ARBITRAGE-FREE IMPLIED VOLATILITY SURFACE RECONSTRUCTION VIA ADDITIVE MARTINGALES

Anonymous authors
Paper under double-blind review

ABSTRACT

This work focuses on reconstructing a continuous implied volatility (IV) surface from a small number of option quotes, a fundamental task for option pricing and risk management. We propose **Additive Martingale** (ADDMART), a few-shot IV surface reconstruction framework that satisfies static no-arbitrage conditions *by construction*. Rather than predicting IVs directly, ADDMART maps sparse quotes to non-negative increment densities of an exponential additive martingale, from which valid risk-neutral marginals and IVs are recovered. By incorporating the no-arbitrage structure directly into its architecture, rather than enforcing it through regularization in the learning objective, ADDMART produces IV surfaces that satisfy static no-arbitrage conditions throughout the domain. Extensive experiments on ten real-option datasets demonstrate that ADDMART achieves competitive reconstruction accuracy with no material static arbitrage violations, while directly enabling the estimation of VaR, CVaR, and crash probabilities.

1 INTRODUCTION

Options are among the most actively traded derivatives in modern financial markets, indispensable for risk transfer, hedging, and price discovery. Within the Black-Scholes-Merton (BSM) framework (Black & Scholes, 1973), market participants analyze and quote an option not through its dollar price but through its *implied volatility* (IV), the volatility input that reconciles the model price with the market price and forms a forward-looking summary of the market's expectation of future fluctuations. Aggregating across contracts yields the *implied-volatility surface* $\mathrm{IV}(k, \tau)$ over log-moneyness $k = \log(K/F)$ and time-to-maturity τ, which is the central pricing and risk-management object on a derivatives desk (Gatheral, 2011). In addition, the IV surface gives access to the risk-neutral density (RND): via Breeden-Litzenberger (Breeden & Litzenberger, 1978), its convexity in strike recovers the full underlying distribution on which downstream tail-risk and exotic pricing depend. Yet at any instant the market quotes only a sparse, discrete set of strikes and maturities, whereas desks need the *entire continuous* surface to price illiquid strikes, hedge complex portfolios, and compute risk measures consistently.

Reconstructing the entire continuous IV surface from sparse option quotes is a long-standing problem. Classical approaches include parametric models such as SVI (Gatheral, 2011), SSVI (Gatheral & Jacquier, 2014) and SABR Hagan et al. (2002), non-parametric arbitrage-aware interpolation and smoothing (Kahalé, 2004; Fengler, 2009), and model-based calibration through local or stochastic volatility dynamics (Dupire, 1997; Heston, 1993). More recently, learning-based methods have demonstrated flexibility of neural networks for surface smoothing Ackerer et al. (2020); Chataigner et al. (2020), generative modeling (Ning et al., 2023; Vuletić & Cont, 2024) and amortized reconstruction across historical surfaces (Zhang et al., 2023; Zhuang & Wu, 2025; Zhang, 2025). In particular, HyperIV (Yang et al., 2025) formulates the problem in the practically important *few-shot, real-time, arbitrage-free* regime: given only 9 reference quotes, it reconstructs the entire IV surface using no-arbitrage regularization in *a single forward pass*. By mapping quotes to a small pointwise network, HyperIV is fast and has competitive cross-asset performance.

Despite advancements, existing learning-based methods for IV surface construction have two major limitations. First, HyperIV and deep-smoothing approaches (Ackerer et al., 2020) enforce static no-arbitrage conditions, such as butterfly and calendar constraints, through penalties evaluated on a finite grid. Such penalties introduce an accuracy-regularization trade-off. In addition, they only

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

provide soft and grid-dependent control to enforce static no-arbitrage conditions. Second, direct IV prediction does not ensure a valid risk-neutral density (RND): applying the Breeden-Litzenberger derivative (Breeden & Litzenberger, 1978) can produce negative densities and consequently invalid tail-risk measures. We address both issues by changing the model output.

To address the issues above, we propose **Additive Martingale** (ADDMART), a learning-based method to reconstruct IV surface satisfying static no-arbitrage conditions *by construction* through a martingale process. As demonstrated in Figure 1, ADDMART uses a permutation-invariant parameterized set encoder (Zaheer et al., 2017) to map a few reference quotes to non-negative increment densities of an exponential additive martingale. Full-support linear convolution then produces valid martingale marginals, from which call prices and IVs are recovered through a non-parametric decoder. Our method connects amortized few-shot reconstruction with the classical density and convex-order literature (Bahra, 1997; Brigo & Mercurio, 2002; Dugas et al., 2000; Hirsch et al., 2011). The properties of the martingales generated by the encoder guarantee that the IV surface satisfies the static no-arbitrage conditions globally (Kellerer, 1972; Hirsch et al., 2011), which enables us to train the models without any additional constraints.

Our contributions are threefold. (1) We introduce AddMart, an amortized framework that maps sparse option quotes to flexible increment distributions and reconstructs IV surfaces through an exponential additive martingale, producing valid risk-neutral marginals and enforcing static no-arbitrage by construction. (2) We establish no-arbitrage guarantees for the full-support decoder and call-price interpolation, and bound the calendar residual induced by support truncation. The resulting marginals directly support option-implied VaR, CVaR, and crash-probability calculations without numerical differentiation. (3) Across ten real-option datasets, AddMart achieves competitive reconstruction accuracy while exhibiting no material butterfly or calendar violations across five seeds and 2,500 audited model-date surfaces over the 16-month test period, without per-surface parameter optimization at inference.

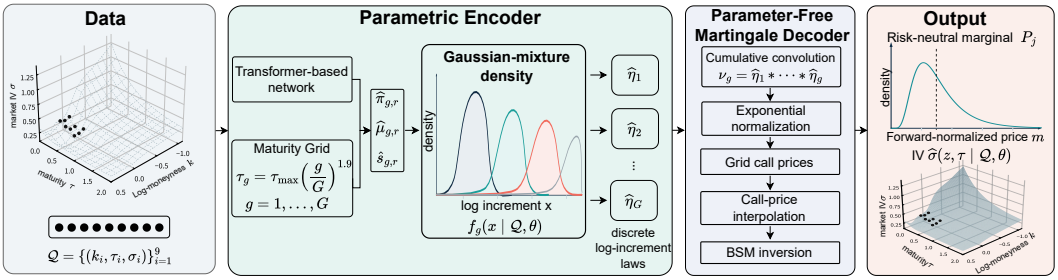

Figure 1: **The ADDMART pipeline.** Reference quotes are encoded into a latent representation for maturity-wise increment prediction, followed by parameter-free deterministic pricing and inversion.

2 RELATED WORK

Arbitrage-Free Surface and Density Models. Risk neutral densities have long been fitted with log-normal mixtures (Bahra, 1997; Brigo & Mercurio, 2002), and convexity architectures make butterfly freeness structural (Dugas et al., 2000; Amos et al., 2017). The most accurate models in practice remain per surface calibrations such as SSVI (Gatheral & Jacquier, 2014) and Deep Smoothing (Ackerer et al., 2020), but both require expensive optimization per surface and meet no arbitrage only softly, as a numerical penalty rather than a guarantee. A complementary route specifies an arbitrage free valuation first and recovers the supporting martingale (Carr & Madan, 2005; Madan & Yor, 2002); modeling the log price by an *additive* process makes this constructive, since the marginals of a genuine martingale increase in convex order, giving calendar freeness, while each marginal is a valid density, giving butterfly freeness (Carr & Torricelli, 2021; Azzone & Torricelli, 2025). Neural variants exist: Ning et al. (2023) train a variational autoencoder over the parameters of additive and Levy models to generate arbitrage free surfaces, and Lin & Liu (2024) fit an additive logistic martingale whose deterministic term structure is a feed forward network. ADDMART shares this additive martingale backbone but is quote conditioned and amortized few shot via a permutation invariant set encoder (Zaheer et al., 2017), places a *nonparametric* mixture density on each

increment rather than a fixed parametric family, which we show is decisive for accuracy, and makes calendar freeness hard by construction through linear convolution rather than a soft penalty.

Few-Shot Amortization. HyperIV (Yang et al., 2025), VolNP (Zhuang & Wu, 2025), and Operator Deep Smoothing (Wiedemann et al., 2025) learn shared mappings from observed quotes to IV surfaces, avoiding per-surface parameter optimization at inference. Their treatment of financial constraints differs: HyperIV and the published Operator Deep Smoothing method use no-arbitrage penalties, whereas VolNP uses SABR-derived pretraining followed by likelihood-based fitting to market quotes. These penalties and learned priors do not provide a global structural guarantee of static no-arbitrage. Consequently, the risk-neutral distributions recovered from their predicted surfaces are not guaranteed to be non-negative. Concurrent work emits neural Gaussian mixture densities with closed form prices (van den Berg, 2026), but in a model surrogate setting rather than from sparse market quotes and without a cross maturity martingale. ADDMART focuses on reconstruction from sparse market quotes, combining amortized inference with a common martingale construction that produces valid marginals and enforces consistency across maturities.

3 PRELIMINARIES

We summarize the frequently used notation in this work in Table 4 of Appendix A. In this work, although listed equity and ETF options are American-style, we use liquid OTM quotes, for which immediate exercise has no value, and express them as European-equivalent Black–76 IVs using parity-implied forwards that account for prevailing carry. This convention limits early-exercise and dividend effects in our contemporaneous reconstruction setting, so the formal guarantee applies to the resulting European-equivalent surface. The Black-Scholes-Merton (BSM) framework provides a closed-form pricing formula for European call options. Given the underlying asset's current price S_0, the strike price K, the risk-free interest rate r, the time-to-maturity τ, and the volatility σ, the call price is given by $C = S_0\mathcal{N}(d_1) - Ke^{-r\tau}\mathcal{N}(d_2)$, where $d_1 = \frac{\ln(S_0/K)+(r+\sigma^2/2)\tau}{\sigma\sqrt{\tau}}$, $d_2 = d_1 - \sigma\sqrt{\tau}$, and $\mathcal{N}$ denotes the cumulative distribution function of the standard normal distribution. Because all parameters except the volatility σ are directly observable in the market, equating the BSM model price to the observed market option price allows us to back out the unique *implied volatility (IV)*.

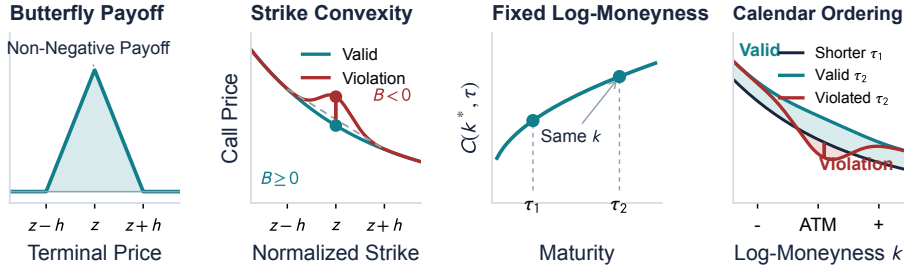

Figure 2: Butterfly and calendar restrictions implied by static no-arbitrage.

To facilitate our further analysis, it is convenient to work with forward-normalized quantities. Let $k = \log(K/F(\tau))$ and $M_\tau = S_\tau/F(\tau)$ where $F(\tau)$ is the parity-implied forward price and S_τ is the asset price at maturity. Under the pricing measure $\mathbb{Q}$, $\mathbb{E}_\mathbb{Q}[M_\tau] = 1$. We define the forward-normalized, undiscounted call price as $C(k,\tau) = \mathbb{E}_\mathbb{Q}[(M_\tau - e^k)^+]$ or equivalently $c(z,\tau) = E_\mathbb{Q}[(M_\tau - z)^+]$ where $z = e^k = K/F(\tau)$ is the normalized strike. Furthermore, we have $\forall\tau$, $c(0,\tau) = 1$ and $\lim_{z\to+\infty} c(z,\tau) = 0$. In this work, we aim to learn a surface mapping from k and τ to implied volatility $\mathrm{IV}(k,\tau)$ based on observable quotes.

Static No-Arbitrage Conditions. A valid option IV surface must satisfy both strike-wise and cross-maturity no-arbitrage conditions (Carr & Madan, 2005; Gatheral & Jacquier, 2014), corresponding to the absence of butterfly and calendar arbitrage, respectively. The butterfly constraint requires the call price $c(\cdot,\tau)$ to be non-increasing and globally convex in normalized strike z, i.e., $\partial c(z,\tau)/\partial z \le 0$ and $\partial^2 c(z,\tau)/\partial z^2 \ge 0$. By the Breeden–Litzenberger theorem (Breeden & Litzenberger, 1978), the distributional second derivative $\partial^2 c/\partial z^2$ is the risk-neutral marginal measure of M_τ and is therefore non-negative. The calendar constraint requires the call price $c(z,\cdot)$ to be non-decreasing in maturity, i.e., $\forall 0 \le \tau_1 \le \tau_2$, $c(z,\tau_1) \le c(z,\tau_2)$. Figure 2 illustrates the butterfly and calendar restrictions implied by static no-arbitrage.

Problem Setting. Given N option quotes for an underlying asset $\mathcal{Q} = \{(k_n, \tau_n, \sigma_n)\}_{n=1}^N$ where σ_n is implied volatility indicated by the market price, our goal is to learn a model $\widehat{\sigma}(z, \tau|\mathcal{Q}, \theta)$ parameterized by θ that reconstruct the full implied-volatility surface subject to the static no-arbitrage conditions discussed above. Unlike per-surface calibration, the learned model should generate the predicted IV in a single forward pass. At a high level, the learning problem is:

$$\begin{aligned} &\min_{\theta} \mathbb{E}_{(k,\tau,\sigma)\sim\mathcal{T}}\, l(\widehat{\sigma}(z, \tau|\mathcal{Q}, \theta) - \sigma) \\ s.t.\ \forall z, \tau, \frac{\partial \widehat{c}(z,\tau)}{\partial z} \leq 0, &\frac{\partial^2 \widehat{c}(z,\tau)}{\partial z^2} \geq 0, \quad \forall z, \tau_1, \tau_2, [\widehat{c}(z, \tau_1) - \widehat{c}(z, \tau_2)](\tau_1 - \tau_2) \geq 0. \end{aligned} \tag{1}$$

where $\mathcal{T}$ denote the training set consisting of sampled quotes to optimize θ, $\widehat{c}(z, \tau)$ is the estimated forward-normalized, undiscounted call price based on $\widehat{\sigma}(z, \tau|\mathcal{Q}, \theta)$ and l is the loss objective function. In general, the learning problem aims to minimize the discrepancy between the market IV and the estimated IV subject to static no-arbitrage conditions.

4 METHODOLOGY: ADDMART

We introduce ADDMART in this section, including the exponential additive-martingale construction (Section 4.1), the resulting by-construction static no-arbitrage guarantees (Section 4.2), the amortized architecture that generates estimated IV from sparse quotes (Section 4.3), and the end-to-end training and inference pipeline (Section 4.4).

4.1 MARTINGALE CONSTRUCTION FOR THE FORWARD-NORMALIZED PRICE

On a maturity grid $0 = \tau_0 < \tau_1 < \ldots < \tau_G$ and a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_{\tau_j}\}_{j=0}^G, \mathbb{Q})$, we consider G independent random variables $\{L_g\}_{g=1}^G$ with probability laws $\{\eta_g\}_{g=1}^G$, each satisfying $\mathbb{E}_{\mathbb{Q}}[e^{L_g}] < \infty$. We define the additive state $A_{\tau_j} := \sum_{g\leq j} L_g$ and its normalized exponential as the normalized terminal price M_{τ_j}. Based on the independent variable, we have:

$$M_{\tau_j} := \frac{e^{A_{\tau_j}}}{\mathbb{E}_{\mathbb{Q}}[e^{A_{\tau_j}}]} = \prod_{g\leq j} \frac{e^{L_g}}{\mathbb{E}_{\mathbb{Q}}[e^{L_g}]}. \tag{2}$$

Each multiplicative $\frac{e^{L_g}}{\mathbb{E}_{\mathbb{Q}}[e^{L_g}]}$ in (2) is non-negative and has unit expectation under $\mathbb{Q}$. In addition, since L_{j+1} is independent of $\mathcal{F}_{\tau_j}$, we have $\mathbb{E}_{\mathbb{Q}}[M_{\tau_{j+1}} \mid \mathcal{F}_{\tau_j}] = M_{\tau_j}$. Therefore, $\{M_{\tau_j}\}_{j=0}^G$ is a non-negative martingale with $\forall j, \mathbb{E}_{\mathbb{Q}}[M_{\tau_j}] = 1$.

Let $\nu_j := \eta_1 * \cdots * \eta_j$ denote the probability law of A_{τ_j} and define $Z_j := \mathbb{E}_{\mathbb{Q}}[e^{A_{\tau_j}}] = \int_{\mathbb{R}} e^x \nu_j(dx)$. Therefore, we can calculate the forward-normalized, undiscounted call price $c(z, \tau_j)$ as follows:

$$c(z, \tau_j) = \mathbb{E}_{\mathbb{Q}}[(M_{\tau_j} - z)^+] = \int_{\mathbb{R}} \left(\frac{e^x}{Z_j} - z\right)^+ \nu_j(dx) \tag{3}$$

Equivalently, the risk-neutral marginal P_j of M_{τ_j} is the pushforward of ν_j through $x \mapsto e^x/Z_j$.

Equation (3) calculates the forward-normalized, undiscounted call price at the maturity grid. For more general maturity $\tau \in (\tau_j, \tau_{j+1})$, we let $\lambda = (\tau - \tau_j)/(\tau_{j+1} - \tau_j)$ and interpolate the prices at the maturity grid by $c(z, \tau) = (1 - \lambda)c(z, \tau_j) + \lambda c(z, \tau_{j+1})$. This is equivalent to using the mixture marginal $\mathsf{P}_\tau = (1 - \lambda)\mathsf{P}_j + \lambda \mathsf{P}_{j+1}$. With $c(z, \tau)$, we can calculate the corresponding $\mathrm{IV}(k, \tau)$ by BSM framework introduced in Section 3.

4.2 ARBITRAGE-FREE BY CONSTRUCTION

Our main theorem indicates the forward-normalized price $c(z, \tau)$ constructed by the martingale above is free of static arbitrage, so we do not need additional constraints to enforce this requirement.

Theorem 1 (Static no-arbitrage on the modeled maturity interval)**.** *Let $L_1, \ldots, L_G$ be independent real-valued random variables satisfying $\mathbb{E}_{\mathbb{Q}}[e^{L_g}] < \infty$. Construct the normalized exponential martingale and its call prices as in Section 4.1, with $M_0 = 1$, and interpolate the grid call prices*

linearly at fixed normalized strike. For every $\tau \in [0, \tau_G]$, the function $c(z, \tau)$ is non-increasing and convex in $z > 0$, with $(1 - z)^+ \leq c(z, \tau) \leq 1$. For every normalized fixed strike $z > 0$, $c(z, \tau)$ is non-decreasing in τ. Thus the surface satisfies both strike-wise and cross-maturity no-arbitrage conditions under the stated pricing assumptions.

To prove the theorem above, we notice that each marginal P_j of M_{τ_j} is a valid probability law and that these marginals increase in convex order. Full-support convolution in the probability law of A_{τ_j} preserves non-negativity and unit mass, while the martingale property in Equation (2) implies convex-order monotonicity (Kellerer, 1972). Both properties hold for every admissible value of the increment laws $\{\eta_g\}_{g=1}^G$. The complete proof is provided in Appendix C.1.

As a direct consequence, each grid-maturity marginal emitted by ADDMART is a valid probability measure. Distributional quantities such as the CDF, survival function, generalized quantiles, VaR, CVaR, and crash probabilities can therefore be computed directly from the predicted marginal. The corresponding formulas are provided in Appendix C.2. Theorem 1 shows that the static no-arbitrage conditions are enforced by the construction in Section 4.1, rather than by the training objective. Therefore, ADDMART requires no butterfly or calendar arbitrage penalty during learning.

