

A Wasserstein Distance

For a Wasserstein distributionally robust optimization problem (1.1), the ambiguity set $B_\delta(\mu)$ is defined as a Wasserstein ball centered at $\mu \in \mathcal{P}(\mathbb{R}^d)$, i.e.,

$$B_\delta(\mu) = \{\eta \in \mathcal{P}(\mathbb{R}^d) : W_p(\mu, \eta) \leq \delta\}. \quad (\text{A.1})$$

For any $p \in (1, \infty)$, the Wasserstein distance $W_p(\mu, \eta)$ between two distributions μ and η is defined as

$$W_p(\mu, \eta) = \left(\inf_{\gamma \in \Pi(\mu, \eta)} \int_{\mathbb{R}^d \times \mathbb{R}^d} d(x, y)^p d\gamma(x, y) \right)^{1/p}, \quad (\text{A.2})$$

where $\Pi(\mu, \eta)$ denotes the set of all joint distributions on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals μ and η and d denotes a metric on \mathbb{R}^d .

For $p = \infty$, the Wasserstein distance becomes the minimal maximal displacement between two distributions

$$W_\infty(\mu, \eta) = \inf_{\gamma \in \Pi(\mu, \eta)} \{\gamma\text{-ess sup } d(x, y)\}. \quad (\text{A.3})$$

Here, $\gamma\text{-ess sup}$ denotes the essential supremum with respect to the measure γ over $\mathbb{R}^d \times \mathbb{R}^d$.

B Deep Hedging Examples

Case 1: Black–Scholes Model with Exponential Loss. In this scenario, we assume that the asset price follows the classical Black–Scholes model

$$dS_t = mS_t dt + \sigma S_t dW_t, \quad (\text{B.1})$$

where m is the drift, σ is the volatility, and W_t is a standard Brownian motion. The process is discretized in time for training the neural network.

Here, as the process is Markovian, we define the information process directly as the price process S_t , which provides all the necessary information to make decisions at time t . The network strategy in (2.1) is simplified to

$$\delta_t = f_{\theta_t}(S_t). \quad (\text{B.2})$$

We consider hedging a European call option with terminal payoff

$$P(S_T) = \max(S_T - K, 0), \quad (\text{B.3})$$

where K is the strike price. To account for risk aversion in the objective function, we adopt the entropic risk measure

$$\rho(Z) = \frac{1}{\lambda} \log \mathbb{E} [e^{-\lambda Z}], \quad (\text{B.4})$$

where $\lambda > 0$ is the risk aversion parameter. This risk measure is commonly used in finance to model the risk preferences of investors [41].

By [2, Example 3.8], the the entropic risk measure admits the OCE form

$$\rho(Z) = \inf_{\omega \in \mathbb{R}} \left\{ \omega + \mathbb{E} \left[\exp(-\lambda(Z + \omega)) - \frac{1 + \log \lambda}{\lambda} \right] \right\}. \quad (\text{B.5})$$

Moreover, the corresponding optimal ω in (B.5) is given by

$$\omega^* = \frac{1}{\lambda} \log \mathbb{E} [\lambda \cdot \exp(-\lambda Z)]. \quad (\text{B.6})$$

The corresponding deep hedging loss is then defined as

$$l_{DH}(\theta, \omega, \mathbf{S}) = \omega - \frac{1 + \log \lambda}{\lambda} + \exp(-\lambda(\text{PnL}(\theta, \mathbf{S}) + \omega)). \quad (\text{B.7})$$

782 **Case 2: Heston Model with Conditional Value-at-Risk (CVaR).** In this scenario, we assume that
 783 the asset price follows the Heston stochastic volatility model:

$$dS_t^1 = mS_t^1 dt + \sqrt{v_t}S_t^1 dW_t^1, \quad dv_t = a(b - v_t) dt + \sigma\sqrt{v_t} dW_t^2, \quad (\text{B.8})$$

784 where v_t is the stochastic variance process, and the Brownian motions W_t^1 and W_t^2 satisfy

$$\mathbb{E}[dW_t^1 dW_t^2] = \rho dt. \quad (\text{B.9})$$

785 The parameters a , b , and σ control the mean reversion speed, long-run variance level, and volatility
 786 of volatility, respectively.

