PIVOT: BRIDGING PRICE AND IMPLIED-VOLATILITY OBJECTIVES VIA DIFFERENTIABLE BLACK–SCHOLES INVERSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern option-learning systems operate in two coordinates: *price space*, where markets quote and no-arbitrage constraints are most naturally enforced, and *implied volatility (IV) space*, where volatility surfaces are smoothed, regularized, and evaluated. The bottleneck is interface, not approximation: Jäckel's seminal *Let's Be Rational* (LBR) solver already inverts the Black–Scholes price to machine precision efficiently. What is missing is a differentiable layer that preserves LBR in the forward pass and avoids backpropagating through its branch logic. Such a layer must also confront the unavoidable singularity of the inverse map in the low-vega regime, where $\partial\sigma/\partial P = 1/\mathcal{V}$ diverges as $\mathcal{V} \to 0$. We close this gap with **PIVOT**, **P**rice–**I**mplied-**V**olatility **O**bjective **T**ranslator. PIVOT keeps the LBR forward pass intact and supplies the backward pass by implicit differentiation through the smooth Black–Scholes/Black-76 price map, with an explicit gating contract: invalid domains return `NaN`, well-conditioned rows receive the exact $1/\mathcal{V}$ gradient, and low-vega rows are attenuated rather than silently regularized. On a single H100, a fused Triton kernel reaches 1.79×10^9 IV/s at machine precision (9.3×10^{-14} max relative error vs. the reference C solver); end-to-end label generation sustains 48.9M/s on synthetic chains and 16.6M/s on SPX OptionMetrics. In a HyperIV-style one-day reproduction on SPX, PIVOT-augmented objectives Pareto-dominate the baselines, reducing held-out price MAE by up to 43.4% and the strongest three-seed gated objective improving price MAE by 38.8% and IV MAE by 21.3% jointly; cross-asset results on RUT, VIX, and NDX show directional price-MAE gains of 40.1%, 24.2%, and 16.7%, while an ungated IV-roundtrip control collapses to a degenerate near-zero surface, confirming the gate as a correctness contract rather than a tuning knob. Code: `https://anonymous.4open.science/r/pivot-anon-iclr2027/README.md`.

1 INTRODUCTION

Implied volatility is the inverse coordinate chart that maps an observed option price into the volatility parameter of the Black–Scholes or Black-76 pricing model (Black & Scholes, 1973; Black, 1976). In practice, option datasets are stored, smoothed, filtered, and learned in IV coordinates. Recent machine-learning work on volatility surfaces makes this dependence explicit: deep smoothing methods construct surfaces from sparse quote clouds (Ackerer et al., 2020; Wiedemann et al., 2025), HyperIV uses a hypernetwork to produce approximately arbitrage-free surfaces in real time (Yang et al., 2025), and generative approaches model distributions over approximately arbitrage-free volatility surfaces (Ning et al., 2023; Vuletić & Cont, 2024). These models differ architecturally, but they share a low-level dependency: millions of market prices $P^\star$ must be converted to accurate pointwise IV labels σ_{imp}, and many objectives compare models in both price and volatility coordinates.

This creates a mismatch between the numerical layer and the ML layer. Jäckel's "Let's Be Rational" (LBR) algorithm is the gold-standard IV solver (Jäckel, 2015; 2013), but the reference implementation is scalar C code exposed in Python through CPU wrappers such as `py_vollib` (Richards, 2023). That design is excellent for single-option analytics and weak for modern workloads: historical option universes contain millions of contracts (S, K, τ, r, q), neural models train on GPU tensors, and differentiable objectives require gradients through the pointwise inverse $P^\star \mapsto \sigma_{\mathrm{imp}}$.

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.

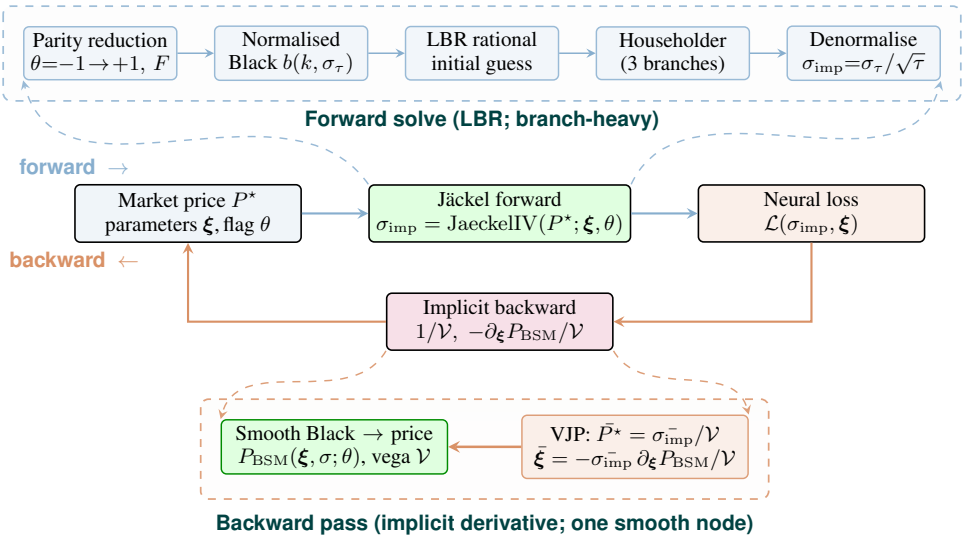


Figure 1: Autograd-native Jäckel IV. **Top:** The compact autograd layer expands into a branch-heavy LBR forward pass that solves $P_{\text{BSM}}(\boldsymbol{\xi}, \sigma_{\text{imp}}; \theta) = P^\star$ for σ_{imp} in normalized Black coordinates (k, σ_τ). **Bottom:** The backward pass bypasses every solver branch and applies the implicit derivative $\partial\sigma_{\text{imp}}/\partial P^\star = 1/\mathcal{V}$ through one smooth Black price node, delivering the implicit gradient.

The contribution of this paper is therefore not a learned replacement for Jäckel's solver. That would be a poor trade: the analytic–numerical solver is already accurate, deterministic, and – once GPU-vectorized – fast. The contribution is instead to make exact pointwise IV inversion an autograd-native systems primitive. The numerically delicate inverse remains in the forward pass; the gradient is supplied analytically by the implicit function theorem.

At a high level, PIVOT separates the two jobs that IV inversion normally entangles. The forward pass delegates each quote to the trusted Jäckel solver, vectorized over modern tensor backends, and returns both the IV value and explicit validity and conditioning information. The backward pass does not differentiate through the solver's rational approximants, branch masks, or Householder updates. Instead, it treats the IV as the exact local inverse of a smooth pricing map and supplies the corresponding analytic sensitivity directly with appropriate boundary behaviour as implied volatility or variance approaches zero. This design keeps the mature numerical inverse intact while making the singular low-vega regime visible to the training loop: well-conditioned quotes receive the exact inverse sensitivity, and ill-conditioned or invalid quotes can be masked or attenuated before their IV-space gradients reach the optimizer.

Contributions. We propose **PIVOT** (**P**rice–**I**mplied-**V**olatility **O**bjective **T**ranslator), which makes exact IV inversion usable as a differentiable bridge between price-space and IV-space objectives. The paper makes three linked contributions.

(C1) Jäckel as a batched GPU primitive. We provide a Jäckel-focused vectorized implementation of pointwise implied-volatility inversion. The forward pass retains LBR's normalized-Black construction $b(k, \sigma_\tau)$ rather than replacing it by a neural surrogate, and parallelizes the solver over heterogeneous option chains on NumPy, PyTorch, JAX, and Triton-style execution paths. The point is not to learn a new scalar IV formula – the analytic inverse is already exact in the bulk and the only loss from a neural surrogate would be in the tails – but to make the trusted numerical inverse available at the scale and interface required by neural pipelines.

(C2) Autograd-native implicit backward. We turn IV inversion into an autograd-native primitive. The PyTorch and JAX operators call the vectorized Jäckel solver in the forward pass and use implicit differentiation through the smooth Black–Scholes / Black-76 price in the backward pass, applying the analytic inverse-price sensitivity directly. This gives the exact inverse-function derivative without backpropagating through rational branch logic, masks, or Householder iterations, and decouples gradient cost from the number of solver iterations.

(C3) Conditioning-aware low-vega gating mechanism. We specify and test the gating mechanism needed to use IV inside losses. In existing pipelines, the standard remedy for the low-vega singularity is to *hand-engineer it out* of the training set – pre-filter low-vega rows, drop short-dated deep wings, generate offline IV labels only on the surviving subset – a workflow that is laborious, dataset-specific, and discards rows that price-space supervision could otherwise use. Our gating mechanism removes that step: below-intrinsic and arbitrage-violating quotes are flagged as invalid and return `NaN` from the inverse, the well-conditioned set $\mathcal{G}_\tau = \{|\mathcal{V}| > \tau\}$ receives the exact $1/\mathcal{V}$ gradient, and rows outside $\mathcal{G}_\tau$ are attenuated by either a hard mask $\mathbf{1}\{|\mathcal{V}| > \tau\}$ or the smooth gate $w_\tau(\mathcal{V}) = \mathcal{V}^2/(\mathcal{V}^2 + \tau^2)$ so that no NaN gradients enter the optimizer. Low-vega rows therefore stay in the batch and continue to contribute through price-space residuals, where the price-to-volatility map is well-conditioned regardless, while their singular IV-space channel is suppressed automatically. The training loop is freed from bespoke filtering and offline label preparation; the gating mechanism handles the conditioning in-line.

Empirical Results Summary. On a single H100 GPU (80 GB) under Linux/Ubuntu, the system produces 48.9 M synthetic IV labels/s and 16.6 M SPX OptionMetrics labels/s on 3.03 M real quotes, with self-consistency round-trip residual Q99 $\leq 1.8\times10^{-7}$ in price units across the 2018–2023 SPX panel. In a controlled HyperIV-style one-day reproduction, a PIVOT-augmented objective combining a price-MSE auxiliary with a w_τ-gated IV-roundtrip term Pareto-dominates the local vanilla baseline on SPX and repeats the directional improvement on RUT, VIX, and NDX, with best-run price MAE reductions of $43.4\%/40.1\%/24.2\%/16.7\%$ at IV MAE preserved within 10% of vanilla. An ungated diagnostic, included as a negative control, collapses to a degenerate near-zero surface (96.5% of test rows below $|\mathcal{V}| \leq 10^{-14}$) and confirms that the gating mechanism is a correctness contract rather than a tuning detail.

2 Background and Preliminaries

Setup and notation. We work with European-style options on the parameter tuple $\boldsymbol{\xi} = (S, K, \tau, r, q)$ – spot, strike, time-to-expiry, risk-free rate, and dividend yield – with put–call flag $\theta \in \{+1, -1\}$ ($+1$ for calls, -1 for puts). Under no arbitrage, the forward is $F = S\,e^{(r-q)\tau}$. We define log-forward-moneyness as $k = \log(K/F)$ and total volatility as $\sigma_\tau = \sigma\sqrt{\tau}$, and write $\Phi(\cdot)$ for the standard normal CDF. The discounted Black–Scholes price of a European call is

$$P_{\mathrm{BSM}}(\boldsymbol{\xi}, \sigma; +1) = S\,e^{-q\tau}\,\Phi(d_1) - K\,e^{-r\tau}\,\Phi(d_2),$$
$$d_{1,2} = \frac{\log(S/K) + (r-q)\tau \pm \frac{1}{2}\sigma_\tau^2}{\sigma_\tau}.$$

The corresponding put price follows from put–call parity: $P_{\mathrm{BSM}}(\boldsymbol{\xi}, \sigma; -1) = P_{\mathrm{BSM}}(\boldsymbol{\xi}, \sigma; +1) - Se^{-q\tau} + Ke^{-r\tau}$. Since $Se^{-q\tau} = Fe^{-r\tau}$, the call formula is equivalently the Black-76 form $e^{-r\tau}[F\Phi(d_1) - K\Phi(d_2)]$, with $d_{1,2} = (-k \pm \frac{1}{2}\sigma_\tau^2)/\sigma_\tau$. Following Jäckel (2015), the inverse problem reduces in forward coordinates to the dimensionless *normalized Black function* $b(k, \sigma_\tau)$, which is bounded in $[0, b_{\max}(k)]$ on out-of-the-money strikes and isolates the volatility-dependent part of the price from the rates and dividend curves. The full closed-form definitions, Black-76 form, and the bounds underpinning Jäckel's region-based branching are collected in Appendix A; the rest of the paper uses this notation throughout.

Implied volatility, vega, and the inverse map. Define $\mathcal{V} := \partial P_{\mathrm{BSM}}/\partial\sigma$. Vega is strictly positive on the admissible domain, so $\sigma \mapsto P_{\mathrm{BSM}}(\boldsymbol{\xi}, \sigma; \theta)$ is a strictly monotone bijection from $\mathbb{R}_+$ onto the no-arbitrage price interval. For an observed mid $P^\star$, the implied volatility σ_{imp} is the unique solution of

$$P_{\mathrm{BSM}}(\boldsymbol{\xi}, \sigma_{\mathrm{imp}}; \theta) = P^\star, \tag{1}$$

existence and uniqueness following directly from the bijection above (Black & Scholes, 1973; Black, 1976). IV is the working coordinate of volatility-surface modelling: it collapses the put–call distinction (a single scalar field $\sigma(k, \tau)$ describes both wings), lies in a narrow band $\sim [0.05, 1.0]$ regardless of moneyness, and exhibits stable structural features (smile, skew, term structure) that price space does not. The expanded justification with the put–call collapse, conditioning, and structural-stability arguments is in Appendix A.5. By the implicit function theorem, whenever $\mathcal{V} \neq 0$ the

Algorithm 1 Autograd-native Jäckel IV (one batched call).

Require: Market mid $P^\star$; parameters $\boldsymbol{\xi} = (S, K, \tau, r, q)$; put–call flag $\theta \in \{+1, -1\}$; gate threshold $\tau > 0$ (overloaded; context-dependent)
Ensure: Implied vol σ_{imp} and VJP $(\bar{P}^\star, \bar{\boldsymbol{\xi}})$ for any upstream cotangent $\bar{\sigma}_{\text{imp}}$

Forward *(vectorized LBR; gradients suppressed)*
1: Normalize: $F \leftarrow S\, e^{(r-q)\tau}$, $k \leftarrow \log(K/F)$, $\beta \leftarrow e^{r\tau} P^\star / \sqrt{FK}$.
2: Solve $\beta = b(k, \sigma_\tau)$ by LBR (rational seed + Householder) and set $\sigma_{\text{imp}} \leftarrow \sigma_\tau / \sqrt{\tau}$.
3: Save $(\boldsymbol{\xi}, \theta, \sigma_{\text{imp}})$; mask invalid rows (below-intrinsic, $\tau \le 0$) to `NaN`.

Backward *(implicit VJP through the smooth Black price)*
4: Recompute $P_{\text{BSM}}(\boldsymbol{\xi}, \sigma_{\text{imp}}; \theta)$ and $\mathcal{V}$ via autograd; partition by $\mathcal{G}_\tau = \{|\mathcal{V}| > \tau\}$.
5: On $\mathcal{G}_\tau$: $\bar{P}^\star = \bar{\sigma}_{\text{imp}} / \mathcal{V}$, $\bar{\boldsymbol{\xi}} = -\bar{\sigma}_{\text{imp}} (\partial P_{\text{BSM}} / \partial \boldsymbol{\xi}) / \mathcal{V}$.
6: Off $\mathcal{G}_\tau$: gate by $w_\tau(\mathcal{V}) = \mathcal{V}^2 / (\mathcal{V}^2 + \tau^2)$ or return `NaN` (§3.3).

inverse map's derivatives are

$$\frac{\partial \sigma_{\text{imp}}}{\partial P^\star} = \frac{1}{\mathcal{V}}, \qquad \frac{\partial \sigma_{\text{imp}}}{\partial \boldsymbol{\xi}} = -\frac{\partial P_{\text{BSM}} / \partial \boldsymbol{\xi}}{\mathcal{V}}, \tag{2}$$

the closed-form gradient an autograd-native IV layer should expose without backpropagating through any solver internals. Equation (2) is the foundation of the autograd-native treatment we develop in §3; it also exposes the only unavoidable failure mode of an exact differentiable IV inverse: as $\mathcal{V} \to 0$ the price-to-IV chart becomes singular and $1/\mathcal{V}$ diverges. We treat this regime explicitly in §3.3.

Why Jäckel as the forward solver. Jäckel's "Let's Be Rational" (LBR) algorithm is the reference scalar IV solver: rational seeds in normalized Black coordinates, region-specific transformed objectives on the bounded domain $0 \le b \le b_{\max}(k) \le 1$, and at most two Householder iterations to machine precision (Jäckel, 2015; 2013). Existing open-source wrappers such as `py_vollib` (Richards, 2023) expose this as scalar CPU code lacking autograd, batched-tensor input, and explicit validity masking, a workflow burden that even recent state-of-the-art neural surface models inherit (Wiedemann et al., 2025); PIVOT closes that interface gap. The full positioning relative to alternative scalar solvers is in Appendix A.6.