4.3 AMORTIZED NETWORK ARCHITECTURE

ADDMART employs a model parameterized by θ to follow the construction process introduced in Section 4.1 so that the generated surface is the arbitrage-free by construction. Figure 1 summarizes the architecture. It consists of a learned set encoder and increment-density head followed by a deterministic martingale decoder.

Parametric Encoder. The encoder maps the input quote reference set $\mathcal{Q} = \{(k_i, \tau_i, \sigma_i)\}_{i=1}^N$ to an estimation of the probability laws $\{\eta_g\}_{g=1}^G$ of log-increments $\{L_g\}_{g=1}^G$ with a model parameterized by θ. Specifically, each quote (k_i, τ_i, σ_i) is mapped to a 128-dimensional feature by a linear transformation, and the resulting sequence is processed by two Transformer encoder layers with two attention heads, followed by mean pooling over the quote dimension and a linear projection to obtain the 64-dimensional representation $h \in \mathbb{R}^{64}$. We use no positional encoding and mean pooling over the quote dimension to make the representation h invariant to the ordering of input quotes. Self-attention allows each quote to be interpreted relative to the remaining maturities and moneyness levels, while the pooled representation summarizes the sparse surface as a whole. We then generate probability laws $\{\eta_g\}_{g=1}^G$ based on learning-based representation h. In particular, we choose $G = 30$ maturity buckets up to $\tau_{\max} = 2$ years, with an exponential interpolation $\tau_g = \tau_{\max}(g/G)^{1.9}$ so that more buckets are allocated near expiry. For each bucket, we concatenate h with time features $(\tau_g, \sqrt{\tau_g})$ and employ a two-layer MLP with width 128 and GELU Hendrycks & Gimpel (2016) activations to map the time-augmented feature $\widehat{h}_g = (h, \tau_g, \sqrt{\tau_g})$ to the 24-dimensional output, which is then organized tuples $\{\widehat{\pi}_{g,r}, \widehat{\mu}_{g,r}, \widehat{s}_{g,r}\}_{r=1}^8$. We then model the corresponding probability law η_g as a Gaussian mixture $f_g(x|\mathcal{Q}, \theta) = \sum_{r=1}^R \pi_{g,r}\mathcal{N}(x|\mu_{g,r}, s_{g,r}^2)$ where $\mathcal{N}(x|\mu_{g,r}, s_{g,r}^2)$ is the probability density of Gaussian distribution with its weight $\pi_{g,r}$, mean $\mu_{g,r}$ and variance $s_{g,r}^2$ calculated as:

$$\pi_{g,r} = \frac{e^{\widehat{\pi}_{g,r}}}{\sum_{r=1}^R e^{\widehat{\pi}_{g,r}}}, \quad \mu_{g,r} = 0.3\sqrt{\Delta\tau_g}\widehat{\mu}_{g,r}, \quad s_{g,r} = \sqrt{\Delta\tau_g}\left[0.02 + \text{softplus}(\widehat{s}_{g,r} - 1)\right]. \quad (4)$$

where $\Delta\tau_g = \tau_g - \tau_{g-1}$. We scale the mean and standard deviation of each Gaussian component by $\sqrt{\Delta\tau_g}$ to impose a natural diffusive time scale. The mixture of eight Gaussian components allows η_g to represent asymmetric and multimodal increment laws. We discretize this law on a fixed, equally spaced grid $\{x_i\}_{i=1}^X \subset [-5, 5]$, with $X = 2048$, by evaluating and normalizing the mixture density:

$$\widehat{\eta}_g = \sum_{i=1}^X m_{g,i}\,\delta_{x_i}, \qquad m_{g,i} = \frac{f_g(x_i \mid \mathcal{Q}, \theta)}{\sum_{\ell=1}^X f_g(x_\ell \mid \mathcal{Q}, \theta)}. \quad (5)$$

Here, δ_{x_i} denotes the Dirac probability measure at x_i. The dependence of $\widehat{\eta}_g$ and $m_{g,i}$ on $\mathcal{Q}$ and θ is omitted for brevity. The resulting discrete law approximates the continuous increment law η_g and enables the convolution and pricing operations in Section 4.1. Its probability masses are non-negative, sum to one, and are differentiable with respect to θ, enabling end-to-end training.

Parameter-free Decoder. Encoder's output $\widehat{\eta}_g(x|\mathcal{Q},\theta)$ are passed to the decoder for construction in Section 4.1. Specifically, we estimate the probability law of A_{τ_j} by convolution, and then the forward-normalized, undiscounted call price $\widehat{c}(z,\tau|\mathcal{Q},\theta)$ by Equation (3) and interpolation. The decoder contains no trainable parameters and supplies the martingale structure of Theorem 1. We apply BSM framework on the log-moneyness $k = \log z$, time-to-maturity τ and estimated call price $\widehat{c}(z,\tau|\mathcal{Q},\theta)$ to obtain the estimated implied volatility, which is a function of z, τ conditioned on $\mathcal{Q}$, θ and denoted as $\widehat{\sigma}(z,\tau|\mathcal{Q},\theta)$.

4.4 Training and Inference

Training Phase. We follow Yang et al. (2025) to include 9 reference quotes as the conditioning set $\mathcal{Q}$ and another 256 quotes sampled with replacement as the training set $\mathcal{T}$ to train the model. Each quote (k_n, τ_n, σ_n) include its log-moneyness $k_n = \log z_n$, time-to-maturity τ_n and market IV σ_n. For each quote in the training set, we apply parametric encoder and non-parametric decoder discussed in Section 4.3 on k_n and τ_n to obtain the estimated IV $\widehat{\sigma}(e^{k_n}, \tau_n|\mathcal{Q},\theta)$. We minimize the estimated discrepancy between the estimated IV and the market IV as follows:

$$\min_{\theta} \mathcal{L}(\theta) = \frac{1}{|\mathcal{T}|} \sum_{n=1}^{|\mathcal{T}|} l_\beta\big(\widehat{\sigma}(e^{k_n}, \tau_n|\mathcal{Q},\theta) - \sigma_n\big) \tag{6}$$

where l_β is a Huber loss Huber (1992) with $\beta = 0.05$. The quadratic part of l_β supplies smooth gradients for accurately reconstructed contracts, while its linear tail limits the influence of noisy or stale quotes. Compared with the generic form in (1), our objective in (6) has no penalty term, regularization term or any constraints, because static no-arbitrage is supplied by the decoder and is therefore independent of the attained loss value. We defer more technical details of the training phase in Appendix D.4.

Inference Phase. In the inference phase, the trained encoder processes the sparse reference quotes $\mathcal{Q}$ once and emits all G increment distributions in a single forward pass. The decoder then compute the estimated IV based on convolution, interpolation and inverse BSM. It returns both the queried IV surface and the underlying risk-neutral marginals. The implementation is detailed in Appendix D.2.

5 Experiments

5.1 Experiment settings

Data and protocol. We use OPRA top-of-book option quotes obtained from a commercial data provider,[1] and price each contract at the bid–ask midpoint. We infer the forward from put–call parity and invert implied volatility under Black–76 (Black, 1976). Each surface contains OTM options and is conditioned on 9 reference quotes: the contracts nearest to three target maturities (7, 30, and 90 days) and three deltas (0.50, 0.25, and -0.25). All remaining contracts are reconstruction targets. We evaluate 7 daily books: SPY, QQQ, IWM, AAPL, GOOGL, META, NVDA, and 3 intraday-minute books: SPY, QQQ, IWM.[2] All datasets use the same strict out-of-time split: surfaces before February 1, 2025 are used for training and validation, and surfaces from that date onward for testing. The resulting ten datasets contain approximately 40.1 million option quotes across 70,027 surfaces. Full preprocessing and dataset statistics are provided in Appendix E.

Evaluation metrics. We report IV-MAE (F.1) over all reconstruction targets (*overall*) and over the liquid region $\{|k| \le 0.1,\ \mathrm{ttm} \ge 14\mathrm{d}\}$ (*reliable*), with mean and standard deviation over five seeds for amortized methods. Static arbitrage is evaluated for every method using the same finite-difference grid. We report violations of calendar monotonicity ($\partial_\tau w \ge 0$) and butterfly positivity ($g(k,\tau) \ge 0$), where $w = \sigma^2\tau$. Metric definitions are given in Appendix F.1.

Baselines. We compare AddMart with the amortized few-shot models, including HyperIV (Yang et al., 2025), GNO (Li et al., 2020), and VAE (Kingma & Welling, 2014), and with the per-surface methods, including SSVI (Gatheral & Jacquier, 2014) and Deep Smoothing (Ackerer et al., 2020). All amortized methods use the same 9 reference quotes, out-of-time split, and five random seeds. Implementation and training details are reported in Appendix F.2.

[1] https://massive.com

[2] SPY, QQQ, and IWM are liquid ETFs tracking the S&P 500, Nasdaq-100, and Russell 2000, respectively.

5.2 RECONSTRUCTION PERFORMANCE

Tables 1 and 2 summarize reconstruction accuracy and static-arbitrage violations under the unified evaluation protocol. All amortized methods are evaluated over five seeds, whereas SSVI and Deep Smoothing are fitted independently to each test surface. On the daily data, ADDMART achieves the lowest overall IV-MAE on six of the seven assets, with HyperIV performing slightly better only on NVDA. The reliable-region comparison is closer: ADDMART leads on SPY, QQQ, and IWM, while HyperIV is marginally better on the four single names. Consequently, the asset-equal daily averages favor ADDMART for overall reconstruction (1.65 versus 1.72 for HyperIV), while HyperIV has a small advantage in the reliable region (1.27 versus 1.29). At the minute scale, ADDMART is best on both metrics for SPY and IWM, whereas GNO performs best on QQQ. Averaged across the three minute-scale assets, ADDMART again gives the lowest reliable and overall errors, at 0.61 and 1.08, respectively. The seed-level results also show that ADDMART is substantially more stable than GNO, whose large standard deviations on SPY and IWM indicate frequent optimization instability.

The arbitrage audit provides a sharper distinction between the methods. On a common 801×30 full-support price grid, ADDMART produces no calendar or butterfly violations beyond numerical precision across any asset or scale. SSVI also attains zero violations, but requires a separate optimization for every surface. Deep Smoothing remains close to arbitrage-free in pairwise terms, although small butterfly violations persist. In contrast, the amortized neural baselines exhibit material violations: on the daily assets, HyperIV violates 0.60–2.62% of calendar pairs and 1.63–8.17% of butterfly pairs, while GNO reaches calendar-violation rates as high as 22.28%. The violations become more pronounced at the minute scale, particularly for GNO and VAE. Thus, among methods that reconstruct a surface in a single amortized forward pass, ADDMART provides the strongest overall accuracy while uniquely preserving the no-arbitrage conditions to numerical precision.

Table 1: Reliable and overall IV-MAE ($\times 10^{-2}$, mean$\pm$std over 5 seeds). Lowest per metric within each asset row in bold. ‡SSVI and Deep Smoothing are per-surface fits (one optimization per surface, not amortized over a training seed), reported as a single value.

		ADDMART		HyperIV		GNO		VAE		SSVI‡		Deep Sm.‡	
Scale	Asset	Reliable	Overall	Reliable	Overall	Reliable	Overall	Reliable	Overall	Reliable	Overall	Reliable	Overall
Daily	SPY	**0.56** $\pm$0.02	**0.89** $\pm$0.05	0.57 $\pm$0.02	0.99 $\pm$0.03	0.62 $\pm$0.05	0.95 $\pm$0.07	1.33 $\pm$0.15	1.75 $\pm$0.11	1.29	2.24	1.07	2.23
	QQQ	**0.68** $\pm$0.03	**1.25** $\pm$0.05	0.74 $\pm$0.03	1.31 $\pm$0.04	0.90 $\pm$0.10	1.45 $\pm$0.13	1.30 $\pm$0.15	1.98 $\pm$0.07	1.20	2.19	1.11	2.57
	IWM	**0.77** $\pm$0.03	**1.17** $\pm$0.03	0.78 $\pm$0.08	1.27 $\pm$0.09	1.09 $\pm$0.53	1.77 $\pm$0.94	1.19 $\pm$0.12	1.93 $\pm$0.14	1.16	1.95	1.20	2.57
	AAPL	**1.18** $\pm$0.07	**1.40** $\pm$0.05	**1.18** $\pm$0.07	1.55 $\pm$0.08	1.52 $\pm$0.25	1.77 $\pm$0.31	1.76 $\pm$0.04	2.37 $\pm$0.10	2.08	3.14	2.90	5.11
	GOOGL	2.22 $\pm$0.14	**2.37** $\pm$0.11	**2.07** $\pm$0.25	2.43 $\pm$0.24	2.59 $\pm$0.32	2.76 $\pm$0.24	3.12 $\pm$0.09	3.42 $\pm$0.10	3.21	4.01	5.87	8.10
	META	1.82 $\pm$0.16	**1.95** $\pm$0.12	**1.79** $\pm$0.07	2.03 $\pm$0.08	2.05 $\pm$0.22	2.30 $\pm$0.26	2.29 $\pm$0.04	2.73 $\pm$0.10	3.69	4.46	6.80	10.40
	NVDA	**1.80** $\pm$0.11	2.54 $\pm$0.07	**1.80** $\pm$0.11	**2.46** $\pm$0.12	2.66 $\pm$0.65	3.08 $\pm$0.55	3.31 $\pm$0.29	3.50 $\pm$0.27	3.83	4.97	5.19	7.72
Minute	SPY	0.56 $\pm$0.03	**0.92** $\pm$0.04	0.57 $\pm$0.03	0.96 $\pm$0.04	**0.55** $\pm$0.08	0.95 $\pm$0.16	1.13 $\pm$0.01	1.46 $\pm$0.03	1.25	2.10	1.07	2.07
	QQQ	0.65 $\pm$0.07	1.23 $\pm$0.07	0.66 $\pm$0.04	1.20 $\pm$0.06	**0.59** $\pm$0.05	**1.01** $\pm$0.06	1.13 $\pm$0.01	1.69 $\pm$0.03	1.20	2.29	1.06	2.44
	IWM	**0.61** $\pm$0.04	**1.09** $\pm$0.05	0.63 $\pm$0.01	1.11 $\pm$0.03	0.77 $\pm$0.15	1.31 $\pm$0.28	1.00 $\pm$0.01	1.51 $\pm$0.00	1.19	2.02	1.16	2.33

Table 2: Calendar and butterfly arbitrage-violation pair rates (%) on the common 801×30 full-support price grid, using a numerical tolerance of 10^{-10}. Amortized methods are pooled over 5 seeds; ‡SSVI and Deep Smoothing are per-surface fits and are reported as a single value. Lowest per metric within each asset row in bold.

		ADDMART		HyperIV		GNO		VAE		SSVI‡		Deep Sm.‡	
Scale	Asset	Calendar	Butterfly	Calendar	Butterfly	Calendar	Butterfly	Calendar	Butterfly	Calendar	Butterfly	Calendar	Butterfly
Daily	SPY	**0**	**0**	0.7310	1.6295	15.6910	4.4026	0.0546	8.4843	**0**	**0**	**0**	0.0113
	QQQ	**0**	**0**	1.0047	2.3451	22.2821	5.2844	0.2145	7.8386	**0**	**0**	**0**	0.0222
	IWM	**0**	**0**	0.6008	4.8701	18.8903	7.6170	0.3043	6.8536	**0**	**0**	**0**	0.0642
	AAPL	**0**	**0**	1.0690	6.6294	20.6356	5.0270	0.0785	5.0351	**0**	**0**	**0**	0.0160
	GOOGL	**0**	**0**	0.9958	7.9404	14.6969	3.4538	0.2408	5.5603	**0**	**0**	0.0051	0.0195
	META	**0**	**0**	2.6205	8.1659	11.6960	3.5165	0.0906	4.4891	**0**	**0**	**0**	0.0108
	NVDA	**0**	**0**	1.6370	7.8105	7.2435	2.3606	0.3796	2.9705	**0**	**0**	**0**	**0**
Minute	SPY	**0**	**0**	0.3779	1.9884	22.4997	8.7293	21.0067	18.5972	**0**	**0**	0.0053	0.0336
	QQQ	**0**	**0**	1.5006	6.1951	25.4820	8.7914	20.2924	13.6071	**0**	**0**	0.0162	0.0274
	IWM	**0**	**0**	0.7306	4.2616	23.2808	8.3368	18.2714	13.4086	**0**	**0**	**0**	0.0745

5.3 DOWNSTREAM DISTRIBUTION VALIDITY.

Figure 3 examines whether the reconstructed surfaces imply coherent risk-neutral distributions. The upper row shows the Breeden–Litzenberger density and the lower row its CDF, with insets highlight-

ing small violations that are obscured at the full-distribution scale. Across the representative SPY and NVDA surfaces, AddMart produces non-negative densities and monotone CDFs by construction. For the baselines that directly predict or fit the IV surface, second differentiation amplifies local curvature errors in the reconstructed surface, producing negative density and locally decreasing CDFs. The effect is particularly pronounced for the one-year NVDA example, where HyperIV has 21.8% integrated negative mass despite competitive IV reconstruction accuracy.

Distributional validity also affects downstream risk estimates. Appendix G.1 evaluates option-implied downside CVaR against a full-information reference fitted using all available quotes. AddMart achieves the lowest CVaR error in five of the six asset–horizon settings, while its predicted marginals remain valid in every evaluated case. The baseline distributions frequently require ex-post clipping and renormalization before CVaR can be computed, making the resulting tail estimates dependent on the repair procedure. These quantities describe risk-neutral tail exposure implied by option prices and should not be interpreted as forecasts of realized losses.

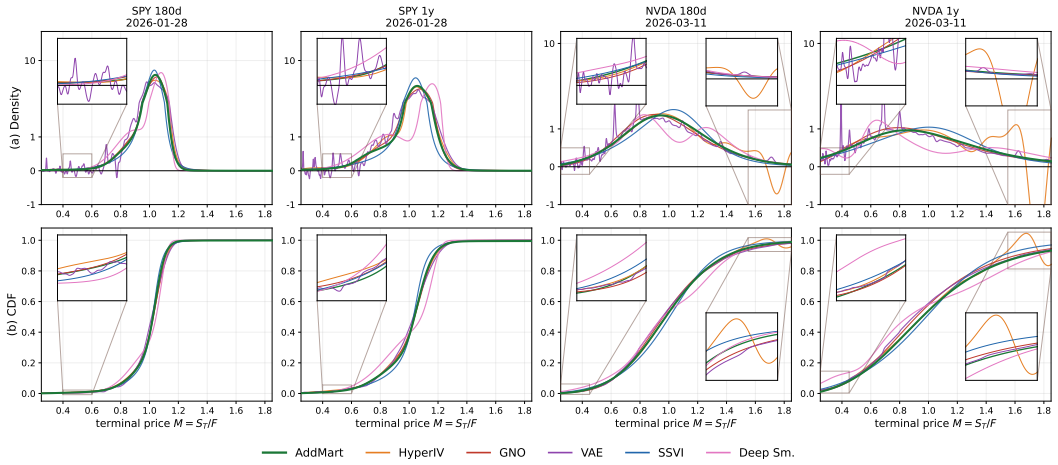

Figure 3: **Risk-neutral densities and CDFs.** AddMart guarantees valid densities and monotone CDFs, while IV prediction-based baselines may violate these properties, particularly for one-year NVDA. Insets highlight the tails.

Table 3: Key ablations on SPY and NVDA daily (mean±std over five seeds). IV-MAE is reported in $\times 10^{-2}$. All mixture and training-loss variants use the exact linear decoder and have zero calendar and butterfly violations.

