Completion of Option Price Surfaces Using Score-based Generative Models with Guided Diffusion

by

Po Chai Wong

A research paper
presented to the University of Waterloo
in fulfillment of the
research paper requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Waterloo, Ontario, Canada, 2026

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Markets only quote option prices at a limited set of strikes and maturities, so a large part of the price surface is usually missing. Reconstructing these unobserved regions is important not only for consistent pricing but also for hedging and for extracting quantities like implied or local volatility. Traditional approaches rely on a chosen asset-price model and often struggle when data are sparse, noisy, or not well explained by the model. Recent machine-learning methods relax these assumptions, but they still do not automatically enforce the no-arbitrage conditions that any valid price surface must satisfy. This project explores the use of score-based generative models (SBGMs) to complete both European and American option price surfaces when only partial market information is available. The main idea is to guide the sampling procedure by adding two types of corrections: (i) no-arbitrage corrections that nudge the generated surface toward economically valid shapes, and (ii) physics-informed corrections that incorporate PDE-based structure under certain modeling assumptions. These additional components help the generative model converge faster and produce surfaces that are much more consistent with option-pricing theory. We tested physics-informed and no-arbitrage correctors on European put options in a predictor-corrector scheme with discretised reverse diffusion using stochastic differential equations. We also combined the no-arbitrage corrector with another annealed Langevin Dynamics to correct American put price surfaces. In general, we attained 10^{-2} of mean average errors for all cases, and different levels of improvement according to moneyness and maturity periods.

Acknowledgements

I would like to sincerely thank my supervisors, Prof. Justin Wan and Prof. James Thompson, for their invaluable guidance and support throughout this research project. I am deeply grateful for the time, trust, and kindness you have shown me. Working with you has been one of the most meaningful experiences of my time at Waterloo. I would also like to thank Prof. Jun Liu for his insightful feedback on this work.

I am profoundly grateful to my family, especially my sister. Your love and support have carried me through every high and low of this journey. There were many moments when I felt overwhelmed or uncertain, yet your belief in me never wavered. I don't think I can ever fully express how much that means to me.

To my friends — Marco, Ivan, Cindy, Niki, and so many others — thank you for being my safe space throughout this program. You listened to my worries, celebrated my small victories, and stayed by my side even on the days when I struggled the most. I could not have come this far without you.

Dedication

This is dedicated to the one I love.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The financial markets provide option prices for a discrete set of strike prices and maturity dates. In practice, however, the availability of market option prices depends on trading activity. When only a small subset of strikes and maturities is actively traded, the observed market data provide only partial information about the full option price surface. This creates the need to estimate the missing prices at untraded points, a task referred to as the option price completion problem. To fill in the missing values on a price surface is to enable more accurate pricing of options and efficient evaluations of traders' portfolios, and a better understanding of the market's expectations regarding the fluctuations of the underlying asset prices.

Completing the option price surface is essential for several reasons. It enables consistent pricing of options across strikes and maturities, supports the assessment and hedging of portfolios containing derivatives, and ensures the enforcement of no-arbitrage conditions. Moreover, once a complete and arbitrage-free price surface is available, implied volatility and local volatility surfaces can be inferred with greater accuracy, allowing for a clearer interpretation of market expectations about future asset movements.

Classical approaches to filling in or extrapolating missing option prices rely on modeling the stochastic behavior of the underlying asset. The Black-Scholes framework [1], for example, assumes the asset prices evolve according to a geometric Brownian motion. More sophisticated alternatives, such as stochastic volatility models [18], local volatility models [11], and rough volatility models [14] introduce additional randomness or path-dependence to better capture market features. Each of these models is governed by a set of parameters that must be calibrated so that the model-generated option prices align with the observed market data under a chosen distance or loss metric. In addition, these methods are model-

dependent and the calibration of the parameters, such as volatility, highly depends on data availability. The noisy and sparse market data therefore may not be in favor of these methods.

Recent developments in machine learning explore model-free and data-driven approaches. For examples, [22] explored the hybrid approach of using recurrent neural network (RNN) with Monte Carlo method to price American options. [9] attempted to use Physics Informed Neural Network to directly incorporate physical laws into the pricing via Partial Differential Equations (PDEs) for both European and American options. The inclusion of machine learning methods improved the pricing speed and accuracy. These methods however do not guarantee to produce arbitrage-free price surfaces. It would be beneficial to identify methods to generate arbitrage-free price surfaces, with the flexibility to include various model assumptions.

Among machine learning models, generative models are designed to generate samples from an unknown data distribution given only samples from the distribution. They can be grouped into three main categories: likelihood-based models, implicit generative models, and the score-based generative models. However, likelihood-based models, such as [24], [10], [41], depend on some known form of the distribution to compute the likelihood computation during training. Implicit generative models, such as [15], contain an adversarial component in the architecture and therefore can be expensive and difficult to train. Score-based generative models (SBGMs), on the other hand, bypass the restrictions on the distribution form and allow flexible model architectures. The earliest work of SBGMs came from [36]. They utilized denoising score-matching with Langevin Dynamics to generate samples from unknown distributions. The technique was widely used in image generation and natural language processing. Later, more recent approaches with SBGMs have been designed to speed up the sampling process with stochastic differential equations (SDEs) and probability flows [39]. Recently, to generate samples with desired quantitative features, physics informed features can be included in the sampling process to guide the diffusion [23], [35], [32]. Partial differential equations can be employed into diffusion models for model calibration and numerical solutions. The idea of utilizing partial information of the solution surface and the coefficient field aligned with our direction of using partial observations of market information in the problem.

In this research project, we navigate methods to generate arbitrage-free price surfaces given partial market information. Arbitrage-free modeling is crucial because it ensures that the generated option price surfaces are financially consistent. Without enforcing no-arbitrage conditions, a model might produce prices that allow riskless profits, which are not observed in real markets and would make any downstream trading, hedging, or risk management strategies invalid. By incorporating arbitrage-free constraints, we can pro-

duce synthetic surfaces that are both realistic and usable for practical applications, such as hedging, pricing exotic derivatives, or calibrating risk models. Previous projects have investigated volatility surface completion for European Options using score-based generative models (SBGMs) [21]. SBGMs generate new samples according to trained samples from an unknown distribution. The distribution is therefore free of model assumption and provide flexibility in the modeling performance over traditional parametric modeling approaches. However the generative methods are also notorious for its slow sampling convergence speed. In this project, we employ techniques to speed up the process using no-arbitrage correctors. We also find that, with additional modeling assumptions, we can define physics-informed correctors which helped with the sampling speed as well. Physics-informed and no-arbitrage corrector were enforced in combination of discretized reverse diffusion to complete European put price surfaces. In particular, American options are notoriously difficult to manage, due to its early exercise property. There does not exist any close-form solutions and thus it relies on numerical methods to solve. The computation of American options is much harder and more expensive than European options. Despite the numerical challenges, it is one of the most commonly traded entity on the market. We attempted SBGMs to complete American put price surfaces in the framework of our designed no-arbitrage corrector with SDE-type predictors, and showed improvement in sampling arbitrage-free surfaces. We identify strike regions where it outperforms the benchmark SBGMs and limitations in the end.

The research project is organized as follows: Chapter 2 provides background concepts of option pricing including European and American options and no-arbitrage conditions. It also introduces SBGMs, the training and sampling setups and their variants, including sampling with SDEs, probability flow and guided diffusion. Chapter 3 lays out the foundations of European Option pricing theories and numerical results. It defines the set-up of no-arbitrage and physics-informed correctors targeting European Options. Chapter 4 formulates the completion problem for American options, with the corresponding correctors.

Chapter 2

Background

In this chapter, we will cover the preliminary background for the project. Concepts of European and American options are introduced. We then define the expected shape and features desired for our final output, by introducing the definitions of arbitrage and conditions for the arbitrage-free price surface to satisfy. Next, we will describe our main tool, the score-based generative model, and how to train such a model and generate samples via Langevin dynamics. Then, we will lay out the predictor-corrector scheme, a method to increase sampling accuracy and to speed up the sampling process, by incorporating stochastic differential equations into the framework.

2.1 Option Pricing

An option is a type of derivative product of an underlying asset. We will consider two types of option in this project: European options and American Options.

2.1.1 European Options

A European call (put) option is a contract that gives the right, not the obligation to buy (sell) 1 unit of an asset, at a predetermined price (strike) $K \geq 0$ at a specified time (maturity) $T > 0$. The seller of a call (put) option can be viewed as an insurance provider from whom the option owner can benefit from the upside (downside) movement of the underlying without bearing the risk from the downside (upside) movement. To acquire the purchasing/ selling right, a price must be paid, which is the option price. The process of

identifying the fair price is called option pricing, making the option fair to everyone in the market.

The pricing takes into account several key elements, including the underlying asset's price S, the strike price K, the maturity T, the interest rate r, the dividend rate q and the volatility σ of the underlying. With estimates of these values, a model can be constructed to analyze and even predict the market behavior. In this project, we will mainly focus on the asset price S and the volatility σ.

A popular choice for modeling European options is the Black-Scholes model [1]. Let S_t be the price of the underlying asset at time t. The Black-Scholes model assumes S_t follows the geometric Brownian motion under the risk neutral measure:

$$dS_t = rS_t dt + \sigma S_t dZ_t, \tag{2.1}$$

where Z_t is the standard Brownian motion. It can then be derived that the put option

$$\begin{aligned} &\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \\ &V(S,T) = \max(K - S, 0), \end{aligned} \tag{2.2}$$

where $V(S,T)$ is the European put option price at time t, S is the price of the underlying at time t. A prominent reason for the popularity of this model is that this provides a closed-form solution for option price:

$$P_{BS}(S_0, K, T, r, \sigma) = -S_0\Phi(-d_1) + Ke^{-rT}\Phi(-d_2), \tag{2.3}$$

where P_{BS} denotes the Black-Scholes European Put option prices, $d_1 = \frac{1}{\sigma\sqrt{T}}\left[\log\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T\right]$, $d_2 = d_1 - \sigma\sqrt{T}$, and Φ is the c.d.f. of the standard normal distribution. Throughout this project, we will only consider put options, but the mechanism works for call options as well. We consider constant risk-free rate r and zero dividends.

To identify a correct model P_{BS} for the market, one needs to estimate the volatility σ, given the observed option price P_{obs} at given n strikes $\{K_i : i = 1, \cdots, n\}$ and m maturities $\{T_j : j = 1, \cdots, m\}$. This process is called *model calibration*. The volatility value such that the model resembles the current market observation on $\{(K_i, T_j)\}_{i,j=1}^{n,m}$ is called **implied volatility** σ_{imp}. In other words, it is the process to find σ_{imp} such that given r, q,

$$P_{BS}(K_i, T_j; \sigma_{imp}, r, q) = P_{obs}(K_i, T_j; \sigma, r, q),$$

for all $i = 1, \cdots, n$ and $j = 1, \cdots, m$. With this estimate of σ_{imp}, one can generate the whole surface with the same option pricer P_{BS}. Although there are no analytic formulas

to recover σ from observed prices, fortunately, the Black-Scholes formula is a monotonic function of the volatility σ. Given its smoothness for $\sigma > 0$, numerical methods, such as Newton's method [12], can be applied to find the root σ for the calibration problem. However, in practice, market data are often sparse, leaving gaps in observed option prices. These gaps prevent direct calibration for all strikes and maturities, motivating methods for completing the price surface before applying volatility calibration. Estimation of the missing price and volatility values is referred to as the completion problem.

Previous work has focused on using SBGMs to complete European implied volatility surface $\sigma_{imp}(K, T)$ [21]. They found that the performance of the model was worst around the boundaries. For this project, we directly compare the performance on the put price surface $P_E(K, T)$.

2.1.2 American Options

In contrast, an American call (put) option allows the option owner to exercise the option anytime on or before the maturity T. This property is called the **early exercise** property. Early exercise can be valuable, for example, for deep in-the-money put options with high interest rates or options approaching maturity. In such cases, exercising early can provide a higher payoff than holding until expiration. Unlike European options, which only take into account the intrinsic value at maturity T, the actual value of an American option at any time before T depends on future potential values. Therefore, the valuation of an American option involves determining not only the expected discounted payoff at maturity but also the optimal exercise strategy over the time interval $[0, T]$. Although both call and put options exist, in the context of American versus European options, American call options on non-dividend-paying stocks are equivalent in value to their European counterparts. Therefore, the early exercise feature is only relevant for puts (or calls on dividend-paying stocks), and we focus on put options in this study.

At each time $t \in [0, T]$, the owner of an American put option either **exercises** the option immediately for an instant payoff $\max(K - S, 0)$, or **continues** holding it for a potentially higher future value. Suppose $C(S, t)$ denotes the **continuation value** of the option at time t given the asset price S, i.e.

$$C(S, t) = \mathbb{E}\big[e^{-r(\tau - t)}(K - S)_+ \mid S_t = S\big],$$

where τ is a *stopping time*, representing the (possibly random) time at which the option is exercised based on information available up to that time. This is the expected discounted value of the option if it is not exercised at time t and the holder follows an optimal exercise

strategy thereafter. This is the expected discounted value of the option if not exercised at time t. Then, the fair price of the American put can be formulated as:

$$P_A(S,t) = \max\left(K - S, C(S,t)\right).$$

If the continuation value exceeds the immediate **exercise value** $\max(K-S,0)$, then it is optimal to continue holding the option. In this region, the option price is computed using the standard Black-Scholes PDE, i.e. Equation (2.2). However, if the immediate exercise value dominates, i.e. $K - S > C(S,t)$, it becomes optimal to exercise the option early. In this *exercise region*, the option value must coincide with the intrinsic value:

$$P_A(S,t) = K - S. \tag{2.4}$$

The difference between the American and European option values is referred to as the **early-exercise premium**. This quantity represents the additional value arising from the flexibility to exercise the option at any time prior to maturity. Formally, the early-exercise premium $\Pi(S,t)$ is defined as

$$\Pi(S,t) = P_A(S,t) - P_E(S,t),$$

where P_E denotes the European put price. Since the holder of an American option can always choose to wait until maturity and receive the European payoff, it follows that $\Pi(S,t) \geq 0$ for all (S,t). The premium therefore quantifies the expected benefit of being able to exercise optimally before maturity. Intuitively, when the underlying asset price drops sufficiently below the strike, the immediate exercise value $K - S$ may exceed the continuation value $C(S,t)$, making early exercise optimal. A key structural reason underpinning this behavior is that the underlying price S is bounded below by zero. When $S \to 0$, the American put's payoff approaches K, while the European put's maximum discounted payoff is only Ke^{-rT}. Thus, deep in-the-money American puts can capture the full strike amount earlier, reinvesting the proceeds at the risk-free rate—an opportunity unavailable to the European counterpart. Conversely, for an American call on a non-dividend-paying stock, early exercise is never optimal. This asymmetry arises precisely because the underlying price has no upper bound: there is no analogous "cap" that would make receiving $S - K$ immediately more valuable than holding the call to maturity. Hence, the early-exercise premium for an American put reflects the value of accessing the intrinsic payoff earlier when the asset price is low, a benefit rooted in the asset's limited downside.

From a mathematical perspective, the early-exercise premium can be interpreted as the expected discounted gain from the possibility of early exercise under the risk-neutral measure. Equivalently, it is the solution to a *free-boundary problem*, where the boundary

separates the continuation and exercise regions. The early-exercise premium thus encapsulates the economic value of flexibility and the analytical complexity introduced by the optimal stopping feature of American options.

Therefore, the early-exercise property makes American put options computationally more challenging than European put options. In particular, there are no analytic formula for the exact price of American options and therefore one can only rely on numerical approximation, such as binomial method [26], [25], finite difference methods [13], [16] and machine learning methods [9].

2.1.3 No-arbitrage Conditions

An arbitrage opportunity refers to the market scenario that one can create a portfolio that for certain will have positive gain. Assuming that there is such an opportunity, it is assumed that such opportunities would be exploited by market participants, eliminating the arbitrage quickly, and therefore the advantage will vanish quickly. Hence, it is reasonable to assume the price surface will contain no arbitrage opportunities. The conditions for such scenarios are called **no-arbitrage conditions**. They are one of the many determining factors of the shape of the price surfaces. In this project, we consider a finite set of strikes. We will use no-arbitrage conditions from [8], [4]:

Theorem 2.1.1 (No-arbitrage Conditions for European Put Options) *Suppose the prices of European Put options with maturity T are given for a finite number of strikes $K_1, \cdots, K_n$. Denote the European put prices at each point (K, T) by $P_E(K, T)$, which is linearly interpolated between the grid points. Then the European put prices are free of arbitrage opportunities if and only if the following are satisfied:*

1. *$P_E(K, T)$ is increasing in K.*
2. *$P_E(K, T)$ is convex in K.*
3. *$e^{-rT}K \geq P_E(K, T) \geq (e^{-rT}K - S_0)_+$, for $K > 0$.*
4. *$\partial_K P(K+, T) < e^{-rT}$, for $K \geq 0$ with $P_E(K, T) > e^{-rT}K - S_0$.*

where S_0 is the current price of the underlying and $\partial_K P_E(K+, T)$ is the derivative in K of P_E from the positive side. For the calendar arbitrage, we impose

5. *$P_E(K, T)$ is non-decreasing in $T > 0$.*

The first three conditions guarantee strike-wise no-arbitrage: put prices must increase and be convex in the strike, lie between their natural lower and upper bounds, and have slopes consistent with risk-neutral probabilities, i.e., between 0 and e^{-rT}. The final condition ensures no calendar (maturity) arbitrage, requiring that option prices increase with time to maturity, since a longer-lived option can never be worth less than a shorter-lived one. Together, these conditions ensure that the price surface could, in principle, arise from a valid risk-neutral distribution of the underlying asset.

For American options, the early-exercise property makes it more complicated. Their no-arbitrage conditions hinge on the European options.

Theorem 2.1.2 (No-arbitrage Conditions for American Put Options) *[7] Suppose the prices of American Put options are given with maturity T are given for finitely many strikes $K_1, \cdots, K_n$. Denote the American put prices at each point (K, T) by $P_A(K, T)$, which is linearly interpolated between these grid points. Assume that the corresponding European Put prices $P_E(K, T)$ satisfy the no-arbitrage conditions in Theorem 2.1.1. Then the American Put price $P_A(K, T)$ is arbitrage-free if and only if:*

1. *$P_A(K, T)$ is increasing in K.*
2. *$P_A(K, T)$ is convex in K.*
3. *$\partial_K P_A(K+, T)K - P_A(K, T) \geq \partial_K P_E(K+, T)K - P_E(K, T)$ for all $K \geq 0$.*
4. *$\max\{(K - S_0)_+, P_E(K, T)\} \leq P_A(K, T) \leq P_E(e^{rT}K, T)$.*

For calendar no-arbitrage, we additionally require:

5. *$P_A(K, T) - P_E(K, T)$ is non-decreasing in $T > 0$.*

The conditions above extend the European no-arbitrage relations by incorporating the early-exercise feature of American options. The first condition ensures convexity and monotonicity with respect to strike, as before. The second condition enforces consistency between the slopes of the American and European price functions, guaranteeing that the early-exercise premium is nonnegative and convex. The third condition bounds the American price between the intrinsic value and the discounted strike, and above its European counterpart. Finally, the last condition prevents calendar arbitrage by requiring that the early-exercise premium increases with maturity. We will explain them in finer details in Chapter 4.

2.2 Score-based Generative Models

Having established the financial context, we now turn to generative modeling techniques that will enable price surface completion. Generative modeling techniques aim to sample from an unknown data distribution given only samples from the distribution. The whole regime of score-based generative models can be boiled down into one single statement. Instead of sampling from the unknown data distribution, noise is added sequentially to the given samples, until they follow a distribution that has a known structure to us. Then one can sample from the output distribution and revert the process to approximate the unknown distribution.

This formulation of the problem gets around several numerical challenges present in other generative methods, such as the estimation of normalizing constant and expensive model component. The model also achieves greater accuracy in prediction over other methods such as conditional GANs [15], U-Nets [34], and even physics-informed neural networks (PINNs) [33] in some cases. In the following, we will describe the training and the inference of SBGMs.

2.2.1 Score Matching

The central advantage that differentiates SBGMs from likelihood-based and implicit adversarial generative methods stems from *score matching* [36]. Suppose our goal is to generate samples $x \in \mathbb{R}^d$ from an unknown distribution D, with a density function $p_0(x)$. Instead of modeling $p_0(x)$ directly, SBGMs aim to model the **score** of the distribution, which is defined as the gradient of the log density function $\nabla_x \log p(x)$ [6].

Modeling on the score avoids the numerical trouble to estimate the normalizing constant, as in the likelihood computation. More precisely, to estimate the probability density $p(x)$, one has to ensure the normalizing property

$$Z := \int_{\mathbb{R}^d} p(x)dx.$$

This poses numerical challenges since the integral approximation requires the estimation over the whole space $\mathbb{R}^d$, which could be expensive in high dimensional cases. Instead, when one estimates the score of the distribution $\nabla_x \log p(x)$, one can skip the computational difficulty in estimating the normalizing constant Z:

$$\nabla_x \log \left(\frac{p(x)}{Z} \right) = \nabla_x \log p(x) - \nabla_x \log Z = \nabla_x \log p(x).$$

Besides the numerical difficulty, the neural network to model the score also refrains from the trouble of including adversarial component in the architecture, and therefore is much easier to train. The process of training a model $s_\theta(x)$ with model parameter θ to learn the score $p(x)$ from the training samples is called **score matching** [6]. One can formulate the training loss as the following:

$$\mathcal{L}(\theta) = \frac{1}{2}\mathbb{E}_{x\sim D}\left[\|s_\theta(x) - \nabla_x \log p(x)\|_2^2\right].$$

In this project, we will use **denoising score matching** to train the model s_θ [42]. There are other formulation of score-matching, which can be found in [38]. The numerical challenge is that the score $\nabla_x \log p(x)$ of the true distribution is still unknown. The idea of denoising score matching is to add predetermined noise z to the true distribution, and then estimate the score of the noise-perturbed distribution, with density $q_\varphi(\cdot|x)$. Formally, suppose we are given some set of data points $x \sim D$ with a probability density $p(x)$. The samples are perturbed via $\widetilde{x} = x + z$ with noise $z \sim \mathcal{N}(0, \varphi^2 I_d)$, where φ refers to the noise level and I_d is the identity matrix in $\mathbb{R}^{d\times d}$. Assume φ is small enough so that the noise perturbed distribution is close to the true distribution, i.e. $q_\varphi(x) \approx p(x)$. Then, the noisy sample $\widetilde{x}$ follows the conditional noise-perturbed distribution

$$q_\varphi(\widetilde{x}) = \int_{x\sim D} q_\varphi(\widetilde{x}|x)p(x)dx. \tag{2.5}$$

Then, the training loss we defined above becomes

$$\mathcal{L}(\theta) = \frac{1}{2}\mathbb{E}_{q_\varphi(\widetilde{x}|x)p(x)}\left[\|s_\theta(\widetilde{x}) - \nabla_{\widetilde{x}} \log q_\varphi(\widetilde{x}|x)\|_2^2\right]. \tag{2.6}$$

In particular, since we define $z \sim \mathcal{N}(0, \varphi^2 I_d)$, and hence $q_\varphi(\widetilde{x}|x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\|\widetilde{x}-x\|^2}{2\varphi^2}\right)$, then it can be shown that

$$\nabla_{\widetilde{x}} \log q_\varphi(\widetilde{x}|x) = -\frac{\widetilde{x} - x}{\varphi^2}, \tag{2.7}$$

which is easily computable, even for high dimensional space. Under this setting, the network, s_{θ^*} with optimal parameter θ^* minimizes Equation (2.6) and consequently $s_{\theta^*}(x) = \nabla_x \log q_\varphi(x) \approx \nabla_x \log p(x)$.

2.2.2 Langevin Dynamics

With the optimal network s_{θ^*}, we can use Langevin dynamics to sample from the density $p(x)$ using the score [36]. More precisely, if we fix a small step size $\alpha > 0$ and the number

of steps M, and suppose we have some initial random noise sample $\widetilde{x}_0 \sim \mathcal{N}(0, I_d)$, the Langevin dynamics gives an iterative algorithm:

$$\widetilde{x}_{k+1} \leftarrow \widetilde{x}_k + \alpha \nabla_x \log p(\widetilde{x}_k) + \sqrt{2\alpha} z_k, \text{ where } z_k \sim \mathcal{N}(0, I_d).$$

With the trained score network $s_{\theta^*}(x) \approx \nabla_x \log p(x)$, all the terms in the algorithm can be computed. In an intuitive sense, it can be thought as providing the most probable direction $\nabla_x \log p(\widetilde{x}_k)$ to the current step $\widetilde{x}_k$, while giving a more generalized exploration from the random noise z_k. It can be shown that under some conditions, the final distribution of $\widetilde{x}_M$ will tend to the true probability density p as $M \longrightarrow \infty$ and $\alpha \longrightarrow 0$.

2.2.3 Noise Conditional Score Network

Note that the current setup of the network s_θ assumes using only one noise level. Two potential problems may arise. The first is slow mixing, especially for multi-modal data distribution. When two modes of data are segregated by a low density region, the algorithm may not correctly recover the relative weights of these modes if φ is too small. The second is inaccurate score estimation. For regions with low data density, score matching may lack sufficient data to estimate score functions accurately [36].

To sidestep these potential difficulties, one can perturb the data using different levels of noise, and train the score network depending on the respective noise level. In other words, define a geometric increasing sequence of noise levels: $\varphi_1 < \varphi_2 < \cdots < \varphi_L$, where L is the number of noise levels considered and the common ratio is $\gamma > 1$. Suppose that φ_L is large enough so that the final noisy samples are diverse enough to cover even regions of low data density; and φ_1 are small enough so that the noise-perturbed distribution is close to the true distribution, i.e. $q_\varphi(x) \approx p(x)$ in Equation (2.5) [37].

Then, we will train the score network $s_\theta(\widetilde{x}, \varphi)$ that is now conditional on the noise level φ, by minimizing the training loss for each φ:

$$\ell(\theta; \varphi) \coloneqq \frac{1}{2}\mathbb{E}_{q_\varphi(\widetilde{x}|x)p(x)} \left[\| s_\theta(\widetilde{x}, \varphi) - \nabla_{\widetilde{x}} \log q_\varphi(\widetilde{x}|x) \|^2 \right]. \tag{2.8}$$

Again, since we use normal distribution for the noise perturbation, by Equation (2.7), the loss becomes

$$\ell(\theta; \varphi) \coloneqq \frac{1}{2}\mathbb{E}_{q_\varphi(\widetilde{x}|x)p(x)} \left[\left\| s_\theta(\widetilde{x}, \varphi) + \frac{\widetilde{x} - x}{\varphi^2} \right\|_2^2 \right]. \tag{2.9}$$

Now, to train on single score network $s_\theta(\cdot,\cdot)$ for the whole sequence of noise levels $\{\varphi_i\}_{i=1}^L$, the training loss is

$$\begin{aligned}\mathcal{L}(\theta;\{\varphi_i\}_{i=1}^L) &\coloneqq \frac{1}{L}\sum_{i=1}^{L}\lambda_i \ell(\theta;\varphi_i)\\ &= \frac{1}{2L}\sum_{i=1}^{L}\lambda_i \mathbb{E}_{q_\varphi(\widetilde{x}|x)p(x)}\left[\left\| s_\theta(\widetilde{x},\varphi_i)+\frac{\widetilde{x}-x}{\varphi_i^2}\right\|_2^2\right].\end{aligned} \tag{2.10}$$

λ_i is set to be φ_i^2 to make the order of magnitude of $\varphi_i^2\ell(\theta;\varphi_i)$ independent of φ_i [36]. This is to balance the training loss among different noise levels so that the model performs evenly well on each noise level. Since the score network $s_\theta(x,\varphi)$ now depends on the current generated sample x and the noise level φ, such network is called **Noise Conditional Score Network**.

Since we consider different noise levels, the Langevin dynamics needs some adjustments as well. It can be thought of as moving from completely noisy samples, following the directions towards clean samples (via the score), from large noise levels gradually to smaller noise levels. The adjusted algorithm is called **Annealed Langevin Dynamics (ALD)** [36]. The central idea of ALD stems from the observation that, a higher noise level φ would introduce better mixing and score matching for the noise-perturbed distribution q_φ, and thus should take a larger step α.

Algorithm 1: Annealed Langevin Dynamics (ALD)

Data: sequence of noise levels $\{\varphi_i\}_{i=1}^L$, number of sampling steps per noise level M, stepsize parameter ϵ, trained noise conditional score network $s_\theta(\cdot,\cdot)$
Result: $\widetilde{x}_0$

Initialize $\widetilde{x}_M \sim \mathcal{N}(0,\varphi_L^2 I_D)$;
for $i = L,\cdots,1$ **do**
  define the stepsize $\alpha_i \leftarrow \epsilon\frac{\varphi_i^2}{\varphi_L^2}$;
  for $t = 1,\cdots,M$ **do**
    sample $z_t \sim \mathcal{N}(0, I_D)$;
    update the new sample via $\widetilde{x}_{t-1} \leftarrow \widetilde{x}_t + \alpha_i s_\theta(\widetilde{x}_t,\varphi_i) + \sqrt{2\alpha_i}z_t$
  $\widetilde{x}_M = \widetilde{x}_0$

Intuitively, the sampling process begins with a highly noisy initialization—a "cloud" of samples— and progressively refines them as the noise level decreases, moving from coarse

to fine details in the data distribution. However, when transitioning between noise levels, the gap must not be too large. The gap here means the spacing between consecutive noise levels. If the noise level of the sample drops too quickly, the sampler cannot smoothly transition from coarse to fine details and may fail to follow the data manifold. otherwise, the sampler may fail to track the true data manifold, leading to instability or divergence in the sampling process. This transition between noise levels is called mixing.

2.2.4 SBGMs with Stochastic Differential Equations

To ensure smoother mixing, one can consider a continuous noise schedule instead. The iterative sampling process can be thought as solving a Stochastic Differential Equation (SDE) [39]. It is possible to employ an infinite number of noise levels to maximize the performance of SBGMs in a continuous sense. We first provide the most general account and then apply to our previous discrete case for comparison.