3 Differentiable Jäckel Solver

Each input row is a market parameter tuple $\boldsymbol{\xi} = (S, K, \tau, r, q)$ with put–call flag $\theta \in \{+1, -1\}$ and observed mid $P^\star$. The forward problem evaluates $P_{\text{BSM}}(\boldsymbol{\xi}, \sigma; \theta)$; the inverse problem solves $P_{\text{BSM}}(\boldsymbol{\xi}, \sigma; \theta) = P^\star$ for σ_{imp} in normalized Black coordinates (k, σ_τ), and (2) supplies the implicit-function-theorem derivative through $\mathcal{V}$. Algorithm 1 stitches these into a single differentiable layer; Figure 1 depicts the forward/backward split graphically.

3.1 Forward Pass: Vectorized Jäckel Inversion

Recall from §2 that, after absorbing the discount factor $e^{-r\tau}$ and the forward $F = S\, e^{(r-q)\tau}$, the normalized Black function (11) reduces the inverse problem to two dimensionless quantities (k, σ_τ). Concretely, given the market mid $P^\star$, Jäckel's algorithm solves $b(k, \sigma_\tau) = e^{r\tau} P^\star / \sqrt{FK}$ for the total volatility σ_τ, returning $\sigma_{\text{imp}} = \sigma_\tau / \sqrt{\tau}$.

The top panel of Figure 1 depicts the full forward pipeline. We delegate the forward solve to the FASTVOLLIB Jäckel backend (Saqur, 2026), which executes the standard LBR construction (Jäckel, 2015) — put–call reduction via parity, coordinate normalization to (k, σ_τ), numerically stable evaluation of $b(k, \sigma_\tau)$ with the asymptotics of (12), region-specific rational seeding, and Householder iterations on the transformed objective — over whole tensors, with NumPy, PyTorch, JAX, and Triton execution paths. The five-stage construction and its numerical analysis are documented in that reference; the contribution of the present paper is the autograd-native wrapper specified in §3.2–3.3 below.

Two properties of the forward solver are load-bearing for the wrapper and worth recording explicitly. First, every input is treated as a tensor: S, K, τ, r, q, and θ may all vary row-by-row, as they do in

batched chains and OptionMetrics panels with multiple expiries — so the backward must vectorize the implicit identity (2) elementwise without any scalar fallback. Second, all internal computation is in float64; the LBR boundary points and asymptotics in (12) make this essential for the singular tails where the cash price spans many orders of magnitude, and the gating mechanism of §3.3 inherits this precision floor when it tests $|\mathcal{V}| > \tau$. Invalid inputs (below-intrinsic, super-arbitrage, $\tau \leq 0$) are masked to `NaN` by the forward solver and propagate to the backward as the invalid branch of the gating contract.

3.2 Backward Pass: Implicit Autograd

The PyTorch implementation uses `torch.autograd.Function`; the JAX implementation uses `jax.custom_vjp`. Both backends share the same idea, motivated by (2): the branch-heavy Jäckel solver is opaque to autodiff, while the smooth discounted Black price $P_{\mathrm{BSM}}(\boldsymbol{\xi}, \sigma; \theta)$ supplies the required vector–Jacobian product. Concretely, given an upstream cotangent $\bar{\sigma}_{\mathrm{imp}}$, the backward returns:

$$\bar{P}^{\star} \;=\; \frac{\bar{\sigma}_{\mathrm{imp}}}{\mathcal{V}}, \qquad \bar{\boldsymbol{\xi}} \;=\; -\,\frac{\bar{\sigma}_{\mathrm{imp}}}{\mathcal{V}}\,\frac{\partial P_{\mathrm{BSM}}}{\partial \boldsymbol{\xi}}\bigg|_{\sigma=\sigma_{\mathrm{imp}}}, \tag{3}$$

where $\partial P_{\mathrm{BSM}}/\partial \boldsymbol{\xi}$ is obtained by ordinary autograd through the analytic Black price helper, evaluated at the recovered σ_{imp}. The expression for $\bar{P}^{\star}$ is the implicit gradient of the inverse map; the expression for $\bar{\boldsymbol{\xi}}$ chains it through the price's dependence on the market parameters.

Three properties follow. *Gradient cost is independent of solver complexity* — the number of Householder iterations, branch choices, and rational-seed regions used in the forward pass do not appear in (3) — so the backward decouples cleanly from any future changes to the LBR construction. *Memory use is low* because no solver trace is stored; the only quantities saved for the backward are $(\boldsymbol{\xi}, \theta, \sigma_{\mathrm{imp}})$. And *the same forward kernel serves both offline labeling and differentiable training*, eliminating the need for parallel CPU/GPU pipelines. The bottom panel of Figure 1 depicts the resulting backward pass: a single smooth Black price evaluation feeds autograd, while the LBR forward pass remains entirely opaque to differentiation.

3.3 Low-Vega Gating Mechanism

Jäckel's region splits make the *forward inversion* numerically reliable, including near price boundaries. They do not remove the mathematical ill-conditioning of the inverse map. As established in (2), when $\mathcal{V} \to 0$ the derivative $\partial \sigma_{\mathrm{imp}}/\partial P^{\star} = 1/\mathcal{V}$ diverges: price changes no longer identify volatility changes. No exact differentiable IV implementation can make this gradient finite without changing the objective being differentiated.

Our differentiable Jäckel operator therefore exposes the conditioning instead of hiding it. For invalid quotes – including below-intrinsic prices and arbitrage-violating mids – the forward returns `NaN`. For low-vega but otherwise valid quotes, the forward returns Jäckel's best estimate, while the backward partitions the batch into the well-conditioned set

$$\mathcal{G}_{\tau} \;=\; \big\{ i : |\mathcal{V}_i| > \tau \big\} \tag{4}$$

and its complement. Rows in $\mathcal{G}_{\tau}$ receive the exact implicit gradient (3). Rows outside $\mathcal{G}_{\tau}$ return a zero cotangent when the upstream gradient is already zero (as in NaN-aware reductions), and return `NaN` when a loss explicitly asks for an ill-conditioned gradient. This behavior is intentional: a downstream training loop that silently absorbs $1/\mathcal{V}$ overflow would corrupt every Adam step through the resulting NaN parameters and is not what an honest differentiable inverse should provide.

This gating mechanism supports two principled training patterns. First, use price-space losses in low-vega regions, where prices remain the observable stable coordinate and the price gradient is computed by ordinary autograd through the smooth Black price. Second, gate IV-space regularizers with a hard mask $\mathbf{1}\{|\mathcal{V}| > \tau\}$ or, preferred for differentiability, with the smooth gate

$$w_{\tau}(\mathcal{V}) \;=\; \frac{\mathcal{V}^2}{\mathcal{V}^2 + \tau^2}, \tag{5}$$

which interpolates monotonically from 0 on the singular tail to 1 on the well-conditioned bulk. This is not an arbitrage-free surface claim and not a learned solver approximation; it is the correct numerical treatment of a singular coordinate chart.

Cleaned production training data (e.g. SPX 2023) sits comfortably in the well-conditioned regime ($|\mathcal{V}| \gtrsim 2\times10^{-2}$); the gate matters when a non-trivial fraction of rows live in the singular tail. The gate's effect on the vega distribution, on the smooth weight w_τ for $\tau \in \{10^{-8}, 10^{-6}, 10^{-4}\}$, and on the (k, DTE) heat-map of the stress chain is illustrated in Fig. 2; the empirical conditioning consequences – ungated training collapses, gated training is stable – are visible directly in the SPX (Table 2, Fig. 17) and cross-asset (Table 3) results below.

4 Experiments

We empirically validate three research questions that the construction in §3 naturally raises. **(RQ1)** Is the differentiable Jäckel primitive (PIVOT) numerically correct and fast enough for ML-scale option universes $\{(\boldsymbol{\xi}_i, \theta_i, P_i^\star)\}_{i=1}^N$? **(RQ2)** Does the implicit backward pass realize the inverse-function-theorem identity $\partial\sigma_{\mathrm{imp}}/\partial P^\star = 1/\mathcal{V}$ of (2) on the well-conditioned set $\mathcal{G}_\tau$? **(RQ3)** Does the primitive change an actual surface-model training objective when price-space and IV-space losses are optimized together? The scaling experiments (§4.1) answer RQ1 and RQ2 on synthetic batches with known volatilities $\sigma^\star$. The real-world experiments (§4.2) answer RQ3 affirmatively: plugging PIVOT into the SOTA IV-smoothing model HyperIV (Yang et al., 2025) on daily SPX option quotes yields a Pareto-dominating point in the price/IV plane relative to the local vanilla baseline. A parallel set of GNO (Wiedemann et al., 2025) ablations and additional cross-asset repeats are deferred to Appendices B.3 and C.

Setup All experiments run on a single H100 (80 GB) under Linux/Ubuntu. Real-world data come from IvyDB OptionMetrics via WRDS (OptionMetrics, 2025; Wharton Research Data Services, 2025); the SPX 1-day flagship spans 2013-01-02 to 2023-08-31 (12.48 M train rows over 2,516 pre-2023 dates, 1.34 M test rows over 167 dates in 2023; HyperIV filters: OTM/ATM, mid $\geq$ \$0.10, $\tau \leq 2$ y, 9-reference-set requirement) and the cross-asset RUT/VIX/NDX repeats are reported in Appendix B.3, Table 4. All rows carry $(\boldsymbol{\xi}, \theta, P^\star)$ with $\boldsymbol{\xi} = (S, K, \tau, r, q)$. The underlying scalar Jäckel routine is `FAST-VOLLIB` (Saqur, 2026) with `backend="jackel"`; PIVOT wraps it in the autograd-native interface of §3.

4.1 Scaling Experiments

We begin with synthetic Black–Scholes / Black-76 batches in which the true volatility $\sigma^\star$ is known by construction. This isolates the numerical primitive from data-cleaning and surface-model effects: any error is attributable to the IV operator rather than to quote conventions, missing rate curves, or model capacity.

Throughput and forward accuracy. Table 1 summarizes the speed, throughput, and forward accuracy of the Jäckel forward solver underneath the PIVOT operator. On a synthetic 10^5-option batch, the vectorized Jäckel forward pass takes 1.39 ms, corresponding to roughly 72 M IV inversions/s. When the entire Jäckel pipeline (preprocessing, boundary computation, Hermite initial guess, three Householder(3) iterations, and postprocessing) is fused into a single Triton kernel that keeps every intermediate in registers and touches HBM only once per element, GPU compute time on the same canonical grid drops to 0.056 ms (1.79×10^9 IV/s) at machine precision (max relative error 9.3×10^{-14} vs. the reference C solver). In a streaming HyperIV-style label-generation loop over 64 batches of 65,536 options, FAST-VOLLIB produces 4.19 M labels in 0.086 s, or 48.9 M labels/s. This is roughly $195\times$ faster than a single-core scalar `py_lets_be_rational` loop on the same problem (or $\sim1905\times$ under the fused Triton kernel above), and is sufficient to move IV inversion from an offline preprocessing step into the inner loop of a GPU training pipeline.

Gradient correctness. We verify equation (2) by generating synthetic BSM prices $P_i^\star = P_{\mathrm{BSM}}(\boldsymbol{\xi}_i, \sigma_i^\star; \theta_i)$, recovering $\sigma_{\mathrm{imp},i}$ through PIVOT, and comparing autograd gradients to the analytic identity $\partial\sigma_{\mathrm{imp}}/\partial P^\star = 1/\mathcal{V}$. The well-conditioned subset uses $|\mathcal{V}| \geq 10^{-6}$ as in (4). On 95.2% of the synthetic batch, the price-gradient absolute error has Q50 6.9×10^{-18} and Q99 1.77×10^{-2}. The raw unmasked maximum is dominated by the low-vega tail, where $1/\mathcal{V}$ overflows to floating-point infinity; this is the expected conditioning singularity from §3.3, not a solver inaccuracy. The JAX and PyTorch implementations agree on mask rates and gradient quantiles up to float64 noise. The full quantile spectrum on the raw, $|\mathcal{V}|>10^{-14}$, and $\mathcal{G}_{10^{-6}}$ subsets is reported in Appendix B.6, Fig. 7.

Experiment	Backend / data	Scale	Result
Forward IV	H100 synthetic	10^5 options	1.39 ms median, 72 M IV/s
Forward IV (Triton fused kernel)	H100 synthetic	10^5 options	0.056 ms GPU compute, 1.79×10^9 IV/s
Label pipeline	H100 synthetic	4.19 M labels	48.9 M labels/s
JAX forward + backward	H100 synthetic	10^5 options	0.45 ms post-JIT
PyTorch forward + backward	H100 synthetic	10^5 options	3.15 s eager-mode
Scalar CPU LBR baseline	`py_lets_be_rational`	4096 options	1.60×10^5 IV/s ($\sim$195$\times$ slower)
SPX 2023	OptionMetrics	3.03 M quotes	0.18 s, 16.6 M IV/s
SPX 2020–2022 robustness	OptionMetrics	4.2–5.1 M quotes/yr	11–13 M IV/s, Q99 self-cons. $\leq 9.1\times10^{-13}$

Table 1: **Scaling experiments** on synthetic and SPX OptionMetrics data. The PyTorch row measures eager-mode autograd; the JAX row measures compiled `custom_vjp` execution. The Triton-fused row reports CUDA-event GPU compute time on the same canonical 10^5-option grid (wall-clock including host-to-device transfer is 2.1 ms; max relative error vs. `py_lets_be_rational` is 9.3×10^{-14}). The CPU baseline is a single-core Python loop over the reference `py_lets_be_rational` solver and is reported as a sanity check on the order-of-magnitude speedup attributable to GPU vectorization (with vendor-IV agreement Q99 1.7×10^{-3} on the well-conditioned subset).

4.2 IV Surface Generation or Smoothing with the Differentiable Jäckel Solver

The previous experiments establish that the differentiable IV operator is accurate and fast. We next ask whether it changes an actual neural volatility-surface training objective. We reproduce the SPX 1-day setting of HyperIV (Yang et al., 2025) using the released architecture pattern: a 9-contract reference set per date feeds a hypernetwork that emits the parameters of a compact surface network $\hat{\sigma}(k,\tau) = h_{\omega(Z)}(k,\tau)$ operating directly on the log-forward-moneyness coordinate $k = \log(K/F_{T_0,T})$ defined in §2. The vanilla objective is the HyperIV-style IV MSE plus the static-arbitrage auxiliary losses. We compare two PIVOT-augmented objectives:

$$\mathcal{L}_{\text{price}}(\hat{\sigma}) = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\left(\frac{P_{\text{BS}}(\boldsymbol{\xi}_i,\hat{\sigma}_i;\theta_i) - P_i^\star}{\frac{1}{|\mathcal{B}|}\sum_{j\in\mathcal{B}}|P_j^\star|}\right)^2, \tag{6}$$

$$\mathcal{L}_{\text{rt}}(\hat{\sigma}) = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} w_\tau(\hat{\mathcal{V}}_i)\left(\text{JaeckelIV}(\tilde{P}_i) - \sigma_i^\star\right)^2, \tag{7}$$

where $\hat{\mathcal{V}}_i = \partial P_{\text{BS}}/\partial\sigma\big|_{\sigma=\hat{\sigma}_i}$, w_τ is the smooth gate (5), $\sigma_i^\star = \text{JaeckelIV}(P_i^\star)$ is the market IV recovered from the observed mid (so (7) compares $\text{JaeckelIV}(\tilde{P}_i)$ to $\text{JaeckelIV}(P_i^\star)$ in IV space), and $\tilde{P}_i$ equals the model price $P_{\text{BS}}(\boldsymbol{\xi}_i,\hat{\sigma}_i;\theta_i)$ on well-conditioned rows $i \in \mathcal{G}_\tau$ and a sentinel market-consistent price on rows with $|\hat{\mathcal{V}}_i| \leq \tau$. The price loss is always computed from the original autograd-tracked model price; the sentinel is used only to protect the singular inverse-IV channel, so low-vega rows continue to contribute through price space. All variants below use the same SPX 1-day dataset described in the setup, the same put-call-parity-derived discount-rate convention, 100 epochs, Adam with learning rate 10^{-3}, batch size 128 dates, and 1024 sampled contracts per date.

Figure 16 in Appendix D visualizes the trade-off.

Both PIVOT-augmented objectives Pareto-dominate the local vanilla baseline on the 3-seed mean. At $\lambda_p = 0.1$ with the price auxiliary alone, per-seed price-MAE improvements range from 32.0% to 44.0% versus the vanilla 3-seed mean of 3.822, and the mean IV MAE improves from 0.017 to 0.015. At $\lambda_p = \lambda_{\text{rt}} = 0.1$ with the gated IV-roundtrip term and $\tau = 10^{-6}$, both axes improve over the price-only auxiliary, giving the strongest 3-seed cell (-38.8% price MAE, -21.3% IV MAE versus vanilla). The best single augmented run uses $\lambda_p = 0.3$ with seed 1. It is evidence that the differentiable PIVOT bridge changes the training objective in the intended direction under a controlled reproduction.

Why the gate matters. The roundtrip term (7) is not new supervision but a differentiable path through the exact inverse price map; the gate of §3.3 simply prevents that path from injecting $1/\mathcal{V}$ poison into the optimizer. Figure 2 anchors this in distributions: cleaned SPX 2023 rows sit comfortably in the well-conditioned regime ($|\mathcal{V}| \gtrsim 2\times10^{-2}$), while a stress chain places 86.5% of

Table 2: SPX 1-day IV-smoothing with HyperIV (Yang et al., 2025) and PIVOT. Lower IV MAE and Price MAE are better (↓). Δ columns are signed relative changes (in %) vs. the vanilla HyperIV reproduction (V0); negative Δ indicates lower error (improvement), positive Δ indicates regression. **Bold** marks the **best** value in each column among the PIVOT-augmented rows. The unsafe row is a diagnostic negative control showing the failure mode of composing the inverse IV map without the low-vega gating mechanism of §3.3.

