Special Purpose Vehicle (SPV) Pricing Using Physics-Informed Neural Networks (PINNs)

Chetan Jadhav
CFA, FRM

Agenda

- Understanding SPVs
- Development SPV pricing function
- Details about neural network to training stategy
- Results

What Is an SPV in the Private Stock Market?

- Multiple investors contribute capital into a single legal entity that in turn holds shares of a private company.
- Typically invests in one private company rather than multiple companies unlike a traditional fund. Cease to exist at the company’s liquidity event.
- Direct share transactions often need company approval but not for the SPV since the SPV is the shareholder on record and not the investor
- Pros: access, pooling, simple settlement process
- Cons: valuation complexity due to fees and opaque underlying stock price

What Makes SPV Valuation Challenging?

- GP’s carried interest (often around 20% of profits), resembles a call option on the profits.
- Absence of observable, market-based pricing prevents accurate assessment of how embedded optionality affects value over time.
- Buyers in secondary transactions must accept existing carry arrangements, which tends to reduce potential upside and makes pricing less transparent.
- Because carry is path-dependent, simple valuation heuristics often fail, and dynamic modeling is needed for reliable valuation.

Addressing Valuation Challenges

- Model carry explicitly as an embedded option in SPV and use the value of the option as discount to underly to get SPV valuation
- Use a stochastic exit model so uncertainty in of the liquidity event is built into embedded option valuation.
- Use Tape D® daily price estimates as source of price and volatility in the valuation model.
- Use physics-informed neural networks to solve this stochastic maturity option pricing model that learns to estimate the embedded option price for an SPV.

Why Using PINNs Helps with SPV Pricing?

Nasdaq Private Market

- When a differential equation describes how SPV value evolves, a PINN can learn that behavior without needing an explicit closed-form formula [1].
- Management fees and carry provisions can be built into the model via modified payoff or dividend terms, so the pricing reflects the real cashflows.
- Even if market data is sparse, PINNs can train effectively using simulated scenarios or partial observations to learn value dynamics.
- PINNs scales polynomially as complexity increases rather than exponentially [2].
- Alternatives like Monte Carlo simulation or PDE solvers may require much more computations, but once a PINN is trained it can deliver pricing results quickly.

SPV Liquidity Events Occur Randomly

- An SPV reaches its end when the underlying company has a liquidity event such as an IPO or acquisition, rather than at a fixed, predictable date.
- The time until that event is assumed to follow an exponential probability distribution, with density given by

$$f(t) = \lambda e^{-\lambda t}$$

- The rate parameter λ is chosen so that the expected waiting time for maturity equals $1/\lambda$ at any moment.
- Assuming λ=0.25, corresponds to 4 years to liquidity event on an average

Pricing the Embedded Short Call in an SPV

- For a small discrete time increment Δt, the short call price is the discounted expected value of the probability weighted next period payoff [3].

$$C_{SPV}(t) = \underbrace{e^{-r\Delta t}}_{\text{Discounting Factor}}\mathbb{E}_t[\lambda\underbrace{\alpha(S_{t+\Delta t} - K)^+}_{\text{Payoff at maturity}} + \underbrace{(1-\lambda)C_{SPV}(t+\Delta t)}_{\text{Continue}}]$$

$$r = Risk\ free\ rate$$
$$S = Stock\ price$$
$$\alpha = Carry$$
$$K = Cost\ basis\ for\ carry\ calculations$$
$$f = Management\ fee$$
$$(S_t - K)^+ = max(S_t - K, 0)$$

Expected Value Expression for the Embedded Short Call

- As $\Delta t \rightarrow 0$, the call price is given by [4],

$$C_{SPV}(S_0) = \int_0^\infty \underbrace{e^{-rt}}_{\text{Discounting Factor}} \quad \underbrace{\lambda e^{-\lambda t}}_{\text{Exponential PDF of maturity}} \quad \underbrace{\mathbb{E}[\alpha(S_t - K)^+]}_{\text{Expected Payoff at maturity}} dt \tag{1}$$

$$C_{SPV}(S_0) = \mathbb{E}\left[\int_0^\infty \lambda e^{-\lambda t} e^{-r} \ \alpha(S_t - K)^+ dt\right] \tag{2}$$

$$C_{SPV}(S_0) = \mathbb{E}[e^{-rT}\alpha(S_T - K)^+] \tag{3}$$

- S_0 is the stock price at time 0 and T is the random Maturity
- Eq 3, was used to generate training and testing data using Monte-Carlo

