac{\partial\, f_{\theta_s}^{\mathrm{surrogate}}(\mathbf{X}^{\mathrm{SK}}, \phi)}{\partial \phi} = \prod_{l=1}^{L} \mathrm{diag}\big(\sigma_l'(\mathbf{h}_l)\big)\, \mathbf{W}_l, \tag{121}$$

where $\mathbf{h}_l$ is the pre-activation vector at layer l, and the product is taken in reverse order (from output to input layer). During SKINNs training, the gradient of the composite loss with respect to ϕ is computed via the vector-Jacobian product:

$$\nabla_\phi \mathcal{L} = \mathbf{J}_s(\phi)^\top \nabla_{g_\phi} \mathcal{L}, \tag{122}$$

which is evaluated efficiently by automatic differentiation in a single backward pass through the frozen surrogate. This ensures that the asymptotic properties established in Section B.2, which require first-order differentiability of g_ϕ with respect to ϕ, are satisfied by construction. The second-order differentiability required by Assumption 7 likewise holds whenever the activation functions $\sigma(\cdot)$ are twice continuously differentiable (e.g., sigmoid, tanh, softplus, GELU), and holds almost everywhere for piecewise-linear activations such as ReLU.

B.6.2 Circumventing the Curse of Dimensionality

A key motivation for the surrogate approach is computational tractability when the structured knowledge is prescribed by high-dimensional SDEs or PDEs. For methods that embed PDE/SDE knowledge directly into the neural network (as in PINNs), the minimum network size required for convergence increases exponentially with both the PDE order and the dimensionality of the state variables (Gao et al., 2023; Song et al., 2024). This renders direct embedding infeasible for the rich multivariate SDE systems commonly encountered in finance, such as the stochastic volatility models in Equations (3)–(5), which involve multiple correlated state variables, jump processes, and ten or more latent parameters.

The deep surrogate strategy decouples the SKINNs optimization from the dimensionality of the underlying SDE/PDE. The computational cost of generating the synthetic training data $\mathcal{D}_{\text{surrogate}}$ (via Monte Carlo simulation or finite-difference methods) is incurred once during the offline pre-training phase. Once trained, the surrogate $f_{\theta_s}^{\text{surrogate}}$ operates as a global function approximator whose parameter count scales polynomially, rather than exponentially, with the input dimension (Grohs et al., 2023). Each evaluation of g_ϕ during SKINNs training requires only a forward pass through the frozen surrogate, at a cost independent of the complexity of the original SDE/PDE system. This ensures that the per-iteration cost of SKINNs training remains computationally tractable even when the embedded structured knowledge originates from high-dimensional theoretical models.

B.7 SKINNs as a Unifying Framework

A distinctive strength of the SKINNs framework is that its composite objective in Equation (16) is not tied to a single methodological tradition. The preceding sections have developed the M-estimator foundations (Section B.2), the GMM interpretation (Section B.3), and the semiparametric efficiency properties (Section B.4). In this section, we show that several well-established paradigms emerge as special cases or limiting configurations of the same composite objective, with the specific instantiation determined by the choice of g_ϕ, the loss structure, and the regularization strength λ. This multiplicity of valid interpretations reflects a deeper architectural point: functional GMM, Bayesian MAP estimation, transfer learning, physics-informed learning, and domain adaptation all reside within the SKINNs framework. We develop each connection below.

B.7.1 Bayesian MAP Estimation

From a Bayesian perspective, the SKINNs objective corresponds to maximum a posteriori (MAP) estimation with a theory-informed prior. Assume independent Gaussian observation noise, $\mathbf{y}_i = f_\theta(\mathbf{X}_i) + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, so that the log-likelihood is proportional to the negative data loss:

$$\log p(\mathcal{D}_N \mid \theta, \phi) = -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (f_\theta(\mathbf{X}_i) - \mathbf{y}_i)^2 + \text{const.} \tag{123}$$