Suppose $x(0)$ are samples from some unknown distribution p_0. The goal is to add noises (diffuse) to the samples up to a point, so that the perturbed samples $x(T)$ follow some distribution p_T, which is independent of p_0 and of a form known to us. Here $x(t)$ refers to the sample after adding noise of level t, with $t \in [0, T]$. We use T, as opposed to L in the previous section, to emphasize that the noise levels are now continuous, but not discrete. An example of such p_T is a Gaussian distribution with some fixed mean and variance. With this setup, since the form of p_T is known, we can sample $x(T)$ from the distribution p_T, and then reverse the diffusion (denoise) to original samples $x(0)$, which should follow the unknown true distribution p_0, which we now call the **prior** distribution.

Formally, we wish to construct a diffusion process $\{x(t)\}_{0 \le t \le T}$ such that $x(0) \sim p_0$, and $x(T) \sim p_T$. There are many different formulation of the noising schedule. In general, consider that the diffusion process follows the Stochastic Differential Equation (SDE):

$$dx = f(x, t)dt + g(t)dw, \tag{2.11}$$

where w is the standard Brownian motion, $f(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ is called the drift coefficient of $x(t)$ and $g(\cdot) : \mathbb{R} \to \mathbb{R}$ is called the diffusion coefficient of $x(t)$. This noise schedule exists and is unique as long as f and g are globally Lipschitz [39].

Suppose the observed data are perturbed following the SDE (2.11), we can then reverse the diffusion from $x(T) \sim p_T$ via

$$dx = [f(x, t) - g(t)^2 \nabla_x \log p_t(x)]dt + g(t)dv, \tag{2.12}$$

where v is another standard Brownian motion and p_t is the probability density of $x(t)$.

To sum up, given data from p_0, one can add noise via SDE (2.11) so that the output samples follow the distribution p_T. Then, we can sample from p_0 by iterating samples from p_T using the reverse diffusion (2.12). Observant readers will notice that the intermediate distribution $p_t, t \in (0, T)$ still depends on the prior p_0. Under the principle of denoising score matching as in the discrete case, we replace it with the transition density $p_{0t}(x(t)|x(0)) = \mathcal{N}(x(t); x(0), \varphi^2(t) I_d)$ for tractability, where the noise level is now a continuous function $\varphi(t)$.

Now, we can compare it to the previous discrete case $\{\varphi_i\}_{i=1}^L$, which is geometric and increasing, with common ratio $\gamma > 1$, as we defined before. Define a function

$$\varphi(t) \coloneqq \varphi_1 \left(\frac{\varphi_L}{\varphi_1}\right)^{(t-1)/(L-1)}, \text{ for } t \in [1, L].$$

Note that with this, we have

$$\varphi(i) = \varphi_1 \left(\frac{\varphi_L}{\varphi_1}\right)^{(i-1)/(L-1)} = \varphi_1 (\gamma^{L-1})^{(i-1)/(L-1)} = \varphi_1 \gamma^i = \varphi_i, \text{ for } i = 1, \cdots, L.$$

So this $\varphi(t)$ is the continuous version that coincides the discrete set of noise levels we discussed before. Now if we fix the smallest and largest noise scales $\varphi_{min} \coloneqq \varphi_1$ and $\varphi_{max} \coloneqq \varphi_L$, and then we increase the number of noise levels L to infinity, we have $\varphi(t) = \varphi_{min}(\frac{\varphi_{max}}{\varphi_{min}})^t$ for $t \in (0, 1]$ [39]. Recall that the discrete noising process can be written as

$$x_i = x_{i-1} + \sqrt{\varphi_i^2 - \varphi_{i-1}^2} z_{i-1}, \text{ for } i = 1 \cdots, L,$$

where $z_{i-1} \sim \mathcal{N}(0, I_d)$ and x_0 is the clean sample. To transition to the continuous version, we consider continuous stochastic processes $\{x(t)\}_{t=0}^1$ and $\{z(t)\}_{t=0}^1$ such that $x(\frac{i}{L}) = x_i$ and $z(\frac{i}{L}) = z_i$ for $i = 0, \cdots, L$. As we increase the number of noise levels L, this becomes:

$$x(t + \Delta t) = x(t) + \sqrt{\varphi^2(t + \Delta t) - \varphi^2(t)}\, z(t) \approx x(t) + \sqrt{\frac{d[\varphi^2(t)]}{dt} \Delta t}\, z(t)$$

As $\Delta t \to 0$, this converges to

$$dx = \sqrt{\frac{d[\varphi^2(t)]}{dt}} dw, \tag{2.13}$$

where w is the standard Brownian motion. The corresponding transition kernel is

$$p_{0t}(x(t)|x(0)) = \mathcal{N}\left(x(t); x(0), \varphi_{min}^2 \left(\frac{\varphi_{max}}{\varphi_{min}}\right)^{2t} I_d\right).$$

Comparing Eq.(2.13) and Eq.(2.11), we set $f(x,t) = 0$ and $g(t) = \sqrt{\frac{d[\varphi^2(t)]}{dt}}$. The corresponding reverse diffusion SDE follows Equation (2.12) to become:

$$dx = -\frac{d[\varphi^2(t)]}{dt}\nabla_x \log p_{0t}(x)dt + \sqrt{\frac{d[\varphi^2(t)]}{dt}}dv. \tag{2.14}$$

The discretised version of the diffusion for $\{\varphi_i\}_{i=1}^L$ from Eq.(2.13) is

$$x_i = x_{i-1} + \sqrt{\varphi_i^2 - \varphi_{i-1}^2}z_{i-1}, \text{ where } z_{i-1} \sim \mathcal{N}(0, I_d), \tag{2.15}$$

with $x_0 \sim p_0$, and the discretised reverse diffusion SDE (2.14) is

$$x_i = x_{i-1} - (\varphi_i^2 - \varphi_{i-1}^2)\nabla_x \log p_{0t}(x_{i-1}) + \sqrt{\varphi_i^2 - \varphi_{i-1}^2}z_{i-1}. \tag{2.16}$$

To train such model, we follow the similar training loss as defined before in Eq.(2.10). We find a score network $s_\theta(x,t)$, so that it minimizes the loss

$$\frac{1}{2}\mathbb{E}_t\left[\lambda(t)\mathbb{E}_{x(0)}\mathbb{E}_{x(t)|x(0)}[\|s_\theta(x(t),t) - \nabla_{x(t)}\log p_{0t}(x(t)|x(0))\|_2^2]\right], \tag{2.17}$$

where t samples from the uniform distribution $\mathcal{U}[0,T]$ and $\lambda(t) : [0,T] \to \mathbb{R}_{>0}$ is a positive weighting function so that the order of magnitude of the residual for each noise level would be similar. Akin to the discrete case, $\lambda(t)$ can be set to be $\varphi^2(t)$.

2.2.5 Predictor-Corrector

At each time step, the numerical solver, SDE (2.16), gives an estimate of the sample at the next time step, serving as the "predictor" of the score. This estimation can be improved since we have trained the model $s_\theta(x,t) \approx \nabla_x \log p_t(x)$, estimating the same distribution. The idea for improvement is to hybridize the samples from either solver with some correction using ALD (Algorithm 1). It was found that the hybrid approach improved the accuracy significantly at the expense of doubling the computational cost [39]. The algorithm is presented in Algorithm 2.

2.2.6 Guided Diffusion

The fundamental assumption imposed on the original formulation of score-based generative models is that the only known information of the data distribution p_0 is the set of data

Algorithm 2: Predictor Corrector Scheme

Data: sequence of noise levels $\{\varphi_i\}_{i=1}^{L}$, number of correction steps per noise level M, step size parameter ϵ, trained noise conditional score network $s_\theta(\cdot,\cdot)$

Initialize $x_L \sim \mathcal{N}(0, \varphi_L^2 I_D)$;
for $i = L-1, \cdots, 1$ **do**
    `/* Predictor Step */`
    Define the step size $\alpha_i \leftarrow \epsilon \frac{\varphi_i^2}{\varphi_L^2}$;
    Sample $z_i \sim \mathcal{N}(0, I_d)$;
    Update the new sample via reverse diffusion in Eq. (2.16):
    $x_i \leftarrow x_{i+1} + (\varphi_{i+1}^2 - \varphi_i^2) s_\theta(x_{i+1}, \varphi_{i+1}) + \sqrt{\varphi_{i+1}^2 - \varphi_i^2} z_i$;
    `/* Corrector Step */`
    for $t = 1, \cdots, M$ **do**
        Sample $z_t \sim \mathcal{N}(0, I_D)$;
        Correct the sample via ALD: $x_i \leftarrow x_i + \alpha_i s_\theta(x_i, \varphi_i) + \sqrt{2\alpha_i} z_t$;

samples. However, in our context, the option price dynamics can be assumed to follow certain models, and hence the price surface distribution can be governed by some physical laws, i.e. the Black-Scholes PDE. In addition, it is also reasonable to assume some values on the price surface are known.

In short, during the inference step, the denoising samples are guided with some sparse observations and relevant PDE constraints, so that the generated samples would be more likely to be plausible and adhere to the imposed constraints. This type of diffusion is called **guided diffusion** [23].

Formally, the distribution of the sample is conditional on physical laws and sparse observation. Then, the score is also conditional on them [5]:

$$\begin{aligned} \nabla_x \log p(x|x_{obs}) &= \nabla_x \log \left[\frac{p(x)p(x_{obs}|x)}{p(x_{obs})} \right] \\ &= \nabla_x \log p(x) + \nabla_x \log p(x_{obs}|x). \end{aligned} \tag{2.18}$$

Note that the first term is the score, which was previously estimated with the score network. The second term accounts for how far off the current sample is from the observed values. Since our diffusion process uses normal distribution, this term is proportional to their Euclidean distance $\|x_{obs} - x\|$ [5].

For the PDE constraints, suppose the generated sample x solves a PDE $\Omega \subset \mathbb{R}^d$:

$$F(x; \eta) = 0 \text{ on } \Omega,$$

where F is some general differential operator on Ω and η is some parameters for the model. At each sampling step, the sample x is modified to slightly reduce the PDE residual r, which is defined to be $\|F(x; \eta)\|_2^2$. To this end, one can simply take a small step to the negative gradient direction $\nabla_x \|F(x; \eta)\|_2^2$ [23]. An advantage of this approach is that it reuses the same trained model. The residual term, though requiring careful discretisation, applies to other PDEs flexibly.

Chapter 3

Price Surface Completion for European Options

In this chapter, we will focus on the European Options and the completion of European Put price surfaces. We first formulate the problem in the discrete setting and describe our measures of arbitrage for a given price surface. We then introduce the method of previous project on arbitrage-free implied volatility surface completion using score-based generative models. We converted this method to complete price surfaces and use this as the benchmark. We then design two correctors and introduce our principle of guided diffusion: no-arbitrage correctors and physics-informed correctors. Afterwards, setups and results of numerical experiments will be shown.

3.1 Problem Formulation

We first describe the problem we intend to solve. We fix the risk-free rate $r = 0.1$ and assume no dividends. Suppose we are given a grid of maturities and strikes $\mathcal{G} \coloneqq \{(K_i, T_j) \colon i = 1, \cdots, n \text{ and } j = 1, \cdots, m\}$. Further suppose we are given observed European Put option prices in some subset of grid points $\mathcal{G}_{obs} \coloneqq \{(K_i, T_j) \in \mathcal{G} \colon P_E(K_i, T_j) \text{ is known}\}$. Our goal is to estimate the European put prices at the remaining grid points $\mathcal{G} \backslash \mathcal{G}_{obs}$, such that the whole surface is arbitrage-free. For convenience, we assume same number of strikes and maturities, $n = m$.

3.2 No-arbitrage Conditions

Under this setting, we first define loss terms to measure how "violated" the current price surface estimate is, in terms of the no-arbitrage conditions in Theorem 2.1.1 we presented before. More precisely, when given a price surface $\mathbf{P} \coloneqq [P_E(K_i, T_j)]_{i,j=1}^n$, we define the corresponding loss as in the set of no-arbitrage conditions in Theorem 2.1.1. The correspondence between each loss and the corresponding condition in Theorem 2.1.1 is shown below.

Monotonicity in K Condition (1) reflects the fact that a put with a higher strike always has a payoff that is at least as large as a lower-strike put. Therefore, put prices must increase with strike. We define a matrix $\mathbf{H}_{mono(K)} \in \mathbb{R}^{n\times n}$ where each entry is the violation of the monotonicity condition:

$$\left[\mathbf{H}_{mono(K)}\right]_{ij} \coloneqq \left(-\partial_K P_E(K_i, T_j)\right)_+, \text{ for } i, j = 1, \ldots, n. \tag{3.1}$$

We then define the corresponding loss $\ell_{mono(K)}$ by averaging these violations across the grid:

$$\ell_{mono(K)}(\mathbf{P}) \coloneqq \frac{1}{n^2} \sum_{i,j=1}^{n} \left[\mathbf{H}_{mono(K)}\right]_{ij}. \tag{3.2}$$

Intuitively, each entry is positive only if the put price decreases with strike, signaling a vertical spread arbitrage opportunity.

Convexity in K Condition (2) is closely linked to the risk-neutral probability distribution of the underlying asset. By the Breeden–Litzenberger identity [3],

$$\partial^2_{KK} P_E(K, T) = e^{-rT} f_{S_T}(K),$$

the second derivative of the put price with respect to strike equals the discounted risk-neutral density f_{S_T}, which is necessarily nonnegative. This requirement forces the put price curve to be convex in strike, ensuring consistency with a valid probability distribution. The slope constraint, requiring the derivative in strike to lie between zero and e^{-rT}, prevents the price from decreasing too quickly or too slowly, ruling out butterfly or vertical spread arbitrage and maintaining consistency with the underlying risk-neutral measure. The convexity violation matrix H_{conv} is

$$[\mathbf{H}_{conv}]_{ij} \coloneqq \left(-\partial^2_{KK} P_E(K_i, T_j)\right)_+, \text{ for } i, j = 1, \ldots, n. \tag{3.3}$$

$\mathbf{H}_{conv}$ captures where the surface is locally concave, which corresponds to a negative implied probability density. The corresponding loss is defined by

$$\ell_{conv}(\mathbf{P}) \coloneqq \frac{1}{n^2} \sum_{i,j=1}^{n} [\mathbf{H}_{conv}]_{ij} \,. \tag{3.4}$$

Bound Loss Condition (3) provides lower and upper bounds of European puts, which restrict the put price to lie between its intrinsic value, $(e^{-rT}K - S_0)_+$, and its maximal discounted payoff, $e^{-rT}K$. We define for $i, j = 1, \ldots, n$,

$$[\mathbf{H}_{bound}]_{ij} \coloneqq \left(P_E(K_i, T_j) - e^{-rT_j}K_i\right)_+ + \left(\max(e^{-rT_j}K_i - S_0, 0) - P_E(K_i, T_j)\right)_+ . \tag{3.5}$$

Note that the violations of upper and lower bounds are considered simultaneously. This measure is still valid because each grid point violates at most one of the two inequalities in condition (3). The corresponding loss term is defined by

$$\ell_{bound}(\mathbf{P}) \coloneqq \frac{1}{n^2} \sum_{i,j=1}^{n} [\mathbf{H}_{bound}]_{ij} \,. \tag{3.6}$$

Derivative Loss Condition (4) enforces an explicit bound on the strike derivative: $\partial_K P_E(K, T) \leq e^{-rT}$. The upper bound arises from put-call parity. Violations are quantified by:

$$[\mathbf{H}_{deri}]_{ij} \coloneqq \left(\partial_K P_E(K_i, T_j) - e^{-rT_j}\right)_+ , \text{ for } i, j = 1, \ldots, n. \tag{3.7}$$

The corresponding loss term is

$$\ell_{deri}(\mathbf{P}) \coloneqq \frac{1}{n^2} \sum_{i,j=1}^{n} [\mathbf{H}_{deri}]_{ij} \,, \tag{3.8}$$

Monotonicity in T Condition (5) ensures that a put option with a longer time to expiration cannot be cheaper than the same option with a shorter time to expiration. This condition prevents calendar-arbitrage opportunities, as longer-lived options contain more optionality and therefore must be at least as valuable as shorter-lived counterparts. The maturity violation matrix $\mathbf{H}_{mono(T)} \in \mathbb{R}^{n \times n}$ is defined by

$$\left[\mathbf{H}_{mono(T)}\right]_{ij} \coloneqq \left(-\partial_T P_E(K_i, T_j)\right)_+ , \text{ for } i, j = 1, \ldots, n. \tag{3.9}$$

This is positive whenever a longer-maturity put is cheaper than a shorter-maturity put. The loss term is then

$$\ell_{mono(T)}(\mathbf{P}) \coloneqq \frac{1}{n^2} \sum_{i,j=1}^{n} \left[\mathbf{H}_{mono(T)}\right]_{ij}. \tag{3.10}$$

Together, these conditions guarantee that the European put price surface is free of static arbitrage across both strike and maturity, and is consistent with a valid risk-neutral distribution of the underlying asset.

Note that this formulation of arbitrage losses relies on approximation of partial derivatives of the price surface, for which we use central difference schemes, with special treatment to the boundary points. For simplicity, we denote $P^{i,j} := P(K_i, T_j)$ for $i, j = 1, \cdots, n$.

$$\partial_T P^{i,j} \approx \begin{cases} \frac{-3P^{i,1}+4P^{i,2}-P^{i,3}}{2\Delta T} & \text{if } j = 1, \\ \frac{P^{i,j+1}-P^{i,j-1}}{2\Delta T} & \text{if } 1 < j < n, \\ \frac{3P^{i,N}-4P^{i,N-1}+P^{i,N-2}}{2\Delta T} & \text{if } j = n \end{cases}$$

$$\partial_K P^{i,j} \approx \begin{cases} \frac{-3P^{1,j}+4P^{2,j}-P^{3,j}}{2\Delta K} & \text{if } i = 1, \\ \frac{P^{i+1,j}-P^{i-1,j}}{2\Delta K} & \text{if } 1 < i < n, \\ \frac{3P^{N,j}-4P^{N-1,j}+P^{N-2,j}}{2\Delta K} & \text{if } i = n \end{cases}$$

$$\partial_{KK} P^{i,j} \approx \begin{cases} \frac{2P^{1,j}-5P^{2,j}+4P^{3,j}-P^{4,j}}{\Delta K^2} & \text{if } i = 1, \\ \frac{P^{i+1,j}-2P^{i,j}+P^{i-1,j}}{\Delta K^2} & \text{if } 1 < i < n, \\ \frac{-2P^{N,j}+5P^{N-1,j}-4P^{N-2,j}+P^{4,j}}{\Delta K^2} & \text{if } i = n. \end{cases}$$

With these approximations, we can measure the arbitrage losses when given a price surface estimate. We would want to minimize these terms during the sampling process.

3.3 Previous Work

In this section, we describe the overall scheme of previous approach [21] in our current context of price surface completion. The central idea is that, during the inference sampling, we only allow updates that roughly reduce the arbitrage losses ℓ we defined before. Formally, suppose x_t is the proposed update for the intermediate estimate at inference step t. We first compute the corresponding losses $\ell^t_{mono(K)}$, ℓ^t_{conv}, ℓ^t_{bound}, ℓ^t_{deri} and $\ell^t_{mono(T)}$. And

then compare the current losses with the losses of the previous step $t-1$ by computing this term β_t[1]:

$$\beta_t \coloneqq \max\left(\frac{\ell^t_{mono(K)}}{\ell^{t-1}_{mono(K)}+\epsilon}, \frac{\ell^t_{conv}}{\ell^{t-1}_{conv}+\epsilon}, \frac{\ell^t_{bound}}{\ell^{t-1}_{bound}+\epsilon}, \frac{\ell^t_{deri}}{\ell^{t-1}_{deri}+\epsilon}, \frac{\ell^t_{mono(T)}}{\ell^{t-1}_{mono(T)}+\epsilon}\right), \tag{3.11}$$

where $\epsilon > 0$ is a small positive constant to prevent division by zero. To make the terms well-defined, set the initial values of all loss terms to be 0. This β_t can be viewed as the *acceptance ratio at step t*. The smaller $\beta_t > 0$ is, the more improvements the proposed step x_t makes over the current step x_{t-1} in terms of arbitrage conditions.

The update rule is formulated as follows. First generate a random number $u \sim \mathcal{U}[0, 1.2]$ as the threshold. If the acceptance ratio $\beta_t \leq u$, then this update x_t is accepted as the next sample. Otherwise, the current step is reused. The upper bound 1.2 is chosen so that the intermediate sample is accepted despite it might be slightly worse in terms of arbitrage losses. The relaxed upper bound mitigates premature rejection caused by noise in loss estimation or discretization error in the arbitrage penalties, thereby preventing oscillatory behavior and improving convergence robustness. In addition, to incorporate the observed values into the process, hard constraint was employed. In particular, recall that $\mathcal{G}_{obs}$ is the set of grid points $\{(K_i, T_j)\} \subset \mathcal{G}$ that are already known. Define a mask $\Xi \in \{0,1\}^{n\times n}$ by

$$\Xi_{ij} \coloneqq \begin{cases} 1 & \text{if } (T_i, K_j) \in \mathcal{G}_{obs} \\ 0 & \text{otherwise.} \end{cases}$$

After every update, we fix the values at the corresponding position to be the observed value. The fundamental reason SBGM can generate surfaces that closely match the ground truth is that it learns a prior over realistic, arbitrage-free price surfaces. The generated surfaces match the observed prices and, under this learned prior, produce an arbitrage-free completion that lies near the true surface in distribution, provided the ground truth is supported by the training data. The whole algorithm can found in Algorithm 3.

[1]The original work is based on volatility surfaces. The no-arbitrage conditions are therefore surrounding the butterfly loss and calendar losses. Here we presented the adaptation of the method to price-based no-arbitrage conditions.

Algorithm 3: Annealed Langevin Dynamics (ALD) for Arbitrage-free Price Surface Completion

Data: sequence of noise levels $\{\varphi_i\}_{i=1}^{L}$, number of correction steps per noise level M, step size parameter ϵ, trained noise conditional score network $s_\theta(\cdot,\cdot)$

Initialize $x_L \sim \mathcal{N}(\varphi_L^2 I_d)$;
for $i = L-1, \cdots, 1$ **do**
  Define the step size $\alpha_i \leftarrow \epsilon \frac{\varphi_i^2}{\varphi_L^2}$;
  for $t = 1, \cdots, M$ **do**
    Sample $z_t \sim \mathcal{N}(0, I_d)$;
    Define the proposed update $x_{tmp} \leftarrow x_i + \alpha_i s_\theta(x_i, \varphi_i) + \sqrt{2\alpha_i} z_t$;
    Compute the arbitrage loss terms via Eq. (3.2), (3.4), (3.6), (3.8) and (3.10);
    Compute the acceptance ratio for the proposed step β_t as in Eq. (3.11);
    Sample a threshold parameter $u \sim \mathcal{U}[0, 1.2]$;
    `/* Accept if the acceptance ratio is small enough */`
    if $\beta_t < u$ **then**
      $x_i \leftarrow x_{tmp}$;
    `/* Hard constraint on the observed values */`
    $x_i \leftarrow x_i \odot (1 - \Xi) + x_{obs} \odot \Xi$, where $\odot$ is element-wise multiplication.;

Observant readers may notice that the main update in the method is driven by the learned model dynamics, whereas the no-arbitrage updates follow a trial-and-error approach. Although the method has been shown to successfully generate arbitrage-free surfaces [21], this typically requires a large number of sampling steps.

To address this issue, we propose an active correction mechanism applied to intermediate samples, designed to produce updates of higher quality with respect to arbitrage constraints.

Concretely, the sampling procedure follows a predictor–corrector framework. The predictor step—implemented using either an SDE or ODE solver—advances the sample from one noise level to the next according to the learned score model. However, due to discretization error and the finite number of predictor steps, the resulting sample may not fully reflect the target distribution at the new noise level. This phenomenon is referred to as slow mixing, meaning that the transition between noise levels is inefficient. By inserting a corrector step between each predictor step, one ensures that samples are fully perturbed to the appropriate noise level before continuing the sampling procedure.

Annealed Langevin Dynamics (ALD) is commonly used as the corrector in the Predictor–Corrector scheme (Algorithm 2). However, the main component of ALD steers samples using the score network, which is trained primarily to match the data distribution and is not explicitly informed of desirable structural properties—such as no-arbitrage conditions.

Therefore, the goal of our thesis is to design arbitrage-aware correctors that remain compatible with the predictor–corrector framework while promoting the generation of arbitrage-free option price surfaces. To understand the effects of our approach, we use the pure predictor scheme as the benchmark, and evaluate the performance when combined with our designed correctors.

Known Values Treatment

In guided diffusion–based sampling, the manner in which known values are enforced plays an important role in both the theoretical consistency of the sampler and its numerical stability. Although hard constraints—where observed grid points are fixed exactly at every iteration—are a natural choice (also used in previous work [21]), they may introduce discontinuities into the sampling dynamics that are not fully aligned with the stochastic differential equation (SDE) underlying the diffusion process. In contrast, soft constraints provide a more principled way to incorporate known values while preserving the smooth evolution of samples across noise levels.

From a theoretical standpoint, score-based diffusion models assume that samples evolve according to continuous drift and diffusion terms determined by the learned score. Enforcing hard constraints can be interpreted as repeatedly projecting the sample onto a lower-dimensional manifold at selected coordinates, effectively overriding the SDE dynamics at those locations. Since this projection is not accounted for by the learned score, it may introduce inconsistencies in the sampling trajectory. Soft constraints, on the other hand, act as a gradual conditioning mechanism and better approximate conditioning on partial observations within the diffusion framework. In addition, the use of soft constraint fits the setting of guided diffusion in Equation (2.18). To this end, we introduce a noise-dependent weighting function

$$\omega_{\text{obs}}(t) \coloneqq \exp\left(-\frac{3(L-t)}{L}\right), \tag{3.12}$$

where $t \in 1, \ldots, L$ indexes the noise level. Let $\Xi \in \{0,1\}^{n\times n}$ denote the indicator mask of observed grid locations, for $i, j = 1, \ldots, n$,

$$\Xi_{ij} \coloneqq \begin{cases} 1 & \text{if } (T_i, K_j) \in \mathcal{G}_{obs} \\ 0 & \text{otherwise.} \end{cases}$$

At the end of each step, the current sample x_i is updated according to

$$x_i \leftarrow \omega_{obs} x_i \odot (1 - \Xi) + (1 - \omega_{obs}) x_{obs} \odot \Xi, \tag{3.13}$$

where $\odot$ denotes elementwise multiplication. This schedule satisfies $\omega_{\text{obs}}(L) = 1$ and decays monotonically as t decreases, ensuring that observed values exert minimal influence at high noise levels while being increasingly enforced as sampling progresses toward lower noise regimes. The decay rate $\alpha = 3$ is chosen to center the transition from prior-driven to observation-driven sampling near the midpoint of the diffusion trajectory. The exponential form provides a smooth transition that avoids abrupt changes in constraint strength. During early sampling steps, the sampler can explore globally consistent configurations under the generative prior with minimal bias from observation ($\omega_{obs} \approx 1$). By the midpoint, the observations exert strong influence, anchoring the emerging surface to the known data before local details are refined by the correctors. In the final steps, $\omega_{obs} \approx 0.05$, assigning approximately 95% of the weight to the observations.

Together with the soft constraint, the complete benchmark algorithm can be found in Algorithm 4.

Algorithm 4: Pure Predictor Scheme: SDE Solver

Data: sequence of noise levels $\{\varphi_i\}_{i=1}^{L}$, number of correction steps per noise level M, step size parameter ϵ, trained noise conditional score network $s_\theta(\cdot, \cdot)$

Initialize $x_L \sim \mathcal{N}(\varphi_L^2 I_d)$;

for $i = L - 1, \cdots, 1$ **do**

  Predictor Step

    Define the step size $\alpha_i \leftarrow \epsilon \frac{\varphi_i^2}{\varphi_L^2}$;

    Sample $z_i \sim \mathcal{N}(0, I_d)$;

    Update the new sample via discretized reverse diffusion in Eq. (2.16):
    $x_i \leftarrow x_{i+1} + (\varphi_{i+1}^2 - \varphi_i^2) s_\theta(x_{i+1}, \varphi_{i+1}) + \sqrt{\varphi_{i+1}^2 - \varphi_i^2} z_i$;

  `/* Soft constraint on the observed values */`

  Define ω_{obs} by Eq.(3.12).;

  $x_i \leftarrow \omega_{obs} x_i \odot (1 - \Xi) + (1 - \omega_{obs}) x_{obs} \odot \Xi$, where $\odot$ is element-wise multiplication.;

3.4 Methodology

We now describe our approach. The main inspiration comes from guided diffusion, where the sampling procedure is steered not only by the generative model but also by additional constraints projected onto an appropriate space.

3.4.1 No-Arbitrage Corrector

In our setting, we would like intermediate samples to satisfy the no-arbitrage conditions. Consequently, the arbitrage losses should decrease throughout the sampling trajectory.