787 As v_t itself is not directly tradable, to hedge the volatility risk, we introduce a second price process
 788 representing a variance swap corresponding to the tradable asset.

789 The variance swap at time t is given by

$$S_t^2 = \int_0^t v_s ds + \frac{v_t - b}{a}(1 - e^{-a(T-t)}) + b(T - t), \quad (\text{B.10})$$

790 where the integral is approximated by the trapezium rule in practice.

791 We then hedge through trading both the underlying asset and the variance swap, i.e., we define the
 792 combined price process as $S_t = (S_t^1, S_t^2)$. Moreover, since the network needs both the price of
 793 the underlying and variance to make decisions, the information process is defined as $I_t = (S_t^1, v_t)$.
 794 The Heston model is Markovian with respect to this information process, so the network strategy
 795 becomes:

$$\delta_t = f_{\theta_t}(S_t^1, v_t). \quad (\text{B.11})$$

796 We again consider hedging a European call option with the same terminal payoff as in (B.3).

797 To evaluate hedging performance under downside risk, we adopt the Conditional Value-at-Risk
 798 (CVaR) risk measure at confidence level $\alpha \in [0, 1)$

$$\text{CVaR}_\alpha(Z) = \frac{1}{1 - \alpha} \int_0^{1-\alpha} \text{VaR}_\gamma(Z) d\gamma \quad (\text{B.12})$$

$$\text{VaR}_\gamma(Z) = \inf \{z \in \mathbb{R} : \mathbb{P}(Z \leq -z) < \gamma\}. \quad (\text{B.13})$$

799 This risk measure captures the expected loss in the worst $1 - \alpha$ fraction of outcomes and is widely
 800 used in risk management [44]. The CVaR can be written in OCE form

$$\text{CVaR}_\alpha(Z) = \inf_{\omega \in \mathbb{R}} \left\{ \omega + \frac{1}{1 - \alpha} \mathbb{E}[\max(-Z - \omega, 0)] \right\}, \quad (\text{B.14})$$

801 where the optimal ω is attained at the α -quantile of Z .

802 We then define the corresponding deep hedging loss as

$$l_{DH}(\theta, \omega, \mathbf{S}^1, \mathbf{v}) = \omega + \frac{1}{1 - \alpha} \max(-\text{PnL}(\theta, \mathbf{S}^1, \mathbf{v}) - \omega, 0). \quad (\text{B.15})$$

803 C Proofs

804 *Proof of Theorem 3.3.* Our proof follows the approach of [23, Theorem 4.1], adapted to the classifi-
 805 cation setting. Specifically, we build on [43, Theorem 2.1] and its proof, which we restate below as a
 806 theorem.

807 **Theorem C.1** (Adapted from [43, 23]). *Under Assumption 3.1 and Assumption 3.2, the following*
 808 *statements hold.*

809 (i) *The first-order sensitivity expansion as $\delta \downarrow 0$ ensures*

$$V(\delta) = V(0) + \delta \Upsilon + o(\delta), \quad \text{where} \quad \Upsilon := \mathbb{E}_{x \sim \mu} [\|\nabla_x l(\theta; x)\|_*^q]^{1/q} \quad (\text{C.1})$$

810 *and q is the conjugate exponent of p , satisfying $1/q + 1/p = 1$.*

811 (ii) Furthermore, $V(\delta)$ can be approximated by

$$V(\delta) = \mathbb{E}_{\eta_\delta}[l(\theta, x)] + o(\delta) \quad \text{as } \delta \downarrow 0 \quad (\text{C.2})$$

812 where the perturbed distribution η_δ is explicitly given by

$$\eta_\delta = [x \mapsto x + \delta \cdot h(\nabla_x l(\theta; x)) \|\nabla_x l(\theta; x)\|_*^{q-1} \Upsilon^{1-q}]_{\#} \mu. \quad (\text{C.3})$$

813 In the data-driven framework we choose $\mu = \frac{1}{N} \sum_{n=1}^N \delta_{X_n}$. Then Υ in (C.1) becomes the average

$$\Upsilon = \left(\frac{1}{N} \sum_{n=1}^N \|\nabla_x l(\theta; X_n)\|_*^q \right)^{1/q} \quad (\text{C.4})$$