Specify the conditional prior $p(\theta \mid \phi)$ as encoding the belief that f_θ should be close to the theoretical model g_ϕ:

$$\log p(\theta \mid \phi) \propto -\frac{\lambda}{2} \, \mathbb{E}\Big[(f_\theta(\tilde{\mathbf{X}}) - g_\phi(\tilde{\mathbf{X}}^{\text{SK}}))^2 \Big], \tag{124}$$

and take the prior on ϕ to be uninformative. The MAP estimator $(\hat{\theta}^{\text{MAP}}, \hat{\phi}^{\text{MAP}}) = \arg\max_{\theta,\phi} \, p(\theta, \phi \mid \mathcal{D}_N)$ then solves, after taking negative logarithms and normalizing:

$$(\hat{\theta}^{\text{MAP}}, \hat{\phi}^{\text{MAP}}) = \arg\min_{\theta,\phi} \left[\frac{1}{N} \sum_{i=1}^{N} (f_\theta(\mathbf{X}_i) - \mathbf{y}_i)^2 + \lambda \, \mathbb{E}\Big[(f_\theta(\tilde{\mathbf{X}}) - g_\phi(\tilde{\mathbf{X}}^{\text{SK}}))^2 \Big] \right], \tag{125}$$

which recovers the SKINNs objective in Equation (16). In this reading, the structured-knowledge loss $\mathcal{L}_{\text{SK}}$ acts as a negative log-prior that softly constrains f_θ to lie near g_ϕ. The regularization parameter λ is the prior precision: a large λ reflects strong prior belief in the theory, while a small λ reflects a diffuse prior. The joint optimization over (θ, ϕ) corresponds to learning both the function and the hyperparameters of the prior center, which is a form of empirical Bayes estimation.

This connection places SKINNs within the tradition of Bayesian neural networks (Neal, 1996; MacKay, 1992), with a crucial distinction: the prior is not generic (e.g., weight decay) but theory-

informed, centered at a scientifically meaningful function g_ϕ whose parameters are themselves learned from data. A fully Bayesian treatment would characterize the entire posterior $p(\theta, \phi \mid \mathcal{D}_N)$ via variational inference (Blundell et al., 2015) or Markov chain Monte Carlo, at substantially greater computational cost. The MAP point estimate provided by SKINNs offers a tractable compromise that captures the posterior mode while remaining computationally feasible.

B.7.2 Physics-Informed Learning and Variational Principles

The SKINNs objective admits a natural reading as a total energy functional:

$$E[f_\theta, g_\phi] = \underbrace{\mathbb{E}\big[(f_\theta(\mathbf{X}) - \mathbf{y})^2\big]}_{E_{\text{data}}} + \lambda \underbrace{\mathbb{E}\Big[(f_\theta(\tilde{\mathbf{X}}) - g_\phi(\tilde{\mathbf{X}}^{\text{SK}}))^2\Big]}_{E_{\text{theory}}}, \tag{126}$$

where E_{data} measures empirical misfit and E_{theory} penalizes deviations from theoretical predictions. Training via gradient descent approximates the continuous-time gradient flow

$$\frac{d\theta}{dt} = -\nabla_\theta E[f_\theta, g_\phi], \qquad \frac{d\phi}{dt} = -\nabla_\phi E[f_\theta, g_\phi], \tag{127}$$

analogous to overdamped Langevin dynamics in statistical physics, in which the system dissipates energy and settles into a local minimum representing a stable equilibrium (Welling and Teh, 2011). Equivalently, the penalized formulation in Equation (16) is dual to a constrained optimization problem

$$\min_{\theta,\phi} \; E_{\text{data}}[f_\theta] \qquad \text{subject to} \qquad E_{\text{theory}}[f_\theta, g_\phi] \leq \epsilon, \tag{128}$$

where λ plays the role of the Lagrange multiplier (Bertsekas, 2014).

The variational perspective clarifies a key distinction between SKINNs and Physics-Informed Neural Networks (PINNs; Raissi et al., 2019). PINNs enforce differential equations as soft constraints by penalizing PDE residuals:

$$\mathcal{L}_{\text{PINN}} = \mathcal{L}_{\text{data}} + \lambda \, \mathbb{E}\big[\|\mathbb{D}[f_\theta] - 0\|^2\big], \tag{129}$$

where $\mathbb{D}[\cdot]$ is a differential operator. Computing $\mathbb{D}[f_\theta]$ requires automatic differentiation of the neural network outputs with respect to its inputs, which can be numerically unstable for high-order derivatives and leads to well-documented spectral biases and gradient pathologies (Wang et al., 2020, 2022). SKINNs instead enforce consistency with the *solution* of a theoretical model, $g_\phi(\mathbf{X}^{\text{SK}})$, rather than with a differential operator. This has two advantages. First, it avoids the numerical instabilities associated with differentiating neural networks with respect to inputs. Second, it allows

g_ϕ to be non-analytical (e.g., a deep surrogate or an auto-encoder), whereas PINNs require explicit differential forms. When g_ϕ is set to the PDE residual operator and $\lambda \to \infty$, SKINNs recover the PINNs objective as a limiting case.