More precisely, we introduce no-arbitrage correctors ν_{NA}, whose purpose is to determine a descent direction that reduces the arbitrage loss at step t. To estimate the descent direction, one can apply finite difference and iterate over each dimension for the surface. A naïve approach would compute the gradient of the arbitrage loss by applying finite differences along every dimension of the surface. However, direct finite-difference gradient computation requires $O(n^2)$ evaluations. This becomes prohibitively expensive in high-dimensional settings. Instead, we adopt simultaneous perturbation stochastic approximation (SPSA) to estimate the gradient efficiently [30]. Let $\ell(x)$ denote the arbitrage loss to reduce ,and $\mathbf{H} \in \mathbb{R}^{n\times n}$ be the corresponding violation matrix as defined before in (3.1),(3.3),(3.5),(3.7), and (3.9). We fix a number of trials n_{trial} and a small positive constant $\widetilde{\epsilon}$. For each trial $i = 1, \cdots, n_{trial}$, we sample a random perturbation direction $\Delta_i \in \mathbb{R}^{n\times n}$ and approximate the gradient by

$$\nu_{NA}^{(i)}(x) \coloneqq -\frac{\ell(x + \widetilde{\epsilon}\Delta_i) - \ell(x - \widetilde{\epsilon}\Delta_i)}{2\widetilde{\epsilon}}\Delta_i.$$

This is an estimator of $-\nabla_x \ell(x)$ and we used Rademacher matrix Δ_i for this thesis, that is, each element in the matrix Δ_i has 0.5 probability of being $+1$ or -1 independently [19]. Averaging over trials gives the final corrector direction:

$$\widetilde{\nu}_{NA}(x) \coloneqq \frac{1}{n_{trial}} \sum_{i=1}^{n_{trial}} \nu_{NA}^{(i)}(x). \tag{3.14}$$

Intuitively, this is to project the descent direction onto several random directions and estimate the true descent by their expectation. At the beginning of the sampling process, arbitrage losses can be very large, which would produce excessively large corrections and potentially destabilize the surface. To address this, we introduce a normalized weight matrix using the violation matrix $\mathbf{H}$:

$$W(x) = \frac{\mathbf{H}}{\|\mathbf{H}\|_2 + \delta}, \tag{3.15}$$

where $\delta = 10^{-8}$ prevents division by zero. This normalization scales the corrections proportionally to the relative severity of violations, ensuring that updates remain stable and balanced across the surface.

Finally, the sample is updated using a weight parameter $\lambda_{NA} > 0$:

$$x \leftarrow x + \lambda_{NA}\, \nu_{NA}(x). \tag{3.16}$$

Here, $\nu_{NA}(x) \coloneqq \widetilde{\nu}_{NA}(x)_{NA} \odot W(x)$ and $\odot$ denotes elementwise multiplication, which emphasizes corrections where violations are largest while keeping the overall step size controlled.

3.4.2 Physics-informed Correctors

Another projection space we can consider, is to assert the modeling assumption to satisfy the Black-Scholes PDE. The ideal European put price surface would attain minimal PDE residual. Formally, the price surface P would satisfy

$$\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} + rS\frac{\partial P}{\partial S} - rP = 0, \tag{3.17}$$

and thus the residual can be defined as

$$\left\| \frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} + rS\frac{\partial P}{\partial S} - rP \right\|_2^2.$$

It is tempting to apply this residual directly on to the price surface $P(K,T)$. However, our price surface lies on (K,T) space, but not (S,t) space. Fortunately, Dupire has shown under the Black Scholes model and fixed S_0, there is a corresponding PDE in the (K,T) space:

$$\frac{\partial P}{\partial T} = \frac{1}{2}\sigma_{LV}^2(K,T)K^2\frac{\partial^2 P}{\partial K^2} - (r-q)K\frac{\partial P}{\partial K}, \tag{3.18}$$

where

$$\sigma_{LV}(K,T) \coloneqq \left(\frac{\partial_T P + rK\partial_K P}{\frac{1}{2}K^2\partial_{KK}P} \right)^{1/2}$$

is called *local volatility*. For a price surface $P \in \mathbb{R}^{n\times n}$, we define a per-grid-point residual for $i,j = 1,\ldots,n$,

$$R(K_i,T_j;P) \coloneqq \frac{\partial P}{\partial T}(K_i,T_j) - \frac{1}{2}\sigma_{LV}^2(K_i,T_j)K_i^2\frac{\partial^2 P}{\partial K^2}(K_i,T_j) + (r-q)K_i\frac{\partial P}{\partial K}(K_i,T_j). \tag{3.19}$$

The global residual can then be expressed as

$$resid(P) = \|R(P)\|_2^2 = \sum_{i,j=1}^{n} |R(K_i, T_j; P)|^2. \tag{3.20}$$

However, this quantity involves several approximations of partial derivatives. It depends on the tangents and curvatures of the surface, which are highly unstable for noisy surfaces, particularly during the early stages of the sampling process. Consequently, the PDE residual computed using Dupire's equation (3.18) can be severely affected by discretization errors. To mitigate this issue, we employ similar normalization as for $\widetilde{\nu}_{NA}$. Define

$$\widetilde{\nu}_{PI}(x) \coloneqq -\frac{1}{n_{trial}} \sum_{i=1}^{n_{trial}} \frac{resid(x + \widetilde{\epsilon}\Delta_i) - resid(x - \widetilde{\epsilon}\Delta_i)}{2\widetilde{\epsilon}} \Delta_i, \tag{3.21}$$

where Δ_i are i.i.d Rademacher matrices in $\mathbb{R}^{n \times n}$. To prevent overly large updates and improve numerical stability, we normalize the gradient and scale it according to the residual magnitude:

$$\nu_{PI}(x) = \frac{\widetilde{\nu}_{PI}(x)}{\|\widetilde{\nu}_{PI}(x)\|_2 + 10^{-8}}. \tag{3.22}$$

The sample x is then corrected with some weight $\lambda_{PI} > 0$ via

$$x \leftarrow x + \lambda_{PI} \nu_{PI}.$$

3.4.3 Physics-informed No-arbitrage Corrector (PINA)

It is obvious that the current setup of the two correctors is agnostic to the noise level. The size of each corrective update plays a critical role in the sampling dynamics: if the update is too large, the sample may overshoot and destroy previously accurate progress; if it is too small, the correction becomes ineffective. Therefore, it is necessary to introduce a mechanism that modulates the effective strength of the correctors throughout the sampling process.

In our implementation, both the no-arbitrage corrector ν_{NA} and the physics-informed corrector ν_{PI} are applied throughout the sampling trajectory. Rather than enforcing these constraints as hard projections at specific stages, we control their influence through normalization and time-dependent annealing of the update magnitude. Specifically, the SPSA-based gradient estimates are normalized to remove scale dependence, and the resulting corrections are multiplied by an annealing factor that decays with the diffusion time.

This design reflects two considerations. First, no-arbitrage conditions constitute hard economic constraints that the final generated surfaces must satisfy. Applying ν_{NA} throughout sampling, while gradually reducing its magnitude, allows the surface to be steered toward arbitrage-free regions without destabilizing early noisy samples. Second, for the physics-informed PDE residual, we observe that this residual is rarely exactly zero even for ground-truth surfaces, due to discretization error and potential model mismatch. Consequently, aggressively enforcing ν_{PI} near the end of sampling may introduce numerical artifacts or bias the surface away from the data. By applying ν_{PI} in a normalized and annealed manner, its effect remains strongest when the sample is highly noisy and gradually diminishes as the surface approaches the target distribution. And for this reason, we decide to apply PI-corrector and then NA-corrector since NA-correction steers towards the hard arbitrage constraints.

Overall, this strategy allows both correctors to act as soft guidance mechanisms rather than hard constraints, stabilizing the sampling process while improving convergence toward economically and physically meaningful option price surfaces. More precisely, we will add additional weights $\omega_{NA}(t)$ and $\omega_{PI}(t)$ to the correctors, which are defined by

$$\omega_{NA}(t) \coloneqq e^{-0.1t}, \tag{3.23}$$

and

$$\omega_{PI}(t) \coloneqq e^{-5t}. \tag{3.24}$$

At early stages of the sampling ($t \approx 0$), both weights are close to 1, allowing the correctors to meaningfully guide the sample towards a realistic, arbitrage-free surface. As the sampling progresses, the weights decay exponentially, preventing the correctors from producing overly large updates once the sample is already near the target distribution. The decay rate for PI-corrector, controlled by the exponent factor 5, is chosen empirically to balance stability and effectiveness across the sampling trajectory. We found that slower decay for the PI-corrector would lead to unstable sampling. For the NA-corrector, since the normalization in (3.15) takes care of the scaling issues and helps stabilize the sampling, and the decay rate can be smaller than that of the PI-corrector. This schedule ensures that both the physics-informed (PI) and no-arbitrage (NA) correctors act primarily in the early stages, providing structural guidance while maintaining stable convergence. We note that the choices of the exponential decay factors (here, 5 and 0.1) are empirical and may not be optimal for all datasets or model configurations. Alternative schedules (e.g. linear decay, different decay rates, or adaptive schemes) could potentially improve convergence or sample quality, but the current setting has been found sufficient in our experiments.

Algorithm 5: SDE Solver + PINA

Data: sequence of noise levels $\{\varphi_i\}_{i=1}^{L}$, number of correction steps per noise level M, step size parameter ϵ, trained noise conditional score network $s_\theta(\cdot,\cdot)$, weights for PINA correctors λ_{NA}, λ_{PI}

Initialize $x_L \sim \mathcal{N}(\varphi_L^2 I_d)$;

for $i = L-1, \cdots, 1$ **do**

  Predictor Step

    Sample $z_i \sim \mathcal{N}(0, I_d)$;

    Update the new sample via discretized reverse diffusion in Eq. (2.16):
    $x_i \leftarrow x_{i+1} + (\varphi_{i+1}^2 - \varphi_i^2) s_\theta(x_{i+1}, \varphi_{i+1}) + \sqrt{\varphi_{i+1}^2 - \varphi_i^2} z_i$;

  Corrector Step

    for $t = 1, \cdots, M$ **do**

      Sample $z_t \sim \mathcal{N}(0, I_d)$;

      Set $x_{tmp} \leftarrow x_i$;

      `/* Attach the zero-maturity boundary */`

      Let $\bar{x}_{tmp} \leftarrow concatenate([x_i, (S_0 - K)_+])$;

      `/* Compute the PINA correctors */`

      PI Corrector

        Set $\nu_{PI}(\bar{x}_{tmp})$ via Eq.(3.22) and weight ω_{PI} by Eq.(3.24);

        Update using this PI corrector $x_{tmp} \leftarrow x_{tmp} + \omega_{PI}\lambda_{PI}\nu_{PI}$;

      NA Corrector

        Set $\nu_{NA}(\bar{x}_{tmp})$ via Eq.(3.16) and weight ω_{NA} by Eq.(3.23);

        Update using this NA corrector $x_{tmp} \leftarrow x_{tmp} + \omega_{NA}\lambda_{NA}\nu_{NA}$;

  `/* Soft constraint on the observed values */`

  Define ω_{obs} by Eq.(3.12).;

  $x_i \leftarrow \omega_{obs} x_i \odot (1 - \Xi) + (1 - \omega_{obs}) x_{obs} \odot \Xi$, where $\odot$ is element-wise multiplication.;

3.4.4 Boundary Treatment

In our experiments, we found that the largest reconstruction errors tend to occur near the boundaries and corners of the surface. One contributing factor is the discretisation error inherent in numerical differentiation. To mitigate this, we employed boundary-attentive finite difference schemes, as discussed in Section 3.2.

Deep out-of-the-money (OTM) European puts correspond to strikes far above the current underlying price, meaning that the probability of the option expiring in the money is extremely low. As a result, their theoretical value under any risk-neutral measure is very close to zero. Negative option prices, however, are economically nonsensical, as they would imply that a trader could receive money for *taking on an obligation* to potentially buy the underlying—creating an immediate arbitrage opportunity.

In practice, our generative model may occasionally produce slightly negative values in these regions due to noising values close to 0. To maintain economic consistency, we clip the generated surface from below at zero.

3.4.5 Network Training

For the architecture of the score network $s_\theta(\cdot, \cdot)$, we employed the U-Net architecture in [36], which is known for its success in semantic segmentation. In particular, we used a simplified version of the RefineNet architecture for $s_\theta(\cdot, \cdot)$ following [21]. The RefineNet architecture came from [29] and was known to improve multi-resolution segmentation results with its ResNet design. Instead, we reduced the number of feature maps for each cascade layer since price surfaces are much simpler than human faces.

To train such network, there are some hyperparameters to tune, including the noise scales $\{\sigma_i\}_{i=1}^L$, sampling step size ϵ and the number of sampling steps M. In this project, we followed several techniques from [37] to improve the results.

Initial noise scale

Recall that the initial noise scale φ_L should be large enough so that the final noisy samples are diverse enough to cover low density regions of the data. For this goal, pick φ_L to be as large as the maximum Euclidean distance between all pairs of training data points. The minimum noise scale φ_1 is set to be 10^{-3}.

Common ratio for the noise scale sequence

The sequence of noise scales $\{\varphi_i\}_{i=1}^{L}$ is an increasing geometric sequence, with common ratio denoted by γ. With fixed φ_1 and φ_L, it remains to determine the number of noise levels L. This number can also be controlled by the common ratio $\gamma > 1$. The closer γ is to 1, the larger L is. Since our price surfaces are of dimension $n \times n$, the common ratio $\gamma > 0$ is selected such that

$$\Phi(\sqrt{2}n(\gamma - 1) + 3\gamma) - \Phi(\sqrt{2}n(\gamma - 1) - 3\gamma) = \frac{1}{2}, \tag{3.25}$$

where Φ is the c.d.f. of the standard normal distribution. This derivation of this equation can be found in [37]. Once γ is found, since $\varphi_L = \varphi_1 \gamma^{L-1}$, we can find the corresponding number of noise levels L.

Sampling step size and Number of Sampling steps

There is a connection between the step size ϵ and the number of sampling steps M. The observation is that since the noise we injected in the diffusion is Gaussian, one can identify a relationship between ϵ and M [37]: let $\alpha = \epsilon\varphi_i^2/\varphi_L^2$ and $s_M := Var(x_M)$, where x_M is the output of ALD after M steps. Then

$$\frac{s_M}{\varphi_i^2} = \left(1 - \frac{\epsilon}{\varphi_1}\right)^{2M} \left(\gamma^2 - \frac{2}{\varphi_1^2 - \varphi_1^2\left(1 - \frac{\epsilon}{\varphi_1^2}\right)^2}\right) + \frac{2\epsilon}{\varphi_1^2 - \varphi_1^2\left(1 - \frac{\epsilon}{\varphi_1^2}\right)^2} \tag{3.26}$$

The motivation of this is to identify an appropriate step size ϵ, so that after M annealing steps, the output sample would smoothly transition across noise levels. In other words, the variance of M-step samples s_M should match that of the noise level φ_i^2. Therefore, one can attempt to select the sampling step M as large as the computational budget allows, then find the corresponding step size ϵ such that the right hand side of Eq.(3.26) is close to 1. We applied grid search over $\epsilon \in [10^{-4}, 1]$ for such selection, as recommended in [37] over using gradient-based optimization methods.

Training Stability

It was found in [37] that the generated image samples sometimes exhibit unstable quality, despite the steady decrease in objective loss during training the score network. To resolve

this, they applied exponential moving average (EMA). Suppose θ_i denotes the parameters of the score network $s_{\theta_i}(\cdot,\cdot)$ after the i-th iteration of training, and suppose θ' be an independent copy of the parameters. The network is updated via $\theta' \leftarrow \eta\theta' + (1-\eta)\theta_i$ after each optimization step, where η is the momentum parameter, for which we set $\eta = 0.999$. Although it slows down the training process, the sampling was shown to stabilize [37].

3.5 Numerical Experiments

After the model is trained, given a set of option prices on an incomplete grid $\mathcal{G}$, the model is able to guide the sampling to complete the rest. In this section, we will describe our experiments, including the data generation, evaluation metrics and numerical results.

3.5.1 Data Generation

To generate a rich collection of training data, we use the Heston model to generate synthetic discrete European put price surfaces. The Heston model is a generalization of the original Black-Scholes Model, with the relaxation of the constant volatility assumption [17]. Instead, Heston assumes the variance variable v_t (square of volatility) to follow a stochastic process:

$$\begin{aligned} dS_t &= rS_t dt + \sqrt{v_t} S_t dZ_t^1, S_{t_0} = S_0, \\ dv_t &= \kappa(\bar{v} - v_t)dt + \xi\sqrt{v_t} dZ_t^2, v_{t_0} = v_0, \\ dZ_t^1 dZ_t^2 &= \rho dt, \end{aligned} \tag{3.27}$$

where Z_t^1 and Z_t^2 are two Brownian motions for the asset price S_t and variance v_t respectively, with the correlation coefficient ρ. $\bar{v}$ is the long term variance, κ is the reversion speed of v_t, and ξ is the volatility of the volatility. It can then be shown that the European put $P_E(S, v, t)$ satisfies

$$\frac{\partial P_E}{\partial t} + (r-q)S\frac{\partial P_E}{\partial S} + \frac{1}{2}vS^2\frac{\partial^2 P_E}{\partial S^2} + \kappa(\bar{v}-v)\frac{\partial P_E}{\partial v} + \frac{1}{2}\xi^2 v\frac{\partial^2 P_E}{\partial v^2} + \rho\xi vS\frac{\partial^2 P_E}{\partial S\,\partial v} - rP_E = 0, \tag{3.28}$$

with some corresponding boundary conditions [27]. This relaxation on the assumption of constant volatility provides a more realistic collection of price surfaces. To solve Equation (3.28), there are no analytical formula. Instead, Monte Carlo methods, numerical integration [28] and COS method [43] are common numerical methods to solve Equation (3.28).

To generate our synthetic training data, we applied numerical integration [28], using strikes $K \in [60, 160]$ and maturities $T \in [0.1, 1]$, evenly spaced in 4×4, 8×8 and 16×16 meshes respectively. We fixed the risk-free rate $r = 0.1$ and zero dividend $q = 0$ for simplicity. The Heston parameters $\{\rho, \bar{v}, \kappa, \xi, v_0\}$ were uniformly sampled from Table 3.1, following [21].

Heston Parameter	Range
ρ	$[-0.95, 0]$
$\bar{v}$	$[0.05, 0.5]$
κ	$[0.1, 2]$
ξ	$[0.05, 0.5]$
v_0	$[0.05, 0.5]$

Table 3.1: Configuration for Heston Parameters

To ensure the instantaneous variance v_t is strictly positive, the Feller's condition $2\kappa\bar{v} > \xi^2$ was imposed [17]. The final training dataset then contains 100K European put surfaces, for each mesh after selecting those that satisfy the no-arbitrage conditions in Theorem 2.1.1.

To enable the model to learn the underlying distribution of option price surfaces, it is trained using the noise-conditioned loss function defined in (2.10). During training, the model assumes fully observed surfaces, since the objective is to learn the complete joint distribution of arbitrage-free surfaces rather than to handle partial observations. Sparse or missing values are introduced only at inference in the numerical experiments, where the model is conditioned on the observed entries. This procedure ensures that the generated surfaces remain consistent with available market data while simultaneously satisfying the prescribed arbitrage conditions. The ground truth is itself a typical surface under the learned distribution in the synthetic experiments. Therefore, comparing the generated surfaces to the ground truth provides a meaningful evaluation metric: it quantifies how well the model recovers a surface that is not only consistent with observations and arbitrage constraints, but also representative of the underlying distribution from which the true surface is drawn.

3.5.2 Comparison Metrics

To demonstrate the completion power of the method, we attempted to complete 100 European put surfaces with $40\%, 60\%$ and 80% missing values, which are selected randomly. For the sampling steps M, we picked $M \in \{1, 2, 5, 10, 20\}$, and determined their corresponding

sequences of noise scales $\{\varphi_i\}_{i=1}^{L}$ and the step size ϵ according to the discussion in Section 3.4.5. The performance metrics include the arbitrage losses $\ell_{mono(K)}$, ℓ_{conv}, ℓ_{bound}, ℓ_{deri} and $\ell_{mono(T)}$ as defined in Equations (3.2), (3.4), (3.6), (3.8) and (3.10). In addition, we evaluated the mean relative error (MRE) and mean absolute error (MAE) on the missing grid points $\mathcal{G}\backslash\mathcal{G}_{obs}$. To be more precise, we define the mean relative error and mean absolute error for a given generated price surface P_{est} and the ground truth P_{true} by

$$MRE := \begin{cases} \frac{1}{|\mathcal{G}\backslash\mathcal{G}_{obs}|}\sum_{(K_i,T_j)\in\mathcal{G}\backslash\mathcal{G}_{obs}} \frac{|P_{est}(K_i,T_j)-P_{true}(K_i,T_j)|}{|P_{true}(K_i,T_j)|}, & \text{if } P_{true}(K_i,T_j)\neq 0 \\ \frac{1}{|\mathcal{G}\backslash\mathcal{G}_{obs}|}\sum_{(K_i,T_j)\in\mathcal{G}\backslash\mathcal{G}_{obs}} \frac{|P_{est}(K_i,T_j)-P_{true}(K_i,T_j)|}{|P_{est}(K_i,T_j)|}, & \text{otherwise} \end{cases} \tag{3.29}$$

and

$$MAE := \frac{1}{|\mathcal{G}\backslash\mathcal{G}_{obs}|}\sum_{(K_i,T_j)\in\mathcal{G}\backslash\mathcal{G}_{obs}} |P_{est}(K_i,T_j) - P_{true}(K_i,T_j)|. \tag{3.30}$$

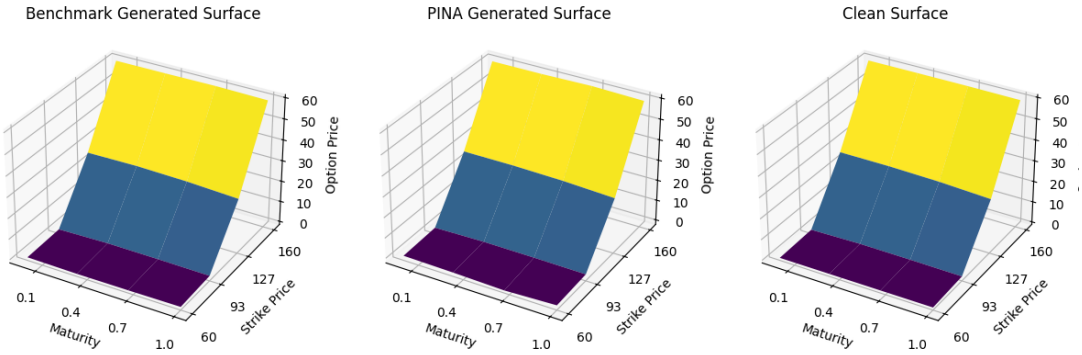
To evaluate the performance in the context of options, we also compare the MRE and MAE over different moneyness regions. We will mainly investigate the improvement of PINA correctors in addition to the pure predictor scheme using SDE solver (Algorithm 4).

3.5.3 Numerical Results

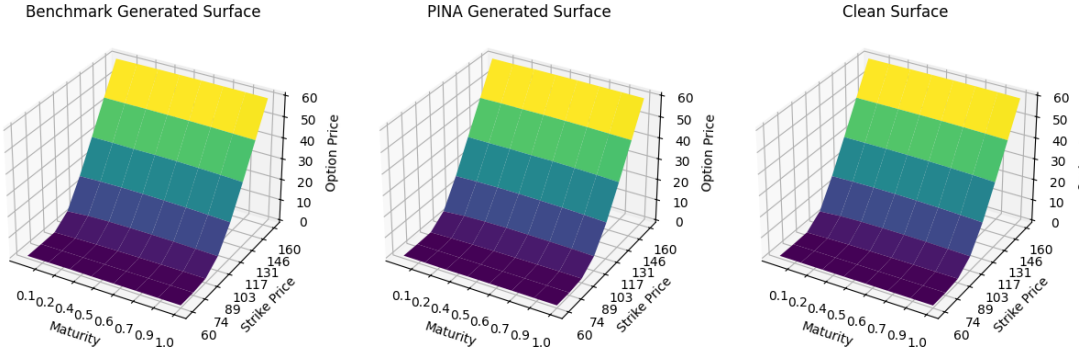
Overall Error Metrics

We first validate our proposed models qualitatively. Figures 3.1a, 3.1b and 3.1c show the European option surfaces where the strike varies between 60 to 160 and the maturity between 0.1 to 1 on a (K, T) grid of size 4×4, 8×8 and 16×16, respectively. Here, we compare the European option surfaces generated by SDE solver and SDE+PINA. We can see that they all show similar result as the ground truth surface.

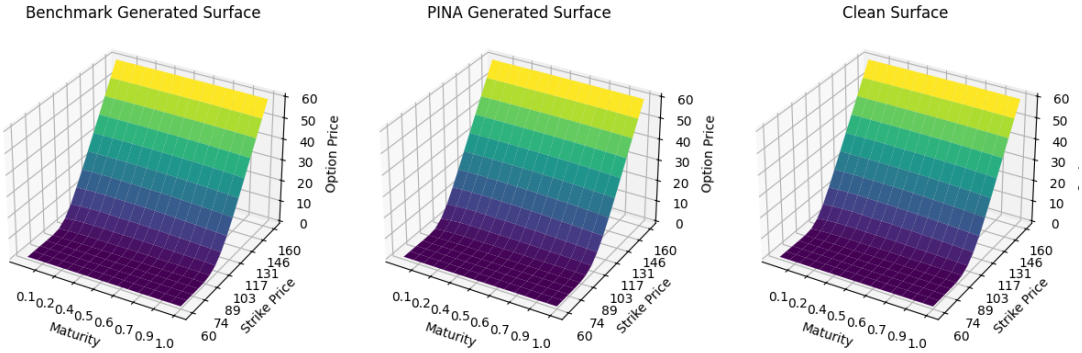
Figures 3.2, 3.3, and 3.4 show the lowest mean absolute error (MAE) obtained after searching over a predefined pool of NA and PI regularization parameters, λ_{NA} and λ_{PI} across different grids. Specifically, for each setting we sample using all combinations of $(\lambda_{NA}, \lambda_{PI})$ in the pool and retained the minimum MAE across runs. This presentation emphasizes the intrinsic capability of the proposed approach, assuming appropriate regularization, rather than the sensitivity to any single hyperparameter choice. The MAE in each setting remains stable across different numbers of annealing steps. This indicates that there exists a limit of such corrections as controlled by M and it seems $M = 5 \sim 10$ would attain such a limit for this experiment. In particular, 4×4 and 8×8 attain lowest MAE of order 10^{-2} for 40% missing, while 16×16 attain lowest MAE of order 10^{-3}.



(a) 4×4



(b) 8×8



(c) 16×16

Figure 3.1: Comparison of generated, clean, and benchmark surfaces across dimensions.

In Figures 3.2, 3.3, and 3.4, there is a clear trend that the errors increase with the amount of missing values. This is expected since a lower missing proportion indicates more observed values to use in the sampling. In addition, in most cases, PINA give lower errors than the benchmark, indicating the improvements from the PI- and NA- correctors, except for 8×8 grid with 40% missing values. In Figures 3.2, 3.3, and 3.4, the gaps between the benchmark MAE (dotted lines) and the PINA MAE (solid lines) become more narrow as the missing percentage decreases from 80% to 40%. This shows that the improvement from PINA corrections depends on the amount of missing values. 80% missing information attains the greatest improvement from PINA, because more uncertainties are presented in the initial samples and therefore more guidance is needed for sampling. Also, finer grids attain lower errors, due to the reduction in discretization errors. Together, 16×16 with 40% missing values attained the lowest MAE (close to order of 10^{-3} in Figure 3.4) while 4×4 with 80% missing values attained the highest MAE (close to order of 10^{-1} in Figure 3.2).

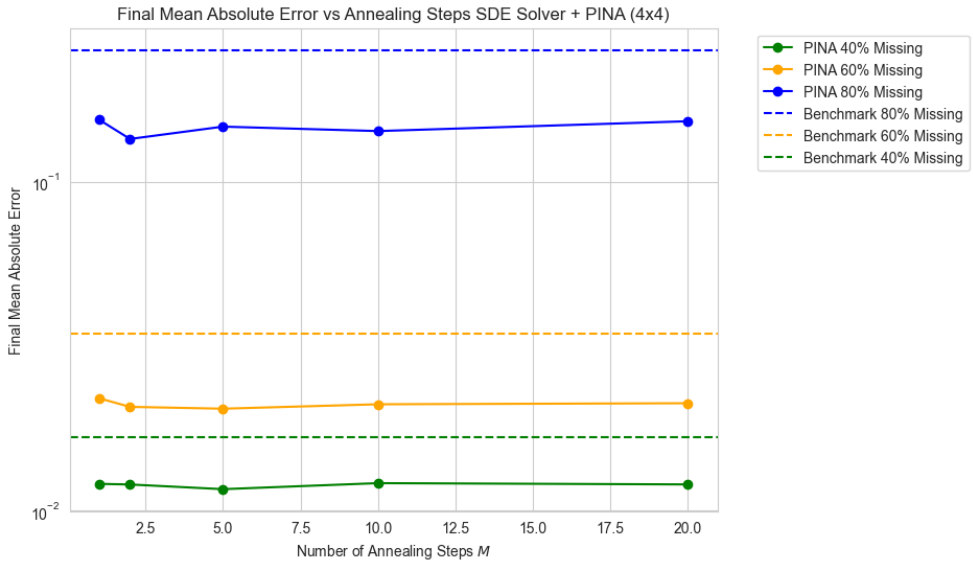


Figure 3.2: European: Final MAE for SDE Solver+PINA (4×4)

3.5.4 Performance w.r.t. Moneyness

The use of options is closely related to the moneyness. For European puts, they are more valuable in in-the-money region ($K > S$) since their prices are higher. They are

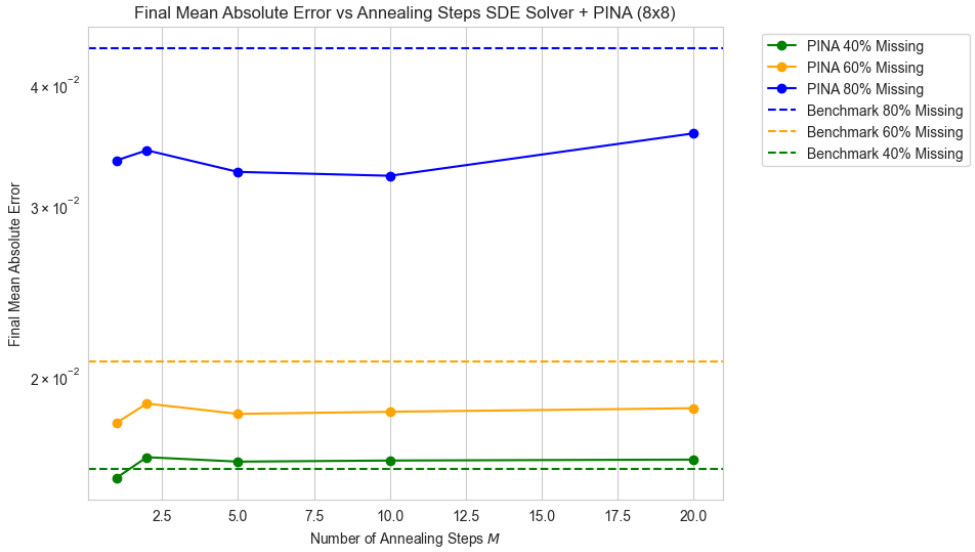


Figure 3.3: European: Final MAE for SDE Solver+PINA (8×8)

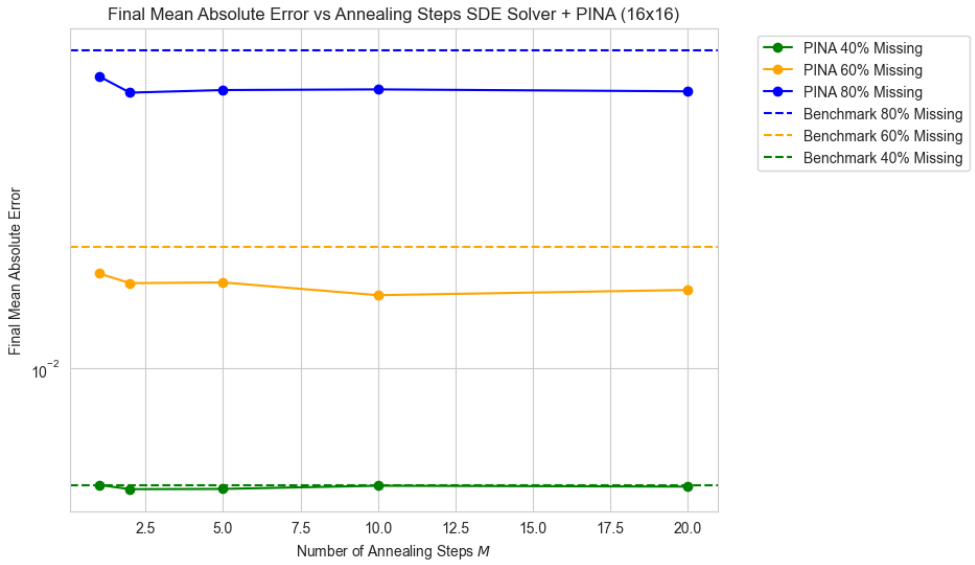


Figure 3.4: European: Final MAE for SDE Solver+PINA (16×16)

also valuable in the OTM and ATM regions for hedging as well. For a more detailed comparison, we consider five moneyness regions for the surface generations according to Table 3.2. We looked at both the mean absolute errors (MAE) and mean relative errors (MRE). The error metrics for European option price reconstruction across different levels of missing data are summarized in Figures 3.5, 3.6 and 3.7 corresponding to 4×4, 8×8 and 16×16 moneyness grids, respectively. Tables 3.3, 3.4, and 3.5 showed the respective cases numerically. For each missing data percentage (80%, 60%, and 40%), the tables report both the mean absolute error (MAE) and the mean relative error (MRE) for PINA and the benchmark method, separated by moneyness categories: deep OTM, OTM, ATM, ITM, and deep ITM (except for the 4×4 grid where only OTM and ITM are available due to the coarseness of the grid).