(a) Convolution

Variant	SPY	NVDA
Reliable IV-MAE		
Linear exact	0.56±0.04	1.84±0.06
Linear crop	0.56±0.02	1.82±0.06
Circular	**0.55±0.04**	**1.76±0.06**
Calendar violation (%)		
Linear exact	**0.000**	**0.000**
Linear crop	3.855	0.023
Circular	3.652	0.070

(b) Architecture and training

Group	Variant	SPY	NVDA
Mixture n_{inc}	1	2.99±0.15	2.11±0.13
	2	0.67±0.05	**1.74±0.07**
	4	**0.55±0.05**	1.75±0.11
	8 (default)	0.56±0.04	1.84±0.06
	16	0.56±0.03	1.75±0.09
Training loss	Implicit IV (default)	0.56±0.04	**1.84±0.06**
	Newton IV	**0.53±0.02**	4.91±0.35
	Price	0.56±0.02	1.88±0.09

5.4 Further Analysis

Ablations. Table 3 examines the main architectural and training choices on the SPY and NVDA daily datasets. The convolution scheme primarily affects the structural guarantee rather than reconstruction accuracy. The exact linear decoder achieves zero calendar violations on both assets, while cropped and circular convolution (Pfister, 2017) introduce measurable violations despite similar reliable-region IV-MAE. In particular, circular convolution attains slightly lower IV-MAE than the exact decoder, but produces calendar-violation rates of 3.652% on SPY and 0.070% on NVDA.

This comparison confirms that the no-arbitrage property follows from the full-support linear construction rather than from reconstruction accuracy alone. The mixture ablation shows that a single component is insufficient, especially on SPY, whereas using two or more components substantially improves accuracy. Increasing the mixture size beyond this point yields only modest and asset-dependent changes. The default n_{inc}=8 therefore provides a practical capacity choice without relying on fine-grained tuning. The training-loss comparison shows a similar pattern. Direct price supervision performs comparably to the implicit-IV objective, while Newton-based IV inversion improves SPY accuracy but becomes unstable on NVDA. We therefore use the implicit-IV objective as the default because it provides consistently strong performance across both assets.

Analysis of Reference Budget. We examine how AddMart uses increasingly informative reference sets by varying the number of observed contracts $K \in \{2, 3, 5, 9, 16, 25\}$. Each setting uses the same architecture, is trained over five seeds, and is evaluated on an identical set of non-reference targets. As shown in Figure 4, additional references generally improve reconstruction accuracy, although the gains are asset-dependent and not strictly monotone: SPY approaches saturation between $K = 16$ and $K = 25$, whereas NVDA performs best at $K = 16$. This suggests that both reference-set size and quote placement affect performance. We use $K = 9$ in the main experiments as a practical balance between accuracy and sparse-market applicability, with a structured 3-maturity $\times$ 3-delta design.

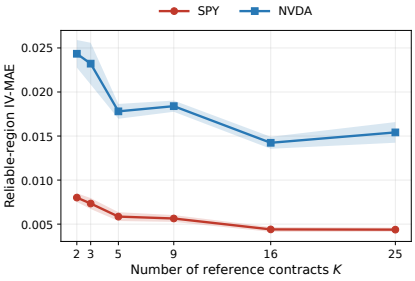

Figure 4: Effect of reference-set size on AddMart for SPY and NVDA on daily data. Lines and shaded regions report the mean and standard deviation of reliable-region IV-MAE over five seeds.

Cross-Asset Performance. We evaluate zero-shot transfer by applying each source-asset checkpoint to the other daily assets, using only the target surface's nine reference quotes and no retraining. Across the 42 directed source–target pairs, AddMart attains the lowest overall mean MAE (5.51×10^{-2}) and outperforms HyperIV on 76.2% of the pairs, while HyperIV is stronger in the reliable region (4.81 versus 5.33×10^{-2}). Transfer is most effective between related broad-market ETFs and deteriorates, often asymmetrically, when volatility levels and smile shapes differ substantially. These results indicate that AddMart learns transferable surface structure, although its representation is not fully asset-invariant. The complete protocol, summary statistics, and directed transfer matrices are reported in Appendix G.4.

6 Conclusion

In conclusion, AddMart reframes few-shot implied volatility surface reconstruction by learning the increment laws of an exponential additive martingale rather than directly predicting an unconstrained surface. This design enforces static no-arbitrage by construction and directly yields valid risk-neutral marginal distributions without additional arbitrage penalties. Across ten real-market datasets, AddMart achieves competitive reconstruction accuracy among amortized methods while enabling downstream distributional risk analysis, demonstrating the benefit of learning structured probability objects rather than only fitting surface values.

The current design nevertheless has several limitations. The additive increment structure induces long-horizon skew flattening, which can reduce accuracy on LEAPS, and the model is less effective for highly asymmetric or heavy-tailed underlyings such as VIX. In addition, serial convolution and implied-volatility inversion limit its suitability for ultra-high-frequency inference, although the resulting latency remains practical at the minute scale. Future work could address these limitations through richer state-dependent or nonstationary martingale constructions that better preserve persistent skew and heavy tails, more efficient convolution and implied-volatility inversion for lower-latency inference, and extensions such as uncertainty quantification, adaptive reference selection, and cross-asset pretraining to improve robustness under sparse, noisy, or out-of-distribution market conditions. Detailed analyses of these failure modes, together with latency and memory measurements, are provided in Appendix G.5.

AI USE STATEMENT

In this work, we used generative AI tools to assist with experimental planning and organization. Additionally, we used generative AI tools for scientific visualization, table formatting, and language editing to improve the readability and presentation of the manuscript. All AI-assisted content, including experimental suggestions, figures, tables, and edited text, was carefully reviewed and verified by the authors for correctness and consistency with the underlying results. Generative AI tools were not used to independently interpret results or draw scientific conclusions. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Damien Ackerer, Natasa Tagasovska, and Thibault Vatter. Deep smoothing of the implied volatility surface. *Advances in Neural Information Processing Systems*, 33:11552–11563, 2020.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Michele Azzone and Lorenzo Torricelli. Explicit option pricing with additive processes. *SIAM Journal on Financial Mathematics*, 16(3):747–802, 2025.

Bhupinder Bahra. Implied risk-neutral probability density functions from option prices: theory and application. Technical report, Bank of England Working Paper No. 66, 1997.

Fischer Black. The pricing of commodity contracts. *Journal of Financial Economics*, 3(1-2):167–179, 1976.

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

Douglas T Breeden and Robert H Litzenberger. Prices of state-contingent claims implicit in option prices. *Journal of business*, pp. 621–651, 1978.

Damiano Brigo and Fabio Mercurio. Lognormal-mixture dynamics and calibration to market volatility smiles. *International Journal of Theoretical and Applied Finance*, 5(4):427–446, 2002.

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

Peter Carr and Lorenzo Torricelli. Additive logistic processes in option pricing. *Finance and Stochastics*, 25(4):689–724, 2021.

Marc Chataigner, Stéphane Crépey, and Matthew Dixon. Deep local volatility. *Risks*, 8(3):82, 2020.

Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2000.

Bruno Dupire. Pricing and hedging with smiles. *Mathematics of derivative securities*, 1(1):103–111, 1997.

Matthias R Fengler. Arbitrage-free smoothing of the implied volatility surface. *Quantitative Finance*, 9(4):417–428, 2009.

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

Jim Gatheral and Antoine Jacquier. Arbitrage-free svi volatility surfaces. *Quantitative Finance*, 14 (1):59–71, 2014.

Patrick S Hagan, Deep Kumar, Andrew S Lesniewski, and Diana E Woodward. Managing smile risk. *The Best of Wilmott*, 1(1):249–296, 2002.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

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.

Francis Hirsch, Christophe Profeta, Bernard Roynette, and Marc Yor. *Peacocks and associated martingales, with explicit constructions*. Springer Science & Business Media, 2011.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Nabil Kahalé. An arbitrage-free interpolation of volatilities. *Risk*, 17(5):102–106, 2004.

Hans G Kellerer. Markov-komposition und eine anwendung auf martingale. *Mathematische Annalen*, 198(3):99–122, 1972.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.

Jimin Lin and Guixin Liu. Neural term structure of additive process for option pricing. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pp. 695–702, 2024.

Dilip B. Madan and Marc Yor. Making markov martingales meet marginals: With explicit constructions. *Bernoulli*, 8(4):509–536, 2002. ISSN 13507265. URL `http://www.jstor.org/stable/3318851`.

Robert C Merton et al. Theory of rational option pricing. 1971.

Brian Ning, Sebastian Jaimungal, Xiaorong Zhang, and Maxime Bergeron. Arbitrage-free implied volatility surface generation with variational autoencoders. *SIAM Journal on Financial Mathematics*, 14(4):1004–1027, 2023.

Henry Pfister. Discrete-time signal processing. *Lecture Note, pfister. ee. duke. edu/courses/ece485/dtsp. pdf*, 4:2, 2017.

R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

Thijs van den Berg. Fast, reliable, and error-bounded option pricing with pretrained neural networks: A gjr–garch study, 2026. URL `https://arxiv.org/abs/2606.15502`.

Milena Vuletić and Rama Cont. Volgan: a generative model for arbitrage-free implied volatility surfaces. *Applied Mathematical Finance*, 31(4):203–238, 2024.

Ruben Wiedemann, Antoine Jack Jacquier, and Lukas Gonon. Operator deep smoothing for implied volatility. In *International Conference on Learning Representations*, volume 2025, pp. 96833–96872, 2025.

Yongxin Yang, Wenqi Chen, Chao Shu, and Timothy Hospedales. HyperIV: Real-time implied volatility smoothing. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Jian'an Zhang. A risk-neutral neural operator for arbitrage-free spx-vix term structures. *arXiv preprint arXiv:2511.06451*, 2025.

Wenyong Zhang, Lingfei Li, and Gongqiu Zhang. A two-step framework for arbitrage-free prediction of the implied volatility surface. *Quantitative Finance*, 23(1):21–34, 2023.

Jirong Zhuang and Xuan Wu. Meta-learning neural process for implied volatility surfaces with sabr-induced priors. *arXiv preprint arXiv:2509.11928*, 2025.

ETHICS STATEMENT

This study uses option-market quotes rather than human-subject or individual-level data. The proposed model produces implied-volatility surfaces and risk-neutral distributions; these are model-based quantities, not forecasts of realized returns. Although the construction enforces static no-arbitrage under its stated assumptions, this property does not eliminate market-data errors, model risk, or uncertainty in downstream decisions. The outputs should therefore not be treated as standalone investment or risk-management advice.

REPRODUCIBILITY STATEMENT

The paper specifies the model architecture, increment-law parameterization, martingale decoder, training objective, data preprocessing, reference-quote selection, chronological data split, baselines, and evaluation metrics in the main text and appendices. Amortized methods are evaluated with five independent random seeds. The experiments use OPRA option quotes obtained through a commercial data provider; access to the underlying data is subject to the provider's licensing terms. These details are intended to support replication by researchers with access to appropriately licensed data.

A NOTATION

Table 4 summarizes the notation used throughout the paper. We use lowercase letters for scalar quantities, bold lowercase letters for vectors, calligraphic letters for sets, and uppercase blackboard-bold letters for probability measures.

Table 4: Notation used throughout the paper.

Symbol	Domain or type	Description
Option market and pricing		
$B(k, s_{\text{tot}})$	$[0,1]$	Forward-normalized Black call price.
$C(k,\tau)$	$[0,1]$	Call price in log-moneyness coordinates, $C(k,\tau) = c(e^k,\tau)$.
$c(z,\tau)$	$[0,1]$	Forward-normalized undiscounted call price, $c(z,\tau) = \mathbb{E}_{\mathbb{Q}}[(M_\tau - z)^+]$.
d_+, d_-	$\mathbb{R}$	Black variables, $d_+ = -k/s_{\text{tot}} + s_{\text{tot}}/2$ and $d_- = -k/s_{\text{tot}} - s_{\text{tot}}/2$.
$\text{DF}(\tau)$	$(0,1]$	Discount factor for maturity τ, defined as $\text{DF}(\tau) = \exp[-r(\tau)\tau]$.
$\mathbb{E}_{\mathbb{Q}}, \mathbb{P}_{\mathbb{Q}}$	–	Expectation and probability under the pricing measure $\mathbb{Q}$.
$F(\tau)$	$\mathbb{R}_+$	Parity-implied forward price of the underlying for maturity τ.
k	$\mathbb{R}$	Log-moneyness, defined as $k = \log(K/F(\tau))$.
K	$\mathbb{R}_+$	Option strike price.
M_τ	$\mathbb{R}_+$	Forward-normalized terminal price, $M_\tau = S_\tau/F(\tau)$, with $\mathbb{E}_{\mathbb{Q}}[M_\tau] = 1$.
Φ, ϕ	–	CDF and density of the standard normal distribution.
q	$\mathbb{R}$	Continuous dividend yield or implied carry.
$\mathbb{Q}$	–	Pricing measure used for normalized option valuation.
$r(\tau)$	$\mathbb{R}$	Continuously compounded risk-free rate.
S_t	$\mathbb{R}_+$	Underlying asset price at observation time t.
S_τ	$\mathbb{R}_+$	Underlying asset price at maturity.

Continued on next page

Table 4 – continued

Symbol	Domain or type	Description
s_{tot}	$\mathbb{R}_+$	Black total volatility, $s_{\text{tot}} = \sigma\sqrt{\tau}$.
$\sigma(k,\tau)$	$\mathbb{R}_+$	Black implied volatility.
t	$\mathbb{R}_+$	Current observation time.
T	$\mathbb{R}_+$	Option expiration time.
$\tau = T - t$	$\mathbb{R}_+$	Time to maturity, measured in years.
$w(k,\tau)$	$\mathbb{R}_+$	Total implied variance, $w(k,\tau) = \sigma(k,\tau)^2\tau$.
$(x)^+$	$\mathbb{R}_+$	Positive part, defined as $(x)^+ = \max(x, 0)$.
$z = e^k$	$\mathbb{R}_+$	Normalized strike, equal to $K/F(\tau)$.
Additive martingale construction		
A_{τ_j}	$\mathbb{R}$	Cumulative additive state, $A_{\tau_j} = \sum_{g\le j} L_g$.
$\Delta\tau_g$	$\mathbb{R}_+$	Width of maturity bucket g, $\Delta\tau_g = \tau_g - \tau_{g-1}$.
η_g	$\mathcal{P}(\mathbb{R})$	Probability law of the additive increment L_g.
G	$\mathbb{N}$	Number of maturity buckets.
L_g	$\mathbb{R}$	Independent additive log-increment in bucket g.
M_{τ_j}	$\mathbb{R}_+$	Normalized exponential martingale, $$M_{\tau_j} = \frac{e^{A_{\tau_j}}}{Z_j} = \prod_{g\le j} Y_g.$$
ν_j	$\mathcal{P}(\mathbb{R})$	Law of A_{τ_j}, given by $\nu_j = \eta_1 * \cdots * \eta_j$.
$\lambda(\tau)$	$[0,1]$	Interpolation weight for $\tau \in (\tau_j, \tau_{j+1})$, $$\lambda(\tau) = \frac{\tau - \tau_j}{\tau_{j+1} - \tau_j}.$$
P_j	$\mathcal{P}(\mathbb{R}_+)$	Risk-neutral marginal law of M_{τ_j}, obtained by pushing ν_j forward through $x \mapsto e^x/Z_j$.
P_τ	$\mathcal{P}(\mathbb{R}_+)$	Interpolated marginal, $$\mathsf{P}_\tau = (1-\lambda)\mathsf{P}_j + \lambda\mathsf{P}_{j+1}.$$
$\preceq_{\text{cx}}$	–	Convex order between probability distributions or random variables.
τ_j	$\mathbb{R}_+$	The j-th maturity-grid point, with $0 = \tau_0 < \tau_1 < \cdots < \tau_G$.
Y_g	$\mathbb{R}_+$	Unit-mean multiplicative increment, $$Y_g = \frac{e^{L_g}}{\mathbb{E}_{\mathbb{Q}}[e^{L_g}]}.$$
Z_j	$\mathbb{R}_+$	Exponential normalizer, $$Z_j = \mathbb{E}_{\mathbb{Q}}[e^{A_{\tau_j}}] = \int_{\mathbb{R}} e^x \nu_j(dx).$$
$(\Omega, \mathcal{F}, \{\mathcal{F}_{\tau_j}\}_{j=0}^G, \mathbb{Q})$	–	Filtered probability space supporting the additive process.
Learned parameterization and grid implementation		
$\mathcal{Q}$	finite set	Sparse reference-quote set used to condition the network.

Continued on next page

Table 4 – continued

Symbol	Domain or type	Description
θ	–	Trainable parameters of the set encoder and increment head.
$f_{\theta,g}(x \mid \mathcal{Q})$	$\mathbb{R}_+$	Continuous Gaussian-mixture density predicted for increment g.
x_i	$\mathbb{R}$	The i-th point of the base log-return grid.
$m_{\theta,g,i}$	$[0,1]$	Normalized probability mass assigned to x_i for increment g, with $\sum_i m_{\theta,g,i} = 1$.
$\widehat{\eta}_{\theta,g}$	$\mathcal{P}(\mathbb{R})$	Discrete increment law used by the decoder, $\widehat{\eta}_{\theta,g} = \sum_i m_{\theta,g,i}\delta_{x_i}$.
$v_j,\ v_j^\star$	probability vectors	Cumulative base and Esscher-tilted log-state mass vectors at maturity τ_j.
$w_{j,i}$	$[0,1]$	Probability mass of the emitted discrete marginal $\widehat{\mathsf{P}}_j$ at atom $M_{j,i}$.
R	$\mathbb{N}$	Number of Gaussian components in each increment density.
X	$\mathbb{N}$	Number of points in the base log-return grid.

B Extended Background on Options, Volatility, and Arbitrage

B.1 Options and the Black Scholes Merton Model

A European call option is a foundational financial contract. It provides the buyer with the right, but absolutely no obligation, to purchase a specific underlying asset (such as a stock or an index) at a predetermined price called the strike price K, on a specific future date called the maturity date T. Let t denote the current time. The duration remaining until the option expires is known as the time to maturity, denoted by $\tau = T - t$. Because this contract limits downside risk while preserving upside potential, it carries a premium (the option price) that must be paid upfront.

Under the standard option pricing framework (Black & Scholes, 1973; Merton et al., 1971), the theoretical price of a European call option can be computed if we assume geometric Brownian motion for the underlying asset. Assuming a constant risk free interest rate r, a continuous dividend yield q, and a constant volatility σ representing the standard deviation of the asset returns, the price of a call option is given by the Black Scholes Merton formula:

$$C_{\mathrm{BSM}}(S_t, K, \tau, r, q, \sigma) = S_t e^{-q\tau}\Phi(d_1) - Ke^{-r\tau}\Phi(d_2)$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. The terms d_1 and d_2 are explicitly defined as:

$$d_1 = \frac{\ln(S_t/K) + (r - q + \frac{1}{2}\sigma^2)\tau}{\sigma\sqrt{\tau}}$$

$$d_2 = d_1 - \sigma\sqrt{\tau}$$

While the Black Scholes Merton mathematical model assumes that volatility σ is a constant parameter, real financial markets behave differently. Empirical market prices systematically deviate from this constant volatility assumption, demonstrating that the market prices different strike prices and different maturity dates with different effective volatilities.

B.2 Implied Volatility and the Volatility Surface

Because all inputs to the Black Scholes Merton formula—the current stock price S_t, the strike price K, the time to maturity τ, the risk free rate r, and the dividend yield q—are directly observable in

the market, practitioners use the formula in reverse. They treat the model as a universal mapping function rather than an absolute pricing truth.