Objective	Runs	IV MAE (↓)	Price MAE (↓)	Δ IV (%)	Δ Price (%)
HyperIV-vanilla (V0)	mean, $n = 3$	0.017 ± 0.001	3.822 ± 0.422	–	–
Price aux. (PIVOT), λ_p=0.1	mean, $n = 3$	0.015 ± 0.002	2.512 ± 0.277	−9.3	−34.3
Price aux. (PIVOT), λ_p=0.3	best	0.014	**1.983**	−17.9	**−48.1**
Price + gated rt. (PIVOT), τ=10^{-6}	mean, $n = 3$	$\mathbf{0.013 \pm 0.001}$	2.340 ± 0.172	**−21.3**	−38.8
Unsafe ungated rt.	diagnostic	0.235	36.937	+1308	+866

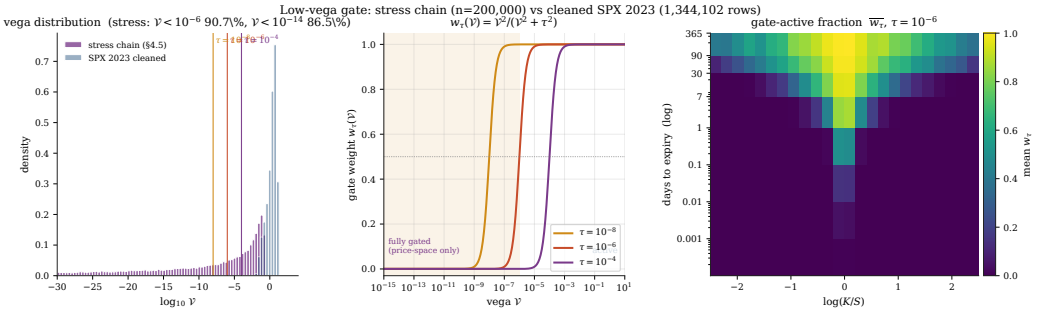


Figure 2: Low-vega gating mechanism in action. **Left:** log-scale vega histogram on the synthetic stress chain ($|\mathcal{V}| < 10^{-6}$ on 90.7% of rows; $|\mathcal{V}| < 10^{-14}$ on 86.5%) versus the cleaned SPX 2023 training panel (no row below $|\mathcal{V}| = 10^{-6}$). **Middle:** smooth gate weight $w_\tau(\mathcal{V}) = \mathcal{V}^2/(\mathcal{V}^2 + \tau^2)$ of (5) for $\tau \in \{10^{-8}, 10^{-6}, 10^{-4}\}$. **Right:** per-cell mean gate weight $\overline{w_\tau}$ at τ=10^{-6} on the stress chain over a (k, DTE) grid; deep wings at very short maturities are fully gated, and IV-space supervision concentrates in the well-conditioned interior.

its mass below $|\mathcal{V}| = 10^{-14}$; the smooth weight w_τ attenuates that tail continuously rather than hard-clipping it. With the sentinel-protected smooth gate, every variant trains with zero NaN-gradient steps; without it, the diagnostic accumulates 999 NaN steps in 50 epochs, test IV MAE 0.235, price MAE 36.94, and collapses to a degenerate near-zero surface (96.5% of test rows below $|\hat{\mathcal{V}}| \le 10^{-14}$); per-step traces are in Appendix B.5. Figure 17 contrasts the predicted IV surfaces of the three trainable variants – which track the OptionMetrics market smile within $\sim$2 vol points – against the ungated diagnostic on a representative SPX 2023 test date.

Discussion. This experiment is a stronger ML-facing demonstration than a pointwise roundtrip test: the differentiable Jäckel layer (PIVOT) is part of a surface-model training objective and changes the out-of-sample price error (**RQ3**). It is also intentionally narrow. We do not claim arbitrage-free IV surface generation – that property is inherited from the HyperIV baseline, which already enforces it. The supported claim is that, at matched conventions, exact differentiable IV makes it possible to train a HyperIV-style smoother with price-space consistency (6) and a gated IV-roundtrip channel (7), improving price MAE without sacrificing IV accuracy.

Cross-asset summary. The PIVOT-augmented construction repeats the SPX result on RUT, VIX, and NDX under matched local conventions (Table 3; full per-asset tables and Pareto views in Appendix B.3). Best-run price-MAE reductions range from 16.7–43.4% at IV MAE preserved within 10% of vanilla; the ungated control collapses on every asset (NaN-gradient steps in the hundreds-to-thousands; price MAE inflated by 9–30×). An analogous PIVOT-loss ablation on a graph neural operator (GNO) smoother (Wiedemann et al., 2025) reproduces the gated-vs-collapse signal on SPX/RUT/VIX but the price-MAE effect (1.7–3.7%) falls below our pre-registered 5% practical threshold; we therefore defer GNO to Appendix C and keep HyperIV as the headline.

Table 3: Cross-asset summary: best PIVOT-augmented HyperIV-style runs versus the local vanilla baseline for the same asset (left), and the ungated diagnostic on the same asset (right). All four augmented best runs preserve IV MAE within the 10% tolerance; the ungated diagnostic collapses on every asset, mirroring SPX.

	Best PIVOT vs. HyperIV (vanilla)		Ungated diagnostic	
Asset	Δ Price MAE	Δ IV MAE	NaN-grad steps	Δ Price MAE
SPX	$\mathbf{-43.4\%}$	-17.9%	999	$+866\%$
RUT	-40.1%	-16.6%	799	$+1054\%$
VIX	-24.2%	-17.0%	1,000	$+923\%$
NDX	-16.7%	$+1.5\%$	999	$+2949\%$

5 Related Work

Implied-volatility inversion. Classical methods solve (1) by Newton, Halley, or bracketed root-finding with problem-specific seeds (Manaster & Koehler, 1982; Brenner & Subrahmanyam, 1988; Corrado & Miller, 1996). Jäckel's LBR algorithm uses rational seeds in normalized Black coordinates on the bounded domain $0 \le b \le b_{\max}(k) \le 1$ and Householder iterations on region-transformed objectives (Jäckel, 2015).

Neural volatility surfaces. Deep smoothing (Ackerer et al., 2020), operator deep smoothing (Wiedemann et al., 2025), HyperIV (Yang et al., 2025), and arbitrage-free generative models (Ning et al., 2023; Vuletić & Cont, 2024) learn maps from sparse quotes, histories, or latent factors to surfaces $\hat{\sigma}(k, \tau)$. These generators sit *above* the differentiable inverse and – once one is available – can incorporate price-space (6) and gated IV-roundtrip (7) objectives. This paper contributes the missing pointwise differentiable layer beneath them.

Price/IV objectives and differentiable finance. Calibration writes losses in price space; neural IV-smoothing optimizes in IV space (§2). PIVOT makes the pointwise bridge between the two exact and differentiable subject to the §3.3 gate. Differentiable pricing libraries (Buehler et al., 2019; Imaki, 2021) supply forward gradients only; extended positioning is in Appendix A.9.

6 Conclusion

PIVOT turns Jäckel's LBR solver into an autograd-native, GPU-batched primitive: forward preserves the solver verbatim; backward applies the implicit derivative (2) through one smooth Black price node; the gating mechanism (§3.3) makes the chart's singularity a documented part of the layer's contract. Plugged into a HyperIV-style smoother it yields a 3-seed Pareto improvement on SPX (Pr. MAE -38.8%, IV MAE -21.3%), repeats on RUT/VIX/NDX (Table 3), and a parallel gating-vs-collapse signal on a graph-operator smoother (Wiedemann et al., 2025); limitations and outlook are in Appendix A.10.

Reproducibility statement

The anonymous code archive includes the operator source, PyTorch/JAX gradient tests, synthetic fixtures, matched benchmark scripts, and downstream training configurations. Its README and validation report separate executable smoke checks from historical aggregate results. Sections 3–4 and Appendices A–C describe the method and experimental protocols. Licensed WRDS/OptionMetrics rows are not redistributed; the synthetic checks do not reproduce the licensed-data results or every experiment reported in the paper.

AI use statement

For this venue conversion, generative AI was used to adapt the manuscript's LaTeX source to the official ICLR 2027 template and to check compilation and formatting. It was not used in this conversion to alter the scientific content, claims, experimental results, or figures. The authors reviewed the AI-assisted changes and take responsibility for the final manuscript.

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.

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.

Menachem Brenner and Marti G. Subrahmanyam. A simple formula to compute the implied standard deviation. *Financial Analysts Journal*, 44(5):80–83, 1988. doi: 10.2469/faj.v44.n5.80.

Hans Buehler, Lukas Gonon, Josef Teichmann, and Ben Wood. Deep hedging. *Quantitative Finance*, 19(8):1271–1291, 2019.

Charles J. Corrado and Thomas W. Miller, Jr. A note on a simple, accurate formula to compute implied standard deviations. *Journal of Banking & Finance*, 20(3):595–603, 1996. doi: 10.1016/0378-4266(95)00014-3.

Jim Gatheral. A parsimonious arbitrage-free implied volatility parameterization with application to the valuation of volatility derivatives. *Presentation at Global Derivatives & Risk Management, Madrid, May 2004*, 2004.

John C Hull and Sankarshan Basu. *Options, futures, and other derivatives*. Pearson Education India, 2016.

Shota Imaki. pfhedge: Deep hedging in pytorch. https://github.com/pfnet-research/pfhedge, 2021.

Peter Jäckel. Implementing "Let's be rational". https://www.jaeckel.org/LetsBeRational.7z, 2013.

Peter Jäckel. Let's be rational. *Wilmott*, 2015(75):40–53, 2015.

Steven Manaster and Gary Koehler. The calculation of implied variances from the black-scholes model: A note. *The Journal of Finance*, 37(1):227–230, 1982. doi: 10.1111/j.1540-6261.1982.tb01105.x.

Brian (Xin) 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.

OptionMetrics. Ivydb us: Historical option prices and smoothed implied volatility surfaces. https://optionmetrics.com/united-states/, 2025. Commercial database (since 1999) providing end-of-day option prices, greeks, and kernel-smoothed implied-volatility surfaces for thousands of names, widely used by funds and researchers. Accessed 2025-10-28.

Larry Richards. py_vollib: A python library for option pricing, implied volatility, and greeks. https://github.com/vollib/py_vollib, 2023. GitHub repository.

Raeid Saqur. Fast-Vollib: A fast implied volatility library for Python with PyTorch, JAX, and CUDA fused-kernel backends, 2026. URL https://arxiv.org/abs/2604.27210.

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

Wharton Research Data Services. Optionmetrics. https://wrds-www.wharton.upenn.edu/pages/about/data-vendors/optionmetrics/, 2025. Vendor summary; accessed 2025-10-28.

Ruben Wiedemann, Antoine Jacquier, and Lukas Gonon. Operator deep smoothing for implied volatility. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DPlUWG4WMw.

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*, volume 267, pp. 70550–70564. PMLR, 2025. URL https://proceedings.mlr.press/v267/yang25d.html.

Appendices

Appendix Contents

A Additional Background and Discussions **12**

A.1 European options and put–call parity 13
A.2 Notation 13
A.3 Black–Scholes, Black-76, and the normalized Black function 13
A.4 Vega and the option Greeks 14
A.5 Implied volatility as the working coordinate 14
A.6 Why Jäckel as the forward solver 15
A.7 Three roles of IV inversion in a modern option-learning pipeline 15
A.8 Forward pass: the five-stage LBR construction 16
A.9 Differentiable finance software (extended related-work note) 16
A.10 Limitations and outlook (extended) 16
A.11 Role in neural surface learning 17

B Additional Experiments and Details **17**

B.1 SPX OptionMetrics: real-data sanity check for the differentiable layer 17
B.2 The gating mechanism is a numerical-correctness requirement: a NaN-poisoning diagnostic 18
B.3 HyperIV: Cross-asset Generalization on RUT, VIX, and NDX 20
B.4 Anatomy of the low-vega gating mechanism 20
B.5 Per-step NaN-gradient trace for the SPX 1-day campaign 21
B.6 Gradient correctness: quantile spectrum on the well-conditioned set 21
B.7 Numerical-correctness diagnostics 21
B.8 On hyperparameter tuning of λ_p, λ_{rt}, and τ 22

C GNO Experiments **25**

C.1 GNO: Controlled SPX one-day loss ablation 25
C.2 GNO: Cross-asset one-day repeats on RUT and VIX 26

D Supplementary visualizations **32**

A Additional Background and Discussions

This appendix provides the full preliminaries and derivations summarised in §2: European-call setup, notation, the Black–Scholes / Black-76 closed-form prices, the normalized Black function $b(k, \sigma_\tau)$, vega and its closed-form expression, the IV inverse problem, the three reasons IV is the working coordinate of volatility-surface modelling, and the positioning of Jäckel's LBR algorithm.

A.1 European options and put–call parity

An option is a financial contract that gives its holder the right, but not the obligation, to transact an underlying asset at a fixed strike price $K \in (0, \infty)$, on a specified maturity date $T \in (0, \infty)$. A call option gives the right to buy the underlying, whereas a put option gives the right to sell it. Underlyings may include equity indices, single stocks, currencies, commodities, futures, or other derivative contracts. For simplicity of exposition, we use "option" to mean a European call option unless stated otherwise; European put prices can be transformed to (mathematically equivalent) call prices through the well-known *put–call parity*.

A.2 Notation

For a given option contract, we use $S \in \mathbb{R}_+$ for the current (aka. spot) price of the underlying asset, $K \in (0, \infty)$ to denote its strike price, τ to denote time-to-expire, with $\tau = T - T_0$, where $T \in (T_0, \infty)$ is the expiry date in the future. In practice, options are traded for fixed expiries $T_1, ..., T_m$; and only a finite range of strikes $K_1^i, ...K_{n_i}^i$ for each expiry T_i is available.

The forward price of the underlying asset for expiry T is denoted $F_{T_0,T} \in \mathbb{R}_+$ and, under the standard no-arbitrage assumption, is given by

$$F_{T_0,T} = S\, e^{(r-q)\tau}, \tag{8}$$

where $r \in \mathbb{R}$ is the risk-free interest rate and $q \in \mathbb{R}$ is the continuous dividend yield (or, more generally, the cost-of-carry) over the period $[T_0, T]$. We then define the *log-forward-moneyness* as $k = \log(K/F_{T_0,T})$, and write $\Phi(\cdot)$ for the standard normal CDF. Throughout this paper, we use $\sigma \in \mathbb{R}_+$ to denote the (annualized) volatility of the underlying, and $\sigma_\tau := \sigma\sqrt{\tau}$ to denote the corresponding *total volatility*, which captures the cumulative volatility over the life of the option.

A.3 Black–Scholes, Black-76, and the normalized Black function

Black–Scholes, Black-76. The seminal Black–Scholes (Black & Scholes, 1973) provides a closed-form, analytic formula for pricing European options – characterized by the single volatility $\sigma \in \mathbb{R}_+$ parameter under the so-called risk-free $\mathbb{Q}$-measure and the assumptions of constant volatility, lognormal returns, and continuous trading. Using the notation introduced above, the Black–Scholes price of a European call is

$$P_{\mathrm{BSM}}(S, K, \tau, r, q, \sigma) \;=\; S\, e^{-q\tau}\, \Phi(d_1) \;-\; K\, e^{-r\tau}\, \Phi(d_2), \tag{9}$$

where $d_{1,2} = \big[\log(S/K) + (r-q)\tau \pm \tfrac{1}{2}\sigma_\tau^2\big]/\sigma_\tau$. The corresponding put price follows from put–call parity.

In *Black-76*, the same equation is written in forward coordinates: the spot is replaced by the forward price F, the carry term cancels, and both option legs are discounted by $e^{-r\tau}$ (Black, 1976). This distinction matters in data pipelines because market prices are discounted cash prices, while many surface models operate on forwards, log forward-moneyness ($k = \log(K/F)$), and implied volatilities. Equivalently, in forward coordinates with log-forward-moneyness $k = \log(K/F_{T_0,T})$, the Black-76 form gives

$$P_{\mathrm{B76}}(F_{T_0,T}, K, \tau, r, \sigma) \;=\; e^{-r\tau}\big[\, F_{T_0,T}\, \Phi(d_1) \;-\; K\, \Phi(d_2)\,\big], \qquad d_{1,2} \;=\; \frac{-k \pm \frac{1}{2}\sigma_\tau^2}{\sigma_\tau}. \tag{10}$$

Normalized Black function. For numerical stability and analytical tractability, it is often preferable to work with a dimensionless form of the Black price rather than the raw cash price. Following Jäckel (2015), and restricting attention to European calls, we work with the *undiscounted* (forward) Black price $B := e^{r\tau}\, P_{\mathrm{B76}}$ and define the *normalized Black function*

$$b(k, \sigma_\tau) \;:=\; \frac{B(F, K, \tau, \sigma)}{\sqrt{FK}} \;=\; e^{-k/2}\, \Phi\!\left(\frac{-k}{\sigma_\tau} + \frac{\sigma_\tau}{2}\right) \;-\; e^{k/2}\, \Phi\!\left(\frac{-k}{\sigma_\tau} - \frac{\sigma_\tau}{2}\right), \tag{11}$$

where $k = \log(K/F_{T_0,T})$ is the log-forward-moneyness and $\sigma_\tau = \sigma\sqrt{\tau}$ is the total volatility. The function b depends only on the two dimensionless quantities (k, σ_τ): the rates curve enters only

through the discount factor $e^{-r\tau}$ and the forward $F = S\,e^{(r-q)\tau}$, both of which are absorbed before b is evaluated.