In Figures 3.5, 3.6 and 3.7, across all grids, PINA consistently outperforms the benchmark in terms of both MAE and MRE, except for MRE for 80% missing in Deep OTM regions. Examining Table 3.5, which presents the 16×16 grid, the MAE of PINA is consistently lower than the benchmark across almost all regions and missing-data levels. In particular, the Deep OTM region shows a decrease in MAE from 2.45×10^{-2} at 80% missing data to 8.47×10^{-4} at 40%, while the benchmark remains higher at 2.56×10^{-2} to 8.52×10^{-4}. Similar trends are observed in ITM and Deep ITM regions, where PINA achieves MAE as low as 2.23×10^{-3} and MRE around 5.69×10^{-5} at 40% missing data. These observations indicate that PINA is particularly effective at reconstructing larger option prices, where relative errors are inherently small. Conversely, MRE values in Deep OTM for high missing percentages are extremely large (e.g., 8.11×10^{7} at 80%), which is a direct consequence of the extremely small option prices in this region (as small as order of 10^{-13}); even minor absolute deviations are amplified when measured relative to such small values.

Looking at the 8×8 grid in Table 3.4, the same pattern emerges. Deep OTM MAE drops from 4.58×10^{-3} at 80% missing to 1.29×10^{-3} at 40%, with MRE similarly decreasing from 9.96×10^{2} to 1.37×10^{-3}. ATM and ITM regions show a steady reduction in MAE and MRE, highlighting that as option values increase toward the money, reconstruction becomes more accurate both in absolute and relative terms. Here, PINA consistently outperforms the benchmark, particularly in MAE, reflecting better fidelity to the underlying price surface.

In the coarsest 4×4 grid shown in Table 3.3, the differences between PINA and the benchmark are even more pronounced under high sparsity. For OTM options with 80% missing data, PINA achieves an MAE of 4.80×10^{-1} compared to the benchmark's 6.91×10^{-1}, while ITM options show a similar advantage (2.58×10^{-1} (PINA) against 3.40×10^{-1} (Benchmark)). The MREs reflect similar patterns: extremely high in OTM at 80% missing for both methods due to small option values, but PINA maintains lower relative errors than the benchmark for ITM options. This demonstrates that even on coarse grids, PINA is

able to reconstruct larger values accurately, whereas extremely small values still pose a challenge in relative terms.

Table 3.2: Strike (moneyness) Regions for Evaluation

Region	**Lower Bound**	**Upper Bound**
Deep OTM	60	80
OTM	80	95
ATM	95	105
ITM	105	120
Deep ITM	120	160

Overall, the phenomenon indicated by these tables is that PINA provides robust and consistent reconstruction across all moneyness regions and missing-data levels, with its advantages more pronounced for larger option values and finer grids. High relative errors in Deep OTM regions arise primarily from the small magnitude of option prices rather than poor model performance, while improvements in MAE and MRE in ATM and ITM regions reflect the model's ability to accurately capture the more substantial price variations. This could come from our current setting of smallest noise scale $\sigma_{min} = 10^{-3}$. Reducing this noise scale and using a finer noise scale could increase the model sensitivity and precision level, at a much higher computational cost. Finer grids, such as the 16×16 mesh, reduce discretization error and improve MAE, but can slightly increase MRE in OTM regions due to capturing more extremely small prices. These observations together suggest that both the absolute and relative performance of PINA is closely tied to the underlying option value distribution across moneyness.

Table 3.3: MAE and MRE for different missing data percentages and moneyness (4×4)

Missing Values	80%		60%		40%	
	PINA	Benchmark	PINA	Benchmark	PINA	Benchmark
OTM MAE	4.80e-1	6.91e-1	2.54e-2	3.53e-2	6.07e-3	8.39e-3
OTM MRE	3.54e+1	1.60e+0	1.71e+1	2.93e+1	1.46e+1	2.79e+1
ITM MAE	2.58e-1	3.40e-1	7.21e-3	1.48e-2	5.29e-3	9.07e-3
ITM MRE	1.02e-2	1.34e-2	2.70e-4	5.54e-4	1.98e-4	3.40e-4

Table 3.4: MAE and MRE for different missing data percentages and moneyness (8×8)

Missing Values	80%		60%		40%	
	PINA	Benchmark	PINA	Benchmark	PINA	Benchmark
DEEP OTM MAE	4.58e-3	4.49e-3	2.77e-3	2.69e-3	1.29e-3	1.39e-3
DEEP OTM MRE	9.96e+2	5.19e+2	1.55e+2	1.91e+2	1.37e-3	8.65e-1
OTM MAE	2.82e-2	3.72e-2	1.60e-2	1.71e-2	1.22e-2	1.24e-2
OTM MRE	1.02e-1	1.37e-1	7.02e-2	7.20e-2	6.85e-2	6.87e-2
ATM MAE	1.08e-1	1.53e-1	5.51e-2	6.44e-2	6.78e-3	6.23e-3
ATM MRE	2.26e-2	3.25e-2	1.07e-2	1.30e-2	1.39e-3	1.32e-3
ITM MAE	7.06e-2	1.01e-1	2.39e-2	2.92e-2	6.63e-3	7.05e-3
ITM MRE	4.31e-3	6.19e-3	1.43e-3	1.76e-3	4.00e-4	4.28e-4
DEEP ITM MAE	1.33e-2	1.51e-2	8.90e-3	9.21e-3	6.01e-3	6.16e-3
DEEP ITM MRE	3.80e-4	4.36e-4	2.44e-4	2.49e-4	1.66e-4	1.70e-4

Table 3.5: MAE and MRE for different missing data percentages and moneyness (16×16)

Missing Values	80%		60%		40%	
	PINA	Benchmark	PINA	Benchmark	PINA	Benchmark
DEEP OTM MAE	2.45e-2	2.56e-2	3.54e-3	4.41e-3	8.47e-4	8.52e-4
DEEP OTM MRE	8.11e+7	6.26e+7	8.67e+2	1.72e+5	1.37e+2	1.70e+5
OTM MAE	3.25e-2	3.64e-2	7.09e-3	8.81e-3	1.46e-3	1.54e-3
OTM MRE	4.00e+0	6.20e+0	1.01e+0	1.25e+0	3.10e-1	3.74e-1
ATM MAE	7.29e-2	8.83e-2	1.20e-2	1.55e-2	1.60e-3	1.70e-3
ATM MRE	2.31e-2	2.94e-2	4.21e-3	5.31e-3	5.23e-4	5.46e-4
ITM MAE	6.75e-2	8.97e-2	1.23e-2	1.61e-2	1.54e-3	1.45e-3
ITM MRE	7.81e-3	1.05e-2	1.48e-3	1.92e-3	1.77e-4	1.63e-4
DEEP ITM MAE	3.30e-2	4.86e-2	8.19e-3	1.18e-2	2.23e-3	2.26e-3
DEEP ITM MRE	9.94e-4	1.48e-3	2.17e-4	3.10e-4	5.69e-5	5.76e-5

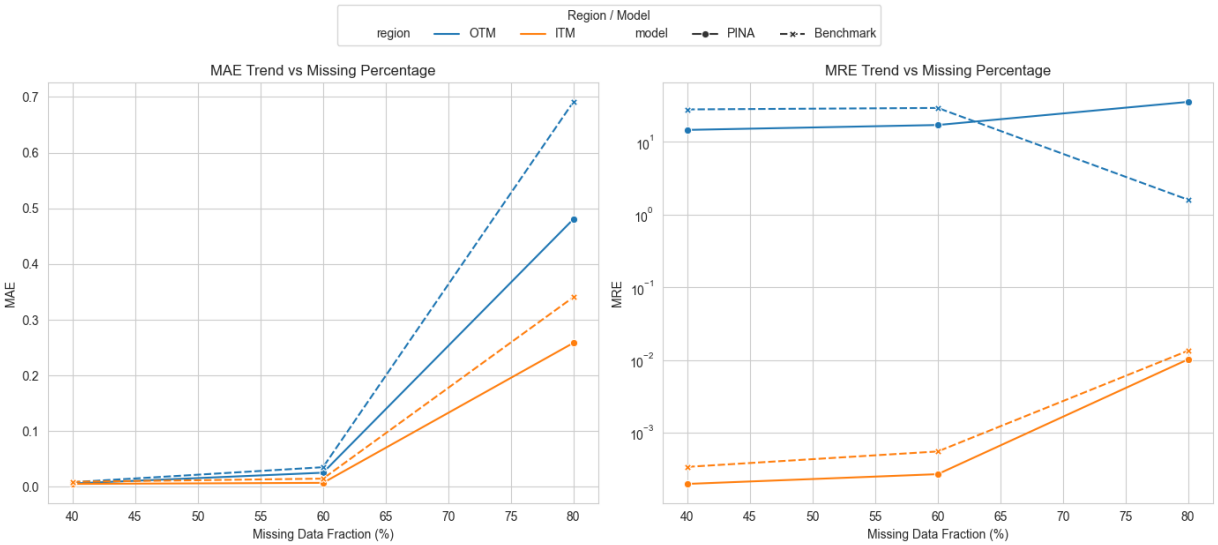


Figure 3.5: European Options: MAE and MRE by Moneyness (4×4)

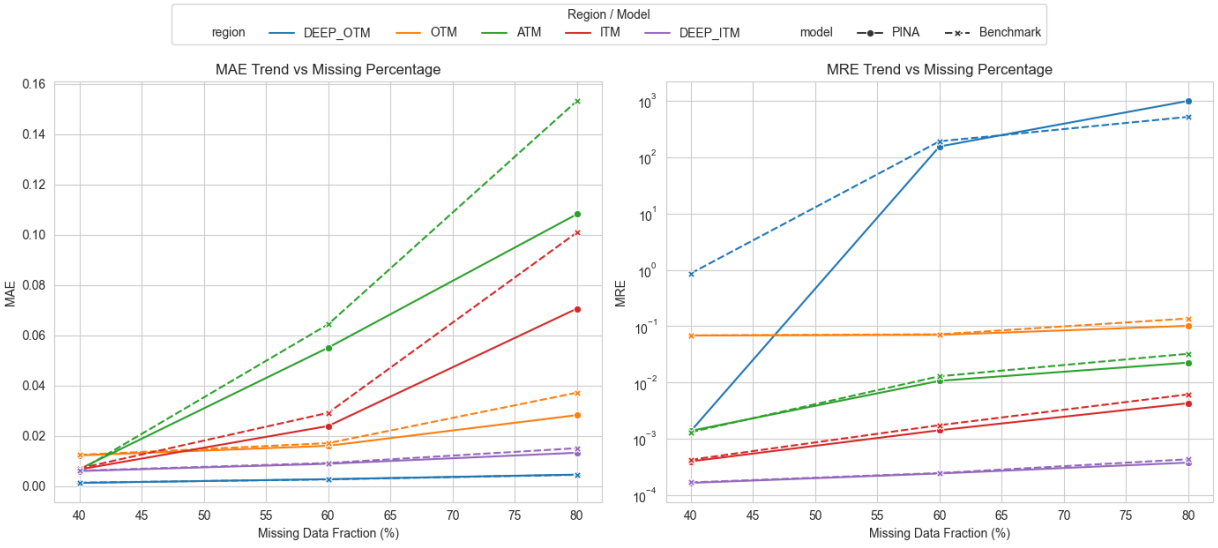


Figure 3.6: European Options: MAE and MRE by Moneyness (8×8)

Weight of NA corrector λ_{NA}

The weight λ_{NA} should not be too large. We found that using values greater than 0.1 may result in unrealistic and nonsmooth surfaces. It is expected that as the percentage of missing values decreases from 80% to 20%, the arbitrage losses reduce in general. We tested different combinations of the weights from $\lambda_{NA} \in \{10^{-1}, 10^{-2}, 10^{-3}\}$ for each mesh. Figures 3.8, 3.9 and 3.10 show the arbitrage losses for 16×16 with different missing percentages.

Here we focus on 16×16 because it bears the lowest discretization errors from the finite difference approximation for the derivatives. The general observation here is that NA-corrector consistently reduces arbitrage losses, with the largest improvement from $\lambda_{NA} = 10^{-1}$. Figure 3.11 shows the MAE comparison for the corresponding choice of λ_{NA} with different number of annealing steps. For $60\% - 80\%$ missing, $\lambda_{NA} = 10^{-1}$ improved the most in terms of arbitrage losses and MAE. However, $\lambda_{NA} = 10^{-1}$ increases the MAE for $M > 5$, even worse than the benchmark. This indicates that as the percentage of missing values reduces from 80% to 20%, the optimal choice for λ_{NA} reduces as well. Indeed, for a small stepsize in arbitrage correction, if there are inaccurate steps in ν_{NA}, small steps can be absorbed into the stochastic component in the SDE component. This means that the inaccurate correction can be corrected by the predictor as well. However, for a large stepsize λ_{NA} in combinations with a large number of annealing steps M, the error can accumulate and become irreversible.

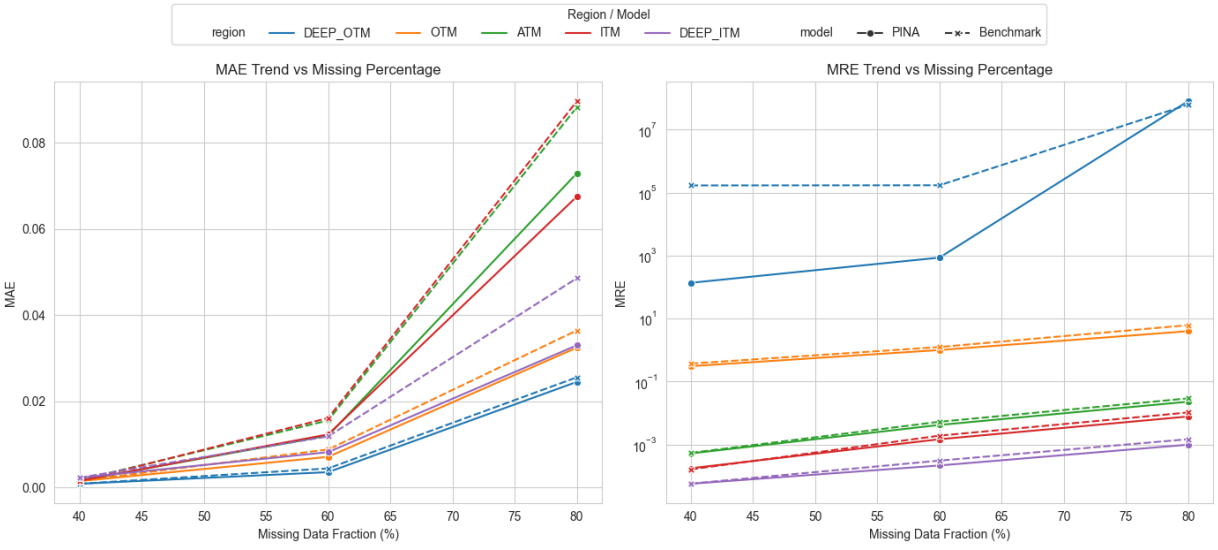


Figure 3.7: European Options: MAE and MRE by Moneyness (16×16)

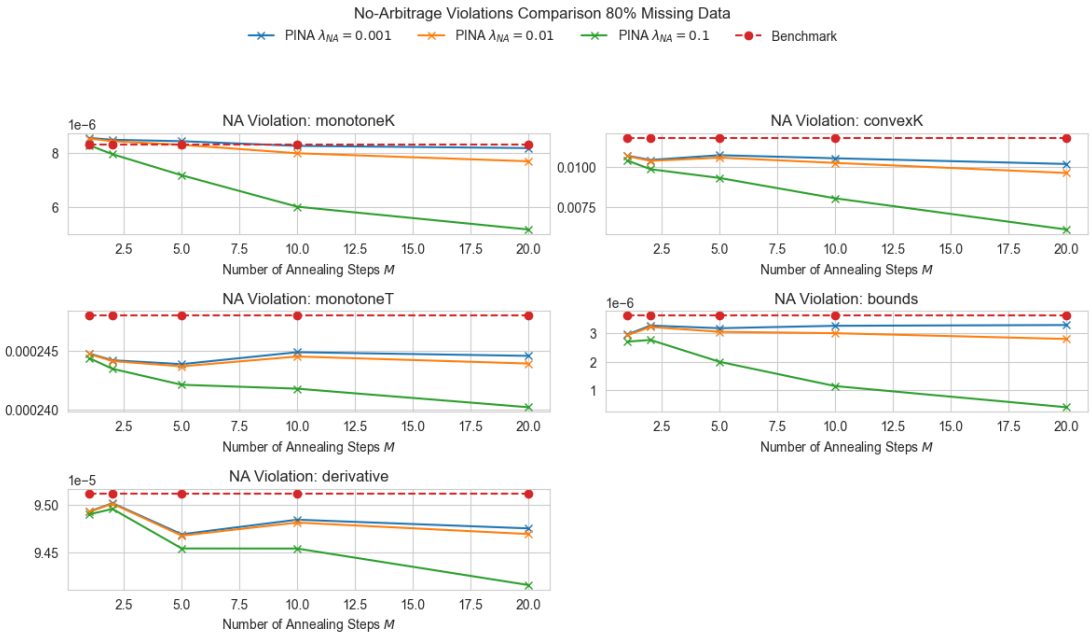


Figure 3.8: European Options: Arbitrage Losses (16×16, 80% Missing)

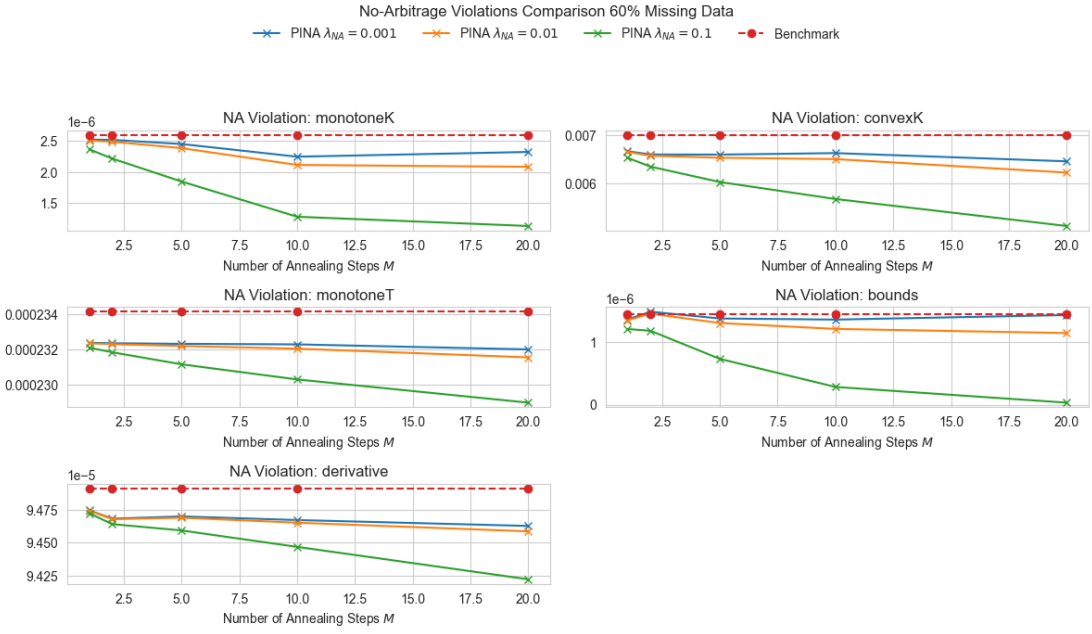


Figure 3.9: European Options: Arbitrage Losses (16×16, 60% Missing)

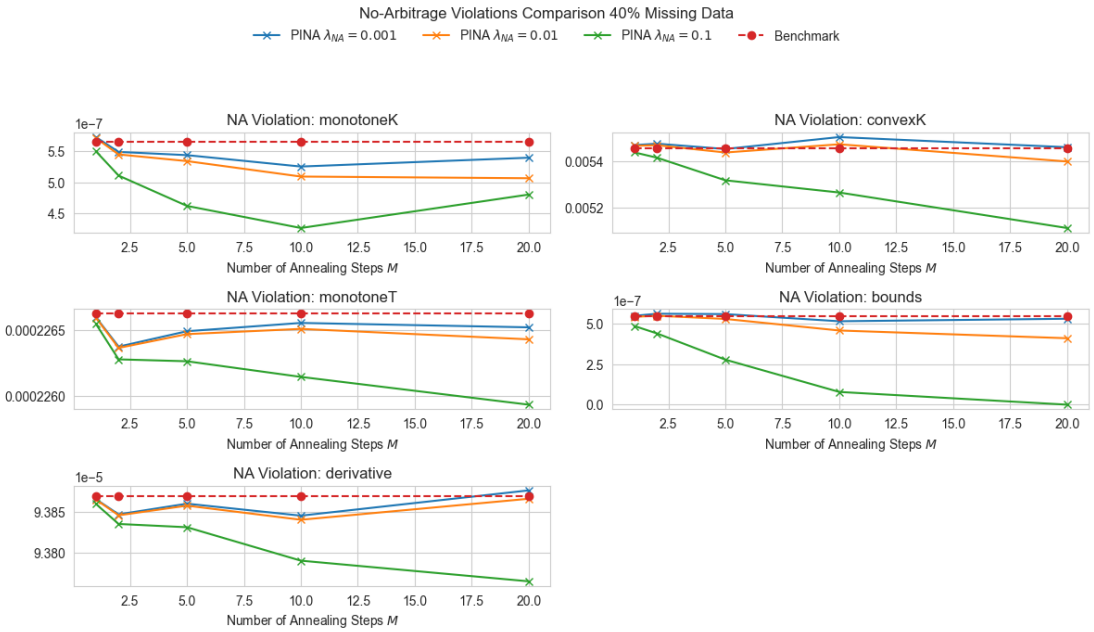


Figure 3.10: European Options: Arbitrage Losses (16×16, 40% Missing)

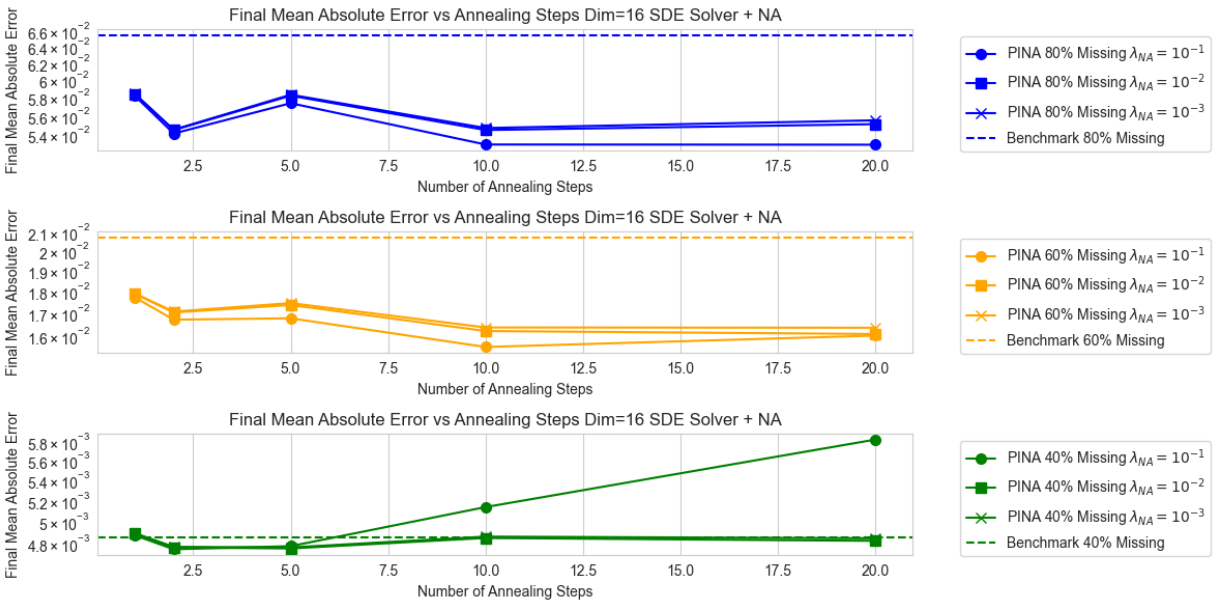


Figure 3.11: European Options: MAE for 16×16 with varying λ_{NA} and $\lambda_{PI} = 0$

Weight of PI corrector λ_{PI}

For λ_{PI}, we also selected from $\lambda_{PI} \in \{10^{-3}, 10^{-2}, 10^{-1}\}$. The choice of λ_{PI} depends on both the percentages of missing values and the number of annealing steps M. Figures 3.12 and 3.13 show the final MAE and MRE for different choices of λ_{PI} for 16×16 with different numbers of annealing steps and missing percentages respectively. Overall, $\lambda_{PI} = 10^{-3}$ gives the most consistently lower MAE. Compared to the NA-corrector, PI-corrector needs much smaller stepsize λ_{PI}. In Figure 3.12, both $\lambda_{PI} \in \{10^{-2}, 10^{-1}\}$ give higher MAE than the benchmark, while $\lambda_{PI} = 10^{-3}$ only gives slightly smaller MAE. As the number of steps M, the gaps between different λ_{PI} widens, favoring smaller values. However, in Figure 3.13, $\lambda_{PI} = 10^{-1}$ gives lowest MRE for $60\% - 80\%$ missing, while smaller $\lambda_{PI} = 10^{-3}$ gives consistently lower MRE than benchmark for 40% missing. Recall in Figure 3.7, MAE is mostly influenced by large values (such as option prices in ITM and ATM) while MRE is mostly influenced by small values (especially prices in Deep OTM). This indicates that, compared to the benchmark, the PI-corrector improves small prices and overcorrects the large prices. This indicates that there is a need to work on the normalization of PI-correctors. The reason is that the current design of the PI-corrector is based on a single quantity—the PDE residual. However, the actual magnitude of correction for each grid point is uniformly scaled by λ_{PI} after normalization, according to (3.22). This uniform scaling does not account for the varying sensitivity of the option price surface across different regions.