814 and the perturbation η_δ in (C.3) becomes a uniform distribution on the perturbed dataset
815 $\{\hat{X}_1, \dots, \hat{X}_N\}$, where each \hat{X}_n satisfies

$$\hat{X}_n = X_n + \delta \cdot h(\nabla_x l(\theta; X_n)) \|\nabla_x l(\theta; X_n)\|_*^{q-1} \Upsilon^{1-q}. \quad (\text{C.5})$$

816 □

817 *Proof of Lemma 3.4.* By Theorem 3.3, η_δ is of the form $\frac{1}{N} \sum_{n=1}^N \delta_{\hat{X}_n}$ and satisfy

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N d(X_i, \hat{X}_i)^p &= \frac{1}{N} \sum_{i=1}^N \|\delta \cdot h(\nabla_x l(\theta; X_n)) \|\nabla_x l(\theta; X_n)\|_*^{q-1} \Upsilon^{1-q}\|^p \\ &= \frac{\delta^p}{\Upsilon^{(q-1)p}} \cdot \frac{1}{N} \sum_{i=1}^N \|h(\nabla_x l(\theta; X_n)) \|\nabla_x l(\theta; X_n)\|_*^{(q-1)p}\|^p \\ &= \frac{\delta^p}{\Upsilon^q} \cdot \frac{1}{N} \sum_{i=1}^N \|\nabla_x l(\theta; X_n)\|_*^q \\ &= \delta^p \end{aligned} \quad (\text{C.6})$$

818 where we use $\|h(x)\| = \sup_{\|x\|_* \leq 1} \langle h(x), x \rangle = \sup_{\|x\|_* \leq 1} \|x\|_* = 1$, $(q-1)p = (1-1/q)pq = q$

819 for exponent conjugate and (C.4). Therefore, we have $\eta_\delta \in \hat{B}_\delta(\mu)$.

820 Moreover, for $\mu = \frac{1}{N} \sum_{n=1}^N \delta_{X_n}$ and $\hat{\mu} = \frac{1}{N} \sum_{n=1}^N \delta_{\hat{X}_n}$, we have

$$\pi = \frac{1}{N} \sum_{n=1}^N \delta_{(X_n, \hat{X}_n)} \in \Pi(\mu, \hat{\mu}). \quad (\text{C.7})$$

821 Therefore, by the definition of the Wasserstein distance,

$$W_p(\mu, \hat{\mu}) \leq \left(\int d(x, y)^p d\pi(x, y) \right)^{1/p} = \left(\frac{1}{N} \sum_{i=1}^N d(X_i, \hat{X}_i)^p \right)^{1/p}. \quad (\text{C.8})$$

822 If $(\frac{1}{N} \sum_{i=1}^N d(X_i, \hat{X}_i)^p)^{1/p} < \delta$, so is $W_p(\mu, \hat{\mu})$, hence $\hat{B}_\delta(\mu) \subset B_\delta(\mu)$.

823 Overall, we proved that $\eta_\delta \in \hat{B}_\delta(\mu) \subset B_\delta(\mu)$. Therefore, $\mathbb{E}_{\eta_\delta}[l(\theta; x)] \leq V_\theta^e(\delta) \leq V_\theta(\delta)$ and

$$0 \leq \frac{1}{\delta} (V_\theta(\delta) - V_\theta^e(\delta)) \leq \frac{1}{\delta} (V_\theta(\delta) - \mathbb{E}_{\eta_\delta}[l(\theta; x)]). \quad (\text{C.9})$$

824 By Theorem 3.3, RHS $\rightarrow 0$ as $\delta \rightarrow 0$, so $\frac{1}{\delta} (V_\theta(\delta) - V_\theta^e(\delta))$ also converges to 0. In other words,

$$V_\theta(\delta) = V_\theta^e(\delta) + o(\delta) \quad \text{as } \delta \downarrow 0. \quad (\text{C.10})$$