This separation of concerns is precisely the appeal of the normalization. Discounting and the forward are determined by the rates and dividend curves — a separate, linear calibration problem — while b isolates the genuinely nonlinear, volatility-dependent part of the price. The function admits the tight bounds

$$0 \;\le\; b(k,\sigma_\tau) \;\le\; b_{\max}(k) \;\le\; 1, \qquad b_{\max}(k) := e^{-k/2}, \quad k \ge 0, \tag{12}$$

for out-of-the-money calls,[1] which keeps b numerically well-behaved across the full range of strikes and maturities encountered in practice — including deep wings and short expiries where the cash price spans many orders of magnitude. The input space dimension is also reduced from the six raw Black-76 parameters $(F, K, \tau, r, q, \sigma)$ to just two (k, σ_τ), which is particularly attractive for learned surface models. For these reasons, the normalized form is the standard parameterization in modern implied-volatility routines (Jäckel, 2015) and in much of the recent ML-for-options literature.

A.4 Vega and the option Greeks

The sensitivities of the option price to its inputs — collectively known as the *Greeks* — play a central role both in risk management and, as we will see, in the inversion problem itself. The most important sensitivity for our purposes is *vega*, the derivative of the option price with respect to volatility:

$$\mathcal{V} \;:=\; \frac{\partial P_{\mathrm{BSM}}}{\partial \sigma}. \tag{13}$$

Under the Black–Scholes model, vega admits the closed form[2]

$$\mathcal{V} \;=\; S\,e^{-q\tau}\,\varphi(d_1)\,\sqrt{\tau} \;=\; F\,e^{-r\tau}\,\varphi(d_1)\,\sqrt{\tau}, \tag{14}$$

where φ is the standard normal density and d_1 is as in (9). Two properties of vega will matter throughout this paper. First, $\mathcal{V} > 0$ strictly, for all admissible (F, K, τ, σ) with $\tau > 0$ and $\sigma > 0$ — the option price is a strictly increasing function of volatility. Second, vega vanishes in the limits $\sigma \to 0$, $\sigma \to \infty$, and $\tau \to 0$ for $K \neq F$, which makes the price-to-volatility map ill-conditioned in those regions and is a recurring source of numerical difficulty for implied-volatility solvers. Other Greeks — delta ($\Delta = \partial P/\partial S$), gamma ($\Gamma = \partial^2 P/\partial S^2$), theta ($\Theta = \partial P/\partial \tau$), and rho ($\rho = \partial P/\partial r$) — appear only incidentally in this work; we refer the reader to standard references (Hull & Basu, 2016) for their definitions.

A.5 Implied volatility as the working coordinate

Let $P_{\mathrm{BSM}}(\boldsymbol{\xi}, \sigma; \theta)$ denote a discounted European option price under Black–Scholes or Black-76, where $\boldsymbol{\xi} := (S, K, \tau, r, q)$ collects the market parameters and $\theta \in \{+1, -1\}$ indicates a call $(+1)$ or put (-1). For an observed market mid-price $P^\star$, the implied volatility σ_{imp} is the unique solution of

$$P_{\mathrm{BSM}}(\boldsymbol{\xi}, \sigma_{\mathrm{imp}}; \theta) = P^\star. \tag{15}$$

Existence and uniqueness of σ_{imp} follow directly from the strict positivity of vega established above: the map $\sigma \mapsto P_{\mathrm{BSM}}(\boldsymbol{\xi}, \sigma; \theta)$ is a strictly monotone bijection from $\mathbb{R}_+$ onto the no-arbitrage price interval, so its inverse is well-defined wherever $P^\star$ is arbitrage-consistent.

Although price is the directly observable market quantity, practitioners and researchers alike work almost exclusively in IV space. This preference is principled rather than conventional, and rests on three properties of the IV representation. First, IV *collapses the put–call distinction*: by put–call parity, a call and put sharing the same (K, τ) admit the same implied volatility, so a single scalar field $\sigma(k, \tau)$ describes both wings of the market simultaneously — in our notation, the θ flag becomes irrelevant after inversion. Second, IV is *numerically better-conditioned* than price as a representation: option prices span many orders of magnitude across strikes and maturities — deep-OTM short-dated options are vanishingly small, while deep-ITM long-dated options approach intrinsic value —

[1] For in-the-money calls ($k < 0$) the put–call invariance maps the problem back to the OTM regime, so it suffices to analyze $k \ge 0$. The bounds in (12) underpin the region-based asymptotic expansions and branching strategy used in our Jäckel solver following the *Let's Be Rational* algorithm (Jäckel, 2015).

[2] More precisely: $\mathcal{V} = S\,e^{-q\tau}\,\varphi(d_1)\,\sqrt{\tau} = S\,e^{-q\tau}\,\varphi(d_1)\,\frac{\partial \sigma_\tau}{\partial \sigma}$,

whereas implied volatilities typically lie in a narrow band of $[0.05, 1.0]$ regardless of moneyness. This bounded, well-scaled range makes IV a far more tractable target for interpolation, regression, and neural-network outputs. We note that this scale advantage does *not* mean equal IV errors correspond to equal pricing errors: by the chain rule, $\Delta P \approx \mathcal{V}\,\Delta\sigma$, and since vega varies by orders of magnitude across the surface, equal-magnitude IV errors translate to highly unequal dollar errors — which is precisely why practitioners often adopt vega-weighted losses or train directly in price space (Yang et al., 2025). Third, the IV surface exhibits *empirically robust structural features*: the volatility smile, the term structure of ATM volatility, and the skew slope persist across instruments and time, which has historically made IV a natural coordinate for arbitrage-free interpolation, smoothing, and cross-sectional comparison once Black–Scholes–Merton made the inversion well-defined in closed form.

For these reasons, IV is not merely a reporting convention but the coordinate system in which option surfaces are quoted, calibrated, smoothed, and compared. Classical no-arbitrage parameterizations such as SVI (Gatheral, 2004) and modern neural smoothers including Deep Smoothing (Ackerer et al., 2020), Operator Deep Smoothing (Wiedemann et al., 2025), and HyperIV (Yang et al., 2025) all rely on pointwise IV labels or IV-space losses. The market observable, however, remains price — a discrepancy that motivates the calibration and inversion machinery developed in the main paper. Crucially, the implicit-function-theorem identity (2) unifies these two perspectives: any IV-space loss can be transparently composed with the price map via autograd, with vega supplying the correct economic reweighting automatically, and obviating the need for hand-engineered vega weights when a model is trained jointly in price and IV coordinates.

The forward and inverse maps are linked through vega via the implicit function theorem. Defining the residual $F(\sigma, \boldsymbol{\xi}, P^\star) := P_{\mathrm{BSM}}(\boldsymbol{\xi}, \sigma) - P^\star$, the condition $F = 0$ implicitly defines σ_{imp} as a function of $(\boldsymbol{\xi}, P^\star)$, and whenever $\mathcal{V} \neq 0$ its derivatives are those of equation (2) in the main text.

A.6 Why Jäckel as the forward solver

The inverse problem in (15) is one-dimensional but numerically delicate. Near intrinsic value, near expiry, and in deep wings, small price perturbations can correspond to large volatility changes. Jäckel's *Let's Be Rational* (LBR) algorithm addresses these difficulties by moving to normalized Black coordinates, using rational approximations to seed the iteration, and applying region-specific transformed objectives so that Newton-like iterations remain well-conditioned across the entire price domain (Jäckel, 2015; 2013). It achieves machine-precision implied volatilities with at most two higher-order (Householder) iterations, and is the de facto standard in production analytics libraries.

Given this, our contribution is deliberately *not* a learned pointwise IV solver: the numerical forward inverse already exists, is fast, deterministic, and accurate, and is precisely the object we want to preserve. The remaining gap is one of *interface* rather than algorithm. Existing open-source implementations such as `py_vollib` (Richards, 2023) predate modern ML frameworks: they expose scalar CPU APIs, lack autograd integration, and require hand-rolled batching, preprocessing (filter out low-vega rows, pre-calculate IV labels for training etc.) to be usable in a training loop — a burden that even recent state-of-the-art surface models inherit (Wiedemann et al., 2025). Neural surface training, however, operates on large heterogeneous tensors and propagates gradients end-to-end. The primitive it needs is not "an IV number for one contract" but a batched, autograd-native operator that can be invoked inside a computation graph, applied to millions of contracts simultaneously, and that explicitly surfaces invalid or ill-conditioned rows rather than silently failing on them. Closing this interface gap is precisely contribution **(C1)** of this paper: a vectorized, GPU-resident Jäckel solver with analytically supplied gradients (per (2)) and explicit validity masking.

A.7 Three roles of IV inversion in a modern option-learning pipeline

The numerical primitive needed in modern ML pipelines is not merely a faster scalar routine. In a modern option-learning pipeline, IV inversion plays three distinct roles. First, it is an offline label-generation step for historical quote universes: batched $P^\star \mapsto \sigma_{\mathrm{imp}}$ over millions of contracts with float64 stability. Second, it is an evaluation coordinate: models may train in price space while practitioners and reviewers report errors in IV. Third, and least supported by existing tooling, it is a differentiable bridge between objectives – a neural model can emit prices or volatilities, and the loss can compare the result in either coordinate, provided the inverse map exposes the gradient required

by the implicit function theorem. These roles impose a stricter interface than traditional analytics libraries: the solver must be vectorized over heterogeneous contracts, robust in the tails, explicit about invalid and low-vega rows, and compatible with autograd. Jäckel's algorithm already supplies the right scalar numerical method (Appendix A.6); the contribution of §3 is to package that method as a batched, GPU-scale, autograd-native layer without replacing it by a learned approximation.

A.8 Forward pass: the five-stage LBR construction

The forward pipeline expands the single solve in Algorithm 1 into a five-stage construction following Jäckel (2015). We delegate this to FAST-VOLLIB (Saqur, 2026), which executes the following stages over whole tensors with NumPy, PyTorch, JAX, and Triton execution paths.

1. **Put–call reduction.** For $\theta = -1$ apply put–call parity to obtain an equivalent call mid; this halves the branch logic and exploits the put–call invariance of b at the price of a single linear correction in $\boldsymbol{\xi}$.
2. **Coordinate normalization.** Map $(\boldsymbol{\xi}, P^\star)$ to the dimensionless pair (k, β) via $F = S\,e^{(r-q)\tau}$, $k = \log(K/F)$, and $\beta = e^{r\tau}P^\star/\sqrt{FK}$. All subsequent stages operate in the (k, σ_τ) chart.
3. **Stable b evaluation.** Evaluate $b(k, \sigma_\tau)$ with the numerically stable branches of (11), including the asymptotic expansions used near the bounds $0 \le b \le b_{\max}(k) \le 1$ (12).
4. **Rational seed and LBR boundary.** Compute the LBR boundary point and the rational initial guess $\sigma_\tau^{(0)}$ for the relevant region (lower-tail, middle, or upper-tail), as in Jäckel (2015).
5. **Householder iterations.** Apply a fixed number of Householder iterations on the region-appropriate transformed objective, denormalize to $\sigma_{\text{imp}} = \sigma_\tau/\sqrt{\tau}$, and mask invalid inputs (below-intrinsic, super-arbitrage, $\tau \le 0$) to `NaN`.

Two design choices follow directly from the heterogeneity of real option data. First, every input is treated as a tensor: S, K, τ, r, q, and θ may all vary row-by-row, as they do in batched chains and OptionMetrics panels with multiple expiries. Second, all internal computation is in float64; the LBR boundary points and asymptotics in (12) make this essential for the singular tails where the cash price spans many orders of magnitude.

A.9 Differentiable finance software (extended related-work note)

Deep hedging and differentiable pricing libraries emphasize gradients through forward pricing maps $(\boldsymbol{\xi}, \sigma) \mapsto P$, often within risk-neutral simulation (Buehler et al., 2019; Imaki, 2021). They generally do not provide a machine-precision differentiable IV inverse $P^\star \mapsto \sigma_{\text{imp}}$, so any objective that mixes price-space and IV-space terms must either fall back on finite-difference gradients (which fail on a non-trivial fraction of every realistic chain, see Appendix B.7) or detach the inverse from the computation graph. The missing primitive is precisely the autograd-native Jäckel operator we contribute, with the implicit backward (3) replacing the brittle differentiation of solver internals.

A.10 Limitations and outlook (extended)

The method is scoped to European Black–Scholes / Black-76 contracts on the parameter tuple $\boldsymbol{\xi} = (S, K, \tau, r, q)$ and assumes $\mathcal{V} > 0$ for meaningful IV gradients. Low-vega regions are not a corner case that can be solved by a better neural architecture: as established in (2) and §3.3, they are singular points of the price-to-IV coordinate chart. PIVOT exposes this through the well-conditioned set $\mathcal{G}_\tau = \{|\mathcal{V}| > \tau\}$, the smooth gate $w_\tau(\mathcal{V}) = \mathcal{V}^2/(\mathcal{V}^2 + \tau^2)$, and NaN-aware gradient semantics on the complement; downstream training pipelines should use price-space losses or w_τ-gated IV losses on the singular tail. The SPX experiment is pointwise and does not prove or enforce static arbitrage-free surface properties; those constraints belong to surface models built on top of the IV primitive, not to the pointwise inverse layer itself. Looking forward, the same operator-level interface generalises naturally to American and barrier payoffs, intraday quote universes, and joint calibration objectives that span price, IV, and Greek targets – all of which become tractable once the price↔IV bridge is exact, batched, and conditioning-aware.

A.11 Role in neural surface learning

Neural IV smoothers and operator models use reference quotes, sparse chains, or latent histories to generate complete volatility surfaces $\hat{\sigma}(k, \tau)$. Their architectural goals differ: Deep Smoothing emphasizes static-arbitrage penalties, OpDS (Wiedemann et al., 2025) learns operators over option clouds, and HyperIV uses a hypernetwork to produce surface parameters in real time. The present paper sits below these architectures and contributes the missing autograd-native primitive of **(C2)**: an implicit-backward operator that lets a surface model compare $\hat{\sigma}$ to IV labels, compare $P_{\mathrm{BSM}}(\hat{\sigma})$ to market prices, and compose $\mathrm{JaeckelIV}(P_{\mathrm{BSM}}(\hat{\sigma}))$ inside the loss while respecting the singularity in (2). The forward solver is opaque to autodiff – by design, since differentiating through Jäckel's branch logic, region masks, and Householder iterations would be both expensive and brittle – and the smooth Black price helper supplies the required vector–Jacobian product analytically through (2). This is the bridge that allows surface generators to train jointly in the price coordinate (where markets quote) and the IV coordinate (where surfaces are reported and regularized), without sacrificing either.

B Additional Experiments and Details

B.1 SPX OptionMetrics: real-data sanity check for the differentiable layer

The synthetic experiments above isolate the implicit-gradient identity (2) and the low-vega gating mechanism of §3.3 on data we generated from a known $\sigma^\star$. This subsection asks the operational question that matters for downstream ML pipelines: *does the same operator survive composition with itself on real SPX quote geometry, where the market parameters $\boldsymbol{\xi} = (S, K, \tau, r, q)$ and quote conventions are not under our control?* We use SPX OptionMetrics 2023 quotes with the OptionMetrics forward $F = S\,e^{(r-q)\tau}$ and a compact filter ($\texttt{min_days} = 14$, $\texttt{max_days} = 120$, $|k| < 0.20$ on the log-forward-moneyness $k = \log(K/F)$); for the per-quote optimization we use Black-76 with the vendor forward and $r = 0$, which is sufficient to exercise the operator and deliberately not an attempt to reproduce vendor IV exactly.

Pointwise label generation. After basic quote filtering, 3.03M rows are inverted in 0.183 s, corresponding to 16.6M labels/s. Round-tripping the recovered IVs through the Black price gives self-consistency residual Q50 1.1×10^{-13} and Q99 9.1×10^{-13} on finite outputs. The run reports a 17.8% invalid-domain rate and a 9.9% low-vega rate at threshold 10^{-6}. We report these rates explicitly because below-intrinsic observations, stale prints, and near-expiry tails are part of the real data interface; the low-vega gating mechanism of §3.3 means each class is handled by a different branch (NaN, exact gradient, gated gradient) rather than silently averaged together. The same pipeline run on SPX 2020–2022 (4.21M, 5.07M, and 4.87M rows respectively) yields aggregate throughputs of 11–13M labels/s and round-trip self-consistency Q99 $\leq 9.1 \times 10^{-13}$ at every year, with an NaN rate that tracks the realized vol regime (2022 8.9% / 12.3% / 2023 17.8% reflects the year-over-year rise in below-intrinsic stale mid-quotes during the 2022 bear market). Vendor-IV agreement is Q99 5×10^{-2} to 9×10^{-2} across these years, attributable to differences in OptionMetrics' discount curve, dividend treatment, and quote conventions rather than solver error (§3.3; full per-year breakdown in supplementary).