Given an observable market price of a call option C_{mkt}, the *implied volatility*, denoted as σ_{imp}, is defined as the single unique scalar value that forces the theoretical model price to exactly equal the observed market price:

$$C_{\text{mkt}} = C_{\text{BSM}}(S_t, K, \tau, r, q, \sigma_{\text{imp}})$$

Because the option price strictly increases as volatility increases (a property known as positive vega, mathematically $\partial C_{\text{BSM}}/\partial\sigma > 0$), this mapping from price to implied volatility is unique and universally invertible. Quoting implied volatility instead of raw dollar prices allows traders to compare options across different underlying assets and different price levels using a standardized metric.

To further standardize the representation of these options, it is standard market practice to map the absolute strike price K into a relative measure called *log moneyness*, denoted by k. Let $F(\tau) = S_t e^{(r-q)\tau}$ denote the forward price of the underlying asset at maturity τ. The log moneyness is defined as:

$$k = \ln\left(\frac{K}{F(\tau)}\right)$$

When $k = 0$, the option is termed "at the money". When we compute the implied volatility across a continuous range of log moneyness k and maturities τ, we construct a two dimensional mathematical function $\text{IV}(k, \tau)$. This continuous geometric map is widely known as the ***Implied Volatility Surface***. It acts as the central pricing tool used by financial institutions to price exotic derivatives, hedge complex portfolios, and calculate future market risk scenarios.

B.3 Static Arbitrage and the Risk Neutral Density

For an implied volatility surface $\text{IV}(k, \tau)$ to be theoretically valid and economically tradable, it must be strictly free of static arbitrage. In finance, an arbitrage is a theoretical mechanism that guarantees a riskless profit with zero net investment. If a model generates a surface containing arbitrage opportunities, it produces mathematically absurd outputs, such as negative probability values.

According to the celebrated Breeden Litzenberger theorem (Breeden & Litzenberger, 1978), the second derivative of the undiscounted call price with respect to the strike price directly yields the market probability mass function, commonly referred to as the risk neutral density.

Complete absence of static arbitrage requires the surface to rigorously satisfy two primary geometric conditions:

1. **Butterfly No Arbitrage:** A butterfly spread is an option strategy involving three different strike prices. For this strategy to never yield a guaranteed riskless payout, the implied risk neutral density must be nonnegative across all possible strikes. Mathematically, this demands that the call price must be globally convex with respect to the strike price K:

   $$\frac{\partial^2 C}{\partial K^2} \geq 0$$

   If this second derivative were negative in any region, it would imply that the market assigns a negative probability to the asset reaching that specific price range, which is fundamentally impossible.

2. **Calendar No Arbitrage:** A calendar spread involves options with the same strike price but different maturity dates. Because an option with a longer time to maturity provides the holder with more time for the asset to move favorably, it must be at least as valuable as an option with a shorter maturity. Therefore, the forward option prices must not decrease as the time to maturity increases:

   $$\frac{\partial C}{\partial \tau} \geq 0$$

   A violation here would imply that time carries negative value.

When translating these concepts into forward normalized spaces, preserving calendar no arbitrage is exactly equivalent to requiring that the sequence of asset distributions over time increases in *convex*

order. In probability theory, a sequence of distributions increasing in convex order forms a martingale structure known as a *peacock* (from the French *processus croissant pour l'ordre convexe*). Therefore, explicitly modeling the implied volatility surface through the lens of additive martingales strictly guarantees that both the butterfly and calendar no arbitrage conditions are satisfied by design.

C Proofs and Additional Theory

Throughout this section, $\eta_1, \ldots, \eta_G$ are probability laws on $\mathbb{R}$, $L_g \sim \eta_g$ are independent under $\mathbb{Q}$ with $\mathbb{E}_{\mathbb{Q}}[e^{L_g}] < \infty$, and

$$Y_g := \frac{e^{L_g}}{\mathbb{E}_{\mathbb{Q}}[e^{L_g}]}, \qquad M_{\tau_j} = \prod_{g \le j} Y_g, \qquad \mathcal{F}_{\tau_j} = \sigma(L_1, \ldots, L_j).$$

C.1 Proof of Theorem 1

Proof. Each Y_g is non-negative and satisfies $\mathbb{E}_{\mathbb{Q}}[Y_g] = 1$. Since Y_{j+1} is independent of $\mathcal{F}_{\tau_j}$ and M_{τ_j} is $\mathcal{F}_{\tau_j}$-measurable,

$$\mathbb{E}_{\mathbb{Q}}\left[M_{\tau_{j+1}} \mid \mathcal{F}_{\tau_j}\right] = M_{\tau_j}\mathbb{E}_{\mathbb{Q}}[Y_{j+1}] = M_{\tau_j}. \tag{7}$$

Hence $\{M_{\tau_j}\}$ is a non-negative martingale with unit expectation.

For any convex function φ for which the expectations exist, conditional Jensen's inequality and Equation 7 give

$$\varphi(M_{\tau_j}) = \varphi\big(\mathbb{E}_{\mathbb{Q}}[M_{\tau_{j+1}} \mid \mathcal{F}_{\tau_j}]\big) \le \mathbb{E}_{\mathbb{Q}}[\varphi(M_{\tau_{j+1}}) \mid \mathcal{F}_{\tau_j}].$$

Taking expectations yields

$$\mathbb{E}_{\mathbb{Q}}[\varphi(M_{\tau_j})] \le \mathbb{E}_{\mathbb{Q}}[\varphi(M_{\tau_{j+1}})]. \tag{8}$$

Thus, $M_{\tau_j} \preceq_{\mathrm{cx}} M_{\tau_{j+1}}$, and the grid-maturity marginals form a peacock (Kellerer, 1972; Hirsch et al., 2011).

For fixed normalized strike $z > 0$, the call payoff $\varphi_z(m) = (m - z)^+$ is convex in m. Applying Equation 8 gives $c(z, \tau_j) \le c(z, \tau_{j+1})$, which is the calendar condition at grid maturities.

For fixed $m \ge 0$, the map $z \mapsto (m - z)^+$ is non-increasing and convex. Integration against the marginal law of M_{τ_j} therefore shows that $z \mapsto c(z, \tau_j)$ is non-increasing and convex. Its distributional second derivative is the non-negative marginal measure:

$$\partial^2_{zz} c(\cdot, \tau_j) = \mathsf{P}_j \ge 0.$$

Moreover,

$$c(0, \tau_j) = \mathbb{E}_{\mathbb{Q}}[M_{\tau_j}] = 1, \qquad \lim_{z \to \infty} c(z, \tau_j) = 0, \qquad c(z, \tau_j) \ge (1 - z)^+.$$

The lower bound follows from Jensen's inequality, and the limit follows from dominated convergence because $(M_{\tau_j} - z)^+ \le M_{\tau_j}$.

For $\tau \in (\tau_j, \tau_{j+1})$, the interpolation defined in Section 4.1 is a convex combination of the endpoint call curves and therefore remains non-increasing and convex in z. Since $c(z, \tau_{j+1}) - c(z, \tau_j) \ge 0$, it is also non-decreasing in τ within each maturity interval. Equivalently, the interpolated marginal $\mathsf{P}_\tau = (1 - \lambda)\mathsf{P}_j + \lambda \mathsf{P}_{j+1}$ is a probability measure with unit mean.

It remains to verify that the arbitrage-free price admits a BSM IV representation. For fixed k and positive maturity, write the normalized Black call price as $B(k, s_{\mathrm{tot}})$, where $s_{\mathrm{tot}} = \sigma\sqrt{\tau}$ is total volatility. The function $s_{\mathrm{tot}} \mapsto B(k, s_{\mathrm{tot}})$ is continuous and strictly increasing because $\partial B / \partial s_{\mathrm{tot}} = \phi(d_+) > 0$. Its limiting values are $B(k, 0^+) = (1 - e^k)^+$ and $B(k, \infty) = 1$. Therefore every interior price $c \in ((1 - e^k)^+, 1)$ has a unique non-negative implied volatility. Combining these properties proves butterfly- and calendar-arbitrage-freeness at every strike and maturity. □

C.2 Distributional Risk Functionals

Proof. At a grid maturity, let

$$\widehat{\mathsf{P}}_j = \sum_i w_{j,i}\delta_{M_{j,i}}, \qquad w_{j,i} \geq 0, \qquad \sum_i w_{j,i} = 1.$$

Then $\widehat{\mathsf{P}}_j$ is a finitely supported probability measure. For every measurable function φ that is integrable under $\widehat{\mathsf{P}}_j$,

$$\mathbb{E}_{\widehat{\mathsf{P}}_j}[\varphi(M_{\tau_j})] = \sum_i w_{j,i}\varphi(M_{j,i}) = \int \varphi \, d\widehat{\mathsf{P}}_j. \tag{9}$$

Its CDF and survival function are

$$F_j(a) = \sum_i w_{j,i}\mathbf{1}\{M_{j,i} \leq a\}, \qquad \overline{F}_j(a) = \sum_i w_{j,i}\mathbf{1}\{M_{j,i} > a\}.$$

They are monotone and take values in $[0, 1]$. The generalized quantile

$$F_j^{-1}(\alpha) = \inf\{a : F_j(a) \geq \alpha\}$$

is therefore well defined. Quantile-based VaR and crash probabilities such $F_j(a) = \sum_i w_{j,i}\mathbf{1}\{M_{j,i} \leq a\}$ and $\overline{F}_j(a) = \sum_i w_{j,i}\mathbf{1}\{M_{j,i} > a\}$, respectively. They are monotone and take values in $[0, 1]$, so the generalized quantile $F_j^{-1}(\alpha) = \inf\{a : F_j(a) \geq \alpha\}$ is well defined.

For any loss variable $\mathsf{L} = \ell(M_{\tau_j})$ that is finite on the emitted atoms, its CVaR has the finite convex representation

$$\mathrm{CVaR}_p(L) = \min_{c\in\mathbb{R}} \left\{ c + \frac{1}{1-p}\sum_i w_{j,i}(L_i - c)^+ \right\}, \qquad L_i = \ell(M_{\tau_j}). \tag{10}$$

following Rockafellar et al. (2000). The objective is convex in c because it is a non-negative weighted sum of convex hinge functions. Every quantity above is evaluated under one valid probability measure and is therefore well defined. □

C.3 Approximation Error of a Cropped Decoder

The formal AddMart decoder uses full-support convolution. A fixed-width training decoder may instead crop and renormalize intermediate convolutions to reduce memory use. A bound based only on discarded probability mass is insufficient because a small amount of far-right mass can have a large effect after exponentiation. The relevant quantity is an exponentially weighted measure discrepancy.

Lemma 2 (Calendar residual under an approximate marginal)**.** *Let ν_j be the exact law of A_{τ_j} and let $\widetilde{\nu}_j$ be any approximate probability law with finite exponential moment. Define*

$$Z_j = \int e^x \nu_j(dx), \qquad \widetilde{Z}_j = \int e^x \widetilde{\nu}_j(dx), \qquad \delta_j = \frac{1}{Z_j}\int e^x \left|\nu_j - \widetilde{\nu}_j\right|(dx),$$

where $|\nu_j - \widetilde{\nu}_j|$ denotes the total-variation measure of the signed difference. Let c_j and $\widetilde{c}_j$ be the corresponding forward-normalized call prices, each using its own exponential normalizer. Then, uniformly over normalized strike z,

$$|c_j(z) - \widetilde{c}_j(z)| \leq 2\delta_j. \tag{11}$$

Consequently,

$$\widetilde{c}_{j+1}(z) - \widetilde{c}_j(z) \geq -2(\delta_j + \delta_{j+1}).$$

Proof. Define $g_z(m) = (m - z)^+$, $\psi_Z(x) = g_z(e^x/Z_j)$, and $\psi_{\widetilde{Z}}(x) = g_z(e^x/\widetilde{Z}_j)$. Since $0 \leq \psi_Z(x) \leq e^x/Z_j$,

$$\left|\int \psi_Z \, d\nu_j - \int \psi_Z \, d\widetilde{\nu}_j\right| \leq \frac{1}{Z_j}\int e^x \left|\nu_j - \widetilde{\nu}_j\right|(dx) = \delta_j.$$

The hinge function is one-Lipschitz, so

$$\left|\psi_Z(x) - \psi_{\widetilde{Z}}(x)\right| \le e^x \left|\frac{1}{Z_j} - \frac{1}{\widetilde{Z}_j}\right|.$$

Integrating under $\widetilde{\nu}_j$ gives

$$\int \left|\psi_Z - \psi_{\widetilde{Z}}\right| d\widetilde{\nu}_j \le \frac{|Z_j - \widetilde{Z}_j|}{Z_j} \le \delta_j.$$

The triangle inequality proves Equation 11. Since the exact prices satisfy $c_{j+1}(z) \ge c_j(z)$ by Theorem 1,

$$\begin{aligned}\widetilde{c}_{j+1}(z) - \widetilde{c}_j(z) &= \left[\widetilde{c}_{j+1}(z) - c_{j+1}(z)\right] + \left[c_{j+1}(z) - c_j(z)\right] + \left[c_j(z) - \widetilde{c}_j(z)\right] \\ &\ge -2\delta_{j+1} - 2\delta_j.\end{aligned}$$

□

The bound converges to zero whenever the approximate marginals converge to the exact marginals in exponentially weighted total variation. The full-support inference decoder has no structural cropping error; only floating-point roundoff remains.

C.4 Long-Horizon Behavior of the Additive Class

Proposition 3 (Standardized log-skew decay)**.** *Suppose $L_g = \mu_g + s\sqrt{\Delta\tau}\,\xi_g$, where $s > 0$ and $\{\xi_g\}$ are i.i.d. random variables with zero mean, unit variance, and $\rho := \mathbb{E}[|\xi_1|^3] < \infty$. Then*

$$\mathrm{Skew}(A_{\tau_j}) = \frac{\mathbb{E}[\xi_1^3]}{\sqrt{j}} = \mathcal{O}(j^{-1/2}). \tag{12}$$

Moreover, the centered and standardized additive state converges in distribution to a standard Gaussian as $j \to \infty$.

Proof. Independence gives

$$\kappa_2(A_{\tau_j}) = js^2\Delta\tau, \qquad \kappa_3(A_{\tau_j}) = js^3(\Delta\tau)^{3/2}\mathbb{E}[\xi_1^3].$$

Therefore,

$$\mathrm{Skew}(A_{\tau_j}) = \frac{\kappa_3(A_{\tau_j})}{\kappa_2(A_{\tau_j})^{3/2}} = \frac{js^3(\Delta\tau)^{3/2}\mathbb{E}[\xi_1^3]}{(js^2\Delta\tau)^{3/2}} = \frac{\mathbb{E}[\xi_1^3]}{\sqrt{j}}.$$

The Lyapunov ratio with exponent one is

$$\frac{\sum_{g\le j}\mathbb{E}[|L_g - \mu_g|^3]}{\left(\sum_{g\le j}\mathrm{Var}(L_g)\right)^{3/2}} = \frac{\rho}{\sqrt{j}} \longrightarrow 0.$$

The Lyapunov central limit theorem therefore yields the stated Gaussian limit. □

Proposition 3 establishes a structural tendency of the additive log-state toward Gaussianity under repeated aggregation. This is consistent with the long-horizon smile flattening discussed in conclusion part; the proposition concerns standardized log-return skewness and does not assume a pointwise convergence rate for the IV skew.

D Architecture, Algorithms, and Implementation Details

D.1 Network Architecture and Increment Parameterization

Each reference quote is represented by $u_i = (k_i, \tau_i, \sigma_i) \in \mathbb{R}^3$. A linear layer first embeds each quote into $\mathbb{R}^{128}$, after which two Transformer encoder layers process the resulting set. Each layer

Table 5: AddMart architecture used in the main experiments. The number of reference quotes K may vary without changing the network.

Component	Tensor mapping	Configuration
Quote embedding	$\mathbb{R}^{K\times 3} \to \mathbb{R}^{K\times 128}$	Linear projection
Set encoder	$\mathbb{R}^{K\times 128} \to \mathbb{R}^{K\times 128}$	2 layers, 2 heads
Surface pooling	$\mathbb{R}^{K\times 128} \to \mathbb{R}^{64}$	Mean pooling, linear map
Bucket conditioning	$\mathbb{R}^{64} \to \mathbb{R}^{66}$	Append $(\tau_g, \sqrt{\tau_g})$
Density head	$\mathbb{R}^{66} \to \mathbb{R}^{3R}$	MLP $(66, 128, 128, 3R)$
Increment law	$\mathbb{R}^{3R} \to \mathbb{R}^{X}$	$R = 8$, $X = 2048$
Martingale decoder	$\{m_g\}_{g=1}^{G} \to \{c(k,\tau)\}$	$G = 30$, linear convolution
IV inversion	$c(k,\tau) \to \widehat{\sigma}(k,\tau)$	Black bisection
Total trainable parameters		236,120

uses two attention heads, a feed-forward width of 128, ReLU activation, zero dropout, residual connections, and layer normalization. We use no positional encoding. Consequently, the Transformer is permutation equivariant over the input quotes, and the subsequent mean pooling makes the surface representation permutation invariant. A final linear map produces $h \in \mathbb{R}^{64}$.

We use a non-uniform maturity grid with $G = 30$ points,

$$\tau_g = \tau_{\max}\left(\frac{g}{G}\right)^{\gamma}, \qquad \tau_{\max} = 2, \qquad \gamma = 1.9,$$

and define $\tau_0 = 0$ and $\Delta\tau_g = \tau_g - \tau_{g-1}$. The power grid allocates more buckets to short maturities, where option prices and implied volatilities vary most rapidly.

For each bucket, the density head receives $\bar{h}_g = [h, \tau_g, \sqrt{\tau_g}] \in \mathbb{R}^{66}$. It consists of two hidden layers of width 128 with GELU activations, followed by an output layer of width $3R$, where $R = 8$. The output is divided into mixture logits $a_{g,r}$, raw locations $\widetilde{\mu}_{g,r}$, and raw scales $\ell_{g,r}$. These are transformed as

$$\pi_{g,r} = \frac{\exp(a_{g,r})}{\sum_{r'=1}^{R}\exp(a_{g,r'})}, \qquad \mu_{g,r} = 0.3\sqrt{\Delta\tau_g}\,\widetilde{\mu}_{g,r},$$

and

$$s_{g,r} = \sqrt{\Delta\tau_g}\,[0.02 + \mathrm{softplus}(\ell_{g,r} - 1)]\,.$$

The factor $\sqrt{\Delta\tau_g}$ provides a diffusive time scale, while the mixture components allow the increment law to be asymmetric and multimodal.

Let $\{x_i\}_{i=1}^{X}$ be an equally spaced grid with $X = 2048$ and $x_i \in [-5, 5]$. Conditional on the reference set $\mathcal{Q}$, the density head defines

$$f_{\theta,g}(x \mid \mathcal{Q}) = \sum_{r=1}^{R}\pi_{g,r}\frac{1}{\sqrt{2\pi}\,s_{g,r}}\exp\left(-\frac{(x-\mu_{g,r})^2}{2s_{g,r}^2}\right).$$

Evaluating this density on the grid and normalizing gives

$$\widetilde{m}_{\theta,g,i} = f_{\theta,g}(x_i \mid \mathcal{Q}), \qquad m_{\theta,g,i} = \frac{\widetilde{m}_{\theta,g,i}}{\sum_{\ell=1}^{X}\widetilde{m}_{\theta,g,\ell}}.$$

Thus, $m_{\theta,g,i} \geq 0$ and $\sum_i m_{\theta,g,i} = 1$. The resulting discrete increment law is

$$\widehat{\eta}_{\theta,g} = \sum_{i=1}^{X} m_{\theta,g,i}\delta_{x_i}.$$

The complete model contains 236,120 trainable parameters. Table 5 summarizes the architecture.