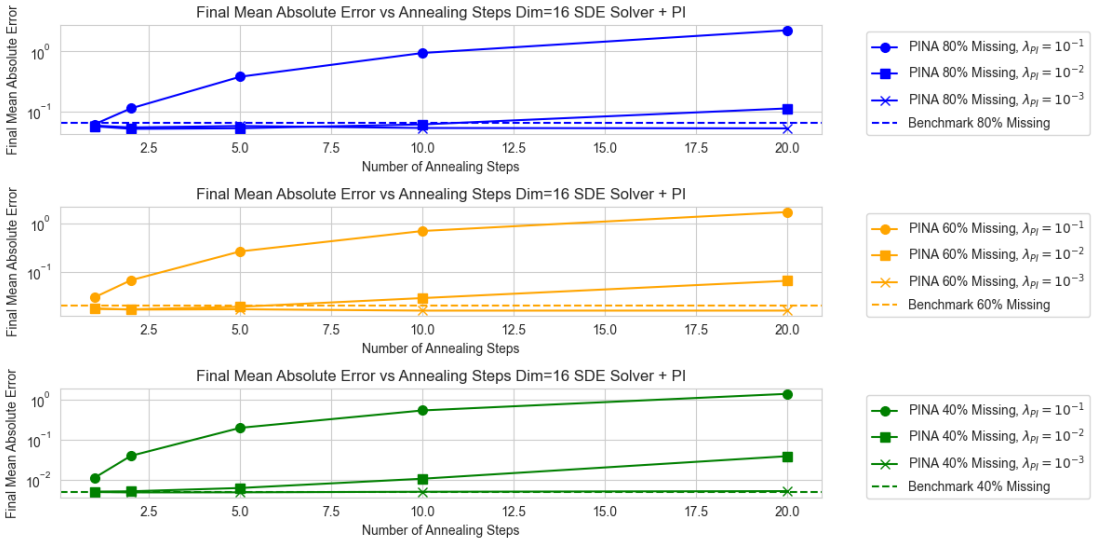


Figure 3.12: European Options: MAE for varying λ_{PI} and $\lambda_{NA} = 0$ (16×16)

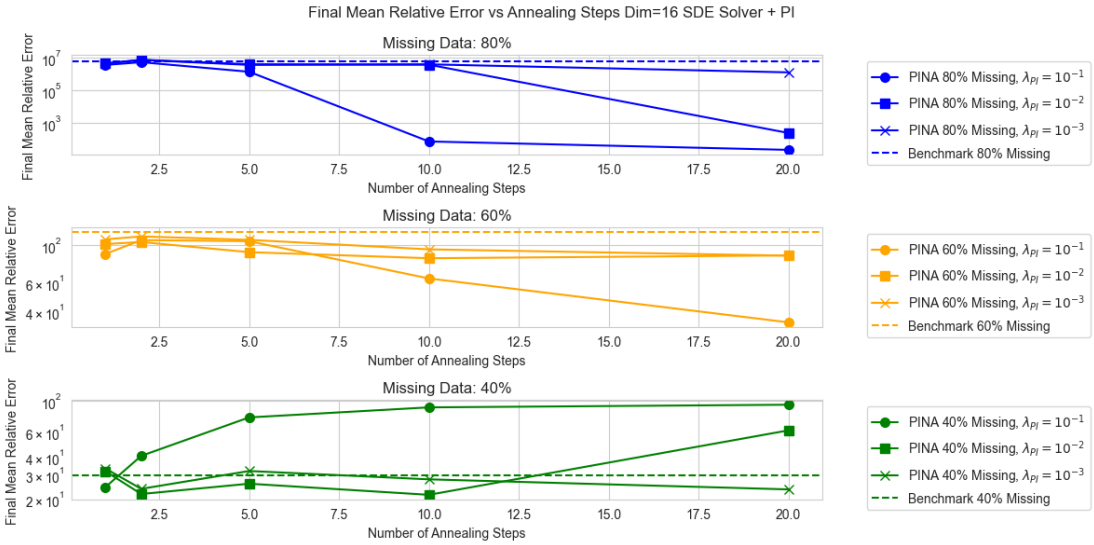


Figure 3.13: European Options: MRE for varying λ_{PI} and $\lambda_{NA} = 0$ (16×16)

Chapter 4

Price Surface Completion for American Put Options

In this chapter, we focus on American Put option price surfaces completion. Unlike European options, the arbitrage-free conditions for American options are different, and must consider the central property of early exercise. In addition, the original framework of using physics-informed and no-arbitrage correctors is not compatible directly with American options, because there does not exist a corresponding PDE describing the dynamics of American puts in the (K, T) space. Indeed, most discussion of American options center around the early exercise property, which mainly concerns the intermediate space (t, S) instead of the (K, T). Therefore, we mainly focus on using no-arbitrage corrector to the SDE-type predictor to complete American put price surfaces. The structure of this chapter therefore follows: we will first discuss the no-arbitrage conditions for American put options and arbitrage losses in section 4.3. Then, the main algorithms we tested for American put completion using the no-arbitrage correctors are described. The numerical results are shown in section at last.

4.1 Pricing American Options

Pricing American options is complex, not only because we need to consider every point until maturity, but also every potential price movement, which is unknown. One can think of it as a grocery store. Customers only can see the price tag and the production label but they probably know nothing about the production pipeline. American options are very

much like the product, tagged with strike K, maturity T and the current spot price S_0; but its actual value would depend on the production cost as well.

Therefore, as we assume Black-Scholes model, we are assuming the production process in order to estimate the option price. For European options, things are easier because one just needs to consider one time point T to exercise the option. In the case of American options, suppose we are at time $0 < t < T$ with an unknown spot price S_t. If S_t is very small, a put option holder will very likely to exercise the option to attain instant huge profit; on the other hand, if S_t is very large, they may wait for a better chance. Between these two spot prices, there has to be a striking point that most affects this decision. This spot price is called the *free boundary*, denoted by $S_f(t)$. Note that this boundary is dependent of t. It is the boundary that separates the two regions and it is from this boundary condition that differentiates American options from European counterparts. It is the critical stock price below which immediate exercise is better than waiting.

For the following, we assume Black-Scholes Model for convenience. Let S_t denote the price of the underlying asset under the risk-neutral measure $\mathbb{Q}$. We assume the dynamics follow a geometric Brownian motion:

$$dS_t = rS_t\,dt + \sigma S_t\,dW_t, \tag{4.1}$$

where r is the risk-free rate, $\sigma > 0$ is the volatility, and W_t is a Brownian motion. For an American put option with payoff $\Psi(S) \coloneqq (K - S)_+$, its arbitrage-free price is given by the discounted expectation

$$V_t = \mathbb{E}^{\mathbb{Q}}\left[e^{-r(T-t)}\Psi(S_T) \,\middle|\, \mathcal{F}_t\right], \tag{4.2}$$

where $\mathcal{F}_t$ is the natural filtration of W_t.

4.1.1 Early Exercise and Optimal Stopping

For an American put option, exercising at time t yields the payoff

$$\Psi(S) = (K - S)_+. \tag{4.3}$$

Since the holder may exercise at any stopping time $\tau \in [0, T]$, the arbitrage-free price satisfies the optimal stopping problem

$$V(t, S) = \sup_{\tau \in [0,T]} \mathbb{E}^{\mathbb{Q}}\left[e^{-r(\tau-t)}\Psi(S_\tau) \,\middle|\, S_t = S\right]. \tag{4.4}$$

This induces:

- a *continuation region*, where it is optimal to hold the option and $V(t,S) > \Psi(S)$;
- an *exercise region*, where immediate exercise is optimal and $V(t,S) = \Psi(S)$.

The boundary between these regions is a time-dependent free boundary $S_f(t)$.

4.1.2 Connection to the Black-Scholes PDE

In the continuation region, the discounted process $e^{-rt}V(t,S_t)$ must be a martingale. Applying Itô's formula yields the Black–Scholes PDE:

$$\partial_t V + \frac{1}{2}\sigma^2 S^2 \partial_{SS} V + rS\partial_S V - rV = 0, \qquad \text{for } S > S_f(t). \tag{4.5}$$

The terminal condition is

$$V(t,S) = \Psi(S). \tag{4.6}$$

But the actual value of the option is different from European put because of the free boundary condition.

4.1.3 Variational Inequality and the LCP Formulation

For convenience, we first define the Black-Scholes operator

$$\mathcal{L}V := \partial_t V + \frac{1}{2}\sigma^2 S^2 \partial_{SS} V + rS\partial_S V - rV. \tag{4.7}$$

The early-exercise constraint requires that the option value dominates its payoff:

$$V(t,S) \geq \Psi(S). \tag{4.8}$$

Continuation vs. Exercise. If at (t,S) the holder chooses *not* to exercise, then the option behaves locally like a European option, and therefore the Black-Scholes PDE must hold:

$$\mathcal{L}V(t,S) = 0 \qquad \text{(continuation region)}. \tag{4.9}$$

If the holder *does* exercise immediately, then

$$V(t,S) = \Psi(S),$$

and the differential operator applied to V need not vanish. In fact, substituting the payoff into the operator yields

$$\mathcal{L}(K - S) = -r(K - S) - rS = -rK < 0,$$

so the early exercise corresponds to

$$\mathcal{L}V(t, S) < 0 \qquad \text{(exercise region)}. \tag{4.10}$$

Complementarity Condition The option holder must be in *one and only one* of the two regimes:

$$\text{either } \; V > \Psi(S) \quad \text{and} \quad \mathcal{L}V = 0, \qquad \text{or } \; V = \Psi(S) \quad \text{and} \quad \mathcal{L}V < 0.$$

That means either of the equality holds, depending on the price S. These mutually exclusive conditions can be written compactly as the *complementarity* condition

$$\big(V - \Psi(S)\big) \cdot \mathcal{L}V = 0. \tag{4.11}$$

The Linear Complementarity Problem. Collecting the inequalities and the complementarity condition yields the Linear Complementarity Problem (LCP):

$$\begin{cases} \mathcal{L}V \le 0, \\ V \ge \Psi(S), & 0 \le t < T, \; S > 0, \\ \big(V - \Psi(S)\big) \cdot \mathcal{L}V = 0, \end{cases} \tag{4.12}$$

with terminal condition $V(T, S) = \Psi(S)$. This system expresses that the option value must stay above the exercise payoff, must satisfy the Black-Scholes PDE whenever the option is held, and must violate the PDE precisely when early exercise is optimal. Equation (4.12) is the standard variational inequality formulation. To solve this, traditional numerical methods include finite-difference [40], penalty [31], and projected-SOR numerical methods [2]. However, these numerical methods work on each (K, T) individually. In our project, we consider all (K, T) altogether by using SBGMs. To demonstrate this in surface completion, we now formulate the problem.

4.2 Problem Formulation

Just like in European case, we assume no dividends here. Suppose we are given a grid of maturities and strikes $\mathcal{G} := \{(K_i, T_j) \colon i = 1, \cdots, n \text{ and } j = 1, \cdots, m\}$. Further suppose we are given observed American Put option prices $P_A(K_i, T_j)$ in some subset of grid points $\mathcal{G}_{obs} := \{(K_i, T_j) \in \mathcal{G} \colon P_A(K_i, T_j) \text{ is known}\}$. Our goal is to estimate the American put prices at the remaining grid points $\mathcal{G} \backslash \mathcal{G}_{obs}$, such that the whole surface is arbitrage-free. For convenience, we assume same number of strikes and maturities, $n = m$.

4.3 No Arbitrage Conditions

We first restate the arbitrage-free conditions for American put options from the Chapter 2 for more detailed discussion. In the same principle of constructing no-arbitrage correctors in the European case, we will translate those arbitrage-free conditions into arbitrage losses. More precisely, when given a price surface $\mathbf{P_E} := [P_E(K_i, T_j)]_{i,j=1}^n$ and $\mathbf{P_A} := [P_A(K_i, T_j)]_{i,j=1}^n$, we define the corresponding losses, by computing the amount of violations.

Theorem 4.3.1 (No-arbitrage Conditions for American Put Options) *[7] Suppose the prices of American Put options P_A with maturity T are given for finitely many strikes $K_1, \cdots, K_n$, and that P_A is linearly interpolated between these strikes. Assume that the corresponding European Put prices P_E satisfy the no-arbitrage conditions in Theorem 2.1.1. Then the American Put price function $P_A(K, T)$ is arbitrage-free if and only if:*

1. *$P_A(K, T)$ is increasing in K.*
2. *$P_A(K, T)$ is convex in K.*
3. *$\partial_K P_A(K+, T)K - P_A(K, T) \geq \partial_K P_E(K+, T)K - P_E(K, T)$ for all $K \geq 0$.*
4. *$\max\{(K - S_0)_+, P_E(K, T)\} \leq P_A(K, T) \leq P_E(e^{rT}K, T)$.*

Here $\partial_K P_E(K+, T)$ and $\partial_K P_A(K+, T)$ are the derivatives in K of P_E and P_A from the positive side respectively. For calendar arbitrage, we additionally require:

5. *$P_A(K, T) - P_E(K, T)$ is non-decreasing in $T > 0$.*

The idea of deducing these conditions is about assuming their violations and developing a corresponding portfolio with nonnegative initial profit and arriving at nonnegative final output, thus creating arbitrage opportunities. The central tool that differentiates this from European options is the freedom of selecting when to exercise the American options in long position that allows arbitrage. Their deductions can be found in [7]. In the following, we will design arbitrage losses and the corresponding correctors. For each no-arbitrage condition c, we define a corresponding violation matrix $\mathbf{H}_c \in \mathbb{R}^{n\times n}$ whose (i,j)-th entry measures the pointwise violation of condition c at $(K_i, T_j) \in \mathcal{G}$ via a hinge loss.

Monotonicity in K Condition (1) requires the American put price surface P_A to satisfy

$$\partial_K P_A(K,T) \geq 0 \quad \text{for all } (K,T) \in \mathcal{G}.$$

To enforce this inequality during sampling, we penalize its pointwise violations. Whenever $\partial_K P_A(K,T) < 0$, the monotonicity condition is violated, and the magnitude of this violation is measured by the hinge loss $(-\partial_K P_A(K,T))_+$. Formally, we define the matrix $\mathbf{H}_{mono(K)} \in \mathbb{R}^{n\times n}$ by

$$\left[\mathbf{H}_{mono(K)}\right]_{ij} \coloneqq \left(-\partial_K P_A(K_i,T_j)\right)_+, \quad \text{for } i,j = 1,\ldots,n.$$

Averaging these penalties over the grid $\mathcal{G}$ yields the monotonicity loss

$$\ell_{\text{mono}(K)} \coloneqq \frac{1}{n^2}\sum_{i,j}\left[\mathbf{H}_{mono(K)}\right]_{ij}. \tag{4.13}$$

The remaining arbitrage conditions are enforced analogously by defining constraint-specific violation matrices and averaging their hinge penalties over the grid $\mathcal{G}$.

Convexity in K Condition (2) requires convexity of P_A in the strike, which is equivalent to

$$\partial^2_{KK} P_A(K,T) \geq 0 \quad \text{for all } (K,T) \in \mathcal{G}.$$

Negative second derivatives correspond to violations of convexity. Following the construction above, these violations are penalized using the hinge loss,

$$[\mathbf{H}_{\text{conv}}]_{ij} \coloneqq \left(-\partial^2_{KK} P_A(K_i,T_j)\right)_+, \text{ for } i,j = 1,\ldots,n,$$

leading to the convexity loss

$$\ell_{conv}(P) \coloneqq \frac{1}{n^2}\sum_{i,j}[\mathbf{H}_{\text{conv}}]_{ij}. \tag{4.14}$$

Derivative bound for P_A Condition (3) concerns the early-exercise feature of American options and constrains the rate at which the early-exercise premium increases with the strike. This condition is equivalent to the pointwise inequality

$$\partial_K P_A(K,T)\,K - P_A(K,T) \;\geq\; \partial_K P_E(K,T)\,K - P_E(K,T), \quad (K,T) \in \mathcal{G}.$$

Violations of this inequality indicate that the marginal benefit of early exercise fails to increase sufficiently with the strike. Following the same construction as above, we define the corresponding violation matrix $\mathbf{H}_{\text{deri}} \in \mathbb{R}^{n\times n}$ by

$$[\mathbf{H}_{\text{deri}}]_{ij} \coloneqq \Big(\partial_K P_E(K_i,T_j)K_i - P_E(K_i,T_j) - \partial_K P_A(K_i,T_j)K_i + P_A(K_i,T_j)\Big)_+, \quad i,j = 1,\dots,n.$$

Averaging its entries over the grid yields the derivative loss

$$\ell_{\text{deri}} \coloneqq \frac{1}{n^2}\sum_{i,j} [\mathbf{H}_{\text{deri}}]_{ij}\,. \tag{4.15}$$

Bounds for P_A Condition (4) imposes both an upper and a lower bound on the American put price, namely,

$$\max\left\{P_E(e^{rT}K,T), (K-S_0)_+\right\} \;\leq\; P_A(K,T) \;\leq\; e^{-rT}K, \quad (K,T) \in \mathcal{G}.$$

Each bound is enforced by penalizing its pointwise violations. Although Condition (4) consists of two separate inequalities, each grid point would violate at most one of the bounds. We therefore enforce the bounds by penalizing the pointwise violations of each inequality and summing their magnitudes into a single scalar loss. We define the corresponding violation matrix $\mathbf{H}_{\text{bound}} \in \mathbb{R}^{n\times n}$ by

$$\begin{aligned}[\mathbf{H}_{\text{bound}}]_{ij} \coloneqq\; & \big(P_A(K_i,T_j) - e^{-rT_j}K_i\big)_+ \\ & + \big(\max\{P_E(e^{rT_j}K_i,T_j), (K_i-S_0)_+\} - P_A(K_i,T_j)\big)_+, \quad i,j = 1,\dots,n.\end{aligned}$$

The bound loss is then given by

$$\ell_{\text{bound}} \coloneqq \frac{1}{n^2}\sum_{i,j} [\mathbf{H}_{\text{bound}}]_{ij}\,. \tag{4.16}$$

Monotonicity in T Condition (5) rules out calendar arbitrage and requires the American put price to be nondecreasing in maturity. Since the European put surface is assumed to be arbitrage-free, this condition is equivalent to

$$\partial_T P_A(K,T) \geq \partial_T P_E(K,T), \quad (K,T) \in \mathcal{G}.$$

Violations correspond to situations in which extending maturity reduces the value of the American option. We define the associated violation matrix $\mathbf{H}_{\text{mono}(T)} \in \mathbb{R}^{n\times n}$ by

$$\left[\mathbf{H}_{\text{mono}(T)}\right]_{ij} := \left(\partial_T P_E(K_i,T_j) - \partial_T P_A(K_i,T_j)\right)_+, \quad i,j = 1,\dots,n.$$

The monotonicity-in-T loss is given by

$$\ell_{\text{mono}(T)} := \frac{1}{n^2}\sum_{i,j}\left[\mathbf{H}_{\text{mono}(T)}\right]_{ij}. \tag{4.17}$$

The numerical approximation of all partial derivatives follows exactly the procedure described in the previous section. The collection of violation matrices $\{\mathbf{H}_c\}_{c\in\mathcal{C}}$ provides a unified representation of arbitrage violations across all conditions, which is subsequently used to construct the no-arbitrage corrector ν_{NA} in (3.16) as in Section 3.4.1. More explicitly, let $\ell(x)$ denote the arbitrage loss to reduce ,and $\mathbf{H} \in \mathbb{R}^{n\times n}$ be the corresponding violation matrix as defined above. Fix a number of trials n_{trial} and a small positive constant $\widetilde{\epsilon}$. For each trial $i = 1,\cdots,n_{trial}$, sample a random perturbation direction $\Delta_i \in \mathbb{R}^{n\times n}$ and approximate the gradient by

$$\nu_{NA}^{(i)}(x) := -\frac{\ell(x+\widetilde{\epsilon}\Delta_i) - \ell(x-\widetilde{\epsilon}\Delta_i)}{2\widetilde{\epsilon}}\Delta_i.$$

with the Rademacher matrix Δ_i. Averaging over trials gives the final corrector direction:

$$\widetilde{\nu}_{NA}(x) := \frac{1}{n_{trial}}\sum_{i=1}^{n_{trial}}\nu_{NA}^{(i)}(x). \tag{4.18}$$

To addition, introduce a normalized weight matrix using the violation matrix $\mathbf{H}$:

$$W(x) = \frac{\mathbf{H}}{\|\mathbf{H}\|_2 + \delta}, \tag{4.19}$$

where $\delta = 10^{-8}$ prevents division by zero. Finally, update the sample using a weight parameter $\lambda_{NA} > 0$:

$$x \leftarrow x + \lambda_{NA}\,\nu_{NA}(x). \tag{4.20}$$

Here, $\nu_{NA}(x) := \widetilde{\nu}_{NA}(x)_{NA} \odot W(x)$ and $\odot$ denotes elementwise multiplication, which emphasizes corrections where violations are largest while keeping the overall step size controlled.

4.4 Physics-informed Correctors

For the physics informed corrector for American puts, we intended to seek any PDEs or dynamical system that provide geometric structures to the price surface for our use. However, our discussion on comparing continuation and exercise values revealed that the pricing of American options lies on the (t, S) space, which is the space of the pricing dynamics. In fact, most of the corresponding PDEs, including the Black-Scholes (2.2), also lie in the (t, S) space. In addition, the solution to the PDE, the option price, has to consider both the continuation and exercise values. While the Black-Scholes prices concerns the discounted future payoffs of the options, we need to consider the exercise value at every time step. Unfortunately, unlike European options, for which there exists the Dupire's Equation (3.18) for the (K, T) space, there does not exist a corresponding equation for American options in the (K, T) space. As such, for American put options, we mainly focus on using no-arbitrage correctors to improve price surface completion.

4.5 Methodology

4.5.1 Algorithms

For this section, we will describe the main algorithms we tested for American put option completion. From the discussion in the European option in the previous chapter, we have seen improvements from PINA correctors. However, for American options, since we did not design a PI-corrector, we look for an alternative corrector to improve the accuracy. From [39], they discussed the original setting for the predictor-corrector scheme, in which they used modified Annealed Langevin Dynamics (ALD) as a corrector, complementing the predictor (either SDE solver or probability flow). The SDE+ALD algorithm is stated in Algorithm 6. The main idea of ALD is to gradually refine the intermediate sample according to the score network to prevent slow mixing. They showed that the ALD corrector, with an appropriate number of corrector steps, outperformed the predictor-only method [39]. Therefore, we explored the case where we combined both no-arbitrage corrector and ALD corrector, to refine the samples from the SDE solver.

Similar to the European case, we use a soft constraint for the observed values: define

$$\omega_{obs}(t) := e^{-\frac{3(L-t)}{L}}, \tag{4.21}$$

as the scaling factor at noise level t, so that this begins with 1 at $t = L$, and monotonically decreases to e^{-3} at $t = 1$. Recall that $\Xi \in \{0, 1\}^{n \times n}$ is the indicator mask defined by

$$\Xi_{ij} := \begin{cases} 1 & \text{if } (T_i, K_j) \in \mathcal{G}_{obs} \\ 0 & \text{otherwise.} \end{cases}$$

At the end of each corrector step,

$$x_i \leftarrow \omega_{obs} x_i \odot (1 - \Xi) + (1 - \omega_{obs}) x_{obs} \odot \Xi, \tag{4.22}$$

so that the sample will gradual move towards the observed values. We only enforce the observed values in the final step of the whole algorithm. The whole algorithm is shown in Algorithm 7.

Algorithm 6: SDE Solver + Annealed Langevin Dynamics (ALD)

Data: sequence of noise levels $\{\varphi_i\}_{i=1}^{L}$, number of correction steps per noise level M, step size parameter ϵ, trained noise conditional score network $s_\theta(\cdot, \cdot)$, weight for NA corrector λ_{NA}

Initialize $x_L \sim \mathcal{N}(\varphi_L^2 I_d)$;
for $i = L - 1, \cdots, 1$ **do**
  Predictor Step
    Define the step size $\alpha_i \leftarrow \epsilon \frac{\varphi_i^2}{\varphi_L^2}$;
    Sample $z_i \sim \mathcal{N}(0, I_d)$;
    Update the new sample via discretized reverse diffusion in Eq.(2.16):
    $x_i \leftarrow x_{i+1} + (\varphi_{i+1}^2 - \varphi_i^2) s_\theta(x_{i+1}, \varphi_{i+1}) + \sqrt{\varphi_{i+1}^2 - \varphi_i^2} z_i$;
  Corrector Step
    for $t = 1, \cdots, M$ **do**
      Sample $z_t \sim \mathcal{N}(0, I_d)$;
      Define the proposed update $x_{tmp} = x_i + \alpha_i s_\theta(x_i, \varphi_i) + \sqrt{2\alpha_i} z_t$;
      $x_i \leftarrow x_{tmp}$;
  `/* Soft constraint on the observed values */`
  Define ω_{obs} by Eq.(4.21).;
  $x_i \leftarrow \omega_{obs} x_i \odot (1 - \Xi) + (1 - \omega_{obs}) x_{obs} \odot \Xi$, where $\odot$ is element-wise multiplication.;

Algorithm 7: SDE Solver + ALD + NA

Data: sequence of noise levels $\{\varphi_i\}_{i=1}^{L}$, number of correction steps per noise level M, step size parameter ϵ, trained noise conditional score network $s_\theta(\cdot,\cdot)$, weight for NA corrector λ_{NA}

Initialize $x_L \sim \mathcal{N}(\varphi_L^2 I_d)$;
for $i = L-1, \cdots, 1$ **do**
  Predictor Step
    Define the step size $\alpha_i \leftarrow \epsilon \frac{\varphi_i^2}{\varphi_L^2}$;
    Sample $z_i \sim \mathcal{N}(0, I_d)$;
    Update the new sample via discretized reverse diffusion in Eq. (2.16):
     $x_i \leftarrow x_{i+1} + (\varphi_{i+1}^2 - \varphi_i^2) s_\theta(x_{i+1}, \varphi_{i+1}) + \sqrt{\varphi_{i+1}^2 - \varphi_i^2} z_i$;
  Corrector Step
    for $t = 1, \cdots, M$ **do**
      Sample $z_t \sim \mathcal{N}(0, I_d)$;
      Define the proposed update $x_{tmp} = x_i + \alpha_i s_\theta(x_i, \varphi_i) + \sqrt{2\alpha_i} z_t$;
      NA Corrector
        Set $\nu_{NA}(x_{tmp})$ via Eq. (4.20) and weight ω_{NA} by equation (4.19);
        Update using this NA corrector $x_{tmp} \leftarrow x_{tmp} + \omega_{NA}\lambda_{NA}\nu_{NA}$;
    $x_i \leftarrow x_{tmp}$;
  `/* Soft constraint on the observed values */`
  Define ω_{obs} by Eq. (4.21).;
  $x_i \leftarrow \omega_{obs} x_i \odot (1 - \Xi) + (1 - \omega_{obs}) x_{obs} \odot \Xi$, where $\odot$ is element-wise multiplication.;

4.5.2 Experiment Setting

The only two parameters we relaxed are σ and r. We sampled uniformly $(\sigma, r) \in [0.05, 0.3] \times [0.1, 0.3]$ and consider a uniform grid on (K, T) where $K \in [80, 120]$ and $T \in [0.1, 2.0]$. We generated 100K American put price surfaces on 4×4, 8×8 and 16×16 grids, using finite difference methods. In particular, we used the Crank-Nicolson scheme to solve the Black-Scholes equation, followed by PSOR algorithm to handle the early exercise condition. Afterwards, we trained a score network using RefineNet and applied reverse diffusion. In particular, we masked 20%, 40%, 60% and 80% of the pixels as missing data. We employed the Predictor-Corrector Scheme, using the discretized SDE solver as the predictor, with

ALD and NA correctors as the corrector, and the latter is weighted by the hyperparameter λ_{NA}. The evaluation of ν_{NA} can then be reflected by the improvement from the corrections. Therefore the SDE solver can be considered a benchmark. We applied M steps of corrector steps (ALD/ ALD+NA) after each predictor step, where $M \in \{1, 2, 5, 10, 20, 40, 80\}$. Note that ALD mainly used the trained score network from the training set distribution to undergo sampling, without the awareness of the PDE residual or arbitrage losses. Therefore, the two correctors could conflict with each other during sampling.

4.6 Numerical Results

We now present our experiments on American options. We first validate our proposed models qualitatively. Figures 4.1, 4.2 and 4.3 show the American option surfaces where the strike varies between 80 to 120 and the maturity between 0.1 to 2 on a (K, T) grid of size 4×4, 8×8 and 16×16, respectively. Here, we compare the American option surfaces generated by SDE, SDE+ALD, and SDE+ALD+NA. We can see that they all show visually similar result as the ground truth surface.

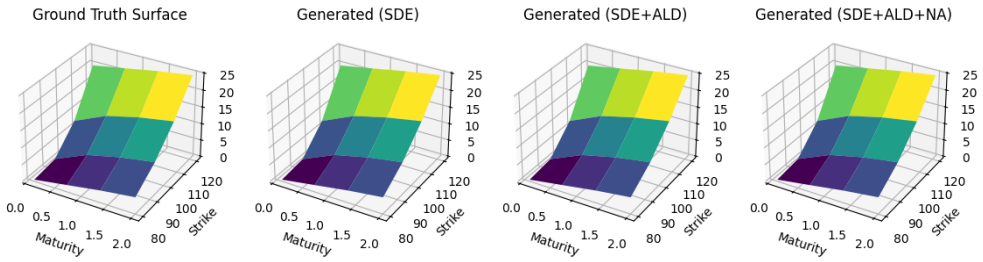


Figure 4.1: American Options: Surface Comparison (4×4)

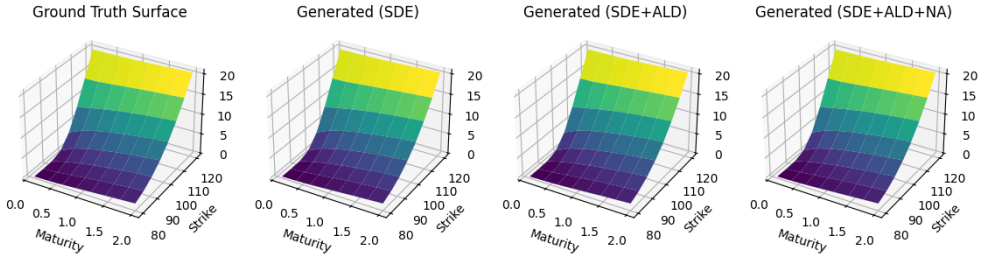


Figure 4.2: American Options: Surface Comparison (8×8)

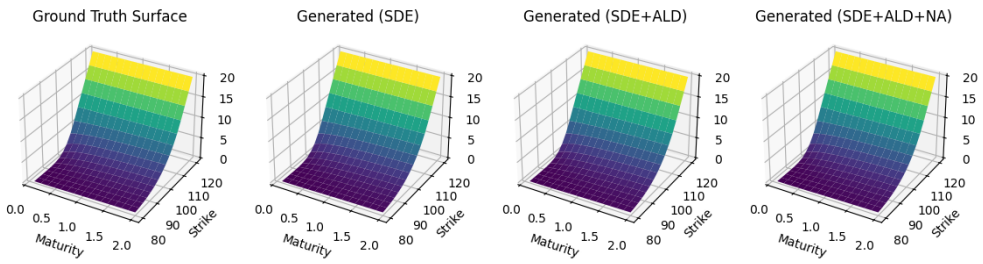


Figure 4.3: American Options: Surface Comparison (16×16)

4.6.1 Regional Performance

Figures 4.4, 4.5, and 4.6 present the Mean Absolute Error (MAE) and Mean Relative Error (MRE) for the benchmark model (pure SDE solver) and the SDE+ALD+NA model. The errors are evaluated over different moneyness and maturity regions defined in Tables 4.1 and 4.2, for grid resolutions 4×4, 8×8, and 16×16, respectively. For the SDE+ALD+NA model, the reported errors correspond to the minimum over all choices of $\lambda_{NA} \in \{10^{-3}, 10^{-2}, 10^{-1}\}$.