Differentiable IV *inside* the loss, illustrated on SPX. We then run a small per-quote optimization that exercises the differentiable layer end-to-end on the same SPX slice. A scalar parameter per quote is reparameterized through softplus to a positive $\hat{\sigma}_i$, mapped through the analytic Black price to $\hat{P}_i = P_{\mathrm{BS}}(\boldsymbol{\xi}_i, \hat{\sigma}_i; \theta_i)$, then mapped back to IV by `implied_volatility_autograd` to produce an *IV round-trip* $\sigma_i^{\mathrm{rt}} = \mathrm{JaeckelIV}(\hat{P}_i)$.

Two losses tap this chain at two different points:

$$\mathcal{L}^{\mathrm{SPX}}_{\mathrm{hybrid}}(\hat{\sigma}) = \underbrace{\frac{1}{|\mathcal{Q}|}\sum_{i\in\mathcal{Q}}\big(\hat{P}_i - P_i^\star\big)^2}_{\text{tap at price}} + \lambda\,\underbrace{\frac{1}{|\mathcal{Q}|}\sum_{i\in\mathcal{Q}}\big(\sigma_i^{\mathrm{rt}} - \sigma_i^\star\big)^2}_{\text{tap at IV, after JaeckelIV}^{-1}}, \tag{16}$$

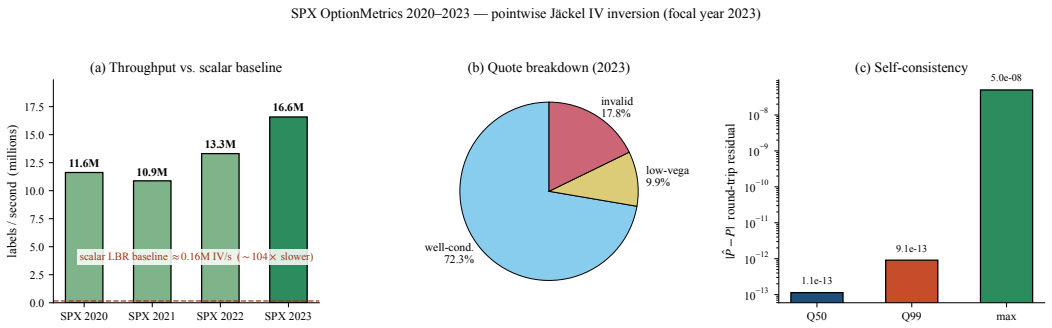


Figure 3: SPX OptionMetrics pointwise IV benchmark (focal year 2023): throughput, quote-class breakdown, and self-consistency residuals. **(a) Throughput** reports labels/s for SPX 2020–2023 under the identical pipeline; the dashed reference line is the single-core `py_lets_be_rational` (LBR) scalar baseline at ~ 0.16M IV/s (Table 1), giving a $\sim 100\times$ speedup at every year. **(b) Quote breakdown** shows the invalid-domain (NaN) and low-vega rates at threshold 10^{-6} on the focal year. **(c) Self-consistency** = the price round-trip residual $|\hat{P} - P|$ from $\hat{P} \to \sigma_{\text{imp}} \to \hat{P}$ on the focal year, i.e. how closely the recovered σ reproduces the input price when re-evaluated through the Black map; bars show **Q50**, **Q99**, and **max**. This is a real-geometry sanity check that the differentiable-IV operator composes with itself stably; it is a label-generation and pointwise differentiable-IV test, not an arbitrage-free surface claim. On the focal SPX slice $\sim 98\%$ of rows lie above the low-vega threshold, so the figure measures the easy regime; the regime where the gate matters is reported in Fig. 4.

where $\sigma_i^\star = \text{JaeckelIV}(P_i^\star)$ is the Jäckel IV recovered from the market mid. Per row i the IV-roundtrip term contributes a chain-rule gradient

$$\frac{\partial}{\partial \hat{\sigma}_i}\left(\sigma_i^{\text{rt}} - \sigma_i^\star\right)^2 = 2\left(\sigma_i^{\text{rt}} - \sigma_i^\star\right) \cdot \underbrace{\frac{1}{\mathcal{V}_i}}_{\text{JaeckelIV}^{-1}\text{ backward}} \cdot \underbrace{\mathcal{V}_i}_{\text{BS forward backward}}, \tag{17}$$

which makes precise the cancellation $1/\mathcal{V} \cdot \mathcal{V} = 1$ used implicitly by (2). Equation (17) is the precise meaning of "differentiable IV inside the loss": the autograd path through JaeckelIV inserts the $1/\mathcal{V}$ factor that the implicit function theorem requires, and the analytic Black backward inserts the $\mathcal{V}$ factor that cancels it. Without the differentiable layer this composition is not expressible; offline label generation gives access to $\sigma^\star$ but not to a gradient that flows through $P^\star \mapsto \sigma_{\text{imp}}(P^\star)$. The two factors cancel *analytically*; numerically they cancel only on the well-conditioned subset $\mathcal{G}_\tau$, which is precisely why the gating mechanism in §3.3 exists. The price-loss term in (16) is $\mathcal{V}$-weighted and well-defined everywhere, so low-vega rows continue to contribute through price space even when their IV-roundtrip channel is gated.

What the SPX hybrid run shows. Figure 3 reports throughput, the quote-class breakdown, and the self-consistency residuals. Over 100 Adam steps on $|\mathcal{Q}|$=2048 SPX quotes the per-row sigma RMSE falls from ≈ 0.10 to ≈ 0.026 and both loss components in (16) drop by 5–6 orders of magnitude in lockstep – the operator survives composition with itself on real geometry without NaN poisoning the shared parameter. For this quote slice the gate-kept rate is `iv_gate_kept` ≈ 0.98 at every step; i.e., only $\sim 2\%$ of rows hit the singular complement of $\mathcal{G}_\tau$. This is the expected behavior on a well-cleaned SPX sample but it also means an ablation on this slice would understate the value of keeping low-vega rows in the loss – the gate is barely active. The ablation that isolates the gate is therefore reported separately, on a synthetic chain engineered as the worst case (§B.2).

B.2 The gating mechanism is a numerical-correctness requirement: a NaN-poisoning diagnostic

The preceding SPX subsection §B.1 shows that the differentiable layer survives composition on real data when the gate is mostly inactive. The complementary question is what happens when it is *not* mostly inactive – when a non-trivial fraction of rows live in the singular tail of the price-to-IV

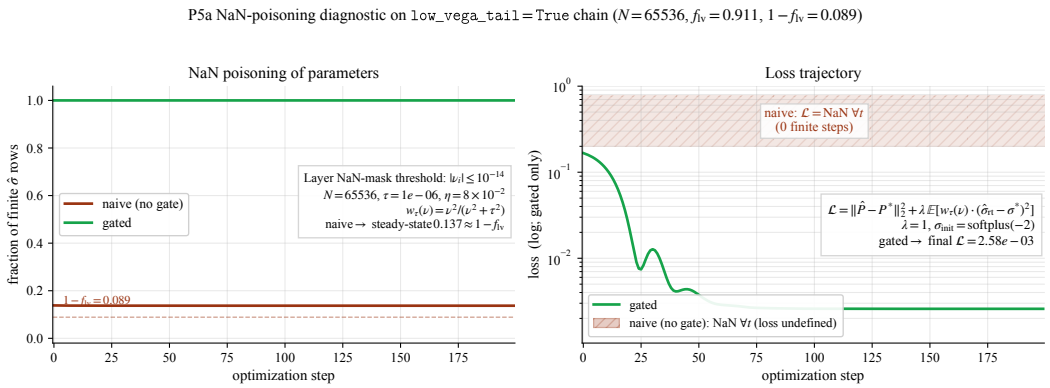


Figure 4: Per-step diagnostic of NaN poisoning on the `low_vega_tail=True` chain ($N = 65{,}536$, seed 31; rows with $|\mathcal{V}| \le 10^{-14}$: 0.911). Naive composes $\sigma_{\text{imp}} \circ P_{\text{BS}}$ in the loss with no gate and no gradient sanitization; gated applies sentinel-replace below a numerical safety floor and weights the per-row IV residual by $w_\tau(\mathcal{V})$. **Left:** fraction of $\hat{\sigma}_i$ rows that remain finite per step. Naive collapses from 1.0 to 0.137 in a single step (slightly above the $1 - f_{\text{lv}} = 0.089$ floor implied by the strict 10^{-14} mask — a small additional set of rows underflows in the implicit $1/\mathcal{V}$ at the singular tail) and stays there; gated stays flat at 1.0 for all 200 steps. **Right:** loss trajectory (log y). The naive objective produces a NaN loss at *every* step (no finite point to plot); we render this explicitly as the hatched red band rather than an empty gap, so the absence of a curve is annotated. Gated decreases monotonically modulo Adam transients (10 increasing steps out of 199, all of magnitude $<10^{-4}$), reaching final loss 2.58×10^{-3}. The figure asserts a correctness property of the layer's documented backward contract, not an optimization speed-up.

chart, i.e. outside the well-conditioned set $\mathcal{G}_\tau$ of (4). A natural first instinct is to compare loss variants on a stress chain by full-population sigma RMSE. We tried this and report the negative result here so that future work does not re-tread it: on a chain extreme enough to make the gate matter (`low_vega_tail=True`: deep wings, $\tau \in [10^{-8}, 10^{-0.5}]$ yr, $|k| \le 2.5$ on the log-forward-moneyness, 91.1% of rows below $|\mathcal{V}| \le 10^{-14}$), the chain is bimodal: only 2.13% of rows fall in the intermediate band $|\mathcal{V}| \in (10^{-6}, 10^{-2}]$ where price-space residuals remain actionable but IV-space gradients are unstable, with the remaining 96% split between trivially recoverable rows ($|\mathcal{V}| > 10^{-2}$, 4.0%) and rows that are numerically unrecoverable by any method ($|\mathcal{V}| \le 10^{-6}$, 93.9%). The middle band is precisely the regime the gate is designed to cushion; when it is empty there is nothing for a quantitative comparison to resolve. A four-variant Adam race on this chain at N=65,536 for 800 steps (price-only, IV-only with deletion, hybrid hard-mask, hybrid smooth-gate w_τ) plateaued at final RMSEs of 0.358, 0.357, 0.358, 0.358 respectively – a spread of 1.95×10^{-3}, well below the $\sim 5 \times 10^{-3}$ noise floor implied by differential exposure to gradient sanitization. The quantitative race is therefore not a defensible value-prop claim; the defensible claim is qualitative.

The qualitative claim. The differentiable layer's documented backward contract returns NaN when $|\mathcal{V}| \le 10^{-14}$ (§3.3); the surrounding training code is responsible for ensuring those NaNs do not enter the optimizer. Without a gate, composing $\sigma_{\text{imp}} \circ P_{\text{BS}}$ inside a loss produces NaN gradients on every row in the singular complement of $\mathcal{G}_\tau$ at every step; Adam propagates these into NaN parameters within one optimizer step; $P_{\text{BS}}(\text{NaN}) = \text{NaN}$ then poisons all downstream forwards; the loss is NaN and stays NaN. *With* the gate (sentinel-replace below a numerical safety floor, smooth weight $w_\tau(\mathcal{V}) = \mathcal{V}^2/(\mathcal{V}^2 + \tau^2)$ on the per-row IV residual), every row's gradient stays finite throughout training and the loss decreases monotonically. This is a binary correctness property of how the layer must be *used*, not an optimization-speed comparison.

Why this is the right framing. A quantitative ranking of synthetic-chain RMSEs is sensitive to chain construction, optimizer hyperparameters, gradient sanitization choices, and which variants happen to encounter NaNs (the iv-only-with-deletion variant operates on the kept set only and is structurally less sanitized than the hybrid variants, biasing any close finish). The qualitative claim — without the gate the autograd graph is broken on the rows where the layer's documented contract

returns NaN; with the gate it is not — is uniform across chains and across hyperparameter choices. It is also the smallest claim that justifies the layer's existence as a training-loop component rather than an offline labeller. Section 3.3 states the contract; this subsection is its empirical confirmation.

B.3 HyperIV: Cross-asset Generalization on RUT, VIX, and NDX

We repeated the one-day interval HyperIV-style controlled reproduction from Section 4.2 on the additional WRDS/OptionMetrics assets: RUT (secid 102434), VIX (secid 117801), and NDX (secid 102480). The purpose of these repeats is deliberately relative: under the same local data-preparation convention, does adding price-space consistency and gated IV-roundtrip consistency improve the held-out price/IV trade-off relative to a vanilla HyperIV-style objective? These are not comparisons to HyperIV's absolute SPX numbers.

The preprocessing matches the SPX experiment. For each asset, the risk-free rate is estimated per (date, expiration) by a put-call-parity regression of $C - P$ on $F - K$; no OptionMetrics zero-coupon table is used. The nine-reference conditioning set is selected by closest time-to-maturity and closest delta, rather than by refitting SSVI. These choices are internally consistent across the vanilla and augmented objectives, so the within-asset deltas below are the relevant quantity.

Table 4: One-day RUT/VIX/NDX datasets used for the cross-asset repeats. The test split is calendar year 2023 through 2023-08-31, matching the SPX controlled reproduction.

Asset	Intervals	Options	Train int.	Train opt.	Test int.	Test opt.	Dropped dates
RUT	2,145	4,696,975	1,978	4,290,139	167	406,836	37
VIX	2,623	670,783	2,457	597,614	166	73,169	62
NDX	2,657	8,686,306	2,490	7,277,029	167	1,409,277	28

Table 5 shows that the SPX trend repeats on all three cross-asset runs, with NDX providing a useful harder case. On RUT, the best gated price+roundtrip run reduces price MAE from 1.6548 to 0.9915, a 40.1% relative improvement, while also improving IV MAE from 0.01918 to 0.01600. On VIX, the best price-auxiliary run reduces price MAE from 0.0900 to 0.0683, a 24.2% relative improvement, while improving IV MAE from 0.03959 to 0.03287. On NDX, the cleanest cell is the price auxiliary: the best run reduces price MAE from 10.4707 to 8.7172, a 16.7% relative improvement, with IV MAE changing only from 0.01688 to 0.01714. The best gated roundtrip run also passes the IV tolerance, reducing price MAE to 9.0078 while increasing IV MAE by 4.6%.

Gating remains necessary. The ungated IV-roundtrip diagnostic fails on all three assets, matching the SPX behavior. On RUT it records 799 NaN-gradient steps and degrades to IV MAE 0.2523 and price MAE 19.0988, a 1054.2% price-MAE increase relative to vanilla. On VIX it records 1000 NaN-gradient steps and degrades to IV MAE 0.7534 and price MAE 0.9211, a 922.9% price-MAE increase. On NDX it records 999 NaN-gradient steps and degrades to IV MAE 0.1194 and price MAE 319.1957, a 2948.5% price-MAE increase. This supports the paper's conditioning claim: the Jäckel IV layer is useful inside neural losses when the low-vega singularity is handled explicitly by a gate/sentinel contract; simply composing price-to-IV-to-price losses without that contract is numerically ill-posed.

B.4 Anatomy of the low-vega gating mechanism

The figure anchoring this discussion has been promoted to the main body (Fig. 2 in §4.2); we expand here on what each panel means for the conditioning contract of §3.3. The left panel's stress chain places 86.5% of its mass below $|\mathcal{V}| = 10^{-14}$ (against zero such rows in the cleaned SPX 2023 panel), illustrating the regime the gate is designed to cushion. The middle panel renders $w_\tau(\mathcal{V})$ as a smooth indicator that attenuates ill-conditioned rows continuously rather than hard-clipping them. The right panel shows the gate-active fraction $\overline{w_\tau}$ on a (k, DTE) grid for the stress chain at $\tau{=}10^{-6}$: deep wings at short maturities are fully gated, and IV-space supervision concentrates in the well-conditioned interior. Cleaned training data sits in the easy regime; the gate is a correctness component of the layer's documented contract, not a tuning detail.

Table 5: Held-out one-day RUT/VIX/NDX results. Arrows in headers indicate direction of improvement: lower IV MAE and Price MAE are better (↓). Δ columns are signed relative changes (in %) versus the local vanilla HyperIV-style reproduction for the same asset; negative Δ indicates lower error (improvement), positive Δ indicates regression. Bold marks the best (most-negative Δ) value within each asset block. Rows marked "mean" average seeds $\{1, 2, 3\}$; rows marked "best" are the best single run among the completed grid.