Time-independent Ordinary Differential Equitation

- The differential equation form given by [4]

$$\left(\frac{1}{2}\right)\sigma^2 S^2 \frac{\partial^2 C_{SPV}}{\partial S^2} + rS\,\frac{\partial C_{SPV}}{\partial S} - (r+\lambda)C_{SPV} + \lambda\,(S - K)^+ = 0 \tag{4}$$

- Incorporating fee (f), as dividend, and carry (α), as partial payoff factor,

$$\left(\frac{1}{2}\right)\sigma^2 S^2 \frac{\partial^2 C_{SPV}}{\partial S^2} + (r - f)S\,\frac{\partial C_{SPV}}{\partial S} - (r+\lambda)C_{SPV} + \alpha\lambda\,(S - K)^+ = 0 \tag{5}$$

- Feynman-Kac method can also be used with Eq 2 to get Eq 5
- Solution for C_{SPV} can be used to estimate the price of SPV, $S_{SPV} = S - C_{SPV}$

Normalizing Price and Strike for SPV Value

- Because S and K vary across large scales relative to other variables, using the raw values can be difficult for neural network to learn.
- Normalized variable, moneyness, *s=S/K*, was introduced in Eq 5

$$c_{spv} = {}^{C_{SPV}}\!/_{K} \text{ and } C_{SPV} = K c_{spv}$$

$$\frac{\partial C_{SPV}}{\partial S} = K \frac{\partial c_{spv}}{\partial S} = K \frac{\partial c_{spv}}{\partial s} \frac{1}{K} = \frac{\partial c_{spv}}{\partial s}$$

$$\frac{\partial^2 C_{SPV}}{\partial S^2} = \frac{\partial^2 c_{spv}}{\partial s^2} \frac{1}{K}$$

Normalizing Price and Strike for SPV Value

- Substituting $Kc_{spv}, \frac{\partial c_{spv}}{\partial s}$ and $\frac{\partial^2 c_{spv}}{\partial s^2}\frac{1}{K}$ in Eq 5

$$\left(\frac{1}{2}\right)\sigma^2 K^2 s^2 \frac{\partial^2 c_{spv}}{\partial s^2}\frac{1}{K} + (r - f)Ks\frac{\partial c_{spv}}{\partial s} - (r + \lambda)Kc_{spv} + \alpha\lambda K(s - 1)^+ = 0 \qquad (6)$$

$$\left(\frac{1}{2}\right)\sigma^2 s^2 \frac{\partial^2 c_{spv}}{\partial s^2} + (r - f)s\frac{\partial c_{SPV}}{\partial s} - (r + \lambda)c_{spv} + \alpha\lambda(s - 1)^+ = 0 \qquad (7)$$

- Eq 7 was used in the loss function of the neural networks as ODE residual loss

Boundary Conditions: SPV Value as Stock Price → 0 or ∞

- When $S \to 0$, $s \to 0$ and SPV is expected to be worthless, hence $C_{SPV} = 0$
- When $S \to \infty$, $(S - K)^+ \to \left(Se^{r-f} - K\right)$ and Eq 2 becomes

$$C_{SPV}(S_0) = \int_0^\infty \lambda e^{-(r+\lambda)t}\,\alpha\left(Se^{(r-f)t} - K\right)dt \tag{8}$$

- After integrating

$$C_{SPV} = \alpha\lambda\left(\frac{1}{f+\lambda}S - \frac{1}{r+\lambda}K\right)$$

- In normalized form

$$c_{spv} = \alpha\lambda\left(\frac{1}{f+\lambda}s - \frac{1}{r+\lambda}\right) \tag{9}$$

Smoothing the Payoff in the Loss Function

- The payoff function $(s - 1)^+$ is not smooth at *s = 1* or *S = K*, which causes difficulties for neural networks during training.
- $(s - 1)^+$ replaced with a smooth Softplus function that approximates payoff structure without discontinuity

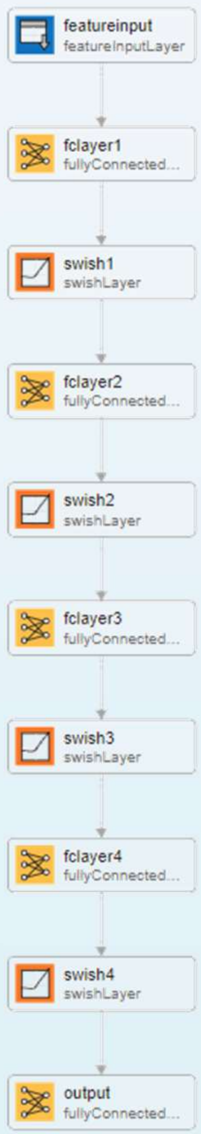
$$\left(\frac{1}{2}\right)\sigma^2 s^2 \frac{\partial^2 c_{spv}}{\partial s^2} + (r - f)s\frac{\partial c_{SPV}}{\partial s} - (r + \lambda)c_{spv} + \alpha\lambda\left(\frac{1}{k} ln\left(1 + e^{k(s-1)}\right)\right) = 0 \quad (10)$$