D.2 Full-Support Convolution and Exact Inference

For readability, write $m_{g,i} := m_{\theta,g,i}$. For each increment law, define its exponential normalizer and Esscher-tilted mass by

$$d_g = \sum_i m_{g,i} e^{x_i}, \qquad m^\star_{g,i} = \frac{m_{g,i} e^{x_i}}{d_g}.$$

At maturity τ_j, the cumulative base and tilted laws are represented by

$$v_j = m_1 *_{\mathrm{lin}} \cdots *_{\mathrm{lin}} m_j, \qquad v^\star_j = m^\star_1 *_{\mathrm{lin}} \cdots *_{\mathrm{lin}} m^\star_j,$$

with exponential normalizer $Z_j = \prod_{g \le j} d_g$.

If each increment law contains X atoms, the cumulative law at bucket j contains $j(X-1)+1$ atoms. The largest required support therefore has

$$L = G(X-1) + 1$$

atoms. The implementation zero-pads every FFT to at least this length, using the next power of two for computational efficiency. Spectral multiplication then implements full linear convolution without circular wraparound or intermediate support truncation.

The exact decoder performs these operations in float64. Inverse FFTs may produce negative coefficients at the level of floating-point roundoff. We clip such coefficients to zero and renormalize each mass vector. This correction changes option prices only at numerical precision while preventing remote-tail errors from being amplified by the exponential transformation.

Directly forming the atoms $e^{A_{\tau_j}}/Z_j$ is numerically unstable in the far-right tail. The tilted representation permits call prices to be computed from tail probabilities instead. For normalized strike $z = e^k$, let $a_j(k) = k + \log Z_j$. Then

$$\begin{aligned} c(z, \tau_j) &= \mathbb{E}_{\mathbb{Q}} \left[\left(\frac{e^{A_{\tau_j}}}{Z_j} - z \right)^+ \right] \\ &= \frac{1}{Z_j} \mathbb{E}_{\mathbb{Q}} \left[e^{A_{\tau_j}} \mathbf{1}\{A_{\tau_j} \ge a_j(k)\} \right] - z\, \mathbb{P}_{\mathbb{Q}} \left(A_{\tau_j} \ge a_j(k) \right) \\ &= \nu^\star_j([a_j(k), \infty)) - z\, \nu_j([a_j(k), \infty)), \end{aligned} \tag{13}$$

where

$$d\nu^\star_j(x) = \frac{e^x}{Z_j}\, d\nu_j(x).$$

After computing suffix sums of v_j and $v^\star_j$, pricing a strike requires one threshold lookup and two tail probabilities. For maturities between grid points, we use the call-price interpolation defined in Section 4.1. The interpolated price is then converted to IV using Black inversion. Algorithm 1 gives the complete inference procedure.

The exact decoder also exposes the complete discrete marginal laws $\{\widehat{\mathsf{P}}_j\}_{j=1}^G$ and their maturity-dependent atom locations. We use these objects to evaluate distributional risk functionals and to audit probability mass, exponential normalization, butterfly convexity, and calendar monotonicity directly in price space.

D.3 Training Protocol

At each timestamp, the main experiments use a fixed 3×3 maturity–delta reference set as input. We independently draw $N = 256$ target contracts with replacement from the complete surface. Reference contracts remain in this sampling pool during training. For validation and testing, however, the nine conditioning contracts are removed from the target set, so the reported metrics measure reconstruction at contracts that were not provided to the network.

Training uses batches of 16 surfaces and the objective in Equation 18. All target contracts receive unit weight; the optional reliable-region weight is set to one. The reliable evaluation region, defined by $|k| \le 0.1$ and $\tau \ge 14$ days, therefore does not receive additional weight during training.

Algorithm 1 Exact AddMart inference

Require: Reference set $\mathcal{Q} = \{(k_i, \tau_i, \sigma_i)\}_{i=1}^{K}$; queries $\mathcal{T} = \{(k_n, \tau_n)\}_{n=1}^{N}$
$h \leftarrow \mathrm{SetEncoder}_\theta(\mathcal{Q})$
for $g = 1, \dots, G$ **do**
  $(\pi_g, \mu_g, s_g) \leftarrow \mathrm{DensityHead}_\theta(h, \tau_g, \sqrt{\tau_g})$
  Evaluate the Gaussian mixture on $\{x_i\}_{i=1}^{X}$ and normalize it to obtain m_g
  $d_g \leftarrow \sum_i m_{g,i} e^{x_i}$
  $m^\star_{g,i} \leftarrow m_{g,i} e^{x_i} / d_g$
end for
$v_0 \leftarrow \delta_0$, $v_0^\star \leftarrow \delta_0$, and $Z_0 \leftarrow 1$
for $j = 1, \dots, G$ **do**
  $v_j \leftarrow v_{j-1} *_{\mathrm{lin}} m_j$
  $v_j^\star \leftarrow v_{j-1}^\star *_{\mathrm{lin}} m_j^\star$
  $Z_j \leftarrow Z_{j-1} d_j$
  Compute suffix sums of v_j and $v_j^\star$
end for
for each query $(k_n, \tau_n) \in \mathcal{T}$ **do**
  Locate the maturity buckets bracketing τ_n
  At each bracketing bucket j, set $a_j \leftarrow k_n + \log Z_j$
  Compute $c_{n,j} \leftarrow \nu_j^\star([a_j, \infty)) - e^{k_n} \nu_j([a_j, \infty))$
  Interpolate the bracketing call prices to obtain c_n
  $\widehat{\sigma}_n \leftarrow \mathrm{BlackInverse}(k_n, \tau_n, c_n)$
end for
return Reconstructed IVs $\{\widehat{\sigma}_n\}_{n=1}^{N}$ and marginal laws $\{\widehat{\mathsf{P}}_j\}_{j=1}^{G}$

We optimize the model with Adam, using an initial learning rate of 10^{-3} and cosine annealing to 10^{-5}. Gradients are clipped to unit norm, and an update is skipped if either the loss or the gradient norm is non-finite. Daily models are trained for at most 400 epochs, evaluated every 20 epochs, and stopped after eight validation evaluations without improvement. Minute models are trained for at most 80 epochs, evaluated every five epochs, and stopped after six validation evaluations without improvement. Checkpoints are selected using validation overall IV-MAE. We train each asset and sampling frequency independently over five random seeds.

Training and inference decoders. The main five-seed checkpoints are optimized with a memory-efficient, fixed-width differentiable convolution. During training, this approximation centrally crops and renormalizes each intermediate convolution. At evaluation, the learned encoder and density head are kept fixed, while all reported predictions and arbitrage audits are recomputed with the full-support decoder of Appendix D.2. The structural guarantee therefore applies to the exact inference map used to generate the reported surfaces. Lemma 2 bounds the approximation introduced during optimization. We additionally evaluate end-to-end training with the exact decoder in the corresponding ablation.

D.4 Implicit Implied-Volatility Layer

The martingale decoder outputs forward-normalized call prices, whereas the training targets and reconstruction metric are expressed in implied volatility. Differentiating through an unrolled Newton solver is unstable in the deep wings, where Vega is small and inaccurate iterates can create large gradients or become trapped at a clipping boundary. We instead solve the Black inverse accurately by bisection in the forward pass and specify its gradient through implicit differentiation.

Forward Black inversion. Let $s_{\mathrm{tot}} = \sigma\sqrt{\tau}$ denote total volatility. The forward-normalized Black call price is

$$B(k, s_{\mathrm{tot}}) = \Phi(d_+) - e^k \Phi(d_-), \qquad d_+ = -\frac{k}{s_{\mathrm{tot}}} + \frac{s_{\mathrm{tot}}}{2}, \qquad d_- = -\frac{k}{s_{\mathrm{tot}}} - \frac{s_{\mathrm{tot}}}{2}. \tag{14}$$

Given a decoder price c, implied total volatility solves $B(k, s_{\text{tot}}) = c$. Before inversion, we project the price into the interior of its no-arbitrage interval:

$$\widetilde{c} = \min\left\{1 - \epsilon,\, \max\left[c,\, (1 - e^k)^+ + \epsilon\right]\right\}, \qquad \epsilon = 10^{-7}. \tag{15}$$

This projection is inactive for a valid interior price and acts only as a numerical safeguard near intrinsic value and the unit upper bound.

During training, we solve $B(k, s_{\text{tot}}) = \widetilde{c}$ using 60 bisection iterations over $s_{\text{tot}} \in [10^{-6}, 6]$. The returned implied volatility is

$$\widehat{\sigma} = \frac{s_{\text{tot}}}{\sqrt{\max(\tau, 10^{-12})}}.$$

Bisection uses the strict monotonicity of B in total volatility and does not require an initial volatility estimate.

Implicit derivative. Define

$$\Psi(\sigma, c; k, \tau) = B(k, \sigma\sqrt{\tau}) - c.$$

At the solution, $\Psi(\widehat{\sigma}, c; k, \tau) = 0$. The implicit-function theorem gives

$$\frac{\partial \widehat{\sigma}}{\partial c} = -\frac{\partial \Psi / \partial c}{\partial \Psi / \partial \sigma} = \frac{1}{V}, \qquad V := \sqrt{\tau}\,\phi(d_+), \tag{16}$$

where V is the Vega of the forward-normalized Black price. Hence, for a model parameter θ,

$$\frac{\partial \widehat{\sigma}}{\partial \theta} = \frac{1}{V}\frac{\partial c_\theta}{\partial \theta}.$$

The inverse $1/V$ becomes unstable as $V \to 0$. We therefore use the Tikhonov-regularized scalar pseudoinverse

$$\frac{\partial \widehat{\sigma}}{\partial c} \approx \frac{V}{V^2 + \lambda_{\text{V}}^2}, \qquad \lambda_{\text{V}} = 0.01. \tag{17}$$

For $V \gg \lambda_{\text{V}}$, this approaches the exact derivative $1/V$. As $V \to 0$, it approaches zero rather than diverging. Its maximum magnitude is $1/(2\lambda_{\text{V}}) = 50$, which explicitly bounds the amplification of price gradients in low-Vega regions. The damping affects only the backward pass; the forward value remains the bisection solution.

Huber residual. For residual $r_n = \widehat{\sigma}_{\theta,n} - \sigma_n$, the elementwise Huber loss is

$$\rho_\beta(r) = \begin{cases} r^2/(2\beta), & |r| \le \beta, \\ |r| - \beta/2, & |r| > \beta, \end{cases} \qquad \beta = 0.05. \tag{18}$$

Its derivative is

$$\rho'_\beta(r) = \begin{cases} r/\beta, & |r| \le \beta, \\ \operatorname{sign}(r), & |r| > \beta. \end{cases}$$

Combining this derivative with the damped implicit inverse yields

$$\frac{\partial \mathcal{L}_{\text{IV}}}{\partial \theta} = \frac{1}{|\mathcal{T}|}\sum_{n \in \mathcal{T}} \rho'_\beta(r_n)\frac{V_n}{V_n^2 + \lambda_{\text{V}}^2}\frac{\partial c_{\theta,n}}{\partial \theta}. \tag{19}$$

The objective is expressed in IV units and directly matches the reported reconstruction metric. The Huber influence function limits the contribution of large residuals, while the damped inverse controls gradient amplification from low-Vega contracts. No gradient is propagated through the bisection iterations; the custom backward pass evaluates Equation 17 once at the converged solution.

Inference-time inversion. Evaluation does not require gradients. We project prices into the same valid numerical interval and apply 50 bisection iterations over $s_{\text{tot}} \in [10^{-3}, 5]$. At this precision, the inversion error is negligible relative to the reported IV-MAE. Training and evaluation therefore use the same normalized Black pricing map, while only the training layer defines a custom implicit backward pass.

E DATA PREPARATION

In this section, we provide detail data processing pipeline with used data scale

E.1 DATA PREPARATION

Asset universe and snapshot construction. We evaluate the actual option tickers rather than relabelling ETF options as index books. At the daily scale, the universe consists of three broad-market ETFs (SPY, QQQ, and IWM) and four liquid single names (AAPL, GOOGL, META, and NVDA). At the intraday scale, we use SPY, QQQ, and IWM. A daily surface is sampled at 15:59 ET, the final minute of continuous trading before the closing auction. Intraday surfaces are sampled every 15 minutes from 09:45 to 15:45 ET, giving at most 25 snapshots per trading day. For each target timestamp T, we retrieve the last OPRA NBBO observed at or before T for every listed contract. The resulting cross-section represents the prevailing option book at T; the distance between the quote timestamp and T measures quote staleness rather than a synchronization offset. We retain quotes updated within 500 ms of T. The underlying price is the contemporaneous midpoint of the Nasdaq BBO from Databento XNAS.ITCH. Thus, the underlying quote is time-aligned with the option snapshot, although it represents the Nasdaq order book rather than the consolidated SIP NBBO.

Quote filtering and price construction. We retain contracts with positive bid and ask prices, positive displayed sizes, and $\mathrm{ask} \geq \mathrm{bid}$. The option price is the arithmetic NBBO midpoint,

$$P_{\mathrm{mid}} = \frac{P_{\mathrm{bid}} + P_{\mathrm{ask}}}{2}.$$

We further require a relative spread $(P_{\mathrm{ask}} - P_{\mathrm{bid}})/P_{\mathrm{mid}} \leq 0.35$, an option price of at least 0.1, and a time to expiry in $[0.5/365, 2]$ years.

The risk-free curve is obtained by linearly interpolating the most recent available Treasury yields across maturities and setting $D(\tau) = \exp(-r(\tau)\tau)$. For each timestamp–expiry slice, the forward is estimated from put–call parity,

$$F_K = K + \frac{C(K) - P(K)}{D(\tau)}.$$

We first take the median across strikes with both call and put quotes, and then recompute the median using strike pairs within 5% of this initial estimate. At least three paired strikes are required. When parity cannot be estimated reliably, we use $F = S\exp((r-q)\tau)$, where S is the aligned Nasdaq-BBO midpoint and q is the median carry inferred from the parity-resolved expiries of the same surface.

Given F, we define log-moneyness as $k = \log(K/F)$ and retain the out-of-the-money side of each strike: puts for $K < F$ and calls for $K \geq F$. Implied volatility is obtained by a vectorized damped-Newton inversion of the Black–76 price, initialized at 50% volatility and constrained to $[10^{-4}, 5]$. Contracts whose reconstructed Black price does not match the observed midpoint within the numerical acceptance tolerance are removed. We use forward delta, $\Delta_C = \Phi(d_+)$ for calls and $\Delta_P = \Phi(d_+) - 1$ for puts.

Surface-level quality control. The remaining contracts undergo the same static-arbitrage cleaning for every asset and scale. Within each maturity, we iteratively remove the interior strike with the largest convexity violation until the normalized call-price curve is convex. Calendar consistency is then enforced in 16 log-moneyness bins by retaining the longest non-decreasing subsequence of median total variance $\sigma^2\tau$ across maturities, followed by a second butterfly-convexity pass. A timestamp is retained only if at least four distinct expiries and 40 contracts remain.

Reference-set selection. Every retained surface provides exactly nine conditioning quotes. We first select three distinct listed maturities nearest to 7, 30, and 90 calendar days. At each selected maturity, we choose three distinct contracts nearest to the forward-delta targets

$$\Delta \in \{0.50,\ 0.25,\ -0.25\},$$

corresponding to an approximately at-the-money call, a 25-delta call, and a 25-delta put. A surface that cannot provide all nine contracts is discarded. Direct inspection of the final files confirms

Table 6: Statistics of the final experimental datasets. "Split" reports the number of train/validation/test surfaces. A surface is one retained quote timestamp, and "Avg. contracts" is the mean number of retained contracts per surface, including the nine reference quotes. Maturity τ is measured in ACT/365 years from the exact snapshot time to expiration at 16:00 ET, and $k = \log(K/F)$.

Scale	Asset	Surfaces	Split (train/val/test)	Contracts	Avg. contracts	τ range	k range
Daily	SPY	992	601/67/324	920,977	928.4	[0.0027, 1.9726]	[−2.482, 0.696]
	QQQ	958	584/65/309	563,724	588.4	[0.0027, 1.9726]	[−1.355, 0.680]
	IWM	727	406/45/276	270,787	372.5	[0.0027, 1.9726]	[−1.418, 0.540]
	AAPL	465	198/22/245	85,619	184.1	[0.0027, 1.9726]	[−2.454, 0.798]
	GOOGL	573	321/36/216	134,580	234.9	[0.0027, 1.9671]	[−1.587, 0.903]
	META	591	282/31/278	190,279	322.0	[0.0027, 1.9726]	[−2.715, 1.488]
	NVDA	797	442/49/306	318,337	399.4	[0.0027, 1.9726]	[−2.916, 1.332]
	Total	**5,103**	**2,834/315/1,954**	**2,484,303**	**486.8**	[0.0027, 1.9726]	[−2.916, 1.488]
15-min	SPY	23,699	14,470/1,608/7,621	18,542,864	782.4	[0.0028, 1.9733]	[−2.551, 0.764]
	QQQ	23,610	14,786/1,643/7,181	13,498,089	571.7	[0.0028, 1.9733]	[−1.393, 0.790]
	IWM	17,615	10,688/1,188/5,739	5,604,565	318.2	[0.0028, 1.9732]	[−1.446, 0.711]
	Total	**64,924**	**39,944/4,439/20,541**	**37,645,518**	**579.8**	[0.0028, 1.9733]	[−2.551, 0.790]

that every retained surface contains exactly nine reference quotes; all remaining contracts form the reconstruction evaluation set.

Temporal split and dataset scale. We use a common chronological cutoff of 1 February 2025. The final 10% of the pre-cutoff timestamps form the validation window, while all timestamps on or after the cutoff are reserved for testing. No surface is shuffled across temporal partitions. The final reported universe contains 70,027 surfaces and 40,129,821 retained contracts. Table 6 reports statistics computed directly from the final Parquet files. Because Meta Platforms changed its ticker from FB to META in June 2022, we include only contracts listed under the META ticker and do not retrospectively relabel FB options. Consequently, the META series begins on June 13, 2022 at the daily scale and June 10, 2022 at the intraday scale, whereas the other assets begin in March 2022.

Short-expiry coverage. The construction filter removes contracts with less than half a calendar day to expiry. Because the latest sampled timestamp is before the 16:00 ET expiration time, this eliminates all same-day-expiry contracts in the final files. The minimum observed maturity is 1.0007 days for the daily data and 1.0104 days for the intraday data, and no retained contract has $\tau < 1$ day. Short-dated contracts beyond one day remain in the dataset: contracts with $1 \le \tau < 7$ days account for 145,312 quotes (5.85%) at the daily scale and 3,419,407 quotes (9.08%) at the intraday scale. The reliable-region metric excludes contracts with fewer than 14 days to maturity, whereas the overall metric and the training data retain them.

F Experimental setup

F.1 Evaluation Metrics

Accuracy. For each held-out surface, every method receives the same nine reference quotes. Accuracy is evaluated on all remaining observed contracts; the reference contracts themselves are excluded from the targets. For a target contract q, with log-forward-moneyness k_q, maturity τ_q, and market implied volatility σ_q, the absolute reconstruction error is

$$e_q = |\hat{\sigma}(k_q, \tau_q) - \sigma_q| .$$

For evaluation region $\mathcal{A}$, we use the contract-weighted IV mean absolute error

$$\text{IV-MAE}_{\mathcal{A}} = \frac{\sum_s \sum_{q \in \mathcal{A}_s} e_q}{\sum_s |\mathcal{A}_s|},$$

where s indexes test surfaces. Results are reported in volatility points ($\times 10^{-2}$). For amortized methods, we first compute the metric separately for each of five independently trained seeds and

then report the mean and standard deviation across seeds. SSVI and Deep Smoothing are optimized independently on every test surface and therefore have no training seed.