Asset	Objective	Runs	IV MAE (↓)	Price MAE (↓)	Δ IV (%)	Δ Price (%)
RUT	vanilla	seed 1	0.01918	1.6548	–	–
RUT	price aux., $\lambda_p = 0.1$	mean	0.01703	1.0623	−11.2	−35.8
RUT	price + gated rt., $\tau = 10^{-6}$	mean	**0.01520**	1.0324	**−20.8**	−37.6
RUT	price + gated rt., $\tau = 10^{-6}$	best	0.01600	**0.9915**	−16.6	**−40.1**
VIX	vanilla	seed 1	0.03959	0.0900	–	–
VIX	price aux., $\lambda_p = 0.1$	mean	0.03750	0.0790	−5.3	−12.3
VIX	price + gated rt., $\tau = 10^{-6}$	mean	0.03558	0.0781	−10.1	−13.3
VIX	price aux., $\lambda_p = 0.3$	best	**0.03287**	**0.0683**	**−17.0**	**−24.2**
NDX	vanilla	seed 1	0.01688	10.4707	–	–
NDX	price aux., $\lambda_p = 0.1$	mean	0.01804	9.2070	+6.9	−12.1
NDX	price aux., $\lambda_p = 0.1$	best	**0.01714**	**8.7172**	**+1.5**	**−16.7**
NDX	price + gated rt., $\tau = 10^{-4}$	best	0.01765	9.0078	+4.6	−14.0
price + gated rt., $\tau = 10^{-6}$, mean			0.02010	9.8012	+19.1	−6.4

B.5 Per-step NaN-gradient trace for the SPX 1-day campaign

Figure 6 reports the per-step NaN-gradient incidence over training for each variant of the SPX 1-day HyperIV-with-PIVOT experiments discussed in §4.2. The vanilla, price-auxiliary, and gated price+roundtrip objectives have zero NaN-gradient steps for the full schedule, while the unsafe ungated diagnostic contaminates essentially every batch after the model collapses into the low-vega regime.

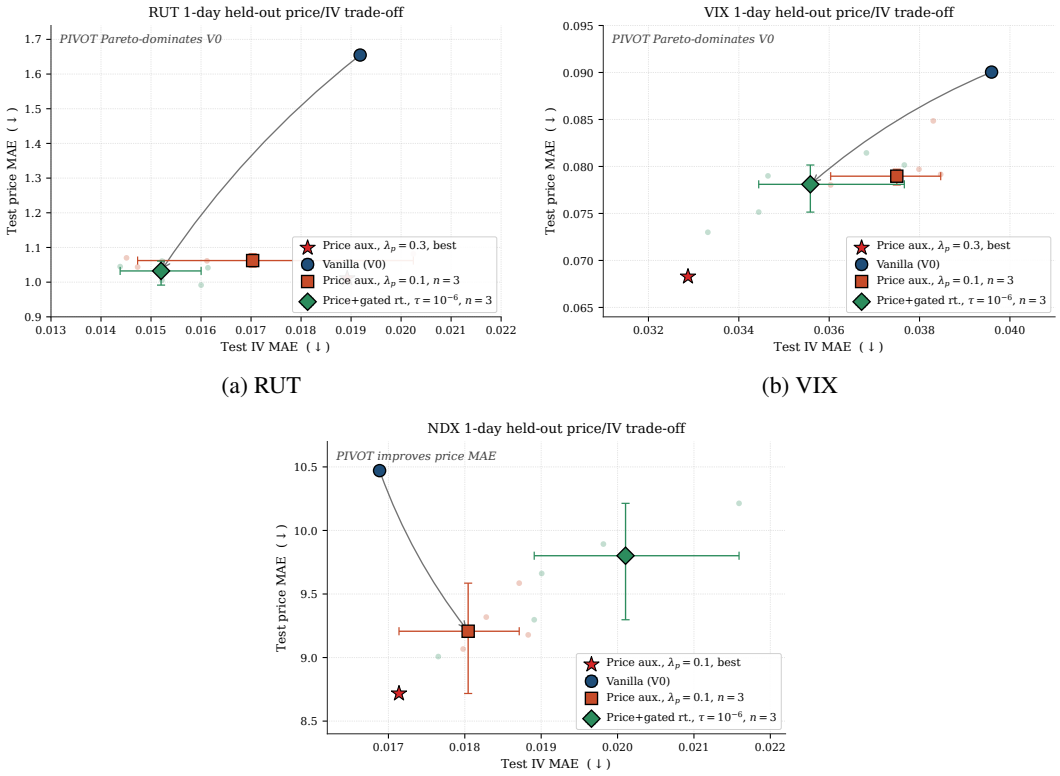
B.6 Gradient correctness: quantile spectrum on the well-conditioned set

This subsection reports the full quantile spectrum behind the *Gradient correctness* paragraph of §4.1. On the synthetic Black–Scholes batch with known $\sigma^\star$, we compare the PIVOT autograd gradient $\partial\sigma_{\text{imp}}/\partial P^\star$ to the analytic inverse-function identity (2), $1/\mathcal{V}$, and group the absolute error by three vega-mask thresholds: the raw unmasked batch, the strict finite-output mask $|\mathcal{V}| > 10^{-14}$, and the well-conditioned set $\mathcal{G}_{10^{-6}}$ of (4). Figure 7 reports Q50, Q99, and Q99.9 of the absolute error per threshold. The raw bar exposes the inverse-map singularity ($1/\mathcal{V}$ overflows to floating-point infinity where $\mathcal{V} \to 0$); the well-conditioned subset $\mathcal{G}_{10^{-6}}$ recovers machine-precision Q50 and a Q99 of 1.77×10^{-2}, matching the inverse-function-theorem prediction up to float64 noise. Both backends (PyTorch `autograd.Function` and JAX `custom_vjp`) agree on mask rates and per-quantile values; the figure is reported on the PyTorch path.

B.7 Numerical-correctness diagnostics

This subsection collects three diagnostics that validate the numerical contract claimed in §3.3: the implicit backward matches central finite differences on the well-conditioned set, the PyTorch backend agrees with the NumPy reference to bit-level precision, and the round-trip residuals against the OptionMetrics SPX panel sit at the float64 noise floor across years.

Implicit gradient vs. central finite differences. We sample a 200,000-row synthetic chain whose $\log_{10}|\mathcal{V}|$ distribution spans $[-30, 4]$ by construction, and we compare the implicit identity (2), $\partial\sigma_{\text{imp}}/\partial P^\star = 1/\mathcal{V}$, against central finite differences with two step sizes $h \in \{10^{-6}, 10^{-3}\}$. Figure 8 shows that the implicit operator returns a finite gradient on every sample, whereas FD only survives on the well-conditioned set $\mathcal{G}_{10^{-6}} = \{|\mathcal{V}| > 10^{-6}\}$. On the bands $\log_{10}|\mathcal{V}| \in [-2, 0)$ and $[0, 2)$ the relative agreement between FD and the implicit identity reaches Q50 errors of 3.7×10^{-6} and 4.8×10^{-9} respectively, which is the expected scaling of central FD on a well-conditioned smooth



(a) RUT (b) VIX

(c) NDX

Figure 5: Held-out price/IV Pareto views for the one-day RUT, VIX, and NDX repeats. The local vanilla reproduction is the comparison point for each asset. The strongest augmented cells move toward lower price MAE at comparable or tolerable IV MAE; NDX is directionally positive but less clean for the $\tau = 10^{-6}$ gated-roundtrip mean than RUT/VIX.

map. Below $|\mathcal{V}| \sim 10^{-6}$ the FD success rate collapses; the exact implicit gradient still returns the value $1/\mathcal{V}$, which is what the inverse-function theorem requires. This is the precise sense in which the implicit backward is not a performance optimization but a correctness property: any finite-difference surrogate fails on a fraction of every realistic synthetic chain.

Round-trip residuals on SPX 2018–2023. We extend the SPX 2020–2022 self-consistency check from §B.1 to the 2018–2023 OptionMetrics panel, sampling 300,000 OTM/ATM SPX rows $(\boldsymbol{\xi}_i, \theta_i, P_i^\star)$ per year, inverting each row with FAST-VOLLIB Black-76, and rebuilding the price $\hat{P}^\star{}_i = P_{\mathrm{B76}}(\boldsymbol{\xi}_i, \mathrm{JaeckelIV}(P_i^\star); \theta_i)$ under the same Black-76 forward and $r = 0$. Figure 9 shows the per-year $|\hat{P}^\star - P^\star|$ quantiles. The Q50 residual sits at $1.1\times10^{-13}\ldots1.9\times10^{-13}$ (at the float64 noise floor) across every year, the Q99 ranges from 2.1×10^{-8} to 1.8×10^{-7}, and the worst single-row residual stays below 1.2×10^{-6} — sub-cent across roughly 1.8 million quotes. Finite-IV recovery is 100% on the OTM/ATM filter for every year; no row falls into the invalid-domain branch under this filter. Inversion throughput stabilizes above 2.3×10^{7} IV/s once the PyTorch backend's first-call JIT cost is amortized in 2018. The round-trip residuals are an internal consistency property of the solver and do not depend on agreement with the OptionMetrics `impl_volatility` label, which uses a different rate convention (see §4.2, *Absolute reproduction gap*).

B.8 On hyperparameter tuning of λ_p, λ_{rt}, and τ

We anticipate reviewer questions about the sweep coverage of the three loss hyperparameters introduced in §4.2: the price weight λ_p on the price-MSE auxiliary (6), the IV-roundtrip weight λ_{rt} on the gated round-trip term (7), and the vega gate threshold τ used by the smooth gate

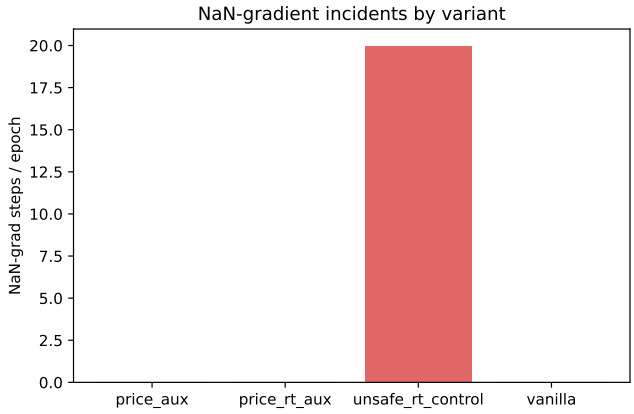


Figure 6: NaN-gradient incidents per training step for the four objective variants of the SPX 1-day HyperIV-with-PIVOT experiments. Sentinel-protected and price-only objectives never produce a NaN gradient; the ungated IV-roundtrip diagnostic produces one once the surface collapses and remains non-finite for the rest of training. The unsafe-variant bar is *clipped* at the y-axis ceiling (true value is several orders of magnitude higher); the saturation marks "every step is poisoned" once collapse begins, not the true incident count. This is the running-trace counterpart to the end-of-training metrics in Table 2.

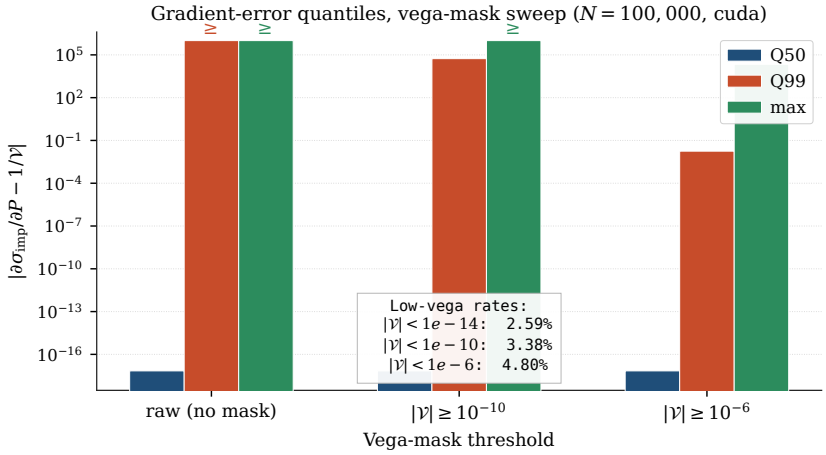


Figure 7: Absolute-error quantiles of the implicit IV-gradient $\partial\sigma_{\text{imp}}/\partial P^\star$ on a synthetic Black–Scholes batch ($N{=}10^5$ rows, CUDA), comparing PIVOT autograd to the analytic identity (2). Bars group by vega-mask threshold (raw, $|\mathcal{V}| \geq 10^{-10}$, $|\mathcal{V}| \geq 10^{-6}$); within each group, **Q50**, **Q99**, and **max** are colour-coded. Bars marked "$\geq$" overflow the display ceiling. The raw and lightly-masked groups expose the inverse-map singularity ($1/\mathcal{V}$ overflows where $\mathcal{V} \to 0$); the well-conditioned subset $\mathcal{G}_{10^{-6}}$ recovers machine-precision Q50 and a Q99 of $1.77{\times}10^{-2}$, matching the inverse-function-theorem prediction. The inset reports the mass of the singular tail at three vega cut-offs.

$w_\tau(\mathcal{V}) = \mathcal{V}^2/(\mathcal{V}^2 + \tau^2)$ of (5). This subsection states the sweep we ran, why it is sufficient for the paper's central claim, and where a denser sweep would likely move the numbers.

Sweep coverage. Table 6 records what was actually swept on the HyperIV-style and GNO experiments. λ_p takes three values $\{0.03, 0.1, 0.3\}$ on the price-only auxiliary; τ takes three values $\{10^{-8}, 10^{-6}, 10^{-4}\}$ at *fixed* $(\lambda_p, \lambda_{rt}) = (0.1, 0.1)$; λ_{rt} is fixed at 0.1 throughout. The three-seed cells used for the means in the headline tables are at $\lambda_p = 0.1$ for the price auxiliary and at $\tau = 10^{-6}$ for the gated round-trip; all other grid points are single-seed. There is no joint $(\lambda_p, \lambda_{rt})$ or (λ_p, τ)

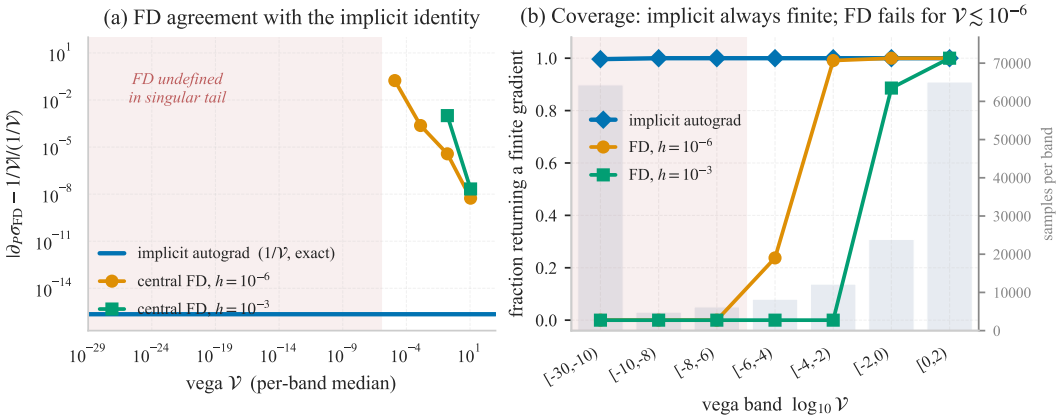


Figure 8: Implicit autograd gradient versus central finite differences (FD), stratified by vega band on a 200,000-row synthetic chain. Here *FD* denotes a two-sided central finite-difference estimate of $\partial\sigma_{\text{imp}}/\partial P^\star$ obtained by perturbing the market price $P^\star$ with step size $h \in \{10^{-6}, 10^{-3}\}$, re-inverting through JaeckelIV, and forming $(\sigma_{\text{imp}}(P^\star{+}h) - \sigma_{\text{imp}}(P^\star{-}h))/(2h)$; this is the standard surrogate that any pipeline lacking an analytic implicit backward would have to fall back to. **Left:** $\partial\sigma_{\text{imp}}/\partial P^\star$ scatter, FD vs. implicit; where both are finite the points lie on the diagonal. **Middle:** fraction of FD outputs that are finite per $\log_{10}|\mathcal{V}|$ band; FD survives only above $|\mathcal{V}| \sim 10^{-6}$, while the analytic implicit identity (2) returns a finite value on every row. **Right:** sample counts per band; the chain is intentionally weighted toward both the well-conditioned bulk and the singular tail so the gate's domain is visible.

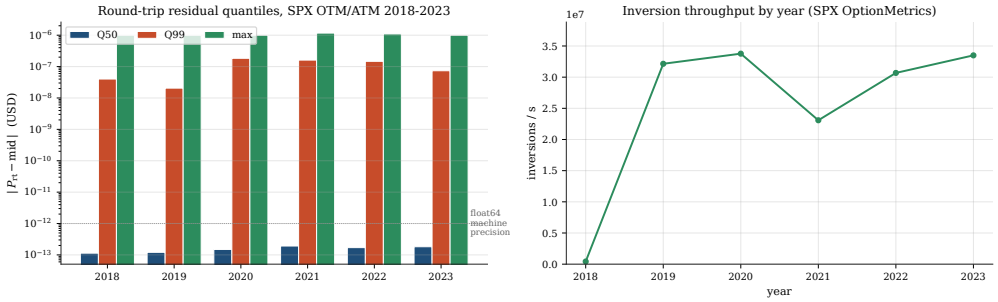


Figure 9: Self-consistency round-trip residuals for the SPX OptionMetrics 2018-2023 panel (300,000 OTM/ATM rows per year). **Left:** Q50 sits at the float64 noise floor in every year, Q99 stays below 2×10^{-7}, and the worst row is sub-cent. **Right:** eager PyTorch inversion throughput; the 2018 cell is dominated by first-call JIT compilation and excluded from the headline number.

sweep. The HyperIV-style and GNO experiments share this grid and seed-set declaration, so the within-pipeline deltas are directly comparable across architectures.