825 □

826 *Proof of Lemma 4.1.* For $g_n^S = \nabla_{\mathbf{S}} l(\theta; \mathbf{S}_n)$, the updates (4.3) can be separated into updates on
827 budget_n and direction_n

$$\Upsilon = \left(\frac{1}{N} \sum_{n=1}^N \|g_n^S\|_*^q \right)^{1/q}, \quad \text{budget}_n = \|g_n^S\|_*^{q-1} \Upsilon^{1-q}, \quad \text{direction}_n = \text{sign}(g_n^S). \quad (\text{C.11})$$

828 By the chain rule, we have $g_n^b = \langle g_n^S, \text{direction}_n \rangle = \|g_n^S\|_*$ and $g_n^d = g_n^S / \text{budget}_n$. The update above
829 becomes

$$\Upsilon = \left(\frac{1}{N} \sum_{n=1}^N (g_n^b)^q \right)^{1/q}, \quad \text{budget}_n = (g_n^b)^{q-1} \Upsilon^{1-q}, \quad \text{direction}_n = \text{sign}(g_n^d), \quad (\text{C.12})$$

830 which proves the lemma. □

831 Corollary 4.2 is a special case of the following more general result, which we will prove next.

832 **Corollary C.2.** Consider the setting of Theorem 3.3 with inputs of the form $x = (x^1, \dots, x^d) \in$
 833 $\mathbb{R}^{d \times (T+1)}$ representing d -dimensional sequences of length $T + 1$. Set the norm as

$$\|x\| := \left(\sum_{i=1}^d (\lambda_i \|x^i\|_\infty)^p \right)^{1/p}, \quad (\text{C.13})$$

834 where $\|\cdot\|_\infty$ is the infinity norm defined in the space of the trajectories \mathbb{R}^{T+1} . Over the input
 835 samples $\{X_1, \dots, X_N\}$, where each sample contains d trajectories $X_n = (X_n^1, \dots, X_n^d)$, we define
 836 $g_n = (g_n^1, \dots, g_n^d) = \nabla_x l(\theta; (X_n^1, \dots, X_n^d))$ to be the gradient with respect to the input.
 837 In this setting, the perturbation (C.5) can be written as

$$\hat{X}_n^i = X_n^i + \frac{1}{\lambda_i} \text{sign}(g_n^i) \left\| \frac{1}{\lambda_i} g_n^i \right\|_1^{q-1} \Upsilon^{1-q} \quad (\text{C.14})$$

838 for $i = 1, \dots, d$ and $n = 1, \dots, N$. Moreover, Υ becomes

$$\Upsilon = \left(\sum_{n=1}^N \sum_{i=1}^d \left\| \frac{1}{\lambda_i} g_n^i \right\|_1^q \right)^{1/q} \quad (\text{C.15})$$

839 where $\|\cdot\|_1$ is the l_1 -norm defined on \mathbb{R}^{T+1} .

840 *Proof.* We first show that the norm in (C.13) has dual norm defined as

$$\|y\|_* = \left(\sum_{i=1}^d \left(\frac{1}{\lambda_i} \|y^i\|_1 \right)^q \right)^{1/q}. \quad (\text{C.16})$$

841 With the standard pairing $\langle x, y \rangle = \sum_i \langle x^i, y^i \rangle$, we estimate

$$|\langle x, y \rangle| \leq \sum_{i=1}^d \|x^i\|_\infty \|y^i\|_1 \leq \left(\sum_{i=1}^d (\lambda_i \|x^i\|_\infty)^p \right)^{1/p} \left(\sum_{i=1}^d \left(\frac{1}{\lambda_i} \|y^i\|_1 \right)^q \right)^{1/q} = \|x\| \|y\|_*, \quad (\text{C.17})$$

842 where the inequalities hold by Hölder's inequality.