Table 4.1: Maturity regions used for error aggregation.

Region	**Lower Bound**	**Upper Bound**
Short	0.10	0.50
Medium	0.50	1.20
Long	1.20	2.00

Table 4.2: Strike (moneyness) regions for evaluation when full strike discretization is available.

Region	**Lower Bound**	**Upper Bound**
Deep OTM	80	90
OTM	90	95
ATM	95	105
ITM	105	110
Deep ITM	110	120

From the figures, we observe that for missing data percentages below 80%, the final MAE across most regions remains below the order of 10^{-2}. When the missing percentage is high

(60% and 80%), the SDE+ALD+NA model consistently achieves lower MAE than the pure SDE benchmark. However, in Figure 4.6, for lower missing percentages (20% and 40%), the improvement is limited in MAE. In some cases the SDE+ALD+NA model performs worse. For example, for 40% missing on the 8×8 grid, SDE+ALD+NA exhibits higher error than the pure SDE solver.

Note that MAE is not scale invariant, unlike MRE. Put prices in shorter maturities and out-of-the-money regions are much smaller, with minimum close to order of 10^{-11}. Therefore in Figures 4.5 and 4.6, for 8×8 and 16×16, the MAE is generally above the order of 10^{-4}, which is much larger than 10^{-11}. Despite the MAE across different maturity and moneyness regions being similar orders of magnitudes, MRE is much higher in OTM and short maturity regions.

The largest MRE values occur in the deep out-of-the-money (OTM) and short-maturity regions, where MRE ranges from approximately 10^{-3} to 10^{2}. This behavior is expected, as American put prices in these regions are close to zero. In fact, the smallest true price in the dataset is on the order of 10^{-11}, which leads to large relative errors even when the absolute error is small. Most training samples have true prices larger than 10^{-5}, and consequently the network struggles to capture these extreme low-price outliers. Another contributing factor is the choice of noise scale in training. The current setup uses a minimum noise level of $\sigma_{\min} = 10^{-3}$, while the American put prices span magnitudes from 10^{-11} to 10^{1}. This mismatch limits the sensitivity of the score network in low-price regions. Reducing $\sigma_{\min}$ and further tuning the noise schedule would likely improve precision for small values. However, in Figure 4.6, for 40% and 20% missing, despite no significant MAE improvement, SDE+ALD+NA reduces the MRE in short-maturity and OTM regions. This indicates that when provided with sufficient observed values, and in lower discretization error, the correctors are able to cater to the small value regions as well.

Overall, the correctors provide the most significant MAE improvement in regions with higher moneyness, longer maturities, and higher missing percentages, with the strongest gains observed for $60\% \sim 80\%$ missing on the 8×8 grid. Except for the extreme case of 80% missing, most regions achieve MAE and MRE on the order of 10^{-2} after corrections. The best-performing regions are in-the-money (ITM and deep ITM) with long maturities. These results indicate that the current model favors the reconstruction of larger option values, while predictions of small prices in OTM and short-maturity regions remain challenging and dependent on factors including discretization errors and observed values.

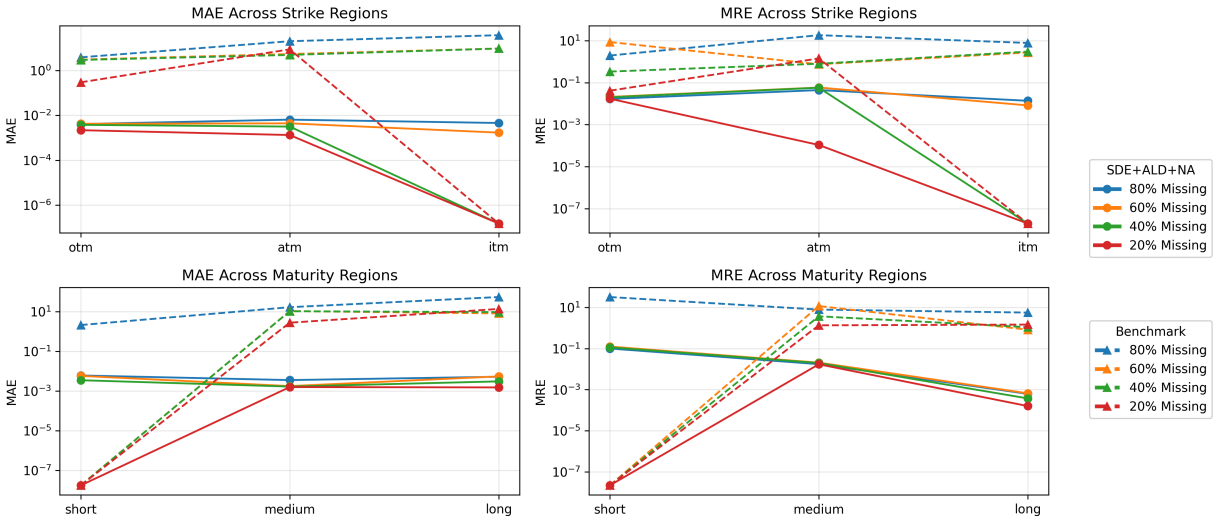


Figure 4.4: American Options: Regional Error Comparison (4×4)

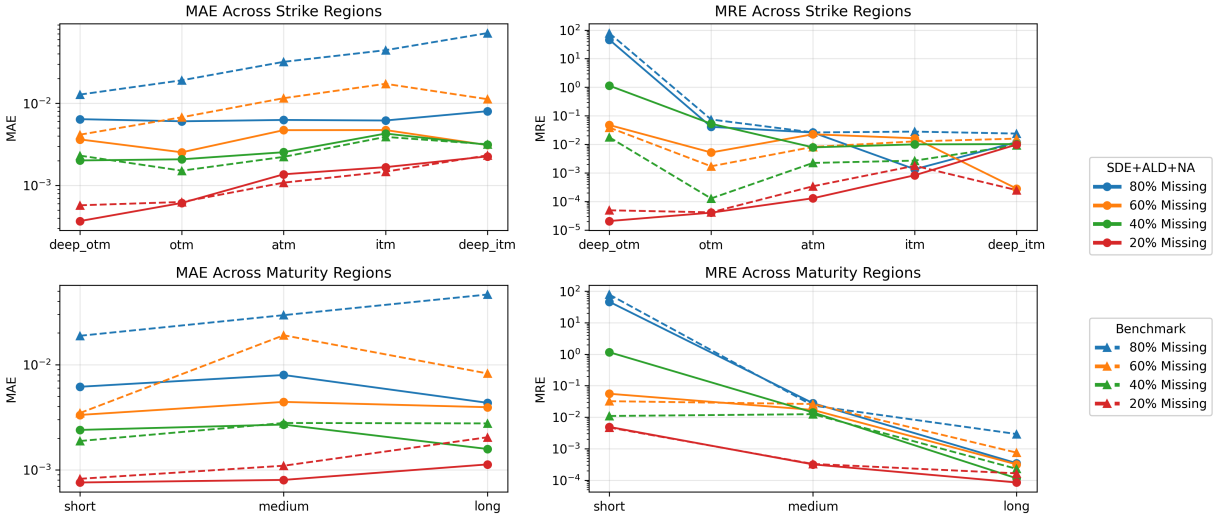


Figure 4.5: American Options: Regional Error Comparison (8×8)

4.6.2 Selection of the Corrector Weight λ_{NA}

The overall performance for 4×4, 8×8, and 16×16 grids across different choices of λ_{NA} can be summarized in Figures 4.7, 4.8, and 4.9 respectively. The figures show the MAE for each $\lambda_{NA} \in \{10^{-3}, 10^{-2}, 10^{-1}\}$ for the weight of the NA corrector, across different numbers

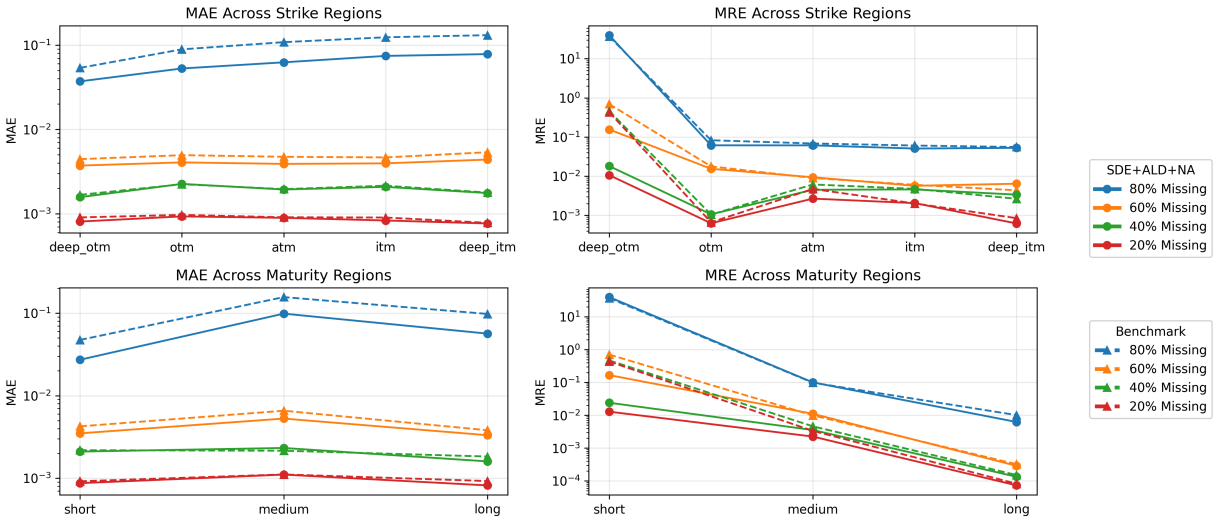


Figure 4.6: American Options: Regional Error Comparison (16×16)

of annealing steps M. Figure 4.10 shows an enlarged MAE comparison for different λ_{NA} on the 16×16 grid. The MAE of the pure SDE method (dotted lines) are compared with the MAE of the SDE+ALD method (dotted lines with markers). Note that even if the points lie on the same line, they represent the final error after adding M correction steps right after each predictor step, so they do not belong to the same sampling trajectory.

In general, there is a decreasing trend in MAE as the number of annealing steps M increases. This trend is the most apparent for 4×4 and 8×8. To compare SDE to SDE+ALD, with large enough number of annealing steps, SDE+ALD attained lower MAE than SDE in all cases, except for $20\% - 40\%$ missing for 8×8 in Figure 4.8. In addition, in Figure 4.10, SDE+ALD attains lower MAE consistently than pure SDE solver for high missing percentages $60\% \sim 80\%$. This can be explained by the nature of ALD, that the ALD corrector introduces a stochastic component at each annealing step. This stochastic component provides advantages in higher missing values. While this is beneficial in cases with many missing values, where exploration of the uncertain surface is necessary, it can occasionally perturb already accurate predictions when the number of missing values is small. This explains why, in the 20% and 40% missing data scenarios, SDE+ALD may perform slightly worse than pure SDE in Figure 4.10.

To compare SDE+ALD with SDE+ALD+NA, Figure 4.10 for 16×16 gives a clear picture. For all missing percentages, $M > 80$ steps gives improvements from the NA correctors in terms of the MAE for some λ_{NA}. There are no big improvements for smaller M because

the NA corrector was not designed to reduce the MAE, but the arbitrage losses. For $M > 20$ and $60 \sim 80\%$ missing, there is a tendency of NA corrector giving higher MAE for $\lambda_{NA} = 10^{-1}$, indicating over-correction. This could come from two sources: (1) for each correction step, ALD introduces white noises, and then is corrected by NA according to no-arbitrage conditions. This could disrupt the correction when the weights of the ALD and NA are not coherent; (2) $\lambda_{NA} = 10^{-1}$ gives more improvement over SDE+ALD for lower missing percentages ($20\% \sim 40\%$). Its performance becomes worse than $\lambda_{NA} \in \{10^{-2}, 10^{-3}\}$ gradually as the proportion of missing values increases to $60\% \sim 80\%$, in which case lower $\lambda_{NA} = 10^{-3}, 10^{-2}$ perform the best. This signifies that λ_{NA} should be decreasing with the percentage of missing values. Indeed, when the proportion of missing data is high, the arbitrage losses are evaluated on a surface with significant uncertainty. In this case, apparent arbitrage violations may arise from reconstruction noise rather than true inconsistencies in the option surface. Applying a large NA correction step can then over-react to these noisy signals, amplifying local errors or distorting nearby regions. Smaller step sizes are therefore required to ensure that the NA corrector performs gradual and stable adjustments that reduce arbitrage violations on average rather than enforcing them too aggressively.

Overall, these results suggest that the optimal choice of λ_{NA} depends on the level of missing data. Larger values of λ_{NA} are more effective when the amount of missing data is small, while smaller values are preferable as the missing percentage increases.

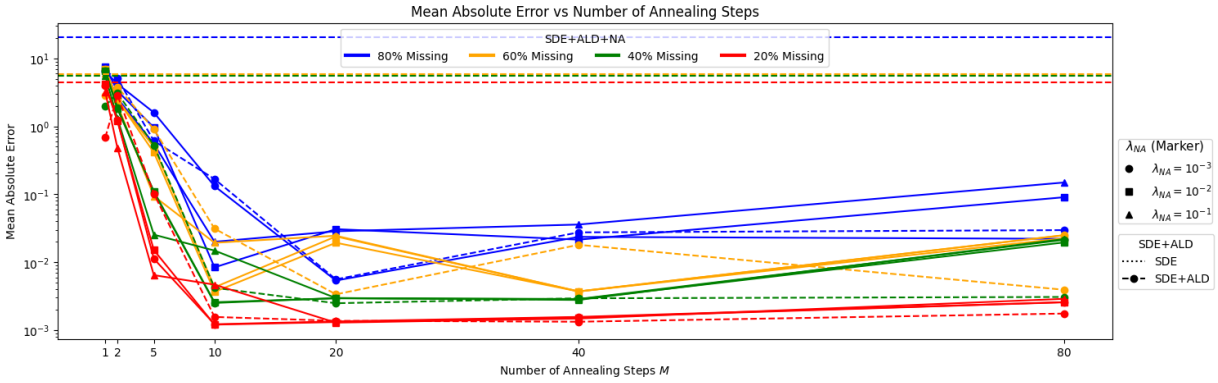


Figure 4.7: American Options: Final MAE Comparison (4×4)

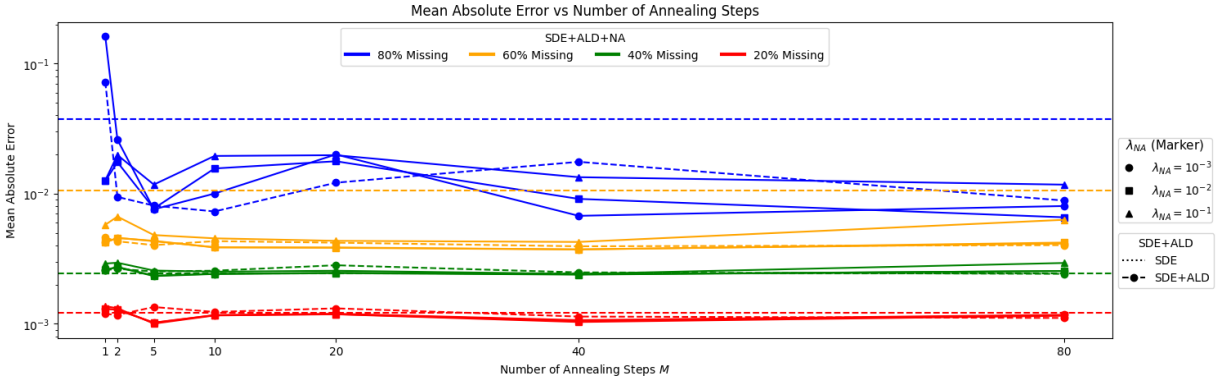


Figure 4.8: American Options: Final MAE Comparison (8×8)

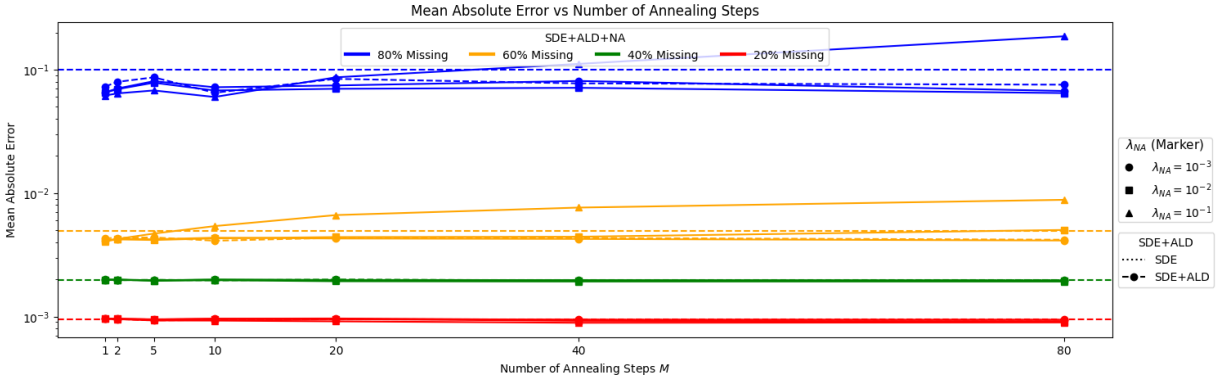


Figure 4.9: American Options: Final MAE Comparison (16×16)

4.6.3 Arbitrage Losses

We now look at the arbitrage losses. Figures 4.11, 4.12, 4.13, 4.14, and 4.15 show the arbitrage losses defined in (4.13), (4.14), (4.15), (4.16), and (4.17), across different numbers of annealing steps M and different missing percentages for 16×16. Here we focus on 16×16 grid mainly, since the finer the grid, the less prone to discretization errors the surfaces become.

First, among the five arbitrage losses, the derivative loss (ℓ_{deri}) is the largest. This is because it requires arbitrage-free structures of more neighboring points compared to other arbitrage losses. We observe that the red dotted line (SDE+ALD) may not improve the arbitrage losses in all cases. In fact, in Figures 4.12 and 4.13, SDE+ALD tends to give higher convexity loss ℓ_{conv} and derivative loss ℓ_{deri} than the pure SDE solver (black dotted

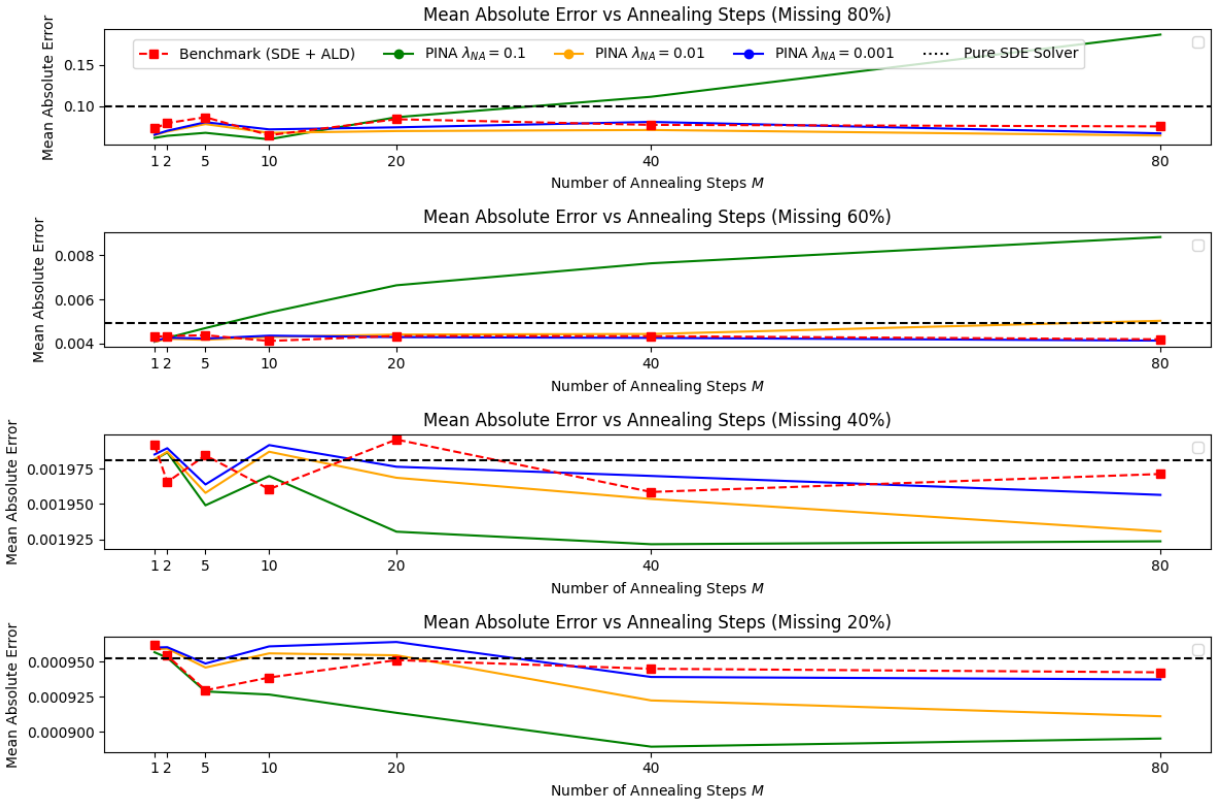


Figure 4.10: American Options: MAE Comparison for 16×16

line), except for the 80% missing case. This is because the ALD corrector was mainly driven by the score network, which learns the price surface distribution while being agnostic to the arbitrage losses, and it introduces white noises in each corrector step. Therefore, even with a large number of annealing steps M, the arbitrage losses using SDE+ALD did not show apparent sign of reduction.

In Figure 4.11, the addition of NA corrector creates oscillations in arbitrage losses for $20\% \sim 40\%$ missing. The NA corrector does not give lower $\ell_{mono(K)}$ in these cases. However, this is not a sign of failed implementation. These stem from the fact that the NA corrector considers all the arbitrage losses at the same time, so it prioritized reducing ℓ_{deri} among others, for it being the relatively large arbitrage loss. In fact, in Figure 4.16, we consider the average of all arbitrage losses, and we can see clearer trend. For $\lambda_{NA} = 10^{-1}$, we observe a steady decreasing trend of average loss as M increases for $20\% \sim 40\%$ missing. It also coheres with our previous conclusion that higher percentage of missing values can

work well with higher λ_{NA}, as the lowest average loss for 80% missing was obtained by $\lambda_{NA} = 10^{-1}$, which gives higher arbitrage losses as the percentage of missing values reduces to 20%. Similar results for 4×4 and 8×8 are included in the appendix.

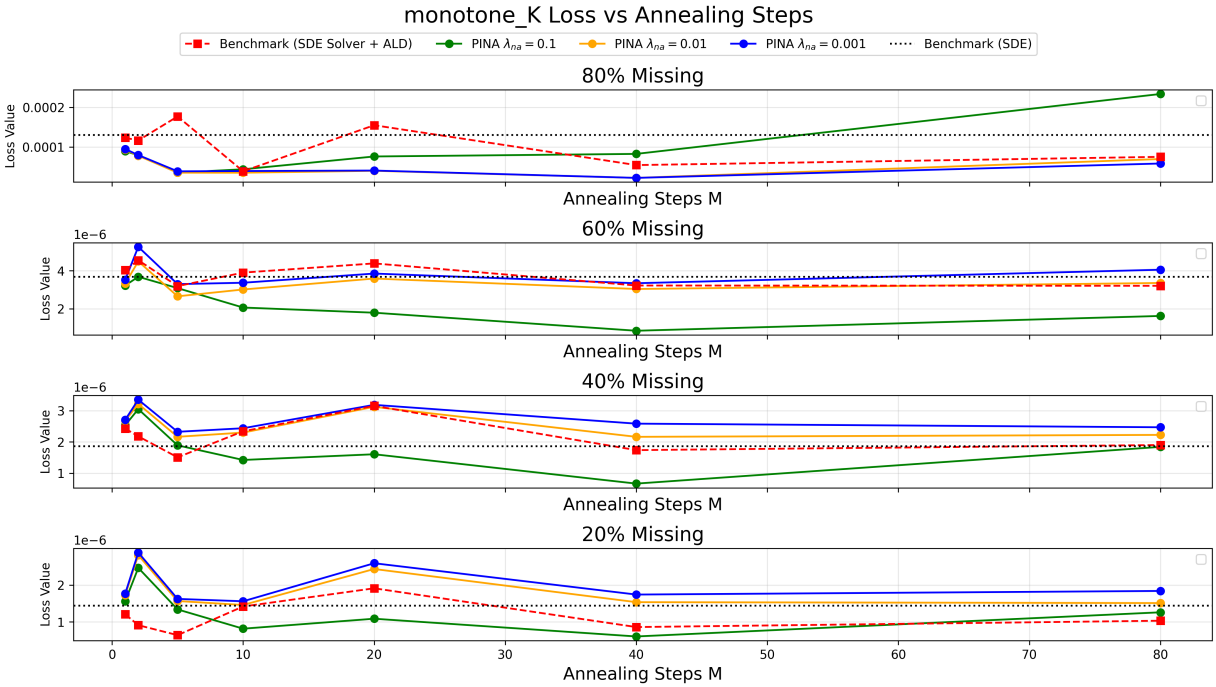


Figure 4.11: American Options: Monotonicity in K Loss (16×16)

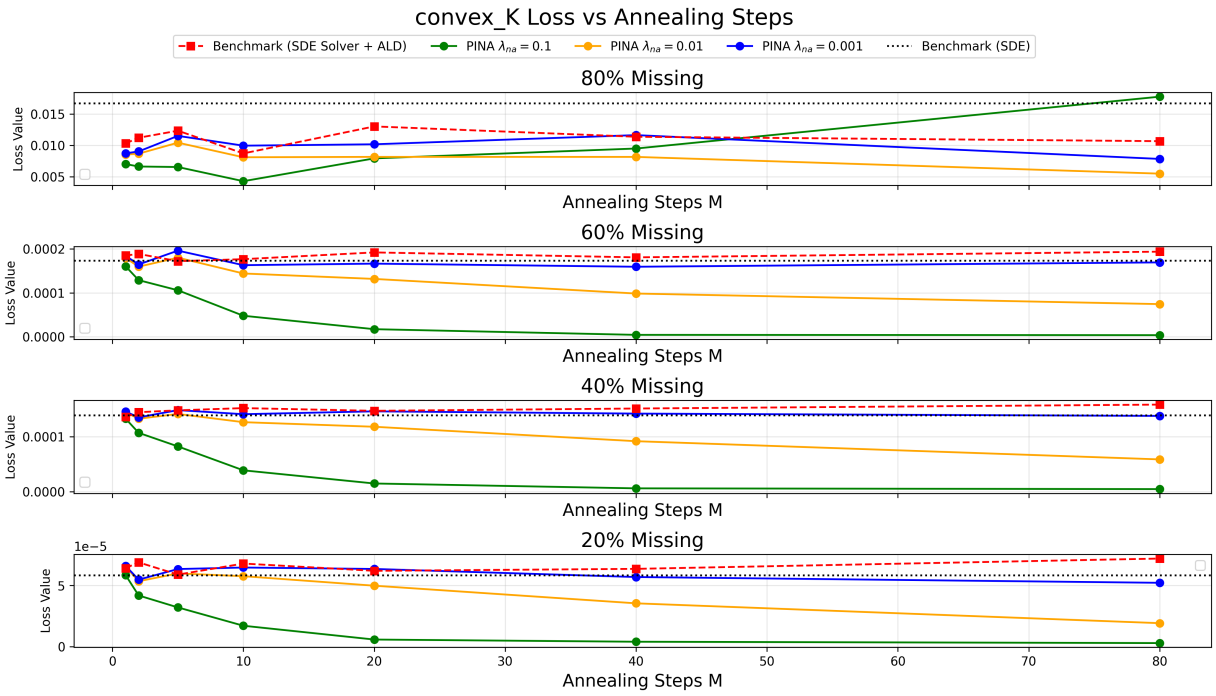


Figure 4.12: American Options: Convexity in K Loss (16×16)