B.7.3 Transfer Learning and Domain Adaptation

SKINNs generalize two-stage transfer learning (Pan and Yang, 2010; Pratt, 1992) in which a model pre-trained on a source task is fine-tuned on a target task. In transfer learning, the source model provides a fixed initialization; the target data then adjust the parameters, but the source model does not adapt. In SKINNs, the structured-knowledge component g_ϕ plays the role of the source model, but with two critical extensions: (i) g_ϕ provides guidance throughout training (not only at initialization), and (ii) its parameters ϕ are jointly updated with the neural network parameters θ, enabling bidirectional knowledge transfer in a single optimization loop. When ϕ is fixed and g_ϕ serves only to initialize f_θ before fine-tuning on $\mathcal{D}_N$, SKINNs reduce to standard transfer learning.

The framework also connects to domain adaptation (Ben-David et al., 2010), where the goal is to learn a function that generalizes across different data distributions. In SKINNs, the structured-knowledge loss enforces alignment between f_θ and g_ϕ over a potentially broad collocation domain $\tilde{\mathbf{X}}$, including regions where observed data are sparse. This promotes generalization to out-of-distribution inputs, as formalized by the target-risk decomposition in Theorem 5: if g_ϕ is portable across distributions, the alignment enforced by $\mathcal{L}_{\text{SK}}$ during training directly controls the risk under the shifted target distribution.

B.7.4 Summary: Limiting Configurations

Table 21 summarizes how several established paradigms emerge as special cases of the SKINNs composite objective.

Table 21: SKINNs as a unifying framework: limiting configurations of the composite objective.

Paradigm	**Configuration**	**Reference**
Plain neural network	$\lambda = 0$	Gu et al. (2020)
PINNs	$g_\phi = \mathbb{D}[f_\theta]$, $\lambda \to \infty$	Raissi et al. (2019)
Transfer learning	ϕ fixed, two-stage training	Pan and Yang (2010)
Bayesian MAP estimation	$\mathcal{L}_{\text{SK}}$ as neg. log-prior	Neal (1996)
Regularized / functional GMM	$\mathcal{L}_{\text{data}}, \mathcal{L}_{\text{SK}}$ as moment conditions	Hansen (1982)
Sieve GMM / Deep GMM	f_θ as neural sieve, optimal $\mathbf{W}$	Ai and Chen (2003)
Structured regularization	$\mathcal{L}_{\text{SK}}$ as domain-adaptive penalty	Ben-David et al. (2010)

Each row corresponds to a specific choice of g_ϕ, loss structure, regularization strength λ, or

training protocol. The full SKINNs framework encompasses all of these configurations simultaneously, while its joint optimization of (θ, ϕ) enables capabilities, such as dynamic latent parameter discovery and bidirectional theory-data reconciliation, that none of the individual paradigms provide in isolation.

References

Aboussalah, A.M., Li, X., Chi, C., Patel, R., 2024. The AI Black-Scholes: Finance-Informed Neural Network. doi:10.48550/arXiv.2412.12213, arXiv:2412.12213.

Ackerer, D., Tagasovska, N., Vatter, T., 2020. Deep smoothing of the implied volatility surface. Advances in Neural Information Processing Systems 33, 11552–11563.

Ai, C., Chen, X., 2003. Efficient estimation of models with conditional moment restrictions containing unknown functions. Econometrica 71, 1795–1843.

Aït-Sahalia, Y., Lo, A.W., 1998. Nonparametric Estimation of State-Price Densities Implicit in Financial Asset Prices. The Journal of Finance 53, 499–547. doi:10.1111/0022-1082.215228.

Anderson, C., 2008. The end of theory: The data deluge makes the scientific method obsolete. Wired magazine 16, 16–07.

Antoine, B., Renault, E., 2009. Efficient gmm with nearly-weak instruments. The Econometrics Journal 12, S135–S171.

Bansal, R., Yaron, A., 2004. Risks for the long run: A potential resolution of asset pricing puzzles. The journal of Finance 59, 1481–1509.

Belloni, A., Chernozhukov, V., Hansen, C., 2014. High-dimensional methods and inference on structural and treatment effects. Journal of Economic Perspectives 28, 29–50.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W., 2010. A theory of learning from different domains. Machine learning 79, 151–175.