843 By setting $\lambda_i \|x^i\|_\infty \propto \frac{1}{\lambda_i} \|y^i\|_1$ and $x^i = \|x^i\|_\infty \text{sign}(y^i)$ for each i , for any y there exists x such
 844 that $\|x\| = 1$ and $|\langle x, y \rangle| = \|x\| \|y\|_*$. Hence (C.16) indeed defines the dual norm by definition. The
 845 corresponding function $h(y)$ such that $\langle h(y), y \rangle = \|y\|_*$ is defined as

$$h(y) = \frac{1}{\|y\|_*^{q-1}} \left(\frac{1}{\lambda_1} \text{sign}(y^1) \left\| \frac{1}{\lambda_1} y^1 \right\|_1^{q-1}, \dots, \frac{1}{\lambda_d} \text{sign}(y^d) \left\| \frac{1}{\lambda_d} y^d \right\|_1^{q-1} \right). \quad (\text{C.18})$$

846 Recall that the perturbation in (C.5) is

$$\hat{X}_n = X_n + \delta \cdot h(g_n) \|g_n\|_*^{q-1} \Upsilon^{1-q}. \quad (\text{C.19})$$

847 By (C.18), $h(g_n)$ becomes,

$$h(g_n) = \frac{1}{\|g_n\|_*^{q-1}} \left(\frac{1}{\lambda_1} \text{sign}(g_n^1) \left\| \frac{1}{\lambda_1} g_n^1 \right\|_1^{q-1}, \dots, \frac{1}{\lambda_d} \text{sign}(g_n^d) \left\| \frac{1}{\lambda_d} g_n^d \right\|_1^{q-1} \right). \quad (\text{C.20})$$

848 Therefore, bringing (C.20) into (C.19), each trajectory in each sample X_n is perturbed to

$$\begin{aligned} \hat{X}_n^i &= X_n^i + \delta \cdot \left(\frac{1}{\|g_n\|_*^{q-1}} \frac{1}{\lambda_i} \text{sign}(g_n^i) \left\| \frac{1}{\lambda_i} g_n^i \right\|_1^{q-1} \right) \cdot \|g_n\|_*^{q-1} \Upsilon^{1-q} \\ &= X_n^i + \frac{\delta}{\lambda_i} \text{sign}(g_n^i) \left\| \frac{1}{\lambda_i} g_n^i \right\|_1^{q-1} \Upsilon^{1-q} \end{aligned} \quad (\text{C.21})$$

849 for $i = 1, \dots, d$ and $n = 1, \dots, N$. Finally,

$$\Upsilon = \left(\sum_{n=1}^N \|g_n\|_*^q \right)^{1/q} = \left(\sum_{n=1}^N \sum_{i=1}^d \left\| \frac{1}{\lambda_i} g_n^i \right\|_1^q \right)^{1/q}. \quad (\text{C.22})$$

850 \square

D Experimental Details

Here we provide additional experimental details regarding the adversarial training introduced in Section 5.2. Readers can refer to the code provided in the supplementary material for a comprehensive implementation.

Network Architecture. The neural network architecture remains consistent with the standard deep hedging framework [2], characterized by decision-making at each time step t through:

$$\delta_t = f_t^{\theta_t}(I_t), \quad (\text{D.1})$$

where I_t encapsulates all relevant information available at step t . In line with [2], each $f_t^{\theta_t}$ comprises two hidden layers, each with 20 neurons, batch normalization, and ReLU activation.

Training Procedure. Our training procedure begins with a preliminary phase of clean training to establish stable initial parameters. Specifically, this phase lasts 100 epochs for the BS model and 300 epochs for the more complex Heston model. Subsequently, the network undergoes adversarial training for an additional 200 epochs (BS) or 400 epochs (Heston), alternating adversarial example generation and optimization of Eq. (5.1). For comparison, we train baseline networks (clean strategies) exclusively with clean training for an equivalent total duration (300 epochs for BS, 700 epochs for Heston).

Optimizer and Learning rate. Optimization utilizes the Adam optimizer, with decaying learning rate—initially set to 0.005 for BS and 0.05 for Heston. The batch size is set to 10,000 unless the dataset size N is smaller, in which case the entire dataset is utilized per batch.

Hyperparameters. Critical adversarial training hyperparameters include α , tested at 0, 1, 10 to gauge the relative influence of clean versus adversarial loss, and perturbation magnitude δ , explored across 0.001, 0.003, 0.005, 0.01, 0.03, 0.05, 0.1, 0.3, 0.5. Hyperparameter selection is performed by evaluating performance on a validation set of size N and selecting the hyperparameters yielding the best validation results.