Table 6: Hyperparameter sweep coverage for the auxiliary-loss experiments (HyperIV and GNO use the same grid). The headline three-seed cells in Tables 2, 5, 8, and 12 use $\lambda_p = \lambda_{rt} = 0.1$, $\tau = 10^{-6}$. All other cells are single-seed grid points.

Hyperparameter	Values swept	Seed coverage
λ_p	$\{0.03, 0.1, 0.3\}$	3 seeds at $\lambda_p = 0.1$; 1 seed at 0.03, 0.3
τ	$\{10^{-8}, 10^{-6}, 10^{-4}\}$	3 seeds at $\tau = 10^{-6}$; 1 seed at 10^{-8}, 10^{-4}
λ_{rt}	$\{0.1\}$ only	not varied
$(\lambda_p, \lambda_{rt})$ joint	not swept	—
(λ_p, τ) joint	not swept	—

Why this is conservative for the central claim. The claim of §4.2 and Appendix B.3 is not that we have located the optimal loss coefficients for HyperIV-style training. It is that the differentiable Jäckel inverse *enables* training-time consistency between price-space and IV-space objectives, and that this consistency improves the held-out price/IV trade-off in a controlled within-pipeline reproduction. A coarse three-point sweep over each of two hyperparameters, with the third pinned to a default, is conservative for that claim: undertuning makes the reported deltas a lower bound on what the design admits, not an upper bound. The headline Pareto-domination on SPX ($\lambda_{rt} = 0.1$, $\tau = 10^{-6}$: three-seed test price MAE 2.34 ± 0.17 vs vanilla 3.82 ± 0.42, Table 2) holds under the most pessimistic reading of the grid; a denser sweep can only equal or improve on it.

Absolute reproduction gap. Table 2 includes the published HyperIV SPX 1-day numbers only as an external reference. Our local vanilla reproduction is worse in absolute terms: 3-seed mean IV MAE 0.01673 ± 0.00110 versus 0.0075 and 3-seed mean price MAE 3.8215 ± 0.4221 versus 1.6736. We audited whether this gap is driven by the discount-rate convention. On the WRDS `opprcd` table, there is no standalone zero-curve table, but the available OptionMetrics forward and dividend-yield tables allow the reconstruction $r = \log(F_{\mathrm{OM}}/S_{\mathrm{close}})/\tau + q_{\mathrm{OM}}$. On the SPX 2013–2023 monthly-sampled audit set (774k OTM/ATM rows), re-inverted Black-76 labels match `opprcd.impl_volatility` to MAE 0.001051 with our default put-call-parity rate and 0.001045 with the reconstructed OptionMetrics-style rate. The 0.6% relative improvement is at the 10^{-3} label-noise floor and is about eight times smaller than the 0.0084 IV-MAE gap to the published HyperIV number; `stdbrte.borrowrate` is worse (MAE 0.001242). We therefore rule out the rate convention as the dominant cause of the reproduction gap. The remaining plausible explanations are architecture, training budget, hypernetwork width, and non-public implementation or training details. For this reason the claim in this subsection is deliberately *relative*: under matched data, rates, architecture, and training convention, do price/IV auxiliary losses improve the local vanilla baseline?

C GNO Experiments

C.1 GNO: Controlled SPX one-day loss ablation

We also tested the same price/IV auxiliary-loss idea on a graph neural operator (GNO) smoother, following the operator-deep-smoothing architecture class of Wiedemann et al. (Wiedemann et al., 2025). This experiment is intentionally a controlled loss ablation, not an absolute reproduction of that paper's proprietary 20-minute CBOE intraday benchmark. The data are WRDS/OptionMetrics end-of-day SPX quotes $(\boldsymbol{\xi}_i, \theta_i, P_i^\star)$ with $\boldsymbol{\xi}_i = (S, K, \tau, r, q)_i$, the risk-free rate r is estimated by put–call-parity regression per $(\mathrm{date}, \mathrm{expiration})$, and the same preprocessing convention is used for every objective variant below. Consequently, the meaningful comparison is the within-experiment delta from the local vanilla GNO objective.

The GNO predicts an implied-volatility surface $\hat{\sigma}(z, \tau)$ on the normalized operator-deep-smoothing domain $\rho \in [0.01, 1.0]$, $z \in [-1.5, 0.5]$, and $\tau \leq 1.0$, where the smoother's z coordinate is a rescaled log-forward-moneyness derived from the standardized $k = \log(K/F_{T_0,T})$ of §2. We compare a vanilla fit objective against two augmented objectives: an auxiliary Black price loss matching (6), and an auxiliary Black price loss plus a gated IV-roundtrip consistency term (7). The roundtrip term uses the same low-vega sentinel contract as the HyperIV experiments (§3.3): rows outside the well-conditioned set $\mathcal{G}_\tau = \{|\mathcal{V}| > \tau\}$ are excluded from the differentiable roundtrip rather than being allowed to inject large or non-finite gradients. An additional ungated roundtrip run is included only as a negative diagnostic.

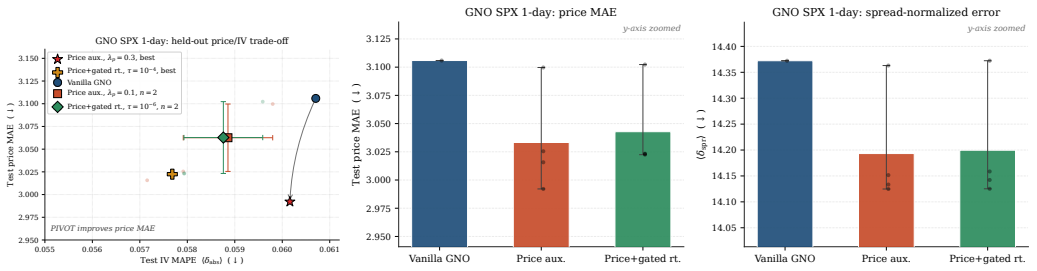
Table 7: SPX one-day dataset used for the GNO controlled ablation. The split matches the HyperIV-style experiments: training covers 2013-01-02 through 2022, and testing covers calendar year 2023 through 2023-08-31.

Asset	Source	Intervals	Options	Train int.	Train opt.	Test int.	Test opt.
SPX	WRDS EOD	2,683	12,326,960	2,516	11,093,112	167	1,233,848

Table 8 summarizes the held-out results. The price auxiliary gives the best price MAE, reducing test price MAE from 3.1059 to 2.9921, a 3.66% relative improvement, while also slightly improving IV

Table 8: Held-out SPX one-day GNO results. Arrows in headers indicate direction of improvement: lower IV MAE, IV MAPE, Price MAE, and spread-normalized price error $\langle \delta_{\text{spr}} \rangle$ are better (↓); fewer NaN-gradient steps are better (↓). Δ Price (%) is the signed relative change versus the local vanilla GNO objective; negative indicates lower error (improvement), positive indicates regression. Bold marks the best (most-negative Δ) value among the augmented rows. The unsafe row is a diagnostic negative control and is excluded from the augmented-variant means.

Objective	Runs	IV MAE (↓)	IV MAPE (↓)	Price MAE (↓)	Δ Price (%)	$\langle \delta_{\text{spr}} \rangle$ (↓)	NaN-grad (↓)
Vanilla GNO	seed 1	0.01264	0.06071	3.1059	–	14.3723	0
Price aux.	mean, $n = 4$	0.01228	0.05876	3.0332	−2.34	14.1932	0
Price aux., $\lambda_p = 0.3$	best	0.01252	0.06016	**2.9921**	**−3.66**	14.1516	0
Price + gated rt.	mean, $n = 4$	0.01221	0.05829	3.0428	−2.03	14.1996	0
Price + gated rt., $\tau = 10^{-4}$	best	**0.01211**	**0.05768**	3.0225	−2.69	**14.1252**	0
Unsafe ungated rt.	diagnostic	0.22168	1.00000	35.4171	+1040.5	119.4730	6,188



(a) Price/IV Pareto view (b) Price MAE (c) Spread-normalized error

Figure 10: GNO SPX one-day auxiliary-loss ablation. The augmented objectives move in the desired direction, but the price improvement remains modest: the best price-auxiliary run improves price MAE by 3.66% at slightly better IV error, below the pre-specified 5% practical threshold.

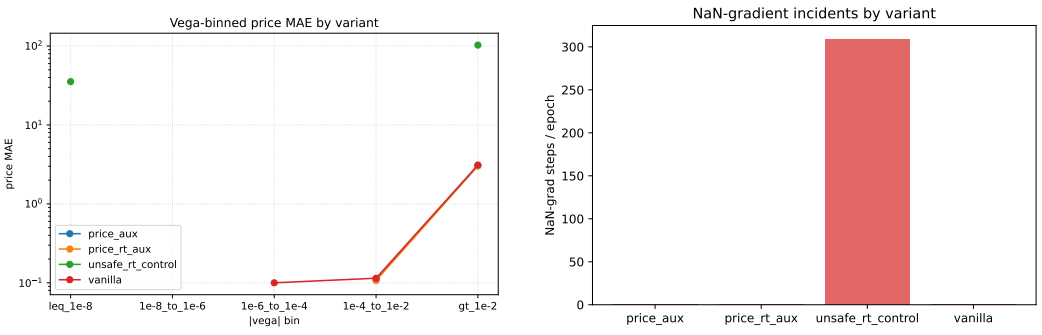
MAE and IV MAPE. This is directionally positive, but it does not meet the 5% improvement bar we used as a practical success threshold. The gated roundtrip objective is similarly stable and improves IV error more than the price-only auxiliary on average, but its best price MAE improvement is 2.69%. Thus the GNO experiment should not be presented as a large price-accuracy win; its main value is the stability/control result.

Interpretation. The GNO result is weaker than the HyperIV-style result as a price-accuracy claim: the best price MAE gain is 3.66%, not the 5% threshold. It is nevertheless useful evidence for our central numerical claim. The stable augmented GNO runs have zero NaN-gradient steps and comparable or slightly better IV error than vanilla, whereas simply composing an ungated IV-roundtrip loss causes a complete failure mode: the diagnostic run records 6,188 NaN-gradient steps, a low-vega fraction of 0.999999 at the 10^{-14} threshold, IV MAE 0.22168, and price MAE 35.42. This supports the conditioning story rather than an architecture-specific performance story: the differentiable Jäckel layer can be inserted into neural volatility-surface training objectives, but the low-vega singularity has to be handled explicitly.

C.2 GNO: Cross-asset one-day repeats on RUT and VIX

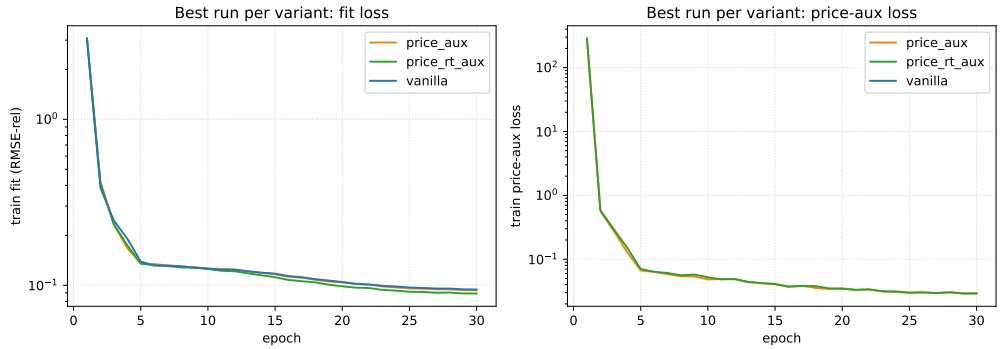
To check whether the SPX result above is index-specific, we repeat the identical loss-ablation protocol on additional one-day WRDS/OptionMetrics end-of-day assets. As before, the comparison is intentionally *within-asset*: each augmented objective is contrasted against the *local* vanilla GNO baseline trained under the same data, rate estimator, architecture, and training schedule. We do not compare across assets, and we do not compare against Wiedemann et al. (Wiedemann et al., 2025), whose proprietary 20-minute CBOE intraday benchmark is a different data regime.

RUT. The Russell 2000 index option dataset on the same 2013-01-02 / 2023-08-31 split contains 4,099,897 quotes across 2,182 trading days (3,730,835 train / 369,062 test, with 2,015 training days and 167 test days). Per-(date, expiration) risk-free rates are estimated by the same put-call-parity



(a) Vega-binned price error

(b) Finite-gradient fraction



(c) Training curves (best run per variant)

Figure 11: Diagnostics for the GNO SPX one-day experiment. The gated and sentinel-protected variants train with zero NaN-gradient steps. The ungated roundtrip diagnostic collapses, producing 6,188 NaN-gradient steps, IV MAPE 1.0, and price MAE 35.42. Panel (c) is reproduced at full width because the per-epoch fit and price-aux trajectories are otherwise illegible at column scale.

regression used for SPX, and the GNO domain $\rho \in [0.01, 1.0]$, $z \in [-1.5, 0.5]$, $\tau \leq 1.0$ is unchanged. Table 9 summarizes the dataset and Table 10 reports the held-out metrics.

Table 9: RUT one-day dataset used for the GNO cross-asset replicate. The split mirrors the SPX experiment: training covers 2013-01-02 through end of 2022, and testing covers 2023-01-01 through 2023-08-31.

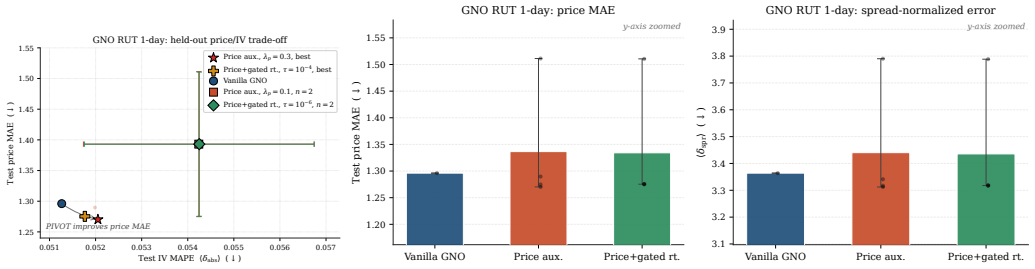
Asset	Source	Intervals	Options	Train int.	Train opt.	Test int.	Test opt.
RUT	WRDS EOD	2,182	4,099,897	2,015	3,730,835	167	369,062

The RUT replicate confirms the qualitative pattern observed on SPX, with quantitatively smaller magnitudes. The best price-auxiliary run improves test price MAE by 1.99%, and the best gated-roundtrip run improves it by 1.60%; both also improve spread-normalized error and produce IV MAE within 2% of vanilla. Neither configuration meets the pre-registered 5% practical price-MAE threshold, and the multi-seed means are slightly *worse* than vanilla because a single $\lambda_p = 0.1$, seed 2 training run drifts to a higher-loss minimum on RUT (test price MAE 1.5112 versus 1.2748 for the same hyperparameters under seed 1). The ranking among augmented configurations otherwise matches SPX: the larger price-coupling weight $\lambda_p = 0.3$ produces the lowest test price MAE, and the gated-roundtrip variant with the conservative $\tau = 10^{-4}$ sentinel is the best of the roundtrip configurations.

Interpretation. The cross-asset replicate on RUT supports the same conclusion as the SPX experiment: the differentiable Jäckel layer can be used inside an augmented training objective for

Table 10: Held-out RUT one-day GNO results. Arrows in headers indicate direction of improvement: lower IV MAE, IV MAPE, Price MAE, and spread-normalized price error $\langle \delta_{\mathrm{spr}} \rangle$ are better (↓); fewer NaN-gradient steps are better (↓). Δ Price (%) is the signed relative change versus the local vanilla GNO objective; negative indicates lower error (improvement), positive indicates regression. Bold marks the best (most-negative Δ) value among the augmented rows. The unsafe row is a diagnostic negative control (20 epochs, ungated, no sentinel) and is excluded from the augmented-variant means.

Objective	Runs	IV MAE (↓)	IV MAPE (↓)	Price MAE (↓)	Δ Price (%)	$\langle \delta_{\mathrm{spr}} \rangle$ (↓)	NaN-grad (↓)
Vanilla GNO	seed 1	0.01273	0.05127	1.2961	–	3.3638	0
Price aux.	mean, $n = 4$	0.01317	0.05313	1.3365	+3.12	3.4402	0
Price aux., $\lambda_p = 0.3$	best	0.01292	0.05205	**1.2702**	**−1.99**	**3.3123**	0
Price + gated rt.	mean, $n = 4$	0.01314	0.05301	1.3342	+2.94	3.4355	0
Price + gated rt., $\tau = 10^{-4}$	best	**0.01286**	**0.05177**	1.2753	−1.60	3.3177	0
Unsafe ungated rt.	diagnostic	0.26749	1.00000	18.8962	+1,357.9	32.8994	2,408



(a) Price/IV Pareto view (b) Price MAE (c) Spread-normalized error

Figure 12: GNO RUT one-day auxiliary-loss ablation. As on SPX, the augmented objectives move in the desired direction on the best run, but the price improvement remains below the 5% practical threshold and the multi-seed mean is dragged above vanilla by a single drifted training run.