Bennett, A., Kallus, N., Schnabel, T., 2019. Deep generalized method of moments for instrumental variable analysis. Advances in neural information processing systems 32.

Bergeron, M., Fung, N., Hull, J., Poulos, Z., 2021. Variational autoencoders: A hands-off approach to volatility. arXiv preprint arXiv:2102.03945 .

Bertsekas, D.P., 2014. Constrained optimization and Lagrange multiplier methods. Academic press.

Bickel, P.J., Klaassen, C.A., Bickel, P.J., Ritov, Y., Klaassen, J., Wellner, J.A., Ritov, Y., 1993. Efficient and adaptive estimation for semiparametric models. volume 4. Springer.

Black, F., Scholes, M., 1973. The pricing of options and corporate liabilities. Journal of political economy 81, 637–654.

Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D., 2015. Weight uncertainty in neural networks, in: International Conference on Machine Learning, PMLR. pp. 1613–1622.

Carr, P., Madan, D., 1999. Option valuation using the fast Fourier transform. The Journal of Computational Finance 2, 61–73. doi:10.21314/JCF.1999.043.

Carrasco, M., Florens, J.P., Renault, E., 2007. Linear inverse problems in structural econometrics estimation based on spectral decomposition and regularization. Handbook of econometrics 6, 5633–5751.

Chataigner, M., Crépey, S., Dixon, M., 2020. Deep local volatility. Risks 8, 82.

Chen, H., Cheng, Y., Liu, Y., Tang, K., 2023. Teaching economics to the machines. Available at SSRN 4642167.

Chen, H., Didisheim, A., Scheidegger, S., 2021. Deep structural estimation: With an application

to option pricing. arXiv preprint arXiv:2102.09209 .

Chen, L., Pelger, M., Zhu, J., 2024. Deep learning in asset pricing. Management Science 70, 714–750.

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

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J., 2018. Double/debiased machine learning for treatment and structural parameters. The Econometrics Journal 21, C1–C68. URL: http://dx.doi.org/10.1111/ectj.12097, doi:10.1111/ectj.12097.

Cochrane, J.H., 2011. Presidential address: Discount rates. The Journal of Finance 66, 1047–1108.

Cohen, S.N., Reisinger, C., Wang, S., 2020. Detecting and repairing arbitrage in traded option prices. Applied Mathematical Finance 27, 345–373.

Cong, L.W., Tang, K., Wang, J., Zhang, Y., 2019. Alphaportfolio: Direct construction through deep reinforcement learning and interpretable ai. Available at SSRN 3554486 .

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems 2, 303–314.

Diebold, F.X., Mariano, R.S., 2002. Comparing predictive accuracy. Journal of Business & economic statistics 20, 134–144.

Duffie, D., Pan, J., Singleton, K., 2000. Transform analysis and asset pricing for affine jump-diffusions. Econometrica 68, 1343–1376.

Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., Garcia, R., 2000. Incorporating second-order functional knowledge for better option pricing. Advances in neural information processing systems 13.

Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., Garcia, R., 2009. Incorporating functional knowledge in neural networks. Journal of Machine Learning Research 10.

Dumas, B., Fleming, J., Whaley, R.E., 1998. Implied volatility functions: Empirical tests. The Journal of Finance 53, 2059–2106.

Fang, F., Oosterlee, C.W., 2009. A Novel Pricing Method for European Options Based on Fourier-Cosine Series Expansions. SIAM Journal on Scientific Computing 31, 826–848. doi:10.1137/080718061.

Farrell, M.H., Liang, T., Misra, S., 2021. Deep neural networks for estimation and inference. Econometrica 89, 181–213.

Feng, G., He, J., Polson, N.G., Xu, J., 2024. Deep learning in characteristics-sorted factor models. Journal of Financial and Quantitative Analysis 59, 3001–3036.

Freire, G., Vladimirov, E., 2023. Autoencoder option pricing models. Available at SSRN .

Gao, Y., Gu, Y., Ng, M., 2023. Gradient descent finds the global optima of two-layer physics-informed neural networks, in: International Conference on Machine Learning, PMLR. pp. 10676–10707.

Garcia, R., Gençay, R., 2000. Pricing and hedging derivative securities with neural networks and

a homogeneity hint. Journal of Econometrics 94, 93–115.

Gatheral, J., 2011. The volatility surface: a practitioner's guide. John Wiley & Sons.

Grohs, P., Hornung, F., Jentzen, A., Von Wurstemberger, P., 2023. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations. volume 284. American Mathematical Society.