Adversarial attack. During the experiment, we employ the WBPGD algorithm detailed in Algorithm 2 for adversarial attacks. We execute this algorithm for 20 iterations, setting the step-size as $\beta = \frac{4}{20}\delta$, which is dependent on the perturbation magnitude δ . Additionally, for the two models considered, input trajectories have identical initial values across all samples. Consequently, we avoid perturbing the initial values by explicitly setting both the perturbation and corresponding gradient components to zero.

Computation time. All computational runs are conducted without GPU on AMD EPYC 7742 or Intel Icelake Xeon Platinum 8358 processors equipped with less than 64GB of memory. For standard adversarial training involving 100,000 sample paths, the computation time is approximately 3 hours for the Black-Scholes model and around 10 hours for the Heston model. In contrast, classical deep hedging without adversarial training requires roughly one-tenth of this computational effort. The increased computational demand for adversarial training is reasonable, as each network update includes an additional 20 iterations of adversarial perturbations, enhancing the network’s robustness.

Cash-invariant property of convex risk measure. By the cash-invariance property [41] of the convex risk measure ρ , we have $\rho(Z + c) = \rho(Z) - c$ for any random variable Z and constant c (representing cash injection). Therefore, in practice, we directly set p_0 in (2.2) to 0 as it does not affect the optimization problem. Note that we are only interested in hedging here; for pricing the an appropriate choice of p_0 could be determined as described in [2].

E Additional Experimental Results

In this section, we provide supplementary experimental results to further validate and contextualize the analyses presented in Section 5.

895 E.1 Autocorrelation function comparison

896 Building upon the analysis presented in Section 5.1, we further examine the impact of adversarial
897 perturbations by comparing the autocorrelation functions (ACFs) of the adversarially perturbed
898 trajectories against those of the original trajectories. For a path $\{x_t\}$, the ACF is defined as:

$$\text{ACF}(x, \text{lag}) = \frac{1}{\sigma^2} \sum_{i=0}^{\text{lag}} \frac{1}{N-i} \sum_{t=1}^{N-i} (x_t - \bar{x})(x_{t+i} - \bar{x}), \quad (\text{E.1})$$

899 where \bar{x} is the empirical mean of the path x , and σ^2 is its empirical variance.

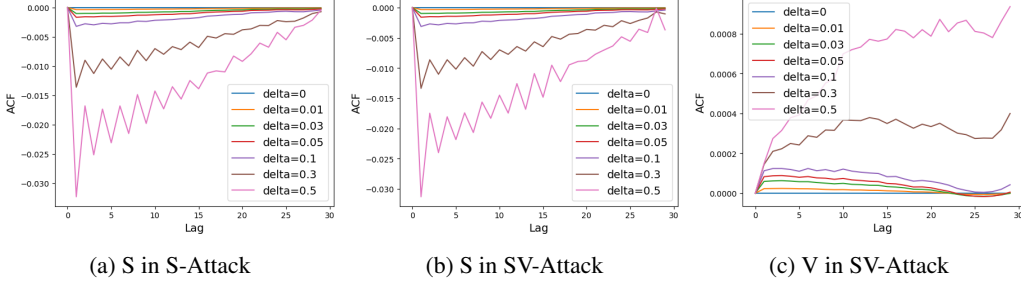


Figure 2: Difference in Autocorrelation function (ACF) of perturbed paths for different δ values and original paths.

900 Fig. 2 illustrates the difference in ACF of perturbed paths and original paths. It reveals minimal
901 deviations (< 0.005) in autocorrelation for perturbations with magnitude $\delta < 0.1$. Despite these
902 modest ACF differences, the corresponding hedging errors are notably large (see Table 1), thereby
903 undermining further that an adversarially perturbed sample path distribution may lead to significant
904 hedging losses, even though comparisons of ACF and covariance matrices suggest that the perturbed
905 distribution is close to the original distribution.