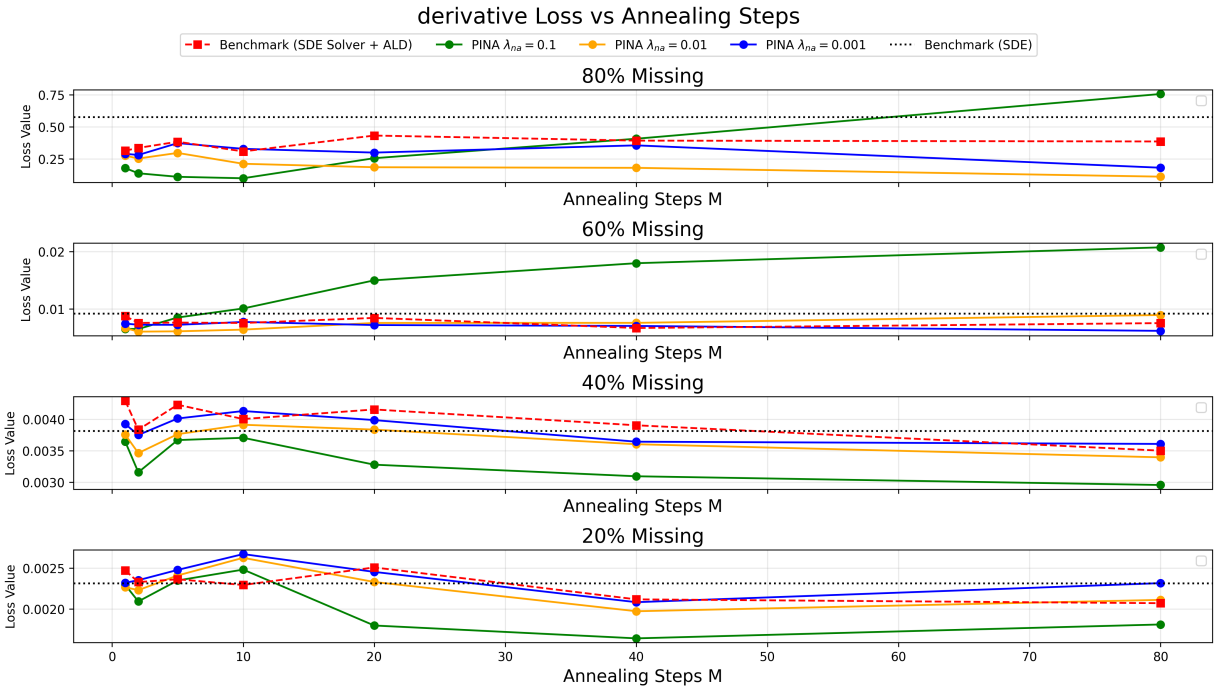


Figure 4.13: American Options: Derivative Loss (16×16)

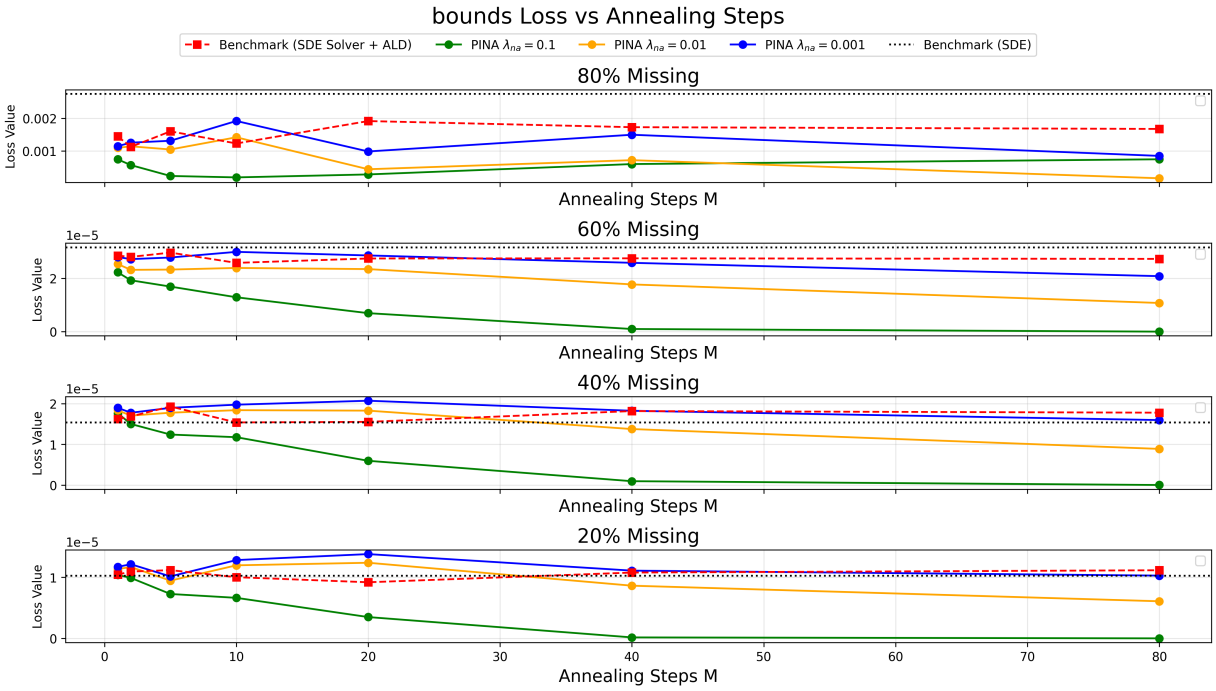


Figure 4.14: American Options: Bound Loss (16×16)

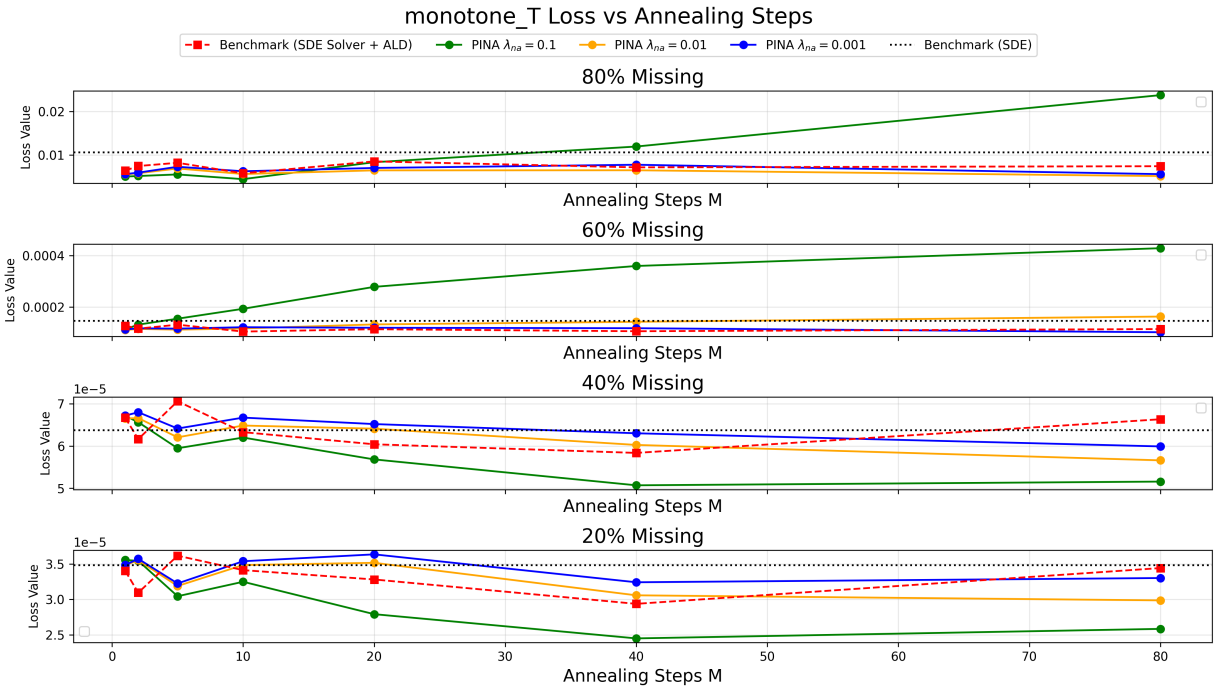


Figure 4.15: American Options: Monotonicity in T (16×16)

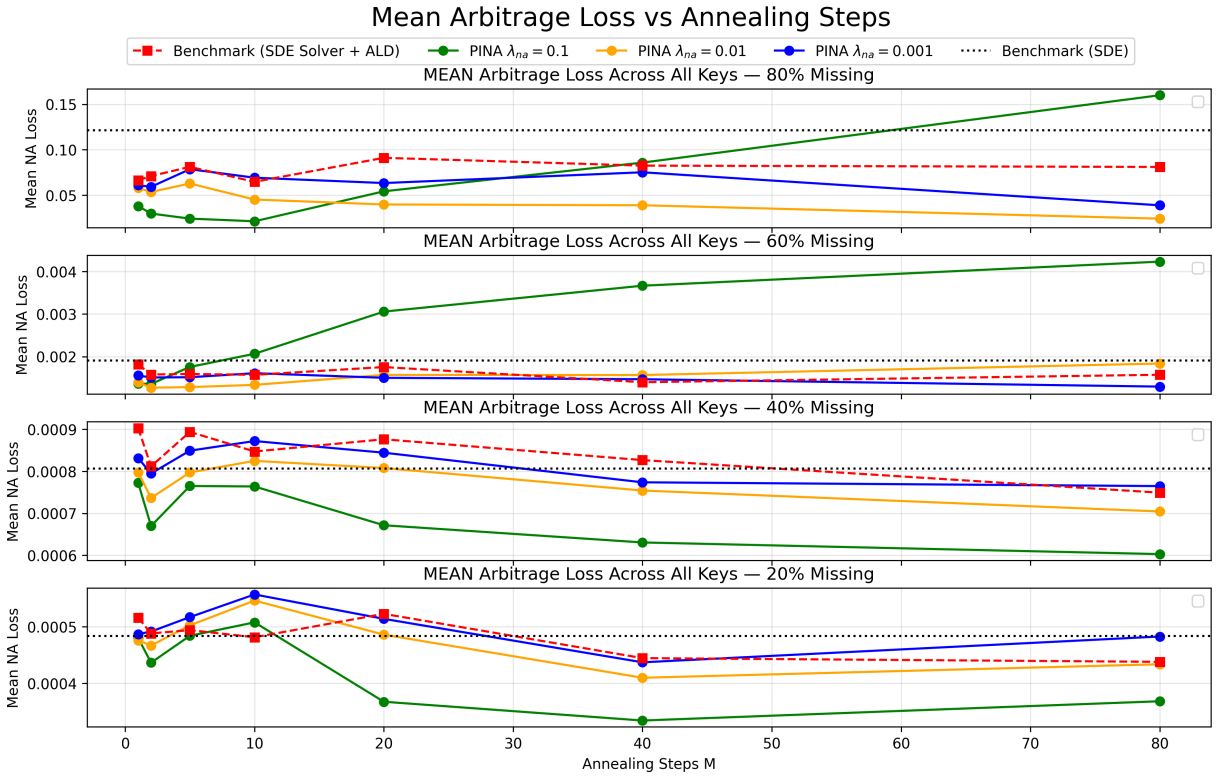


Figure 4.16: American Options: Mean Arbitrage Losses (16×16)

Chapter 5

Conclusions

In this thesis, we investigated several approaches to solve the price surface completion problem for European and American put options on the strike–maturity space (K, T). We extended previous work that treats surface completion as an image inpainting problem and applied it to both European and American options. Working directly with price surfaces is particularly advantageous for American options, as closed-form solutions do not exist. However, enforcing arbitrage-free conditions is more nuanced for American options due to the early-exercise feature, which introduces additional challenges for numerical methods. The image-based approach mitigates these challenges by treating each surface as a high-dimensional image. To ensure that the completed surfaces remain economically consistent, we converted arbitrage-free conditions into loss functions and designed correctors to guide the sampling process toward no-arbitrage solutions.

For European options, we incorporated physics-informed corrector (PI) into the sampling process to accelerate convergence, in addition to the no-arbitrage (NA) correctors. We investigated the tuning of their weights and analyzed their interaction with the stochastic differential equation (SDE) predictor within a predictor–corrector scheme. The approach achieved a mean absolute error (MAE) at the 10^{-2} level even with 80% missing values across different grids, while the no-arbitrage corrector effectively reduced arbitrage violations and improved sampling efficiency.

For American options, due to the absence of a corresponding PDE in (K, T) space, we focused on the NA corrector in combination with the Annealed Langevin Dynamics (ALD) corrector, carefully studying their interactions. We found that the proposed method is most effective with higher missing proportions and carefully tuned correction weights.

The proposed methodology has several advantages. First, the score-based generative model

treats price surfaces as images without imposing specific model assumptions. Physics-informed correctors, when applicable, provide the flexibility to incorporate any PDE-based knowledge. Second, the NA correctors are purely numerical and can be applied using the same score model employed in the predictor to actively reduce arbitrage violations. Finally, compared to previous work on volatility surface completion, the predictor–corrector scheme allows a substantial reduction in sampling steps; with only 20–40 steps and properly tuned corrector weights, we can efficiently generate completed surfaces.

We also explored the limitations of this approach. The method is sensitive to the magnitude of option prices because working directly on price surfaces introduces a wide dynamic range, requiring finer noise scales for training. Another challenge is the tuning of the NA corrector weight, λ_{NA}, which interacts with the stochastic component in ALD, the number of annealing steps M, and the imputation of observed values. However, an adaptive schedule for λ_{NA} remains an open question for future work.

Potential future directions include extending the method to American put options in the (t, S) space. The lack of a PDE in (K, T) space makes the PINA approach incompatible for American options, but most theoretical results on American options concern the (t, S) space due to its natural handling of free-boundary problems. Incorporating physics-informed residuals into the guided diffusion process in (t, S) space could enable efficient surface completion and generation, and could also facilitate identification of the free boundary using score-based generative models—an area that is computationally challenging with traditional numerical methods.

References

[1] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[2] Artan Borici and H Luthi. Fast solutions of complementarity formulations in american put pricing. *Journal of Computational Finance*, 9(1):63, 2005.

[3] Douglas T. Breeden and Robert H. Litzenberger. Prices of state-contingent claims implicit in option prices. *The Journal of Business*, 51(4):621–651, 1978.

[4] Peter Carr and Dilip B. Madan. A note on sufficient conditions for no arbitrage. *Finance Research Letters*, 2(3):125–130, 2005.

[5] Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems, 2024.

[6] Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A kernel test of goodness of fit, 2016.

[7] Alexander M. G. Cox and Christoph Hoeggerl. Model-independent no-arbitrage conditions on american put options, 2013.

[8] Mark H. A. Davis and David G. Hobson. The range of traded option prices. *Mathematical Finance*, 17(1):1–14, 2007.

[9] Ashish Dhiman and Yibei Hu. Physics informed neural network for option pricing, 2023.

[10] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2015.

[11] Bruno Dupire. Pricing with a smile. 1994.

[12] Karol Duris, Shih-Hau Tan, Choi-Hong Lai, and Daniel Sevcovic. Comparison of the analytical approximation formula and newton's method for solving a class of nonlinear black-scholes parabolic equations, 2015.

[13] Peter Forsyth and George Labahn. Numerical methods for controlled hamilton-jacobi-bellman pdes in finance. *J. Comput. Finance*, 11, 10 2007.

[14] Jim Gatheral, Thibault Jaisson, and Mathieu Rosenbaum. Volatility is rough, 2014.

[15] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[16] Julien Guyon and Pierre Henry-Labordere. *Nonlinear option pricing*. Chapman & Hall/CRC Financial Mathematics Series. Chapman and Hall/CRC, an imprint of Taylor and Francis, Boca Raton, FL, first edition. edition, 2014 - 2013.

[17] Steven Heston. *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, pages 382–397. 03 2005.

[18] Steven L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343, 1993.

[19] Paweł Hitczenko and Stanisław Kwapień. On the rademacher series. In Jørgen Hoffmann-Jørgensen, James Kuelbs, and Michael B. Marcus, editors, *Probability in Banach Spaces, 9*, pages 31–36, Boston, MA, 1994. Birkhäuser Boston.

[20] Blanka Horvath, Aitor Muguruza, and Mehdi Tomas. Deep learning volatility, 2019.

[21] Ying Kit Hui. Volatility surface completion using score-based generative models. *Thesis, University of Waterloo*, 2023.

[22] Codruț-Florin Ivașcu. Option pricing using machine learning. *Expert Systems with Applications*, 163:113799, 2021.

[23] Christian Jacobsen, Yilin Zhuang, and Karthik Duraisamy. Cocogen: Physically-consistent and conditioned score-based generative models for forward and inverse problems, 2024.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[25] Damien Lamberton. On the binomial approximation of the american put, 2018.

[26] Dietmar PJ Leisen. Pricing the american put option: A detailed convergence analysis for binomial models. *Journal of economic dynamics and control*, 22(8-9):1419–1444, 1998.

[27] Alan Lewis. *Option Valuation Under Stochastic Volatility*. 01 2000.

[28] Alan Lewis. A simple option formula for general jump-diffusion and other exponential levy processes. *SSRN Electronic Journal*, 05 2002.

[29] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation, 2016.

[30] J.L. Maryak and D.C. Chin. Global random optimization by simultaneous perturbation stochastic approximation. In *Proceeding of the 2001 Winter Simulation Conference (Cat. No.01CH37304)*, volume 1, pages 307–312 vol.1, 2001.

[31] Bjørn Fredrik Nielsen, Ola Skavhaug, and Aslak Tveito. Penalty methods for the numerical solution of american multi-asset option problems. *Journal of Computational and Applied Mathematics*, 222(1):3–16, 2008. Special Issue: Numerical PDE Methods in Finance.

[32] Tianyu Pang, Cheng Lu, Chao Du, Min Lin, Shuicheng Yan, and Zhijie Deng. On calibrating diffusion probabilistic models, 2023.

[33] M. Raissi, P. Perdikaris, and G.E. Karniadakis. 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, 2019.

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[35] Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, 2023.

[36] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

[37] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models, 2020.

[38] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation, 2019.

[39] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020.

[40] D.Y. Tangman, A. Gopaul, and M. Bhuruth. A fast high-order finite difference algorithm for pricing american options. *Journal of Computational and Applied Mathematics*, 222(1):17–29, 2008. Special Issue: Numerical PDE Methods in Finance.

[41] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks, 2016.

[42] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[43] Chunfa Wang. Pricing european options by stable fourier-cosine series expansions, 2017.

APPENDICES

Appendix A

Supplementary Plots

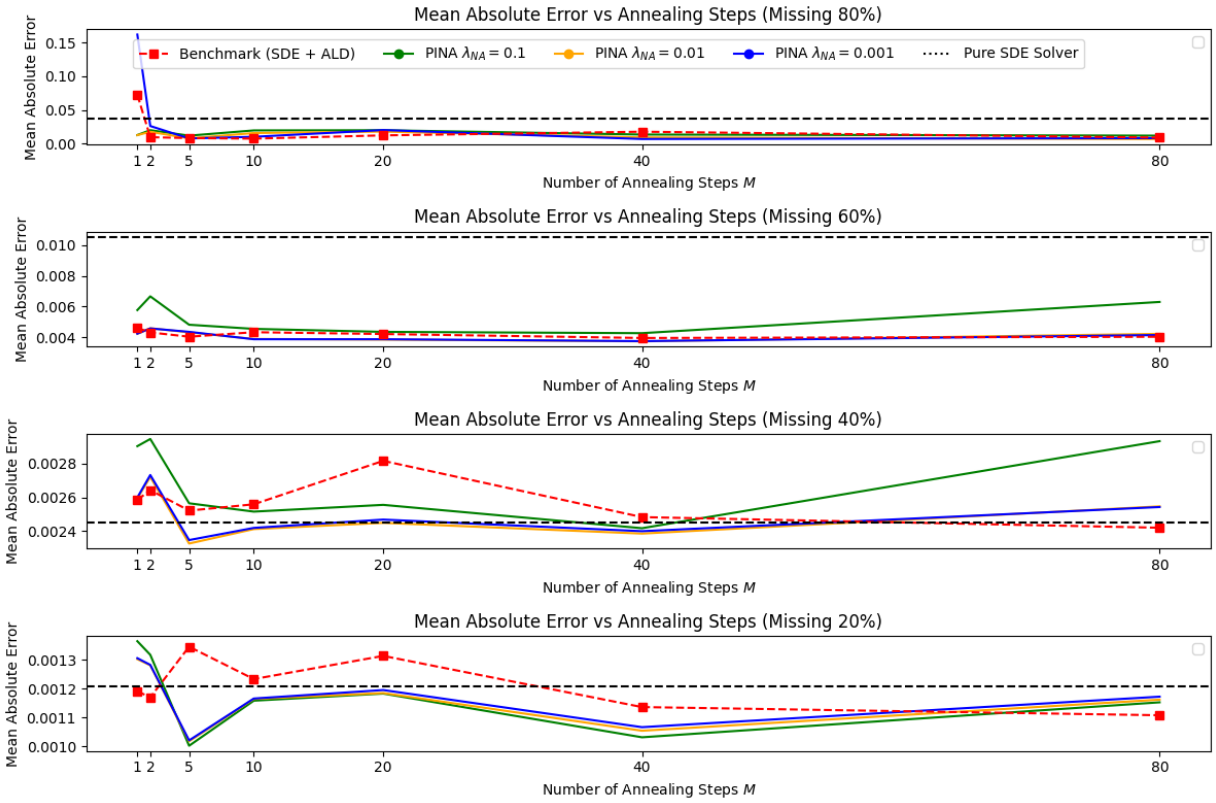


Figure A.1: American Options: MAE Comparison (8×8)

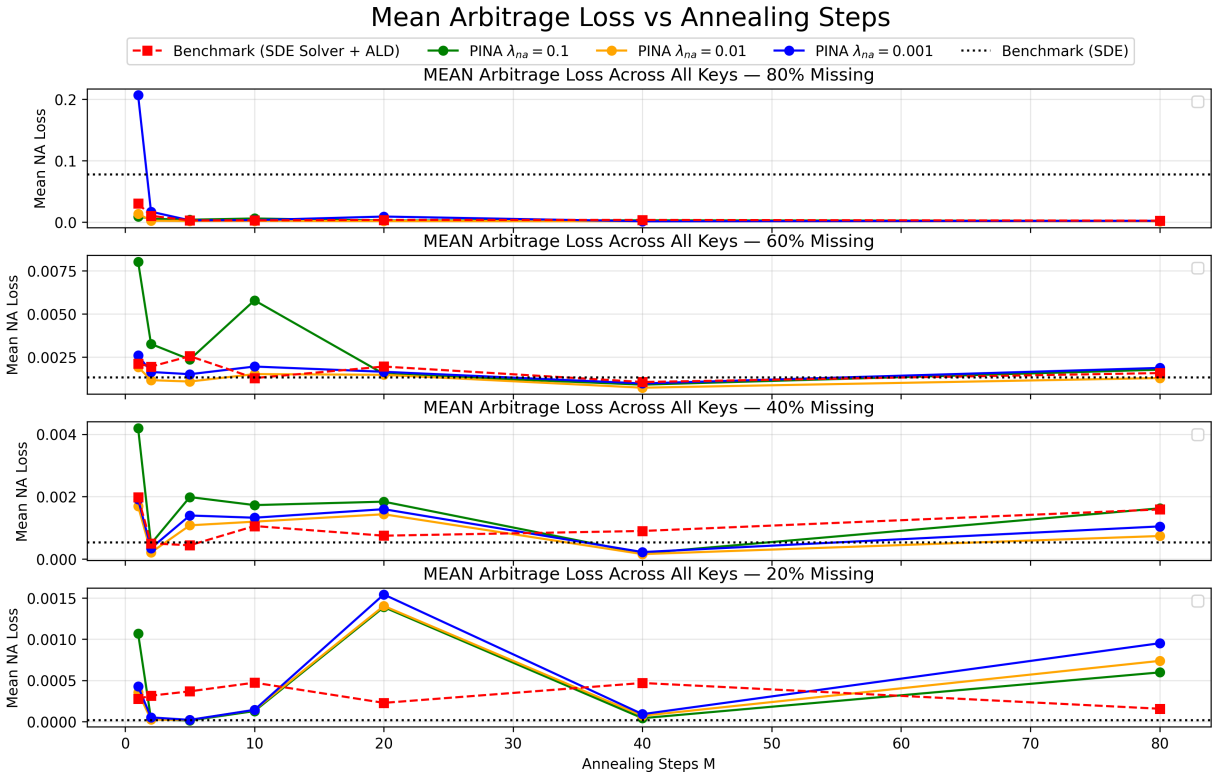


Figure A.2: American Options: Mean Arbitrage Losses (8×8)

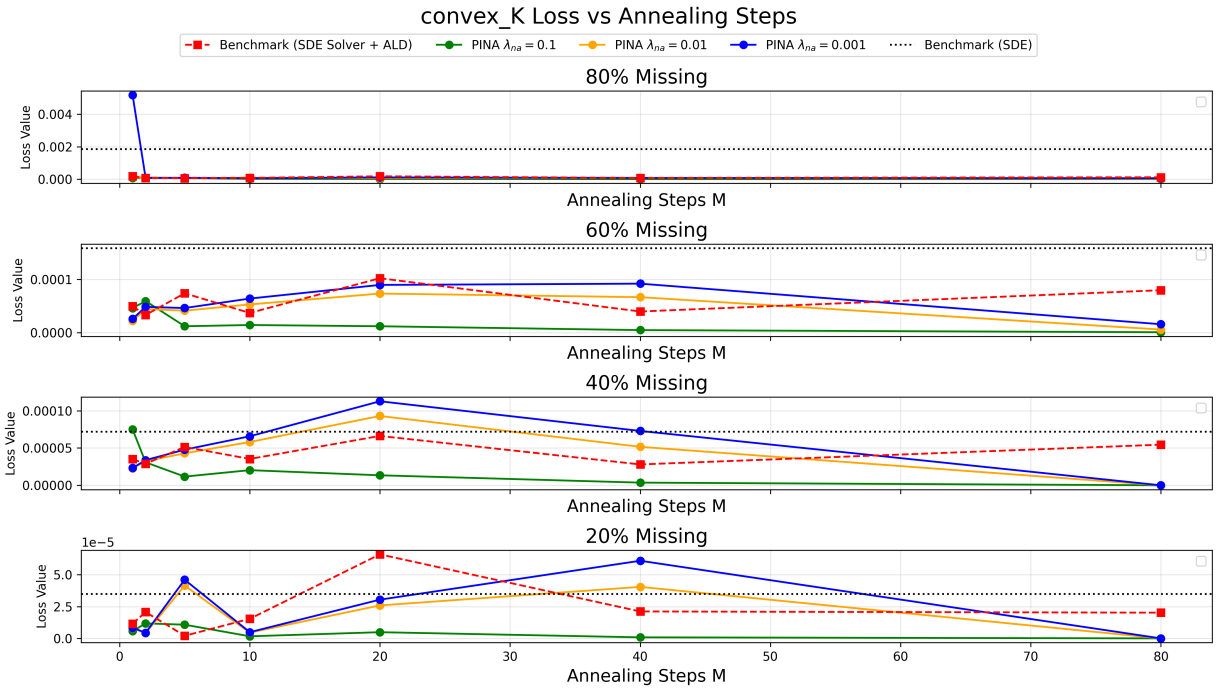


Figure A.3: American Options: Convexity in K (8×8)

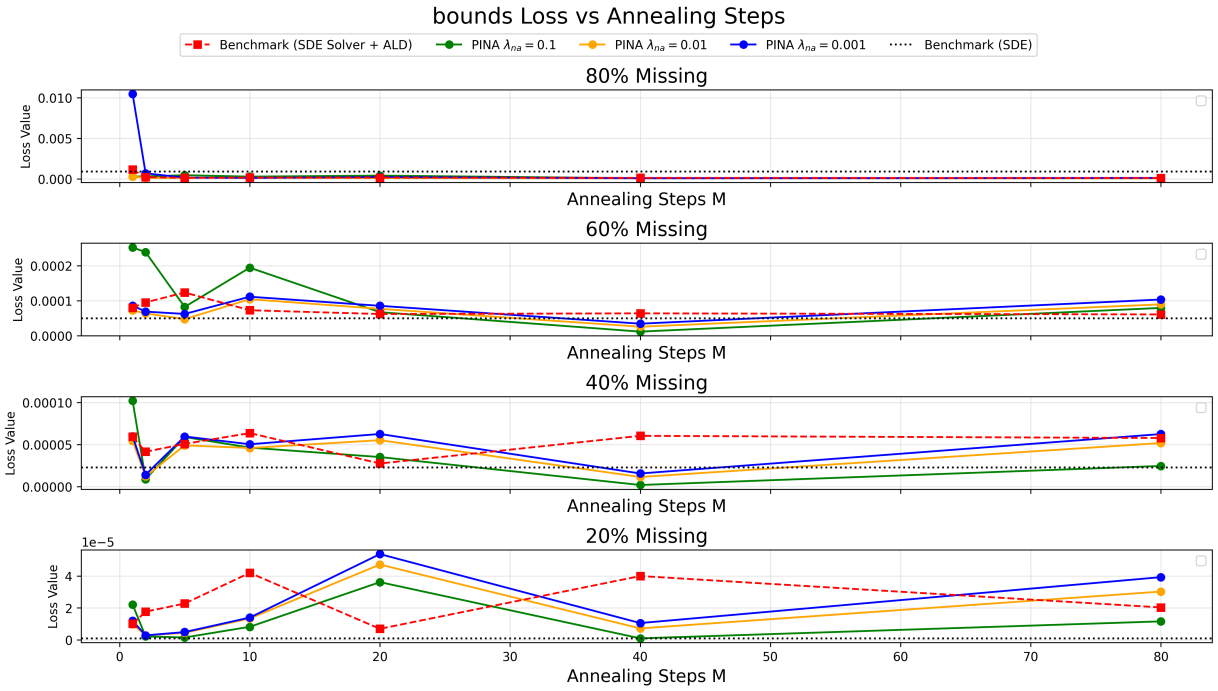


Figure A.4: American Options: Bound Loss (8×8)