- Sharpness parameter *k = 37*

Neural Network Specification and Parameter Summary

- The network uses four fully connected layers with Swish activations, together with input and output layers, making ten layers in total.
- Weights are initialized using Xavier initialization and biases are set to zero which helps ensure gradient stays stable during training [6]
- Swish activation is chosen because smooth activation functions often produce better training convergence and performance for PINNs [5]

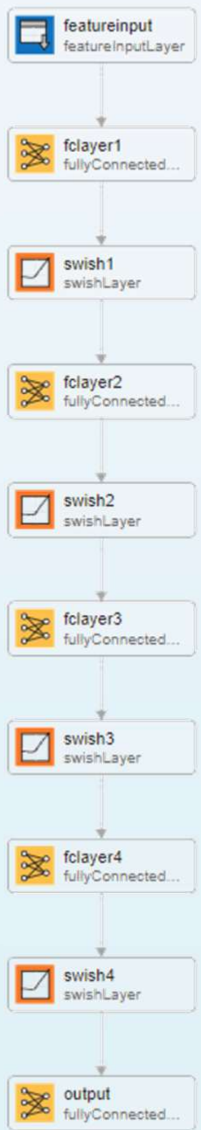
$$swish(x) = \frac{x}{(1 + e^{x})}$$



NPM
Nasdaq Private Market

Neural Network Specification and Parameter Summary

Layer	Name	Type	Activations	Weights	Bias	Number of Learnables
1	featureinput	Feature Input	5 × 1			
2	fclayer1	Fully Connected	128 × 1	128 × 5	128 × 1	768
3	swish1	Swish	128 × 1			
4	fclayer2	Fully Connected	128 × 1	128 × 128	128 × 1	16512
5	swish2	Swish	128 × 1			
6	fclayer3	Fully Connected	128 × 1	128 × 128	128 × 1	16512
7	swish3	Swish	128 × 1			
8	fclayer4	Fully Connected	128 × 1	128 × 128	128 × 1	16512
9	swish4	Swish	128 × 1			
10	output	Fully Connected	1 × 1	1 × 128	1 × 1	129
					total	50,433



Generating Training Data for the Model

- Real SPV trades are limited and noisy, so they are only used to choose realistic ranges for parameters and to validate out-of-sample performance
- One million samples are drawn across those ranges to train the model
- Sampled moneyness s from Gamma(2,1) to obtain more points near *s=1*, near the discontinuity and where errors tend to be largest.
- Using the Gamma(2, 1) for sampling s helps stabilize learning in non-smooth regions of the payoff.
- Monte Carlo simulation is used to estimate call prices for the training set

Training Strategy and Hyperparameter Setup

- The training uses the Adam optimizer over 200 epochs to update model parameters adaptively
- Trained in mini-batches of size 1,000, and with random shuffling after every epoch
- Loss function includes Boundary condition losses (Eq 8 & 9), the PDE residual loss (Eq 10) and MSE against MC calculated call price
- Gradient clipping used to prevent exploding gradients, and a learning rate schedule to help convergence

PINN vs Data-Driven Models: Comparative Performance

- Only 65 good trades available for the comparative analysis
- Regression Learner App was used to train 28 model to compare against
- Ensemble model with 30 learners, minimum leaf size of 8, and learning rate of 0.1 had the least RMSE and was selected for comparison

	PINN	Ensemble
MAE	11.46	4.82
RMSE	19.80	6.61
R-Squared	0.83	0.82

- Apparent better performance for Ensemble compared to PINN with trade data
- Bias, Actual > PINN in data and confirmed by t-test on Actual - predicted

Evaluating PINN Predictions via OLS Regression

Nasdaq Private Market

- OLS regression carried out between PINN prediction and actual trade prices

	Estimate	SE	tStat	pValue
(Intercept)	-1.5099	1.2351	-1.2224	0.22609
PINN Call Price	5.0074	0.2838	17.644	4.59E-26

Number of observations: 65, Error degrees of freedom: 63

Root Mean Squared Error: 7.32

R-squared: 0.832, Adjusted R-Squared: 0.829

F-statistic vs. constant model: 312, p-value = 4.48e-26

- Coefficient for the PINN predicted was 5 & statistically significant
- High coefficient value is another evidence of bias in data suggesting trades at a deep discounts to fair value of embedded call

PINN vs Data-Driven Models: Comparative Performance

- A testing data set was generated using MC and both models were tested

	PINN	Ensemble
MAE	0.023	12.990
RMSE	0.038	21.500
R-Squared	100	62.60
Mean Difference	0	-1.612

- Mean difference was calculated between actual and prediction to test for bias
- A two-sample t-test showed that this difference was statistically significant, suggesting the data driven model overestimated the embedded call prices as it was trained on biased data.