906 E.2 Results on adversarial training

907 **Optimal hyperparameter choice.** As detailed in Section 3 and Appendix D, our robust training
908 framework introduces two critical hyperparameters: the attack radius δ controlling perturbation
909 magnitude, and the balance weight α modulating between nominal and adversarial losses. Through
910 grid search across $(\delta, \alpha) \in [0.001, 0.5] \times \{0, 1, 10\}$, we identify optimal configurations that maximize
911 validation performance for each training set size N . The optimal hyperparameters for the Heston
912 model are presented in Table 4a where we can observe that the optimal δ decreases from 0.5 at
913 $N = 5,000$ to 0.005 at $N = 100,000$. This phenomenon arises because smaller training sizes induce
914 greater divergence between the empirical distribution μ_N and the true data-generating distribution μ ,
915 thus larger adversarial perturbations (δ) are required to bridge this distributional gap. The optimal
916 parameters for the Black-Scholes model show a similar pattern, see Table 4b.

917 **Detailed Heston results.** Table 5 provides detailed out-of-sample performance results, supplement-
918 ing the information shown in Figure 1a. Specifically, the table shows improvements in robust strategy
919 performance as the sample size becomes large and demonstrates that the SV-Attack strategy exhibits
920 lower variance compared to the S-Attack strategy.

921 **Black-Scholes results.** Figure 3 and Table 6 show comprehensive results for the Black-Scholes
922 model, revealing patterns analogous to the Heston model. However, the gap between robust and
923 clean strategies is relatively smaller than in Heston. This aligns with expectations - the simpler
924 Black-Scholes model offers fewer exploitable gaps for adversarial training to mitigate, particularly in
925 volatility dynamics.

926 F Extension to other models

927 In Section 3, we introduced distributional adversarial attack algorithms specifically for the Black-
928 Scholes model, with input $\mathbf{I} = \mathbf{S}$, and the Heston model, with input $\mathbf{I} = (\mathbf{S}^1, \mathbf{v})$. In this section, we
929 generalize these approaches to an arbitrary model characterized by input trajectories $\mathbf{I} = (\mathbf{I}^1, \dots, \mathbf{I}^d)$.

Table 4: Hyperparameter Choices for Heston and Black-Scholes Models

(a) Heston Model Hyperparameters					(b) Black-Scholes Model Hyperparameters		
Training Samples (N)	S-Attack		SV-Attack		Training Samples (N)	δ	α
	δ	α	δ	α			
5,000	0.3	1	0.5	1	5,000	0.01	10
10,000	0.1	10	1.0	10	10,000	0.005	10
20,000	0.05	1	0.1	1	20,000	0.003	10
50,000	0.03	0	0.03	0	50,000	0.001	1
100,000	0.01	0	0.005	0	100,000	0.001	0

Table 5: Detailed out-of-sample performance across sample sizes (N) for SV-Attack, S-Attack, and Clean strategies on Heston model

Strategy	N	Avg Loss	Min Loss	Max Loss	Variance
SV-Attack	5,000	2.8644	2.7386	3.5975	0.0346
SV-Attack	10,000	2.5460	2.5173	2.6087	0.0008
SV-Attack	20,000	2.1063	2.0928	2.1282	0.0002
SV-Attack	50,000	1.9706	1.9669	1.9742	2.6e-5
SV-Attack	100,000	1.9469	1.9469	1.9469	–
S-Attack	5,000	2.9646	2.7605	4.5912	0.1574
S-Attack	10,000	2.5287	2.4694	2.6629	0.0028
S-Attack	20,000	2.1259	2.1027	2.1489	0.0004
S-Attack	50,000	1.9705	1.9665	1.9745	3.2e-5
S-Attack	100,000	1.9472	1.9472	1.9472	–
Clean	5,000	6.2095	4.7379	10.6187	2.0887
Clean	10,000	3.0000	2.8068	3.1755	0.0129
Clean	20,000	2.1955	2.1329	2.2266	0.0014
Clean	50,000	1.9773	1.9705	1.9841	9.2e-5
Clean	100,000	1.9503	1.9503	1.9503	–

Table 6: Detailed BS model performance metrics across sample sizes (N) for robust and Clean strategies