Reliable and overall regions. We report accuracy on two fixed evaluation regions:

- **Reliable** contains contracts satisfying
$$|k| \le 0.1, \qquad \mathrm{TTM} \ge 14 \text{ days}.$$
Since $k = \log(K/F)$, the moneyness restriction corresponds approximately to $K/F \in [0.905, 1.105]$. It retains the central portion of the smile while excluding very short-dated contracts.
- **Overall** contains every observed non-reference target contract, including short maturities and deep wings.

The reliable region is intended as a high-confidence evaluation subset rather than as an additional data filter. Near the money, option vega is sufficiently large that the mapping from price to implied volatility is comparatively stable. At very short maturities or in the deep wings, vega can be close to zero, so a small price difference or even one bid–ask tick can translate into a large IV difference. Quotes in those regions are also less liquid. In our final dataset, the relative bid–ask spread
$$\frac{\mathrm{ask} - \mathrm{bid}}{(\mathrm{ask} + \mathrm{bid})/2}$$
has a quote-weighted median of 1.25% and a 90th percentile of 6.90% overall, compared with 0.81% and 2.54%, respectively, inside the reliable region. The reliable metric therefore measures accuracy where the market mid and its implied volatility are most statistically meaningful. The region is defined once and applied identically to every asset, scale, and method; it is not selected using model performance. Reporting the overall metric alongside it prevents a method from appearing strong merely by performing well near the money. The two metrics consequently answer different questions: reliable MAE measures reconstruction in the liquid interior, whereas overall MAE measures whether the fitted surface remains well behaved under short-end and wing extrapolation.

Static-arbitrage validity. A surface can fit observed IV quotes accurately while assigning inconsistent prices to nearby strikes or maturities. We test two elementary forms of static arbitrage. A *calendar violation* occurs when a longer-maturity call is cheaper than a shorter-maturity call at the same forward moneyness. A *butterfly violation* occurs when call price is locally non-convex in strike; this would permit a butterfly portfolio with a non-negative terminal payoff and a negative initial cost, and corresponds to a negative discrete risk-neutral probability.

For each asset and scale, we select 50 deterministic, time-spanning surfaces from the out-of-time test period. Every method is evaluated on 801 log-moneyness points $k_i \in [-1, 1]$ and the same 30 maturity nodes τ_j used by AddMart. Let $M_i = e^{k_i} = K_i/F$ and let $c_{j,i} = C(K_i, \tau_j)/(D(\tau_j)F(\tau_j))$ denote normalized forward call price. HyperIV, GNO, VAE, SSVI, and Deep Smoothing are converted from predicted IV to $c_{j,i}$ using Black–76. For ADDMART, we directly evaluate the normalized prices produced by its exact decoder. All methods are then tested on the same grid with the same price-space inequalities.

Calendar validity requires adjacent-maturity prices to be non-decreasing. Butterfly validity requires adjacent strike slopes to be non-decreasing:
$$\Delta^{\mathrm{cal}}_{j,i} = c_{j+1,i} - c_{j,i} \ge 0, \tag{20}$$
$$\Delta^{\mathrm{bf}}_{j,i} = \frac{c_{j,i+2} - c_{j,i+1}}{M_{i+2} - M_{i+1}} - \frac{c_{j,i+1} - c_{j,i}}{M_{i+1} - M_i} \ge 0. \tag{21}$$
Each surface therefore contributes 29×801 calendar pairs and 30×799 butterfly pairs. For either condition $a \in \{\mathrm{cal}, \mathrm{bf}\}$, the reported violation rate is
$$R_a = 100\, \frac{\sum_{r,s,j,i} \mathbf{1}[\Delta^{a,(r,s)}_{j,i} < -\varepsilon]}{\sum_{r,s} N^{(r,s)}_a},$$
where r indexes training seeds and s indexes audited surfaces. Learned methods use five seeds, whereas per-surface baselines have no seed dimension. Thus, a reported rate of 0.5% means that 0.5% of all evaluated price pairs violate the corresponding inequality; it does not mean that only 0.5% of complete surfaces contain a violation.

Table 7: Implementation settings for the amortized baselines. Epoch counts are maximum values for daily/intraday experiments.

Method	Trainable parameters	Batch	Queries/surface	Epochs (daily/intraday)
HyperIV	243,153	16	1024	1500/150
GNO	102,529	20	1024	300/60
VAE	332,015	20	1024	200/60

Numerical tolerance. Finite-precision price evaluation and differencing can produce tiny negative residuals around an exact zero. The headline pair-rate audit uses $\varepsilon_{\text{pair}} = 10^{-10}$. We separately define a surface as having a materially meaningful violation only if its minimum residual is below -10^{-6}. Consequently, a residual such as -10^{-9} is visible in the strict pair diagnostic but is not interpreted as a material arbitrage failure. The spatial heatmaps and convolution ablations use $\varepsilon_{\text{vis}} = 10^{-7}$ to suppress isolated numerical noise; their percentages are therefore kept distinct from the stricter headline pair rates.

The exact AddMart decoder is audited in both float32 and float64. Both precisions produce zero calendar and butterfly pair violations in the formal audit; the smallest residuals are approximately 10^{-13}, consistent with floating-point error. We additionally check for non-finite prices, invalid price bounds, and failed surface evaluations before aggregation.

F.2 Baselines and Reproduction Details

We compare AddMart with three amortized neural baselines and two per-surface calibration methods. All methods receive information from the same nine observed reference contracts. Accuracy is evaluated on the remaining, non-reference contracts of each surface.

Common protocol. Each amortized model is trained independently for every asset and sampling frequency using seeds $\{0, 1, 2, 3, 4\}$. We use the chronological split described in Appendix E.1: the test period begins on February 1, 2025, while the last 10% of the pre-test timestamps form the validation set. The test period is never used for hyperparameter tuning or checkpoint selection.

During training, each surface is represented by the nine reference tuples (k_i, τ_i, σ_i). We sample $N = 1024$ query contracts from the same surface with replacement and minimize the corresponding reconstruction objective. HyperIV uses a batch size of 16 surfaces, while GNO and VAE use 20. All three neural baselines are optimized with Adam at an initial learning rate of 10^{-3} and cosine annealing to 10^{-5}. The checkpoint with the lowest validation overall IV-MAE is retained. At test time, the nine conditioning quotes are excluded from both reliable-region and overall accuracy metrics. The reliable region is

$$\mathcal{R} = \{(k, \tau) : |k| \le 0.1,\ \tau \ge 14/365\}.$$

We report the mean and standard deviation across the five independently trained seeds. SSVI and Deep Smoothing perform a new optimization for every test surface and are therefore reported as single values.

- **HyperIV** (Yang et al., 2025). We retain the released set-embedding and hypernetwork construction and adapt its data interface to our nine-reference protocol. The generated surface-specific IV network has architecture $2 \to 16 \to 16 \to 1$, with `tanh` hidden activations and a `softplus` output. It contains 337 generated weights. A permutation-invariant set encoder maps the nine (k, τ, σ) tuples to these weights and contains 243,153 trainable parameters.

  Our reproduction uses the HyperIV reconstruction loss together with its three soft regularizers: calendar monotonicity, the Gatheral–Jacquier butterfly condition, and density normalization. These terms are evaluated by automatic differentiation on a 31×9 auxiliary grid over $k \in [-1.5, 0.5]$ and $\tau \in [0.01, 2]$. All four loss components have unit weight. We replace non-finite gradients by zero and clip the global gradient norm to one, which prevents occasional spikes produced by the second-derivative penalties. Validation is performed every 50 epochs for the daily experiments and every five epochs for the intraday experiments.

- **GNO** (Li et al., 2020; Wiedemann et al., 2025). Our GNO baseline uses the published Operator Deep Smoothing `GNOVolSmoother` architecture, adapted so that the nine reference contracts constitute the observed nodes and arbitrary held-out contracts constitute the query nodes. Following the published operator coordinates, we transform each query to
$$(\rho, z) = \left(\sqrt{\tau},\, \frac{k}{\sqrt{\tau}}\right).$$
The model uses three graph-neural-operator layers, 16 operator channels, a 64-dimensional pointwise network, radius 10, and subsample size 16, for a total of $102{,}529$ trainable parameters. It is trained only with query-IV MSE; no calendar or butterfly penalty is added.
The original positive-output layer can enter a numerically saturated region when fitting low-volatility assets. In our adapter, both reference and target IVs are divided by the median IV of the training set before being passed to the network, and predictions are multiplied by the same constant afterward. The scale is estimated exclusively from the training set and stored in the checkpoint. This transformation leaves the relative reconstruction problem unchanged and prevents seed-dependent zero-output collapse without modifying the released GNO implementation.
- **VAE** (Kingma & Welling, 2014). The VAE requires a fixed ordering of its nine-dimensional input. Because the three selected market maturities vary across timestamps, we first calibrate the four-parameter SSVI form using only the nine observed references. We then evaluate that fitted surface at a fixed virtual grid consisting of 7-, 30-, and 90-day maturities crossed with a 25-delta put, ATM, and 25-delta call. This produces a consistently ordered nine-dimensional IV vector without using any reconstruction target.
The encoder has two width-322 ReLU layers and outputs the mean and log variance of a 16-dimensional latent variable. The pointwise decoder receives the latent variable together with (k, τ) and uses three width-322 ReLU layers followed by a `softplus` output. The resulting model contains $332{,}015$ trainable parameters. We minimize
$$\mathcal{L}_{\mathrm{VAE}} = \mathrm{MSE}(\widehat{\sigma}, \sigma) + 10^{-3} D_{\mathrm{KL}}\big(q_\phi(z \mid \sigma_{\mathrm{ref}}) \,\|\, \mathcal{N}(0, I)\big).$$
Latent samples are used during training, while the posterior mean is used for validation and testing.
- **SSVI**[‡] (Gatheral & Jacquier, 2014). For every test surface, we fit the four-parameter power-law SSVI parameterization
$$\theta(\tau) = \sigma_0^2 \tau, \qquad \phi(\theta) = \eta \theta^{-\gamma},$$
using only the same nine reference quotes. The parameters satisfy $\sigma_0 > 0$, $0 < \gamma < 1/2$, $\eta > 0$, and $-1 < \rho < 1$. Calibration minimizes IV-MSE with SLSQP and imposes the Gatheral–Jacquier sufficient butterfly-arbitrage inequalities over the maturity range of the surface. We use three initializations, at most 1000 SLSQP iterations per initialization, and retain the feasible solution with the lowest reference-point objective. SSVI has no learned parameters, training split, validation selection, or seed averaging.
- **Deep Smoothing**[‡] (Ackerer et al., 2020). For each test surface, Deep Smoothing independently fits a network $2 \to 32 \to 32 \to 1$ with `tanh` hidden activations and a `softplus` IV output. Its objective is
$$\mathcal{L}_{\mathrm{DS}} = \mathrm{MSE}_{\mathrm{ref}} + \lambda \left[\mathbb{E}\,\mathrm{ReLU}(-\partial_\tau w) + \mathbb{E}\,\mathrm{ReLU}(-g)\right], \qquad \lambda = 1,$$
where $w(k, \tau) = \sigma(k, \tau)^2 \tau$ and g is the Gatheral–Jacquier butterfly function. The penalties are evaluated on a 41×16 collocation grid whose range is determined from the test asset's observed (k, τ) support. Each surface is optimized for 1000 Adam steps with learning rate 10^{-2}.
For computational efficiency, the test surfaces of an asset are optimized simultaneously on a GPU. Every surface retains its own parameters and its own loss; this batching changes only execution time and does not amortize the per-surface optimization.

G Additional Results and Analysis

G.1 Downstream Tail-Risk Evaluation

The discrete marginal distribution produced by AddMart can be used directly for downstream risk calculations. We examine this property through option-implied downside CVaR at tail probabilities

$\alpha \in \{1\%, 5\%, 10\%\}$. For each asset, we select 50 dates spanning the out-of-time test period and evaluate the expiries closest to 180 days and one year. The amortized models observe only the nine reference contracts used in the reconstruction task.

Full-information reference. Since the physical return distribution is unobserved, we construct an option-implied reference from all available quotes at the selected expiry. Let $M_T = S_T/F_T$ and $z = K/F_T$. After converting OTM puts to equivalent calls using put–call parity, the normalized option price satisfies

$$\widetilde{C}(z) = \mathbb{E}\big[(M_T - z)^+\big].$$

We fit a nonnegative discrete distribution on $M_T \in [0.01, 4]$ by matching these normalized prices, with quote weights derived from the bid–ask spread and a second-difference penalty that suppresses unidentified grid-scale oscillations. Unit mass and the martingale moment are imposed numerically,

$$\sum_i p_i = 1, \qquad \sum_i M_i p_i = 1.$$

The resulting distribution uses approximately 30–36 quotes per expiry and serves as a full-information option-implied reference. It is not treated as physical-distribution ground truth.

For a probability mass function p on normalized terminal prices, we define downside CVaR by

$$\mathrm{CVaR}_\alpha = 1 - \frac{1}{\alpha}\int_0^\alpha F_M^{-1}(u)\, du,$$

which measures the average normalized loss in the lower α-tail. For AddMart, this quantity is computed directly from its predicted marginal. For IV-only baselines, we recover the density by finite differences of the predicted call curve. Negative density is clipped and the remaining mass is renormalized before CVaR is evaluated; we separately record how often this repair is required.

Table 8: Option-implied downside-CVaR error on daily test surfaces. Entries are CVaR-MAE ($\times 10^{-2}$), averaged over 50 test dates and $\alpha \in \{1\%, 5\%, 10\%\}$, with mean±std over five seeds. Lowest error in each column is bold.

	SPY		QQQ		NVDA	
Method	180d	1y	180d	1y	180d	1y
AddMart	**1.87±.37**	**2.27±.35**	3.05±.27	**3.12±.19**	**3.86±.20**	**2.42±.21**
HyperIV	1.89±.18	4.76±.66	**2.88±.24**	4.65±.39	4.72±.93	5.39±1.91
GNO	3.47±2.48	4.23±.97	19.74±9.79	6.73±2.57	5.23±3.74	3.44±.72
VAE	12.51±3.78	6.18±1.44	10.09±5.42	14.79±5.32	12.19±5.00	3.25±1.22

Table 8 shows that AddMart obtains the lowest CVaR error in five of the six asset–horizon settings. On SPY at 180 days it is nearly tied with HyperIV, while on QQQ at 180 days HyperIV is marginally better after averaging the three tail levels. The latter advantage is concentrated at $\alpha = 1\%$; AddMart is more accurate at the 5% and 10% tails. At the one-year horizon, AddMart is best on all three assets.

Figure 5 exposes information hidden by the aggregate MAE. The AddMart residuals remain relatively stable over time, with a moderate tendency to underestimate the most extreme 1% tail. HyperIV exhibits a stronger negative bias at long horizons, particularly on QQQ and NVDA. GNO substantially overestimates QQQ tail risk, whereas VAE exhibits persistent positive bias on SPY and QQQ. These patterns indicate systematic calibration differences rather than errors driven by a small number of test dates.

The distributional validity check further separates the methods. The raw marginals of AddMart are valid on every evaluated asset, date, maturity, and seed. Before post-processing, the invalid-density rate ranges from 10% to 100% for HyperIV and from 5.2% to 100% for GNO, and is 100% for VAE. Their CVaR values in Table 8 therefore depend on ex-post density repair, while AddMart provides the risk functional directly from its predicted martingale marginal.

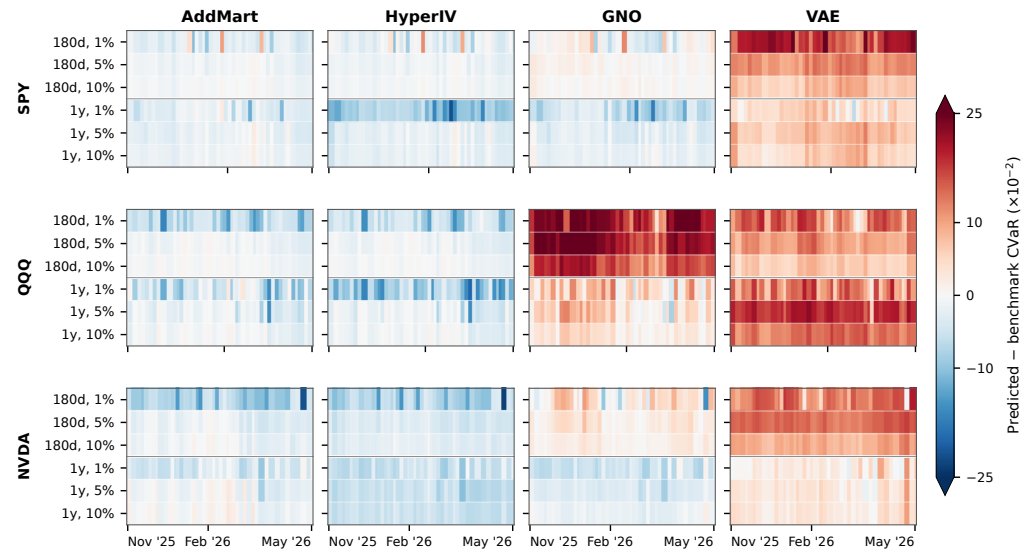

Figure 5: Signed CVaR residuals over the test period. Columns denote reconstruction methods and rows denote assets. Within each panel, the heatmap rows combine the 180-day and one-year horizons with tail probabilities of 1%, 5%, and 10%. Blue indicates that the model underestimates the full-information option-implied CVaR, while red indicates overestimation. Values are averaged over five seeds.

Interpretation. The experiment evaluates risk-neutral tail exposure implied by option prices. It does not constitute a forecast of realized portfolio losses. The 5% and 10% results are better identified by the observed strike range; the 1% results should be interpreted as an extreme-tail stress test because they are more sensitive to sparse wing quotes and distributional extrapolation.

G.2 Stress Tests

Protocol. We evaluate all amortized methods during two pre-specified SPY market-stress windows: April 1–10, 2025 (8 trading days) and March 23–31, 2026 (7 trading days). To avoid look-ahead bias, we use a separate set of checkpoints whose out-of-time test period begins on February 1, 2025. Each model is trained once on the preceding data and then kept frozen through both events. The reference set, reconstruction targets, and reliable-region definition are identical to the main experiments. The two windows contain 10,273 and 14,557 held-out contracts, respectively.

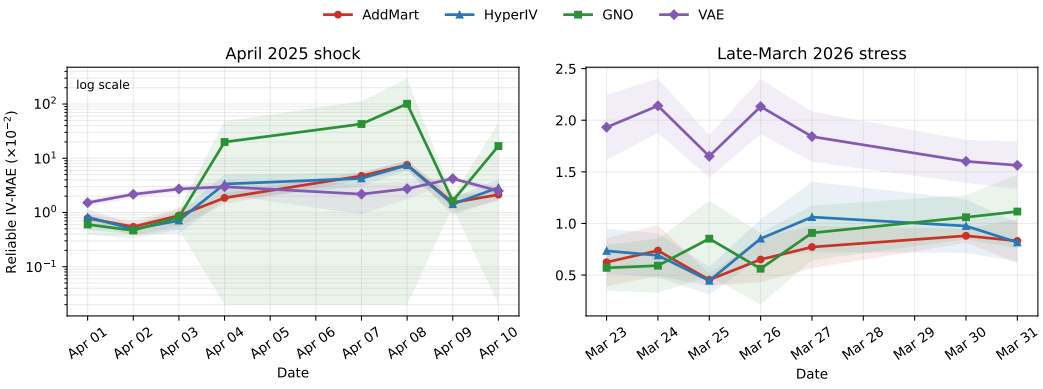

Figure 6: Daily reliable-region IV-MAE on the two SPY stress windows. Lines and shaded regions denote the five-seed mean and standard deviation. The April panel uses a logarithmic vertical scale because one GNO seed diverges during the most volatile days.

