#12 Derivatives: Perpetual Futures

Lecture Notes for CS190N: Blockchain Technologies and Security — November 10, 2025

This lecture explores perpetual futures, the dominant financial instrument in cryptocurrency derivatives. We will demystify the core mechanisms that make them work, including the funding rate that anchors them to reality and the leverage that makes them both powerful and risky. We will focus on the essential concepts, using analogies and diagrams to build a clear mental model, and conclude with a look at how these instruments are implemented in a decentralized context through the case study of GMX and Hyperliquid.

1 THE PROBLEM: TRADING AN ECHO

1.1 Why Derivatives Exist

Imagine trying to build a financial system where the value of your measuring tape, the dollar, stretches and shrinks unpredictably every day. This is the challenge of building with volatile assets like Bitcoin (BTC) and Ether (ETH). Stablecoins were the first solution, creating a fixed, reliable unit of account. They are the solid ground in the volatile world of crypto.

However, stability is only half the story. A mature financial market also needs tools to manage, and speculate on, the volatility itself. This is where derivatives come in. Instead of trading the asset, you trade a contract that **derives** its value from the asset. You're not trading the object itself, but its echo. This allows for complex strategies like hedging (protecting your holdings from price drops) and leveraged speculation.

1.2 Spot vs. Futures vs. Perpetuals

To understand what makes perpetual futures special, let's quickly compare the three main ways to trade.

- **Spot Trading:** The simplest form. You buy 1 BTC, you own 1 BTC. It's a direct exchange of assets for immediate ownership.
- **Traditional Futures:** A contract to buy or sell an asset at a set price on a future date. Think of it as pre-ordering a new phone. You agree on the price today, but the transaction happens later. The fixed expiration date is crucial, as it forces the contract's price to converge with the asset's actual price on that day.
- **Perpetual Futures (Perps):** The innovation that took over crypto. A perpetual is a futures contract with a superpower: **it never expires**. You can hold a position indefinitely. This creates a seamless trading experience but introduces a new, critical problem: if there's no expiration date to force the price to converge, what stops the contract's price from drifting away from the real asset's price forever?

Table 1. Comparison of Trading Models

Feature	Spot Trading	Traditional Futures	Perpetual Futures
Expiration Date	None	Fixed Date	**None**
Leverage	Low/None	Yes	High
Asset Ownership	Direct	Contract Only	Contract Only
Price Anchor	Market Price	Expiration Date	**Funding Rate**

2 THE CORE MECHANISM: THE FUNDING RATE

2.1 The Problem: Keeping the Echo in Sync

The lack of an expiration date is the perpetual's greatest strength and its greatest engineering challenge. Without a future settlement date to act as an anchor, the price of a BTC perpetual contract (P_{perp}) could drift miles away from the actual spot price of BTC (P_{spot}). The contract would become a meaningless echo, completely disconnected from the asset it's supposed to represent.

The solution to this is an elegant mechanism called the **funding rate** [2].

2.2 The Solution: A Financial Tug-of-War

The funding rate is a periodic payment exchanged directly between traders holding long positions and those holding short positions. The exchange doesn't take a cut; it just facilitates the transfer. Its sole purpose is to create a financial incentive that pulls the perpetual price back towards the spot price.

Think of it as a constant tug-of-war, where the funding rate is the force that keeps the rope centered over the spot price.

- **When $P_{\text{perp}} > P_{\text{spot}}$ (Perp is expensive):** The market is bullish, and there are more longs than shorts. The funding rate becomes **positive**. To rebalance, longs must pay shorts. This makes it expensive to be long and profitable to be short, encouraging traders to sell the perpetual, pushing its price down towards the spot price.
- **When $P_{\text{perp}} < P_{\text{spot}}$ (Perp is cheap):** The market is bearish, with more shorts than longs. The funding rate becomes **negative**. To rebalance, shorts must pay longs. This punishes short positions and rewards long positions, encouraging traders to buy the perpetual, pushing its price up towards the spot price.

This balancing act is illustrated in Figure 1.

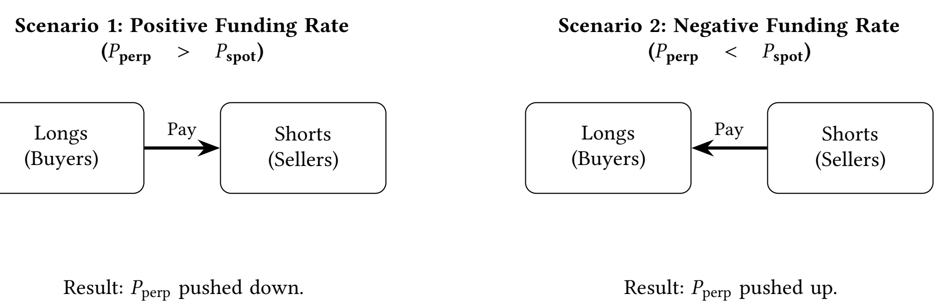

Fig. 1. The funding rate mechanism acts as a balancing force. Payments flow from the side of the market with the majority of positions to the minority side, restoring the perpetual price's alignment with the spot price.