Strategy	N	Avg Loss	Min Loss	Max Loss
Robust	5,000	2.4109	2.4040	2.4195
Robust	10,000	2.3947	2.3920	2.3976
Robust	20,000	2.3855	2.3852	2.3861
Robust	50,000	2.3798	2.3794	2.3802
Robust	100,000	2.3769	2.3769	2.3769
Clean	5,000	2.4136	2.4055	2.4208
Clean	10,000	2.3976	2.3947	2.3993
Clean	20,000	2.3860	2.3854	2.3866
Clean	50,000	2.3798	2.3794	2.3801
Clean	100,000	2.3772	2.3772	2.3772

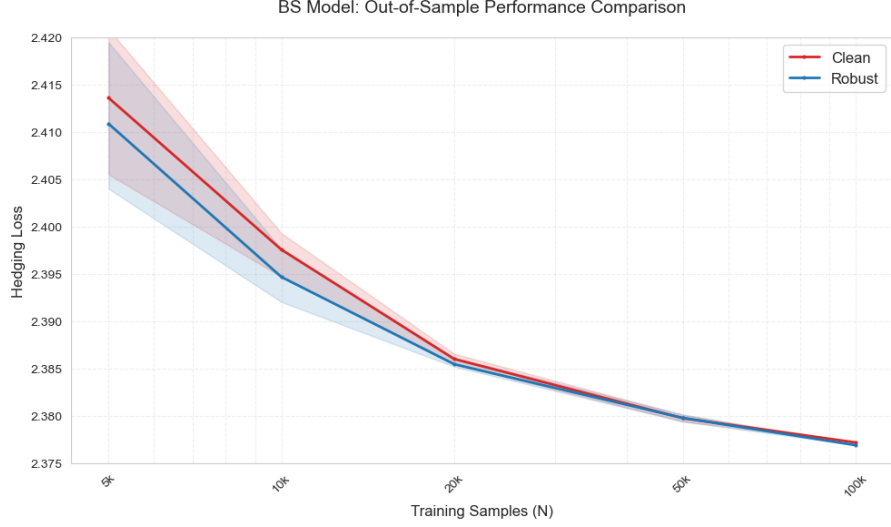


Figure 3: Out-of-sample hedging performance comparison under Black-Scholes dynamics, comparing robust training with clean strategies. Shaded regions indicate min-max ranges across training partitions.

930 We begin by defining a distance measure on the general input space

$$d(\mathbf{I}, \hat{\mathbf{I}}) = \left(\sum_{i=1}^d (\lambda_i \cdot \max_t |I_t^i - \hat{I}_t^i|)^p \right)^{1/p}, \quad (\text{F.1})$$

931 analogous to the distance measure introduced for the Heston model in (4.8). This definition allows
 932 distinct perturbation scales for each trajectory, consistent with the Heston model framework.

933 Utilizing Corollary C.2, a direct generalization of Corollary 4.2, the update rule for each scaled
 934 trajectory is given by:

$$\lambda_i \hat{\mathbf{I}}_n^i = \lambda_i \mathbf{I}_n^i + \delta \cdot \text{sign}\left(\frac{1}{\lambda_i} g_n^i\right) \left\| \frac{1}{\lambda_i} g_n^i \right\|_1^{q-1} \Upsilon^{1-q}, \quad (\text{F.2})$$

935 which closely parallels the update step for the complete sample set

$$\hat{\mathbf{I}}_n = \mathbf{I}_n + \delta \cdot h(g_n) \|g_n\|_*^{q-1} \Upsilon^{1-q}. \quad (\text{F.3})$$

936 Furthermore, the total distance $\sum_{n=1}^N d(\mathbf{I}_n, \hat{\mathbf{I}}_n)^p$ and the term Υ^p used during the iterative update
 937 process can naturally be decomposed into sums across both trajectory dimensions ($i = 1, \dots, d$) and
 938 individual samples ($n = 1, \dots, N$).

939 Consequently, similar to the approach in the Heston model, each trajectory will be independently
 940 perturbed according to an l_∞ -norm metric, effectively transforming the original set of samples into a
 941 structured set of scaled trajectories $\{\lambda_i \mathbf{I}_n^i\}_{n=1, \dots, N}^{i=1, \dots, d}$.