Build, Scale, Deploy: MATLAB on AWS

- Very detailed documentation with examples to get started
- Deep learning toolbox offers quick way to build POC due to less moving parts
- Scaling on CPU+GPU using Parallel computing toolbox
- Matlab docker image on AWS Sagemaker allows the use CPU or GPU optimized instances as needed
- Complier SDK allows rapid deployment of Matlab docker image and use it as Microservice on AWS App Runner
- Matlab technical support for expedited trouble-shooting

Conclusion

- PINN is able to price SPV structures by learning embedded optionality without the need of clean training data.
- Testing against real trades confirms that PINN not only captures price directionality but does so with statistical significance and avoids overfitting
- NPM's propriety daily source for stock prices and volatility, Tape D®, can be used to price SPVs, offering confidence in this innovative pricing solution
- This framework can offer transparency to market participants and help reduce information asymmetry in private stock markets.

Reference

1. Dong, Y. (2024, October 22). Physics-informed neural networks for option pricing [Video]. MathWorks Finance Conference. MathWorks.
2. De Ryck, T., & Mishra, S. (2021, June 28; revised July 10). Error analysis for physics informed neural networks (PINNs) approximating Kolmogorov PDEs (arXiv:2106.14473). arXiv. https://arxiv.org/abs/2106.14473
3. Kwok (HKUST), “Pricing of defaultable claims using the intensity approach,” MATH685R lecture slides
4. Carr, Peter. “Randomization and the American Put.” The Review of Financial Studies 11, no. 3 (1998): 597–626. https://doi.org/10.1093/rfs/11.3.597
5. Al-Safwan, A., Song, C., & bin Waheed, U. (2021). Is it time to swish? Comparing activation functions in solving the Helmholtz equation using physics-informed neural networks. arXiv:2110.07721. https://doi.org/10.48550/arXiv.2110.07721
6. Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010) (pp. 249–256). Proceedings of Machine Learning Research, 9. https://proceedings.mlr.press/v9/glorot10a.html

Please Read These Important Legal Notices + Disclosures

THE INFORMATION CONTAINED HEREIN IS PROVIDED FOR INFORMATIONAL AND EDUCATIONAL PURPOSES ONLY. NONE OF THE INFORMATION PROVIDED REPRESENTS AN OFFER TO BUY OR SELL, OR THE SOLICITATION OF AN OFFER TO BUY OR SELL ANY SECURITY. NOR DOES IT CONSTITUTE AN OFFER TO PROVIDE LEGAL, TAX, FINANCIAL OR INVESTMENT ADVICE OR SERVICES

NASDAQ PRIVATE MARKET, LLC IS NOT: (A) A REGISTERED EXCHANGE UNDER THE SECURITIES EXCHANGE ACT OF 1934; (B) A REGISTERED INVESTMENT ADVISOR UNDER THE INVESTMENT ADVISORS ACT OF 1940; OR (C) A FINANCIAL OR TAX PLANNER, AND DOES NOT OFFER LEGAL, FINANCIAL, INVESTMENT OR TAX ADVICE. SECURITIES RELATED SERVICES ARE OFFERED THROUGH NPM SECURITIES, LLC, A WHOLLY OWNED SUBSIDIARY OF NASDAQ PRIVATE MARKET, AND A REGISTERED BROKER-DEALER AND ALTERNATIVE TRADING SYSTEM, WHICH IS A MEMBER FINRA/SIPC. NASDAQ PRIVATE MARKET IS OPERATIONALLY INDEPENDENT AND DISTINCT FROM THE NASDAQ STOCK MARKET. LLC. TRANSACTIONS IN SECURITIES CONDUCTED THROUGH NPM SECURITIES, LLC ARE NOT LISTED OR TRADED ON THE NASDAQ STOCK MARKET LLC, NOR ARE THE SECURITIES SUBJECT TO THE SAME LISTING OR QUALIFICATION STANDARDS APPLICABLE TO SECURITIES LISTED OR TRADED ON THE NASDAQ STOCK MARKET LLC, NOR DO TRANSACTIONS IN SECURITIES CONDUCTED THROUGH NPM SECURITIES, LLC AFFECT AN ISSUER'S ELIGIBILITY TO LIST ITS SECURITIES FOR TRADING ON THE NASDAQ STOCK MARKET, LLC.