Results. All methods deteriorate during the April 2025 shock, confirming that the event induces a substantial distribution shift rather than a model-specific failure. Relative to March 2025, ADDMART's overall and reliable MAE increase by $5.9\times$ and $4.0\times$, respectively. Nevertheless, it retains the lowest reliable-region error and remains essentially tied with HyperIV in overall MAE. VAE obtains a slightly lower overall error, but remains worse in the liquid reliable region. GNO is considerably less stable: four seeds produce window-level overall MAE between 4.5 and 8.2×10^{-2}, while one seed rises to 91.2×10^{-2}, causing the large reported variance.

Table 9: SPY performance during two out-of-sample market-stress windows. IV-MAE is reported in $\times 10^{-2}$ as mean±std over five seeds. Best results are shown in bold.

	Apr. 1–10, 2025		Mar. 23–31, 2026	
Method	Reliable	Overall	Reliable	Overall
ADDMART	$\mathbf{2.03}\pm\mathbf{.23}$	$4.92\pm.84$	$\mathbf{0.72}\pm\mathbf{.13}$	$\mathbf{0.98}\pm\mathbf{.09}$
HyperIV	$2.23\pm.45$	$4.99\pm.86$	$0.80\pm.12$	$1.58\pm.19$
GNO	16.74 ± 28.25	23.85 ± 37.66	$0.82\pm.20$	$1.19\pm.16$
VAE	$2.68\pm.45$	$\mathbf{4.76}\pm\mathbf{.53}$	$1.83\pm.24$	$2.06\pm.19$

The late-March 2026 window is less severe but occurs more than one year after the training cutoff. Here, ADDMART achieves the lowest error on both metrics. Its overall MAE is approximately 18% below GNO, 38% below HyperIV, and 53% below VAE. The daily curves in Figure 6 also show that its seed dispersion remains controlled throughout the event. Relative to February 2026, ADDMART's overall MAE increases by 35%, compared with 71% for GNO and 86% for HyperIV. VAE exhibits a smaller relative increase, but from a substantially higher absolute-error level.

These results suggest that the additive construction acts as a useful structural regularizer under abrupt surface shifts: its accuracy still degrades when the market leaves the training distribution, but it does not exhibit the catastrophic seed failure observed for GNO. Moreover, the exact decoder preserves its calendar and butterfly constraints independently of the market regime. The stress tests therefore support robustness in the sense of controlled degradation and retained validity, rather than invariance of reconstruction error during extreme events.

G.3 COMPUTATIONAL PERFORMANCE

We benchmark the amortized models on a single NVIDIA RTX 5090 D using nine reference contracts and 300 reconstruction targets per surface. Each measurement uses 20 warm-up iterations followed by 100 synchronized iterations. We report the median across seven daily assets and three minute-scale assets using the seed-0 checkpoints. Preprocessing includes the construction of model-specific inputs but excludes disk I/O. The batch-one end-to-end latency is estimated as preprocessing time plus neural forward time. SSVI and Deep Smoothing are excluded because they solve a separate optimization problem for every input surface, whereas the learned models perform amortized inference with fixed trained weights.

Table 10: Single-GPU inference performance. Forward and end-to-end latency are reported in milliseconds per surface. Throughput measures neural forward inference at batch size 32. Peak memory is the maximum allocated GPU memory for batch sizes 1 and 32, respectively.

Scale	Method	B1 forward	B1 end-to-end	B32 forward	B32 surface/s	Peak MiB B1/B32	Weights MiB
Daily	ADDMART	6.98	7.60	1.040	961	127.64 / 2939.44	0.918
	HyperIV	0.58	**1.19**	0.019	53,201	32.99 / 34.84	0.928
	GNO	1.99	2.60	1.874	533	36.69 / 36.85	**0.402**
	VAE	**0.20**	29.60	**0.009**	**113,918**	34.04 / 58.05	1.272
Minute	ADDMART	6.80	7.43	1.042	960	127.64 / 2939.44	0.918
	HyperIV	0.57	**1.21**	0.019	52,994	32.99 / 34.84	0.928
	GNO	1.99	2.63	1.883	531	36.69 / 36.85	**0.402**
	VAE	**0.20**	33.69	**0.009**	**114,110**	34.04 / 58.05	1.272

For single-surface inference, ADDMART requires approximately 7 ms for the forward pass and 7.4–7.6 ms including preprocessing. Although slower than HyperIV and GNO due to full-support convolution and implied-volatility inversion, this latency remains below 10 ms and is sufficient for minute-scale surface updates and conventional market-making applications. Batch inference further reduces the cost to approximately 1.04 ms per surface, or 960 surfaces per second. The model weights oc-

cupy less than 1 MiB, while batch-one inference uses approximately 128 MiB of GPU memory. In return for this moderate computational overhead, AddMart achieves the lowest asset-averaged overall IV-MAE at both daily and minute scales while preserving structural no-arbitrage guarantees.

G.4 Cross-Asset Performance

We next examine whether a surface model trained on one underlying can be applied to another without retraining. For each ordered source–target pair, we retain the source-asset checkpoint and provide only the target surface's nine reference quotes at inference time. All remaining target contracts are used for evaluation under the same out-of-time split beginning on February 1, 2025. The experiment covers seven daily assets and therefore contains 7×7 directed pairs per method. Learned models are evaluated over five seeds; SSVI is fitted independently on each target surface and is included only as a target-calibrated reference.

Table 11 shows that transfer is substantially more difficult than in-domain reconstruction. The overall MAE of AddMart increases from 1.65 to 5.51 after excluding diagonal pairs, corresponding to a $3.34\times$ transfer penalty. Nevertheless, it retains the lowest cross-asset overall mean and outperforms HyperIV on 76.2% of the 42 directed pairs when each pair is first averaged over seeds. HyperIV is stronger on the reliable region after transfer, attaining a mean of 4.81 versus AddMart's 5.33. Thus, the in-domain advantage of AddMart does not uniformly extend to zero-shot accuracy: its benefit is clearer under the overall metric, while HyperIV transfers better within the liquid interior.

Table 11: Daily cross-asset transfer over the seven-asset universe. We report IV-MAE ($\times 10^{-2}$). In-domain columns average diagonal source–target pairs, while cross-asset columns aggregate the 42 directed off-diagonal pairs. All methods are evaluated over five seeds. The lowest error in each column is shown in bold.

	In-domain		Cross-asset overall		Cross-asset reliable		Transfer penalty
Method	Overall	Reliable	Mean	Median	Mean	Median	
AddMart	**1.65**	1.29	**5.51**	4.67	5.33	4.25	3.34×
HyperIV	1.72	**1.27**	5.53	**4.24**	**4.81**	**3.52**	3.21×
GNO	2.01	1.63	6.99×10^{3}	5.04	7.62×10^{3}	4.15	2921.6×
VAE	2.52	2.04	6.03	5.57	5.21	5.04	2.39×

Table 12: Daily cross-asset reliable-region IV-MAE ($\times 10^{-2}$, five-seed mean). Rows denote training assets and columns denote evaluation assets. Gray cells indicate in-domain evaluations, and the lowest value in each target column is shown in bold.

(a) AddMart

Train \ Test	SPY	QQQ	IWM	AAPL	GOOGL	META	NVDA
SPY	0.56	0.76	0.97	2.05	6.41	7.03	15.61
QQQ	1.25	0.68	0.79	1.55	4.03	4.84	11.14
IWM	1.87	0.82	0.77	1.70	4.46	4.97	10.02
AAPL	2.83	1.44	1.12	1.18	2.27	2.75	5.81
GOOGL	6.12	3.01	2.26	1.52	2.22	2.60	5.76
META	13.87	8.25	6.22	5.47	**2.18**	1.82	2.65
NVDA	19.66	14.65	12.81	11.11	4.79	4.43	**1.80**

(b) HyperIV

Train \ Test	SPY	QQQ	IWM	AAPL	GOOGL	META	NVDA
SPY	0.57	0.77	1.00	1.97	6.24	6.86	14.81
QQQ	1.40	0.74	0.76	1.55	3.69	4.41	9.14
IWM	1.65	0.85	0.78	1.57	4.35	4.99	10.29
AAPL	2.49	1.49	1.11	1.18	2.31	2.87	5.60
GOOGL	4.36	2.66	2.11	1.43	2.07	2.47	5.18
META	11.48	7.24	6.30	4.95	2.16	1.79	2.74
NVDA	16.53	12.11	10.57	9.33	4.24	3.79	1.80

The directed matrices in Tables 12 reveal that asset similarity dominates transfer quality. For AddMart, QQQ→IWM and IWM→QQQ attain overall MAEs of 1.30 and 1.33, respectively, close to their in-domain errors. Transfers between broad-market ETFs and NVDA are much harder: SPY→NVDA reaches a reliable MAE of 15.61, while the reverse NVDA→SPY direction reaches 19.66. Transfer is therefore highly asymmetric, and reversing the source and target can substantially change the error. The matrices also show that the best source need not be the target itself; for example, META transfers slightly better than GOOGL itself when AddMart is evaluated on GOOGL.

Finally, GNO illustrates why both means and medians are necessary in cross-asset evaluation. Its median off-diagonal MAE remains comparable to the other neural baselines, but 13 of its 210 seed–direction cells exceed an MAE of 1, eight exceed 10, and the resulting mean becomes several orders

of magnitude larger. These failures indicate unstable zero-shot extrapolation rather than uniformly poor reconstruction. Overall, the experiment shows that AddMart transfers reliably between related markets but is not asset-invariant; large shifts in volatility level and smile asymmetry remain a substantial source of error.

G.5 Analysis on LEAPS and VIX

We further examine two regimes that depart substantially from the liquid equity-option surfaces used in the main experiments: long-dated LEAPS and options on VIX. These experiments characterize the scope of the additive martingale assumption rather than extend the main benchmark universe.

Long-dated LEAPS. Figure 7 localizes the long-horizon limitation to the evolution of skew across maturity. On SPY, AddMart follows the market ATM-skew term structure reasonably well through approximately one year, but its predicted skew subsequently decays faster than observed. The same behavior is visible in the 554-day AAPL slice, where the predicted smile is substantially flatter than the market quotes, including within the reliable region. This is consistent with the repeated convolution of conditionally independent increments progressively smoothing the terminal distribution. At sufficiently long horizons, the construction therefore has difficulty retaining persistent asymmetry and tail shape. HyperIV is more flexible at these maturities and recovers more of the observed skew, although its unconstrained extrapolation becomes unstable away from the quoted region.

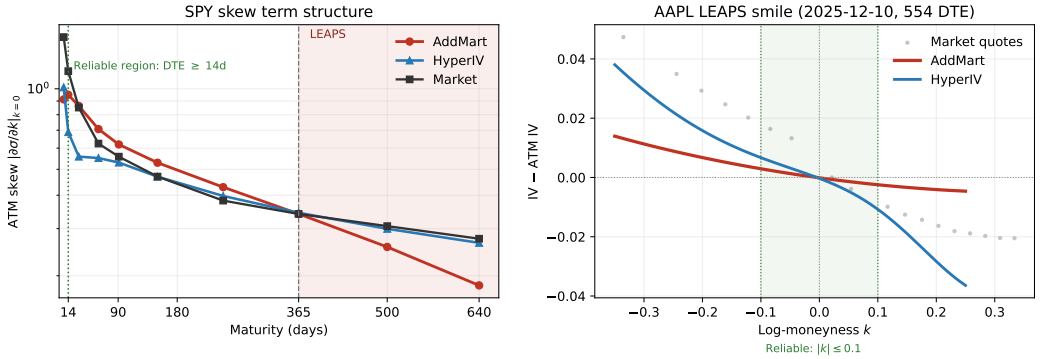

Figure 7: Long-horizon behavior on LEAPS. Left: average ATM-skew term structure on the SPY test set, with the reliable-region cutoff at 14 days and the LEAPS region beginning at 365 days. Right: a representative AAPL LEAPS slice observed on December 10, 2025 (554 DTE; 18 market quotes). The shaded interval denotes the reliable moneyness region $|k| \le 0.1$. AddMart increasingly underestimates the persistent long-dated skew, whereas HyperIV retains more of the observed skew but extrapolates more aggressively outside the reliable region.

VIX options. VIX produces a distinct short-horizon failure mode. Figure 8 shows that most of the accuracy gap is concentrated at the shortest maturity and decreases rapidly along the term structure. VIX options inherit strong mean reversion, jumps, volatility-of-volatility, and level-dependent skew. Moreover, different maturities are linked to different points of the VIX-futures term structure. Representing these marginals through a shared sequence of additive increments consequently imposes a stronger cross-maturity restriction than it does for equity options. The consistency of this pattern across seeds suggests a representation limitation rather than training instability. Despite the larger IV error, AddMart retains zero measured calendar and butterfly violations on the VIX test set; the failure concerns market fit rather than measure validity.

Practical impact. These limitations have limited effect on the primary use case studied in this paper. Most liquid equity and ETF option activity is concentrated in shorter maturities and near the central moneyness region, where AddMart remains accurate and the hard no-arbitrage construction is most valuable. LEAPS quotes are comparatively sparse and have wider bid–ask spreads, while VIX options constitute a different underlying process from the equity and ETF surfaces in the main benchmark. Consequently, neither stress regime materially drives the aggregate reconstruction results. The limitation is nevertheless relevant for applications devoted specifically to long-horizon

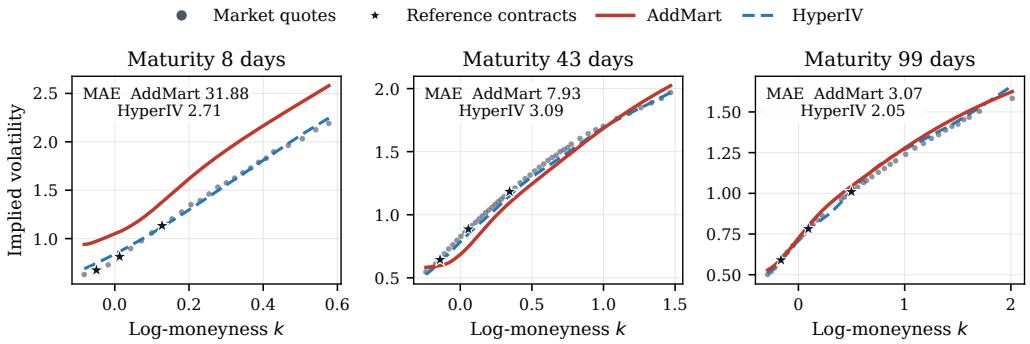

Figure 8: Representative failure case on a VIX daily test surface observed on September 9, 2025. The surface is selected by the median per-surface MAE gap between AddMart and HyperIV rather than by an extreme error. AddMart substantially overestimates the short-maturity smile, while the discrepancy decreases with maturity.

tail hedging or short-dated VIX trading, for which the current additive increment parameterization should not be used as a stand-alone pricing model.

G.6 Detailed Comparison

Figures 9 compares reconstruction errors along three complementary dimensions: test time, log-moneyness, and maturity. At the daily scale, AddMart achieves the lowest or near-lowest error across most assets, market regimes, and contract buckets. The temporal profiles show that several error spikes occur simultaneously across methods, indicating periods of greater market-wide surface complexity. GNO exhibits the clearest method-specific instability on IWM, while SSVI deteriorates substantially in the deep wings and near the maturity boundaries. HyperIV remains competitive on several assets and performs slightly better in parts of the NVDA surface. At the minute scale, AddMart is particularly strong on SPY and IWM and remains competitive on QQQ. Across both sampling frequencies, the shortest maturities and sparsely populated wings are consistently the most difficult regions.

Figures 10 and 12 further localize reconstruction error jointly over log-moneyness and maturity. At the daily scale, AddMart maintains low error throughout the reliable region across all seven assets, showing that its aggregate performance is not driven by a small collection of favorable contracts. Errors increase mainly at very short maturities and in the deep wings, where quotes are sparse and implied-volatility inversion is less stable. The competing methods exhibit more distinct spatial failure patterns: HyperIV deteriorates near some short-end boundaries, GNO produces pronounced wing errors—most visibly on IWM—and VAE has broader regions of elevated error. SSVI is accurate near the liquid interior but extrapolates poorly toward sparse wings and longer maturities. The minute-scale comparison is tighter, although AddMart continues to degrade smoothly outside the reliable region. Figures 11 and 13 show that reconstruction accuracy and structural validity are separate properties. Each cell reports the pooled violation rate over calendar and butterfly price-grid pairs. The exact AddMart decoder has zero violations throughout the evaluated domain, including short maturities and deep wings where reconstruction is more difficult. HyperIV violations are concentrated mainly at short maturities and in the wings, GNO exhibits extensive violations under extrapolation, and VAE violations extend closer to the liquid interior. The constrained per-surface SSVI fit is also arbitrage-free in this audit, but its error maps reveal the accuracy cost of its restrictive parametric form.

The representative surfaces in Figures 14 and 15 illustrate these aggregate patterns at the individual-contract level. The SPY surface on April 7 is especially challenging at the shortest maturity, where the methods disagree substantially in the left wing. Their predictions become much closer around the reference quotes and on the later test dates. On May 2 and June 10, AddMart closely tracks the dense non-reference quotes across the medium and long maturities. NVDA presents a more difficult cross-sectional fit, and no method dominates every slice. GNO and SSVI can depart materially from

the observed wing shape, while HyperIV develops an oscillatory long-maturity wing on April 7. ADDMART produces smoother extrapolations and competitive fits, although HyperIV retains a small aggregate advantage on NVDA. Together, the temporal profiles, joint heatmaps, and representative slices show that ADDMART combines stable reconstruction accuracy across market regimes with exact static-arbitrage control over the full surface.

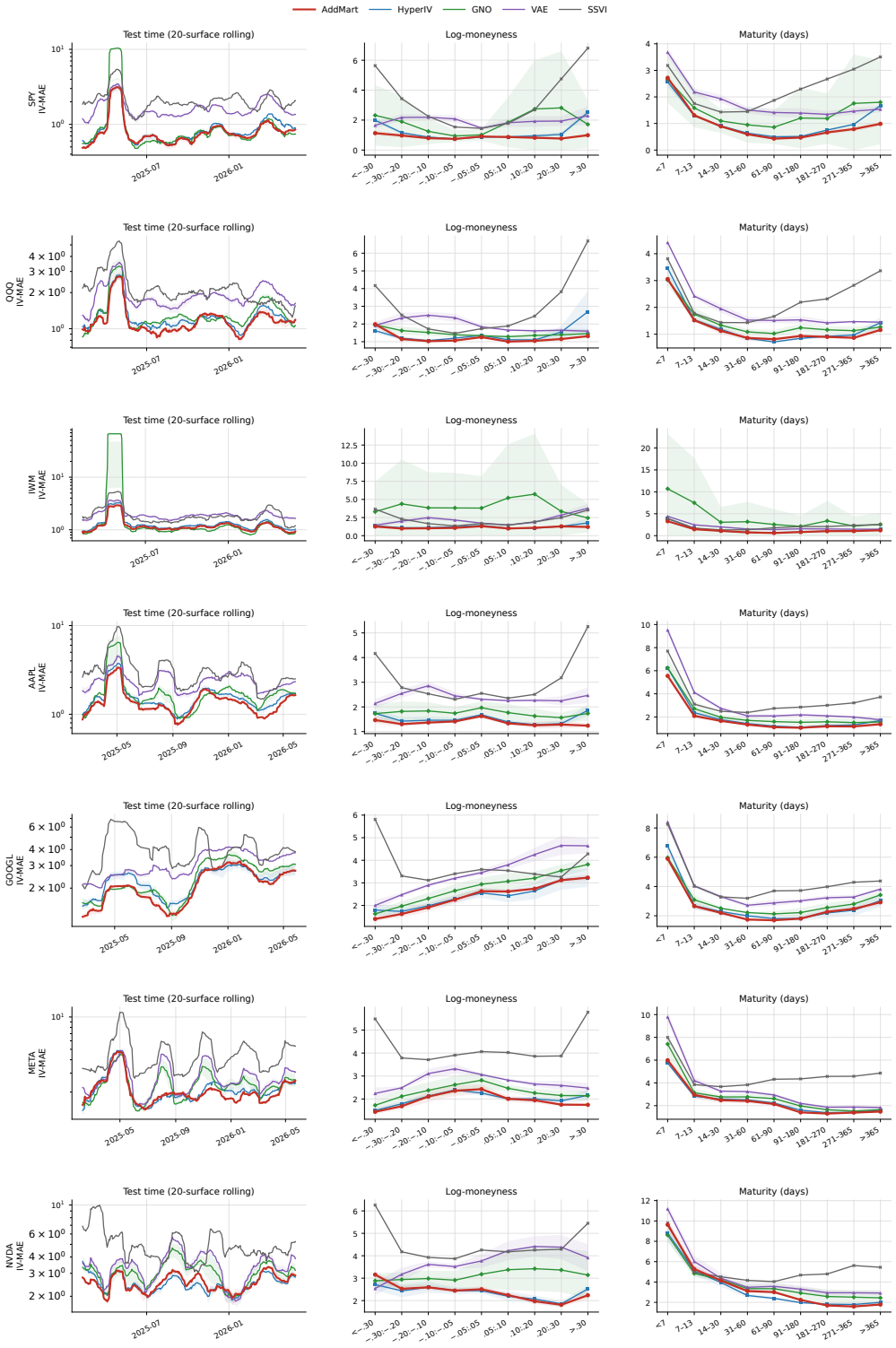

Figure 9: Daily-data error profiles for SPY, QQQ, IWM, AAPL, GOOGL, META, and NVDA. Rows correspond to underlying assets. The columns report 20-surface rolling IV-MAE over the test period, IV-MAE by log-moneyness, and IV-MAE by maturity, respectively. IV-MAE is reported in $\times 10^{-2}$. For amortized methods, lines and shaded regions denote the mean and standard deviation over five seeds; SSVI is fitted independently to each surface.