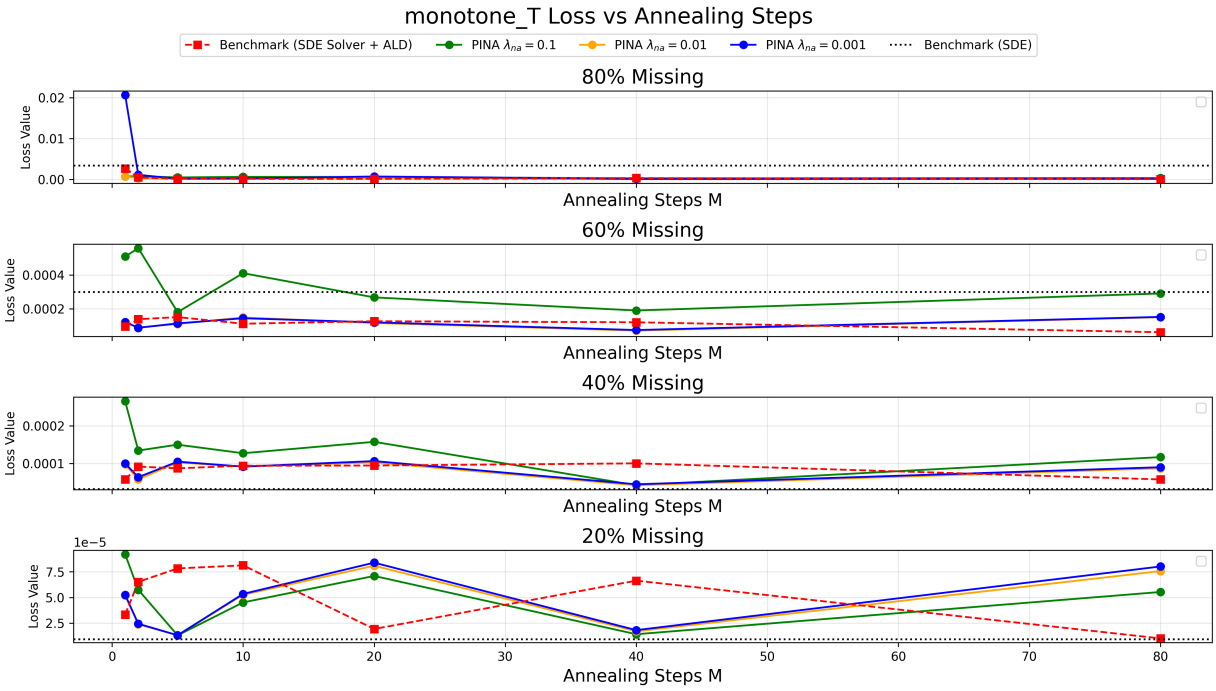


Figure A.5: American Options: Monotonicity in T (8×8)

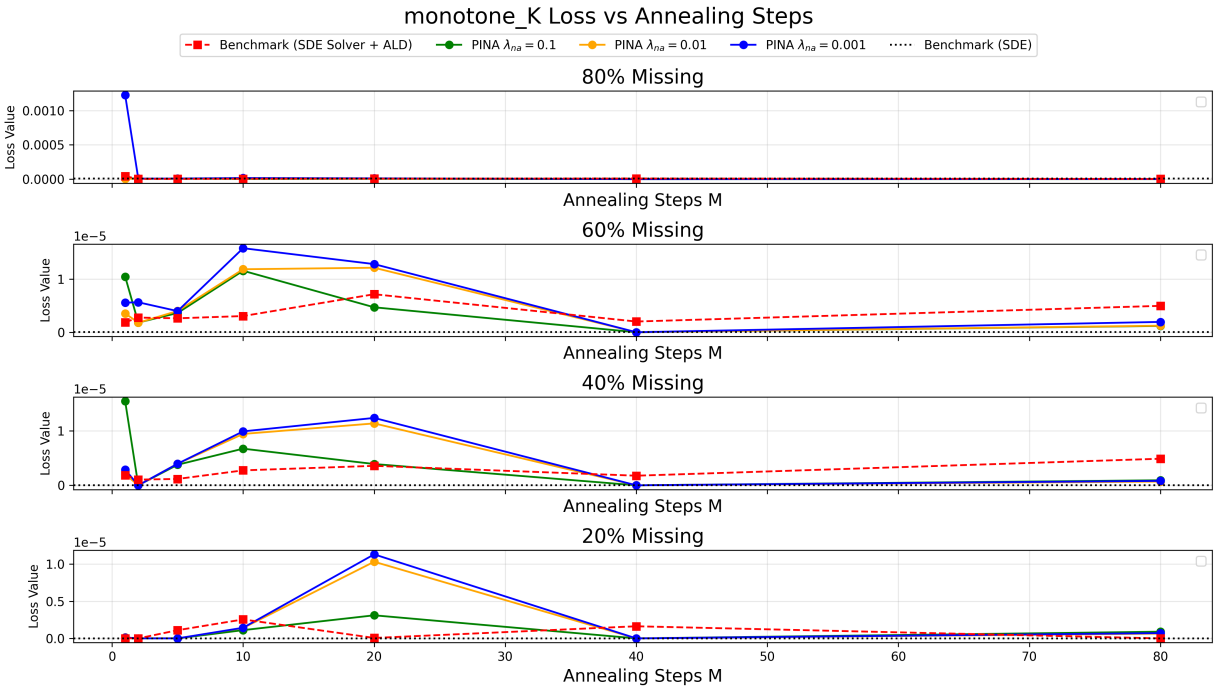


Figure A.6: American Options: Monotonicity in K (8×8)

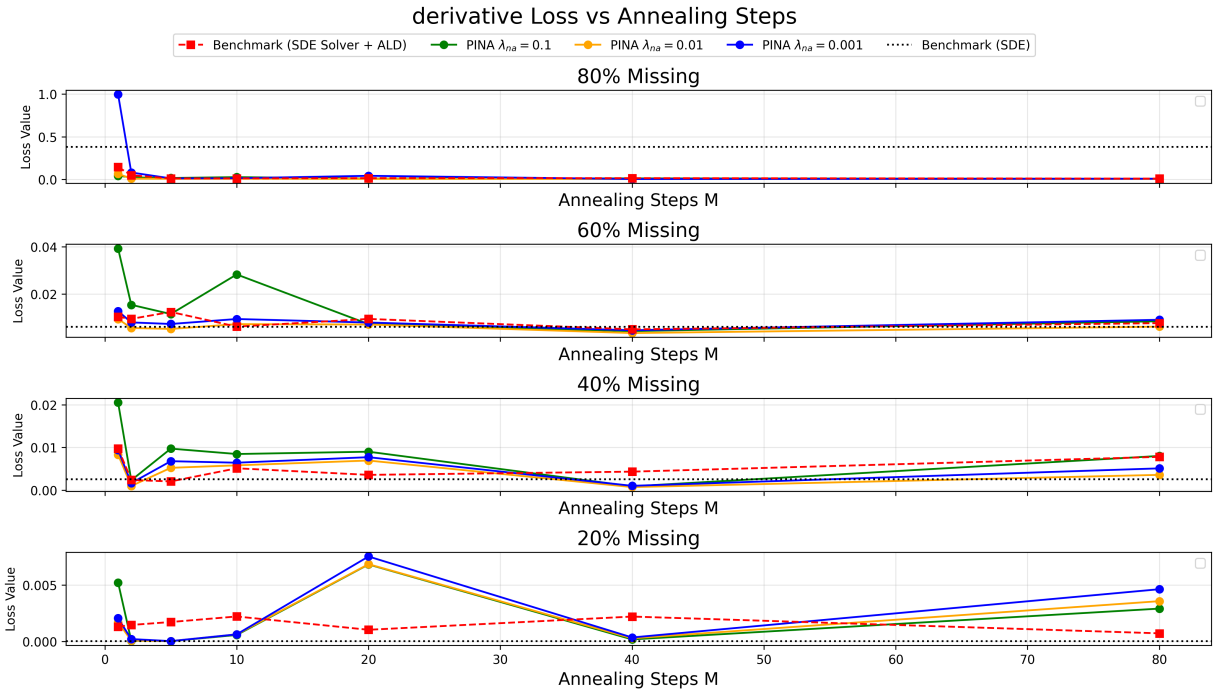


Figure A.7: American Options: Derivative Loss (8×8)

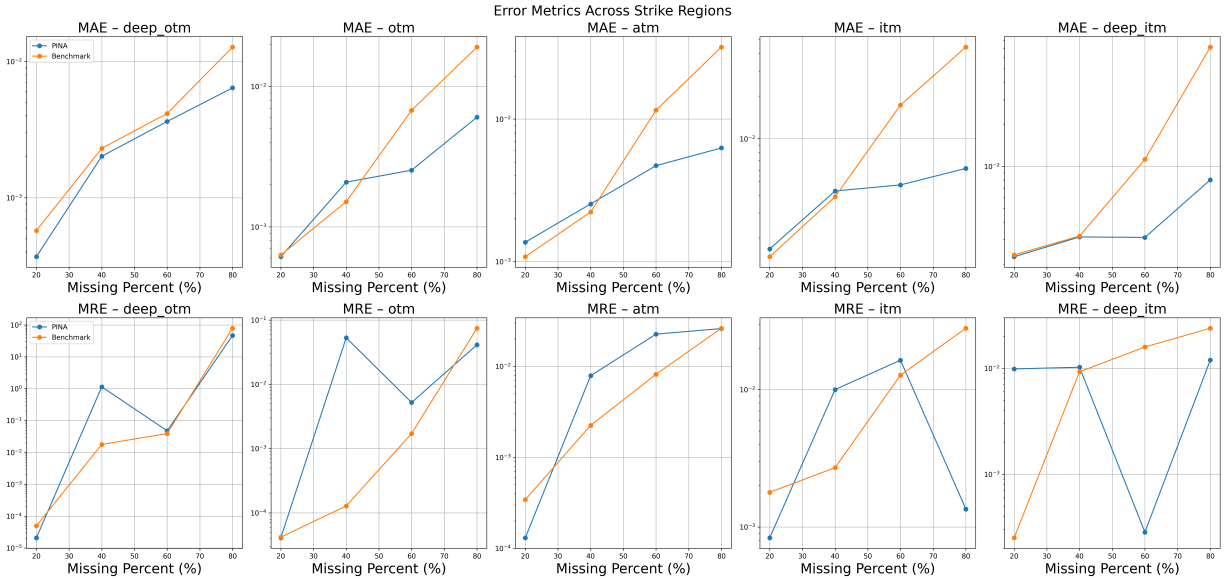


Figure A.8: American Options: Error Comparison for Varying Missing Percentages (8×8)

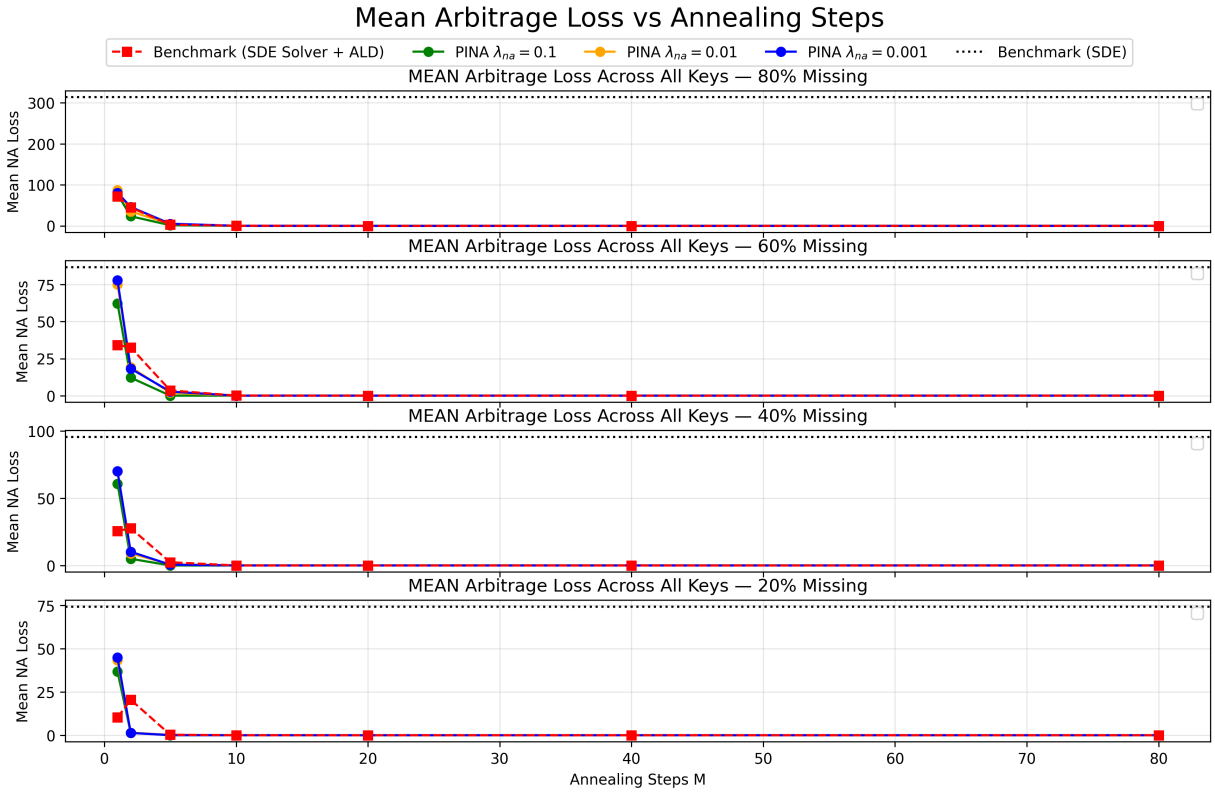


Figure A.9: American Options: Mean Arbitrage Losses (4×4)

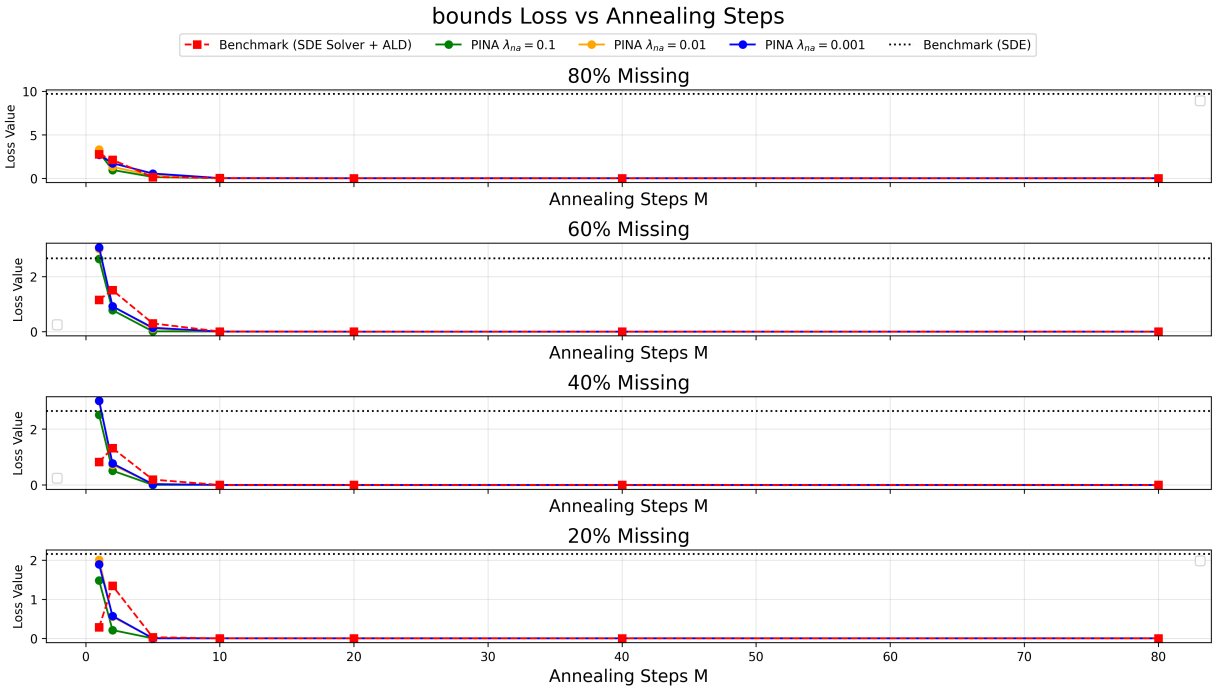


Figure A.10: American Options: Bound loss (4×4)

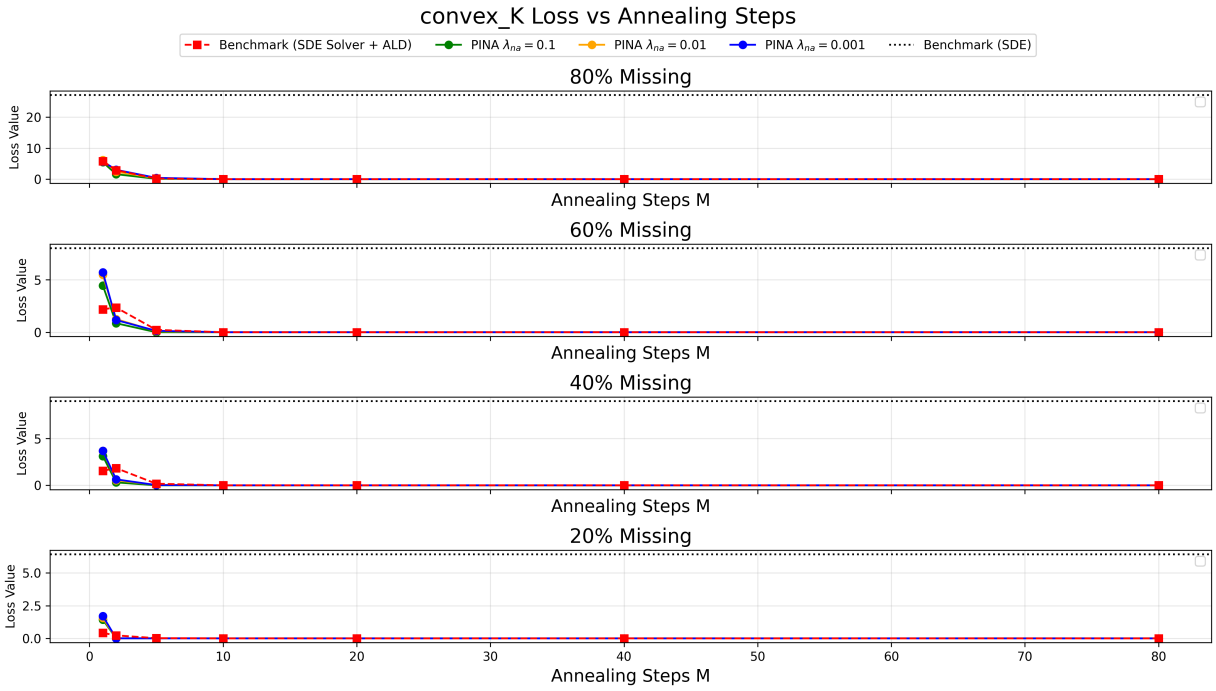


Figure A.11: American Options: Convexity in K (4×4)

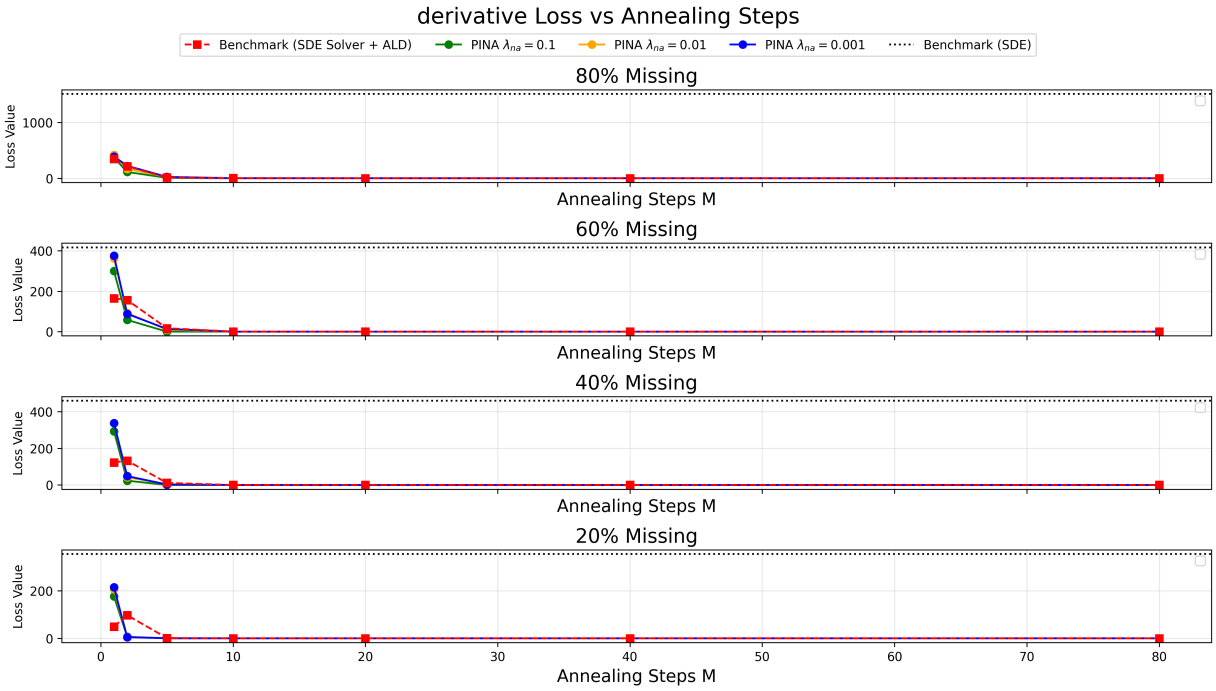


Figure A.12: American Options: Derivative Loss (4×4)

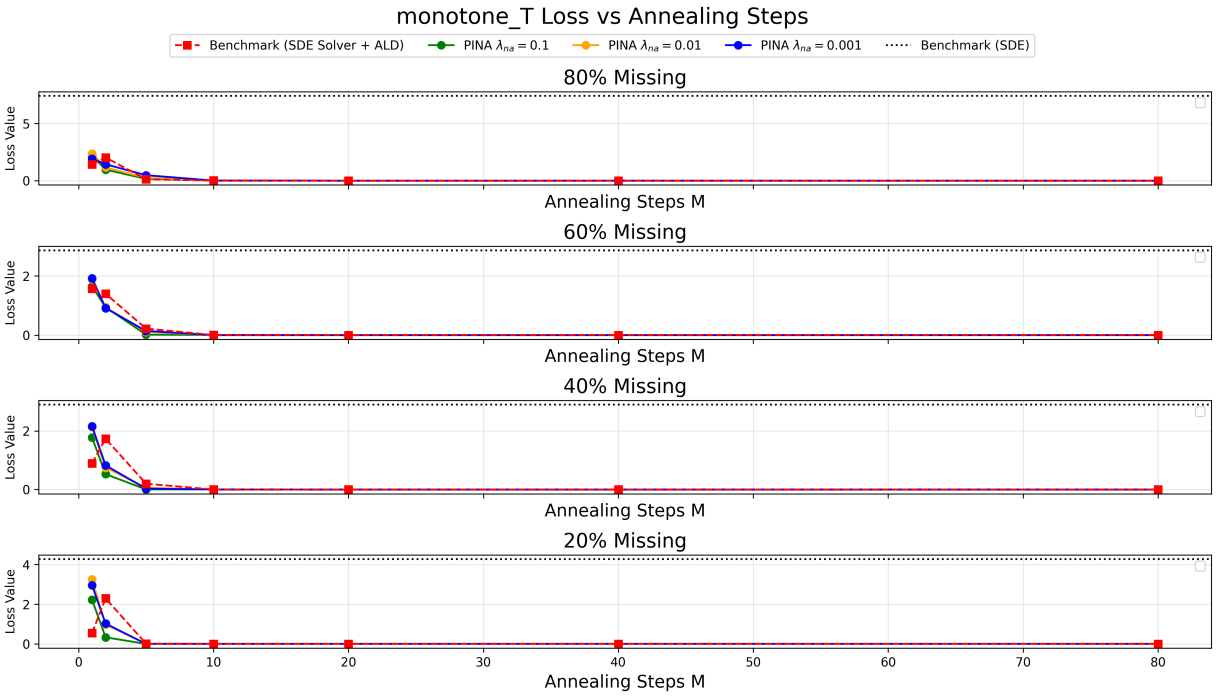


Figure A.13: American Options: Monotonicity in T (4×4)

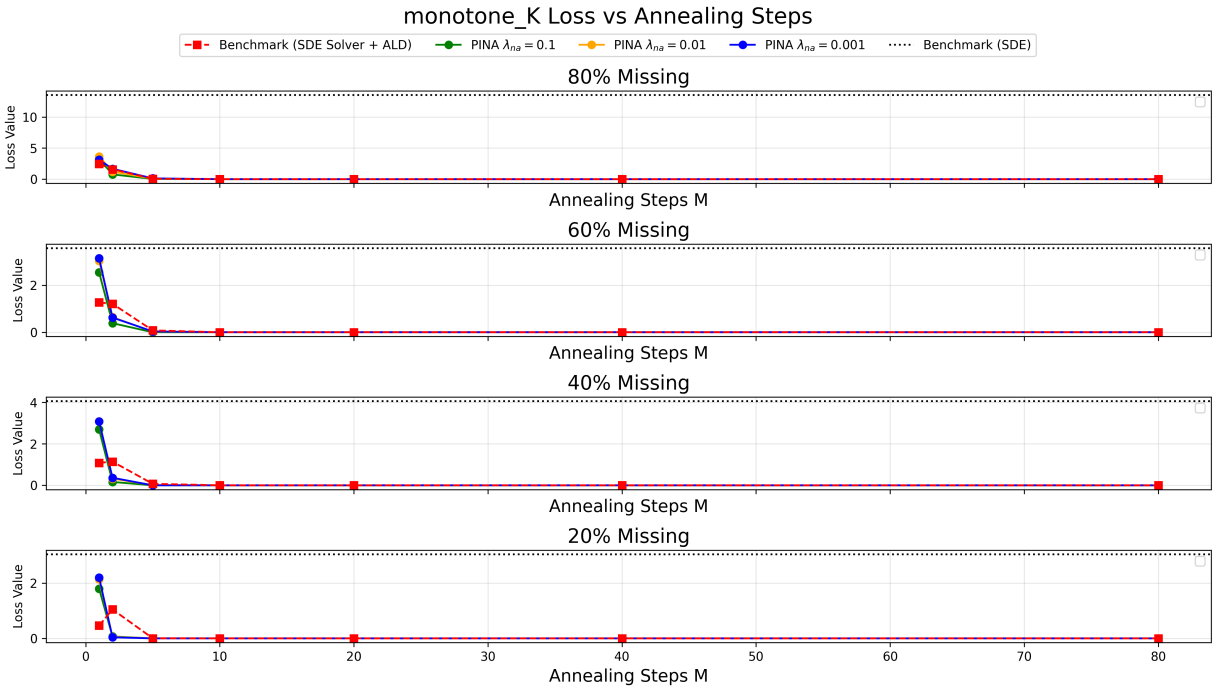


Figure A.14: American Options: Monotonicity in K (4×4)

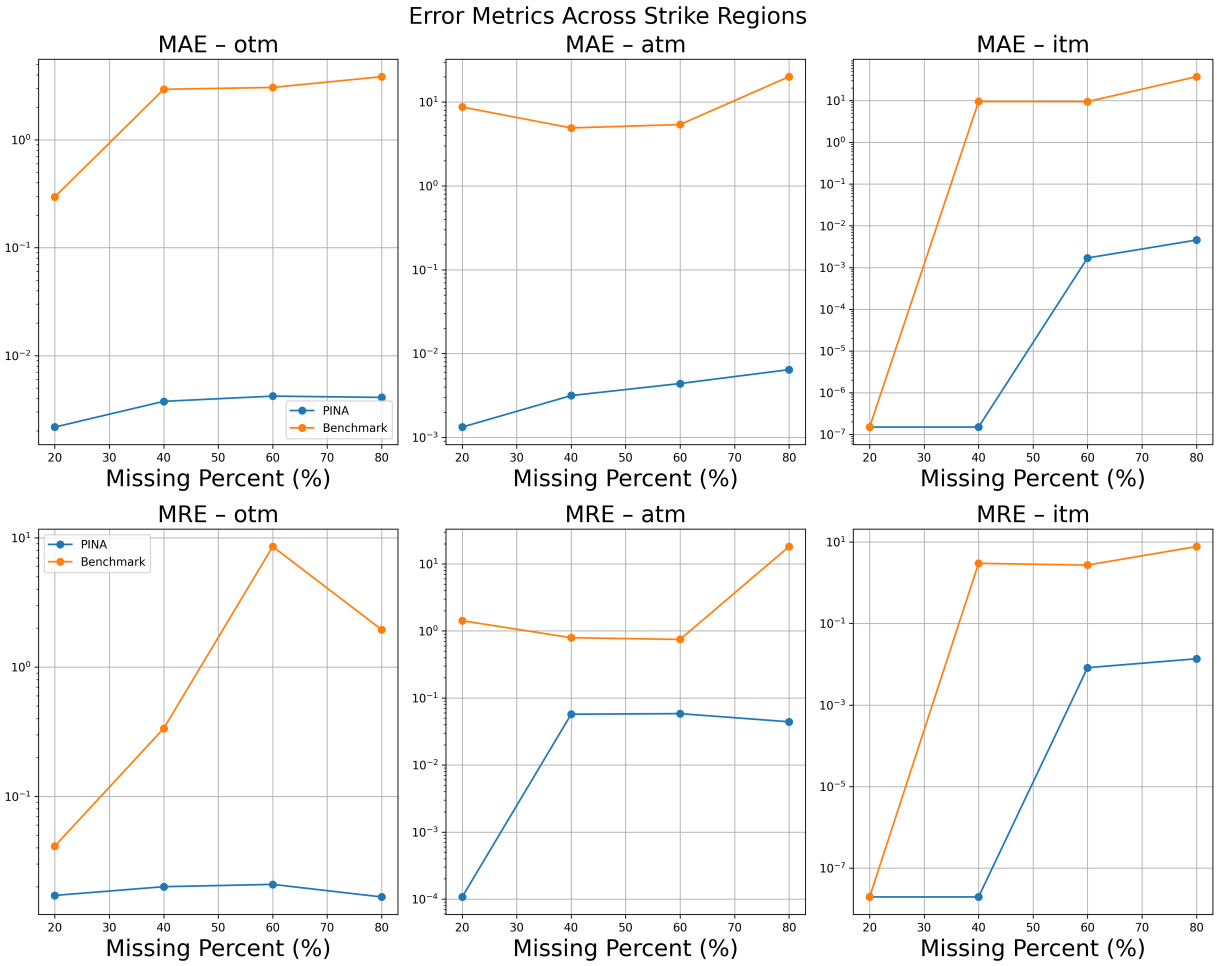


Figure A.15: American Options: Error Comparison for Varying Missing Percentages (4×4)