Gu, S., Kelly, B., Xiu, D., 2020. Empirical asset pricing via machine learning. The Review of Financial Studies 33, 2223–2273.

Gu, S., Kelly, B.T., Xiu, D., 2019. Autoencoder asset pricing models .

Hagan, P.S., Kumar, D., Lesniewski, A.S., Woodward, D.E., 2002. Managing smile risk. The Best of Wilmott 1, 249–296.

Hanin, B., 2019. Universal function approximation by deep neural nets with bounded width and relu activations. Mathematics 7, 992.

Hansen, L.P., 1982. Large sample properties of generalized method of moments estimators. Econometrica: Journal of the econometric society , 1029–1054.

Harvey, C.R., Liu, Y., Zhu, H., 2016. ⋯and the cross-section of expected returns. The Review of Financial Studies 29, 5–68.

Heston, S.L., 1993. A closed-form solution for options with stochastic volatility with applications to bond and currency options. The review of financial studies 6, 327–343.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural networks 2, 359–366.

Huber, P.J., et al., 1967. The behavior of maximum likelihood estimates under nonstandard conditions, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Berkeley, CA: University of California Press. pp. 221–233.

Hutchinson, J.M., Lo, A.W., Poggio, T., 1994. A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks. The Journal of Finance 49, 851–889. doi:10.2307/2329209, arXiv:2329209.

Kaeck, A., Alexander, C., 2012. Volatility dynamics for the s&p 500: Further evidence from non-affine, multi-factor jump diffusions. Journal of Banking & Finance 36, 3110–3121.

Kou, S.G., 2002. A jump-diffusion model for option pricing. Management science 48, 1086–1101.

MacKay, D.J., 1992. Bayesian interpolation. Neural computation 4, 415–447.

Merton, R.C., 1973. Theory of rational option pricing. The Bell Journal of Economics and Management Science 4, 141. URL: http://dx.doi.org/10.2307/3003143, doi:10.2307/3003143.

Neal, R.M., 1996. Bayesian Learning for Neural Networks. volume 118 of *Lecture Notes in Statistics*. Springer, New York.

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

Osajima, Y., 2007. The asymptotic expansion formula of implied volatility for dynamic sabr model and fx hybrid model. Available at SSRN 965265 .

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on Knowledge and

Data Engineering 22, 1345–1359.

Pratt, L.Y., 1992. Discriminability-based transfer between neural networks. Advances in neural information processing systems 5.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2017a. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561 .

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2017b. Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations. doi:10.48550/arXiv.1711.10566, arXiv:1711.10566.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics 378, 686–707.

Ruf, J., Wang, W., 2019. Neural networks for option pricing and hedging: a literature review. arXiv preprint arXiv:1911.05620 .

Song, C., Park, Y., Kang, M., 2024. How does pde order affect the convergence of pinns? Advances in Neural Information Processing Systems 37, 73–131.

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

Vapnik, V.N., Chervonenkis, A.Y., 1971. On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability & Its Applications 16, 264–280.

Wang, S., Teng, Y., Perdikaris, P., 2020. Understanding and mitigating gradient pathologies in physics-informed neural networks. doi:10.48550/arXiv.2001.04536, arXiv:2001.04536.

Wang, S., Teng, Y., Perdikaris, P., 2021. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM Journal on Scientific Computing 43, A3055–A3081.

Wang, S., Yu, X., Perdikaris, P., 2022. When and why PINNs fail to train: A neural tangent kernel perspective. Journal of Computational Physics 449, 110768. doi:10.1016/j.jcp.2021.110768.

Welling, M., Teh, Y.W., 2011. Bayesian learning via stochastic gradient langevin dynamics, in: Proceedings of the 28th international conference on machine learning (ICML-11), pp. 681–688.

White, H., 1980. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. Econometrica: journal of the Econometric Society , 817–838.

White, H., 1982. Maximum likelihood estimation of misspecified models. Econometrica: Journal of the econometric society , 1–25.

Wilcoxon, F., 1945. Individual comparisons by ranking methods. Biometrics bulletin 1, 80–83.

Zhang, W., Li, L., Zhang, G., 2023. A two-step framework for arbitrage-free prediction of the implied volatility surface. Quantitative Finance 23, 21–34.

Zheng, Y., Yang, Y., Chen, B., 2021. Incorporating prior financial domain knowledge into neural networks for implied volatility surface prediction. doi:10.48550/arXiv.1904.12834.