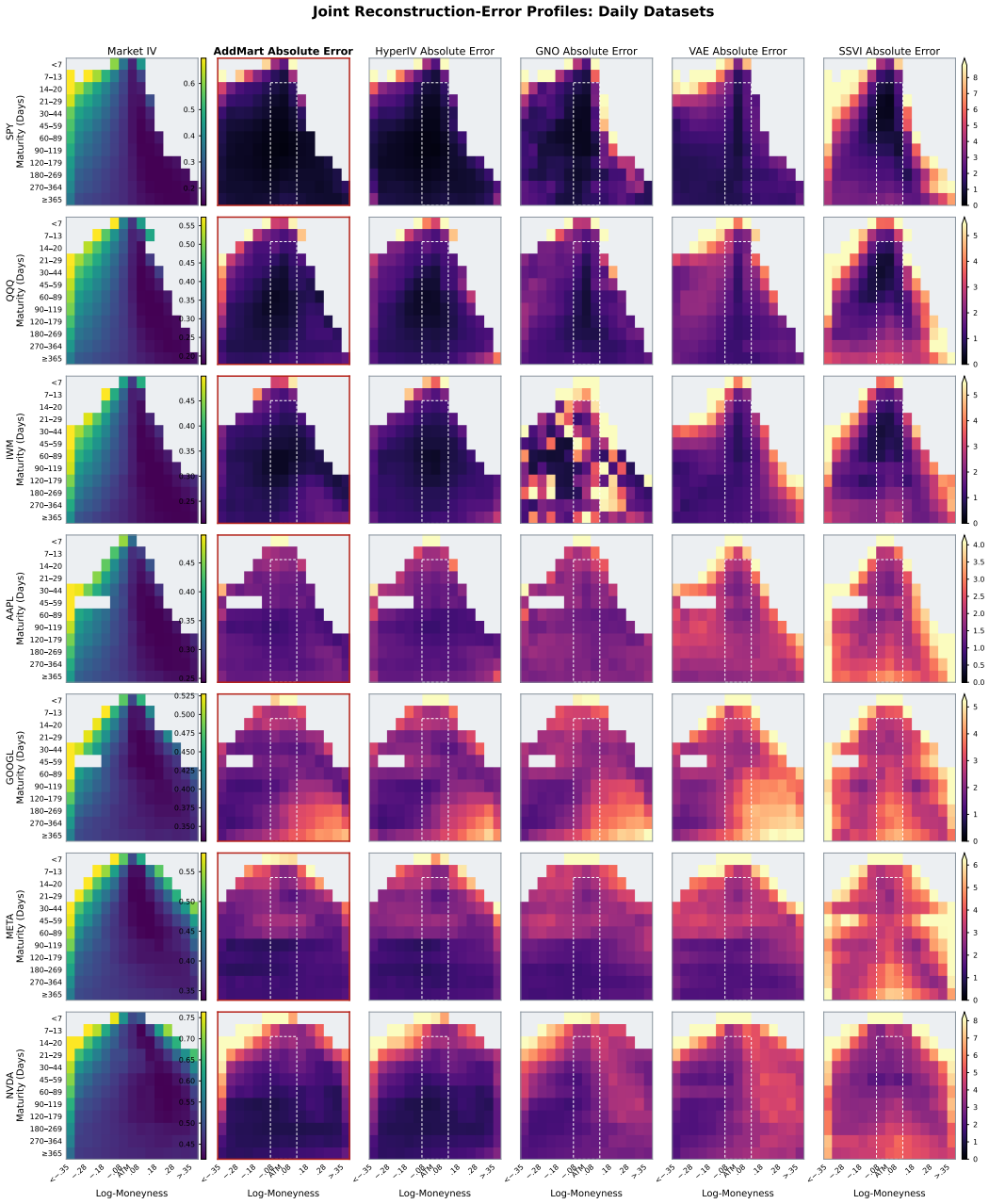

Figure 10: Joint reconstruction-error profiles across the seven daily datasets. The first column shows market IV; the remaining columns show IV-MAE ($\times 10^{-2}$) aggregated by log-moneyness and maturity. Error color scales are shared across methods within each asset row. The dashed box denotes the reliable region, and gray cells contain insufficient observations for reliable aggregation.

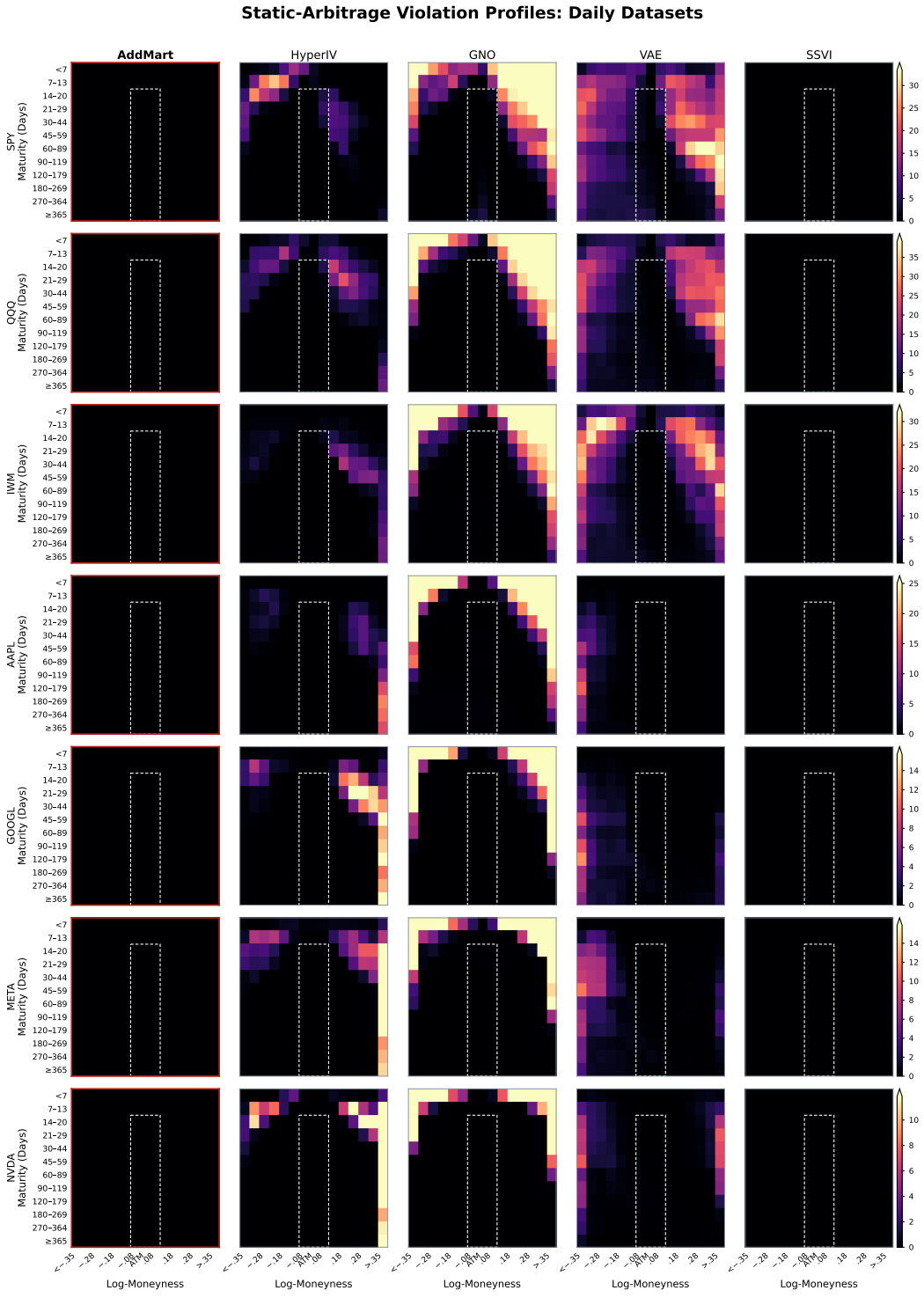

Figure 11: Static-arbitrage violation profiles across the seven daily datasets. Each cell reports the pooled calendar and butterfly pair violation rate (%) on a dense price grid, using a tolerance of 10^{-7}. Color scales are shared across methods within each asset row; the dashed box denotes the reliable region.

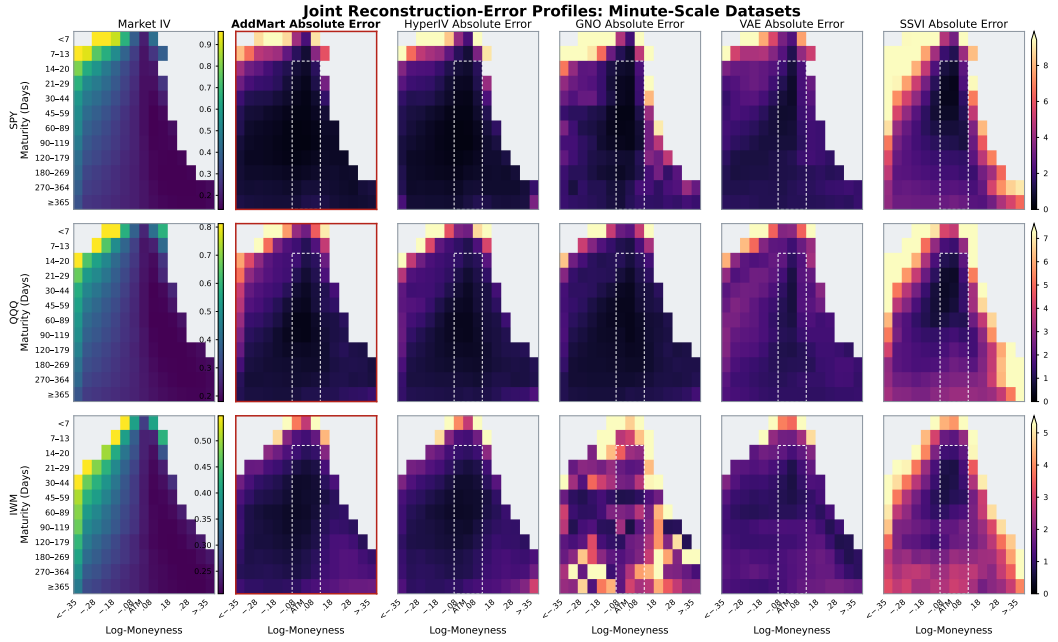

Figure 12: Joint reconstruction-error profiles on the SPY, QQQ, and IWM minute-scale datasets. Each heatmap reports IV-MAE ($\times 10^{-2}$) by log-moneyness and maturity; notation and masking follow Figure 10.

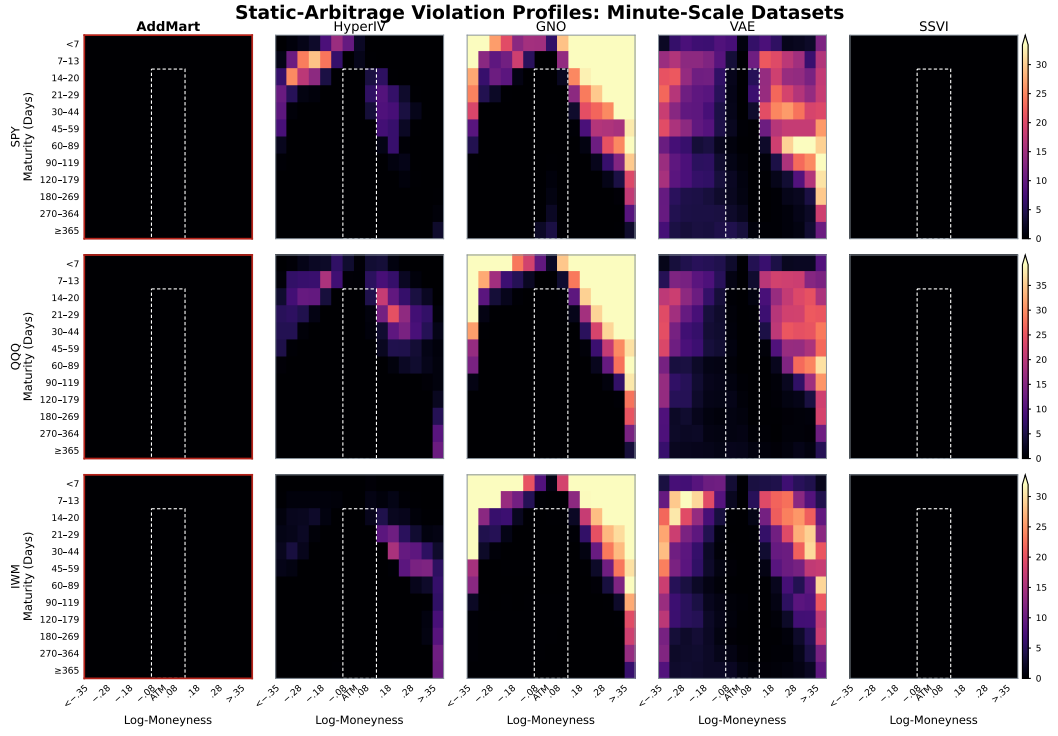

Figure 13: Static-arbitrage violation profiles on the SPY, QQQ, and IWM minute-scale datasets. Evaluation and notation follow Figure 11.

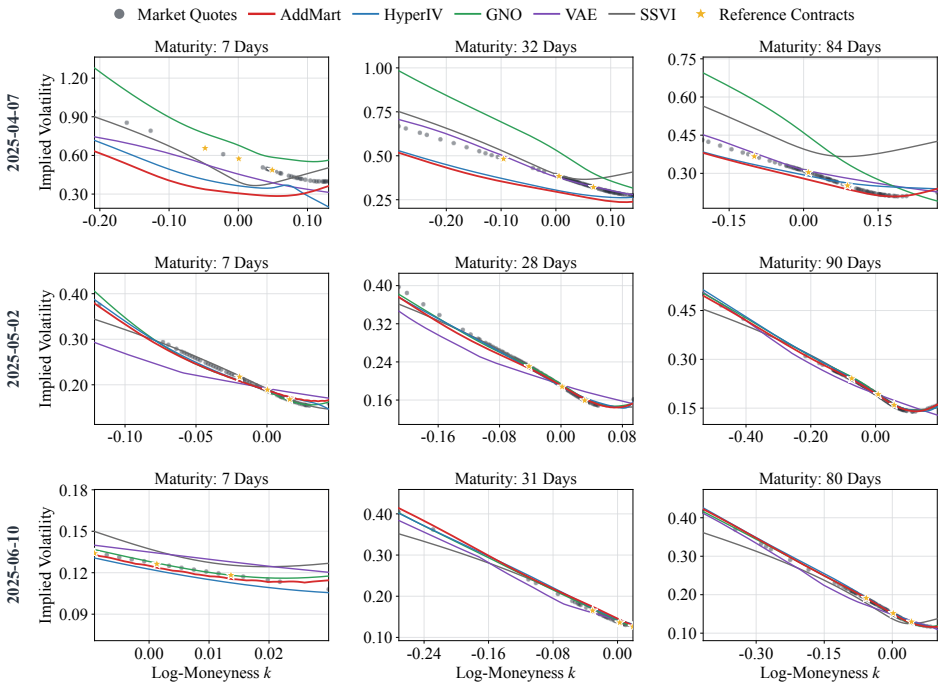

Figure 14: SPY implied-volatility slices on 7 April, 2 May, and 10 June 2025. Each row corresponds to one trading day and each column to one reference maturity. Gray dots denote market quotes and gold stars denote the nine reference contracts used by AddMart, HyperIV, GNO, VAE, and SSVI.

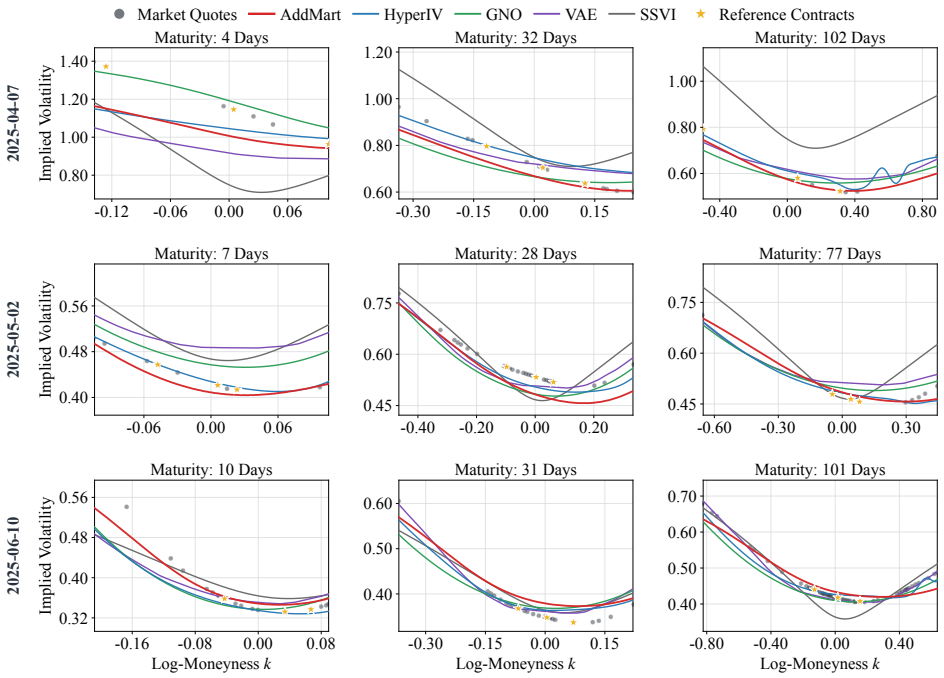

Figure 15: NVDA implied-volatility slices on 7 April, 2 May, and 10 June 2025. Each row corresponds to one trading day and each column to one reference maturity. Gray dots denote market quotes and gold stars denote the nine reference contracts used by AddMart, HyperIV, GNO, VAE, and SSVI.