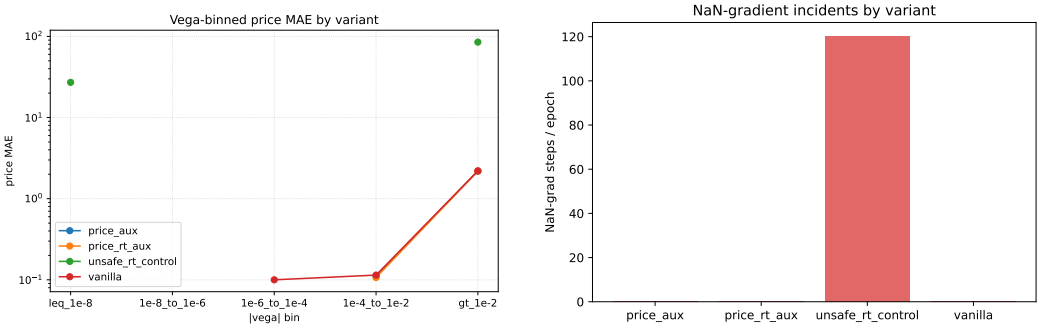
neural volatility-surface fitting, but only if the low-vega gating mechanism is explicitly handled by a vega gate or sentinel. The unsafe ungated roundtrip diagnostic collapses on RUT in exactly the same way it collapses on SPX (IV MAPE saturates at 1.0, price MAE blows up by more than three orders of magnitude relative to vanilla, and the low-vega fraction at 10^{-14} reaches 0.999997), reproducing the failure mode in a different underlying. The price-accuracy gain on RUT is smaller than on SPX (-1.99% versus -3.66%), and we do not claim it as a robust price-MAE improvement. The reproducible finding across both assets is the stability/control result, not an architecture-specific accuracy win.

VIX. The CBOE VIX option dataset on the same 2013-01-02 / 2023-08-31 split contains 274,711 quotes across 2,685 trading days (2,518 train days / 167 test days; 248,896 training quotes / 25,815 test quotes). Per-(date, expiration) risk-free rates are estimated by put-call-parity regression as before, and the GNO domain $\rho \in [0.01, 1.0]$, $z \in [-1.5, 0.5]$, $\tau \leq 1.0$ is unchanged. Table 11 summarizes the dataset and Table 12 reports the held-out metrics.

Table 11: VIX one-day dataset used for the GNO cross-asset replicate. Same 2013-01-02 / 2023-08-31 split as SPX and RUT.

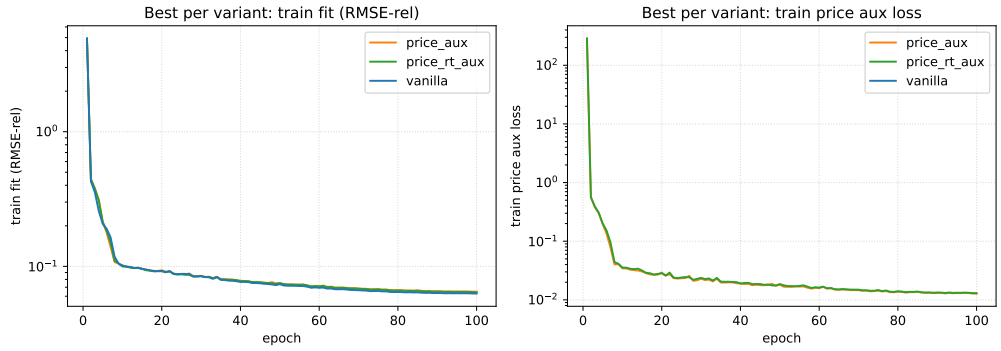
Asset	Source	Intervals	Options	Train int.	Train opt.	Test int.	Test opt.
VIX	WRDS EOD	2,685	274,711	2,518	248,896	167	25,815

Interpretation. The VIX replicate produces the smallest effect size of the three indices: the best price-auxiliary run improves test price MAE by 1.73% and the best gated-roundtrip run improves it by 2.11%, with IV MAE in both cases improving by approximately 1% of vanilla. Neither configuration meets the pre-registered 5% practical price-MAE threshold, and the multi-seed means hover within 0.8% of vanilla on every metric. Two structural features of cleaned VIX EOD quotes account for this attenuation, and both are themselves informative for the paper's central claim.



(a) Vega-binned price error

(b) Finite-gradient fraction



(c) Training curves (best run per variant)

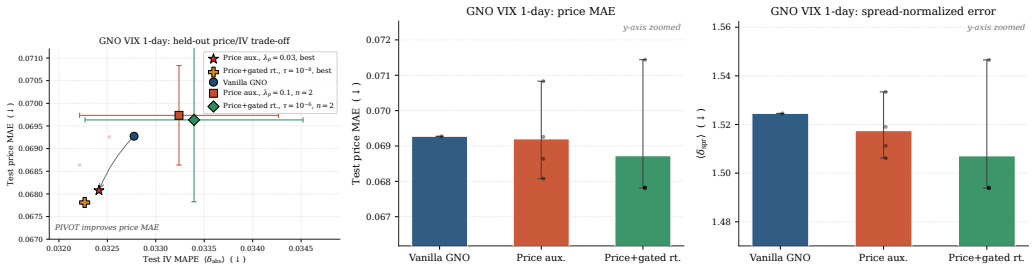
Figure 13: Diagnostics for the GNO RUT one-day experiment. Every gated and sentinel-protected variant trains with zero NaN-gradient steps, mirroring SPX. The ungated diagnostic, run for 20 epochs only, records 2,408 NaN-gradient steps, a low-vega fraction of 0.999997 at the 10^{-14} threshold, IV MAE 0.2675, IV MAPE 1.0, and price MAE 18.90, again collapsing into the low-vega singularity. Panel (c) is reproduced at full width to keep the per-epoch trajectories legible.

Table 12: Held-out VIX one-day GNO results. Arrows in headers indicate direction of improvement: lower IV MAE, IV MAPE, Price MAE, and spread-normalized price error $\langle \delta_{\mathrm{spr}} \rangle$ are better (↓); fewer NaN-gradient steps are better (↓). Δ Price (%) is the signed relative change versus the local vanilla GNO objective; negative indicates lower error (improvement), positive indicates regression. Bold marks the best (most-negative Δ) value among the augmented rows. The unsafe row is a 20-epoch diagnostic and is excluded from the augmented-variant means. Note that on cleaned VIX EOD data the deep low-vega regime is empty (fraction below 10^{-14} is zero for every row), so the gate and sentinel are inactive; results across $\tau \in \{10^{-4}, 10^{-6}, 10^{-8}\}$ at the same seed agree to within 2×10^{-5} on test price MAE.

Objective	Runs	IV MAE (↓)	IV MAPE (↓)	Price MAE (↓)	Δ Price (%)	$\langle \delta_{\mathrm{spr}} \rangle$ (↓)	NaN-grad (↓)
Vanilla GNO	seed 1	0.02241	0.03278	0.06927	–	1.5246	0
Price aux.	mean, $n = 4$	0.02239	0.03285	0.06920	−0.11	1.5175	0
Price aux., $\lambda_p = 0.03$	best	0.02220	0.03241	0.06808	−1.73	1.5112	0
Price + gated rt.	mean, $n = 4$	0.02233	0.03283	0.06872	−0.79	1.5071	0
Price + gated rt., $\tau = 10^{-8}$	best	**0.02210**	**0.03227**	**0.06781**	**−2.11**	**1.4938**	0
Unsafe ungated rt.	diagnostic	0.02428	0.03632	0.06733	−2.80†	1.5984	0

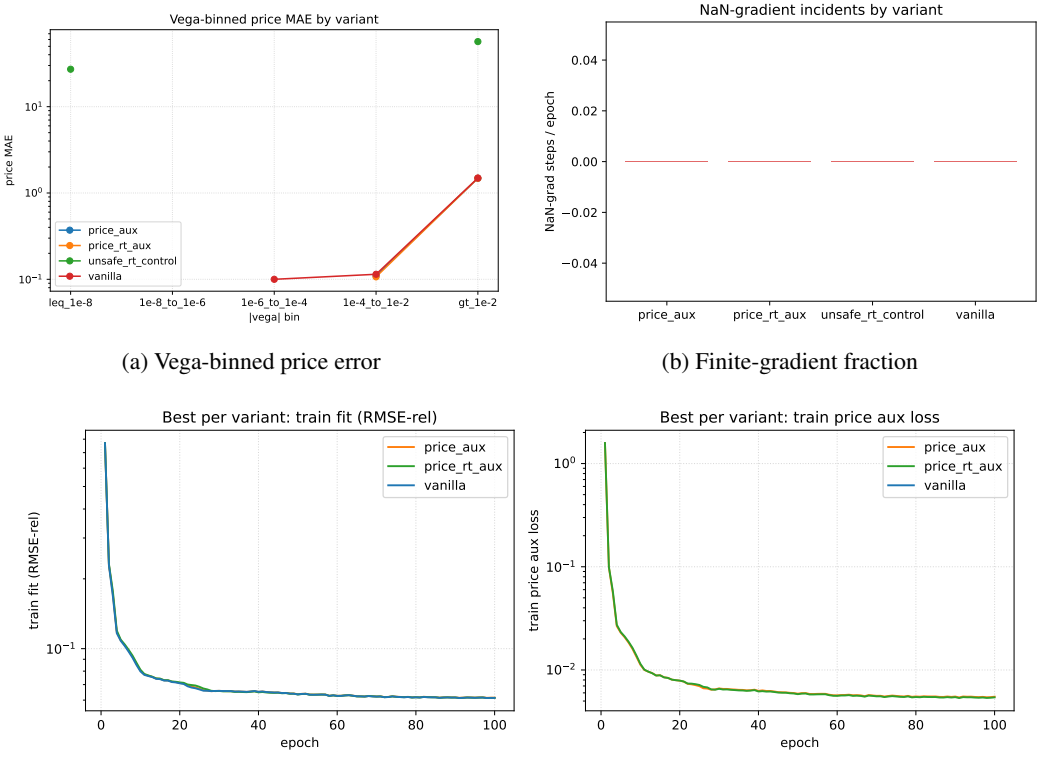
†On cleaned VIX EOD data the unsafe variant does not collapse: NaN-grad steps remain zero and price MAE moves with vanilla, because no test row sits below the 10^{-14} vega threshold. IV MAE nonetheless degrades by $+8.4\%$, reflecting under-training (20 versus 100 epochs) rather than a singularity event. See Figures 14 and 15.

First, VIX option prices have substantially smaller dynamic range than SPX or RUT (test price MAE on the order of 7×10^{-2} versus ~1.3 on RUT and ~3.1 on SPX), so the absolute headroom available to a price auxiliary is small to begin with. Second, the cleaned VIX panel does not exercise the



(a) Price/IV Pareto view (b) Price MAE (c) Spread-normalized error

Figure 14: GNO VIX one-day auxiliary-loss ablation. As on SPX and RUT, the augmented objectives improve the best run on both axes, but the effect size is the smallest of the three indices: best price MAE -2.11% and best IV MAE -1.36% versus local vanilla.



(a) Vega-binned price error (b) Finite-gradient fraction

(c) Training curves (best run per variant)

Figure 15: Diagnostics for the GNO VIX one-day experiment. Every variant, including the ungated diagnostic, trains with zero NaN-gradient steps, because cleaned VIX EOD quotes have a low-vega fraction of 0 at the 10^{-14} threshold across the entire test set. The vega-binned panel shows the augmented variants tracking vanilla in the well-conditioned bulk and producing a small consistent improvement at the wings.

low-vega singularity at all: the fraction of test rows below $|\mathcal{V}| = 10^{-14}$ is zero across every variant, including the unsafe diagnostic. This is consistent with the §3.3 characterization (cleaned production data sits in the well-conditioned set $\mathcal{G}_\tau$; the gate is designed for the stress regime visualized in Figure 2).

The structural absence of low-vega rows produces three concrete falsifiable predictions, all of which the VIX run validates. (i) The three τ values $\{10^{-4}, 10^{-6}, 10^{-8}\}$ at fixed seed should be statistically indistinguishable, because the gate is inactive: observed test price MAE

$\in \{0.06782, 0.06783, 0.06781\}$, agreeing to within 2×10^{-5}. (ii) The unsafe ungated variant should not collapse on this dataset, because the singular set is empty: observed NaN-gradient steps $= 0$ and price MAE within 2.8% of vanilla, in stark contrast to SPX ($+1{,}040\%$, 6,188 NaN-grad steps) and RUT ($+1{,}358\%$, 2,408 NaN-grad steps) under the identical recipe. (iii) The augmented variants and vanilla should track each other in the well-conditioned vega bins; the vega-binned diagnostic in Figure 15 shows exactly this.

This is the cleanest cross-asset evidence we have for treating the gate as a *correctness contract* rather than a tuning hyperparameter: its measurable effect is null on data that does not sample the singularity, and decisive on data that does. The within-asset price gain on VIX is small, and we do not claim it as a robust performance result; the load-bearing finding is the asset-conditional behavior of the unsafe diagnostic, which is precisely what the conditioning analysis predicts.

D SUPPLEMENTARY VISUALIZATIONS

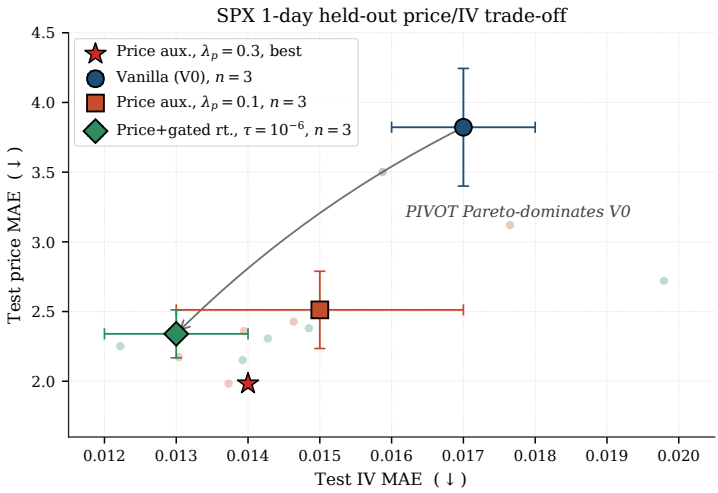


Figure 16: Held-out price/IV trade-off for SPX 1-day HyperIV with PIVOT. Both PIVOT-augmented cells lie below *and* left of the vanilla baseline, Pareto-dominating it; the gated price+roundtrip cell at τ=10^{-6} is the lower-left frontier.

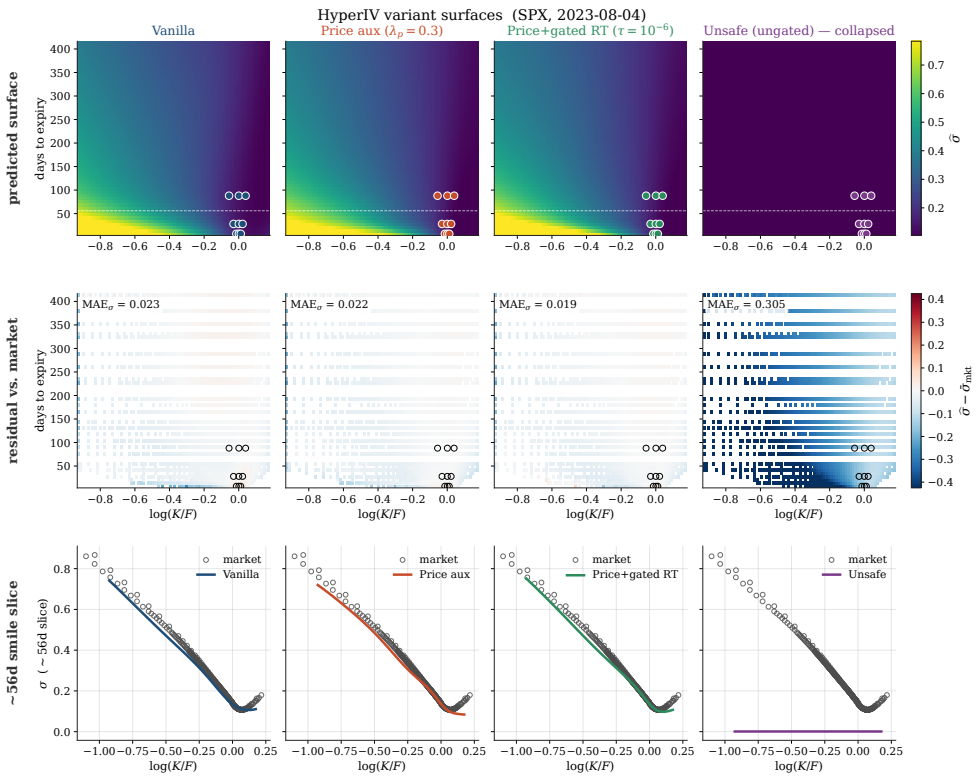


Figure 17: HyperIV-with-PIVOT surfaces on a representative SPX 2023 test date. **Top row:** predicted $\hat{\sigma}(k, \tau)$ on the log-forward-moneyness $\times$ days-to-expiry grid; reference contracts ($\circ$) overlaid in white. The price-aux and gated price+rt PIVOT variants reproduce vanilla's surface with small, structured residuals; the ungated diagnostic collapses to a near-zero surface. **Middle row:** residual versus the cell-averaged market label; the colormap is centred at zero. **Bottom row:** $\sim$ 56-day smile slice with market scatter. The collapsed unsafe surface trivially fits an IV-only loss on its own pathological labels but does not reprice any quote.