2.3 The Funding Rate Formula

The funding payment is calculated periodically (e.g., every 8 hours). The amount a trader pays or receives is determined by the notional value of their position (in the quote currency, e.g., USD) and the funding rate [2]:

$$\text{Funding Payment (USD)} = \text{Position Notional (USD)} \times \text{Funding Rate (\%)} \tag{1}$$

The **Funding Rate** itself is designed to pull the perpetual price towards the spot price. It is composed of two main parts:

$$\text{Funding Rate (F)} = \text{Interest (I)} + \text{Premium (P)} \tag{2}$$

- **Interest (I):** This is a small, usually fixed rate meant to account for the difference in interest rates between the base (e.g., BTC) and quote (e.g., USD) currencies. For simplicity, we can assume this is a constant, like **0.01%** per 8-hour period.
- **Premium (P):** This is the most important part. It measures the **percentage difference** between the perpetual price and the spot price. To prevent manipulation, this is averaged over the funding period (e.g., as a Time-Weighted Average Price or TWAP).

$$P = \text{Time-Weighted Average}\left(\frac{P_{\text{perp}} - P_{\text{spot}}}{P_{\text{spot}}}\right) \tag{3}$$

In simple terms: if the perpetual price is consistently higher than the spot price (a positive premium), P will be positive, leading to a positive funding rate. If the perp price is consistently lower, P will be negative, which in turn **pushes** the funding rate to become negative (or at least smaller, as it may be offset by the positive Interest I).

The **Interest (I)** exists to balance the difference in borrowing interest rates between the base currency (e.g., BTC) and the quote currency (e.g., USD). In a standard BTC/USD contract, holding a BTC long position is conceptually like "borrowing" USD to buy BTC, which has a cost. This 0.01% rate represents that cost. Therefore, even if P_{perp} and P_{spot} are exactly equal (meaning $P = 0$), the funding rate F will still be 0.01%. This results in longs paying shorts a small interest payment, reflecting the cost of holding the position.

Example 1: Positive Funding Rate. Let's walk through a calculation for a BTC/USD perpetual, assuming an 8-hour funding period.

- **Trader's Position:** Long 1 BTC
- **Spot Price (P_{spot}):** \$100,000
- **Perpetual Price (P_{perp}):** \$100,120 (The market is bullish, so the perp is trading at a premium)
- **Position Notional:** 1 BTC × \$100,120 (using perp price) = \$100,120
- **Interest (I):** 0.01% (the fixed rate for this 8-hour period)

First, we calculate the **Premium (P)**. For this example, we'll assume the time-weighted average premium is equal to the current premium.

$$P = \frac{\$100,120 - \$100,000}{\$100,000} = \frac{\$120}{\$100,000} = 0.0012 = 0.12\%$$

Next, we calculate the **Funding Rate (F)** for this 8-hour period.

$$\begin{aligned} F &\approx P + I \\ &= 0.12\% + 0.01\% \\ &= 0.13\% \end{aligned}$$

The resulting funding rate is **positive (0.13%) for this 8-hour period**. Finally, we calculate the payment:

$$\text{Funding Payment} = \text{Position Notional} \times F = \$100,120 \times 0.13\% \approx \$130.16$$

Because the funding rate is positive, our trader, who is **long**, must **pay \$130.16** to a short-position holder for this funding period. This payment makes it more expensive to be long, incentivizing selling and pushing P_{perp} back down towards P_{spot}.

Example 2: Negative Funding Rate. Now, let's consider a bearish market where the perpetual is trading at a discount.

- **Trader's Position:** Long 1 BTC
- **Spot Price (P_{spot}):** $100,000
- **Perpetual Price (P_{perp}):** $99,900 (The market is bearish, perp trades at a discount)
- **Position Notional:** 1 BTC × $99,900 (using perp price) = $99,900
- **Interest (I):** 0.01% (the fixed rate for this 8-hour period)

First, we calculate the **Premium (P)**.

$$P = \frac{\$99,900 - \$100,000}{\$100,000} = \frac{-\$100}{\$100,000} = -0.001 = -0.1\%$$

Next, we calculate the **Funding Rate (F)** for this 8-hour period.

$$\begin{aligned} F &\approx P + I \\ &= -0.1\% + 0.01\% \\ &= -0.09\% \end{aligned}$$

The resulting funding rate is **negative (-0.09%) for this 8-hour period**. Let's calculate the payment for our trader:

$$\text{Funding Payment} = \text{Position Notional} \times F = \$99,900 \times (-0.09\%) \approx -\$89.91$$

The negative result indicates a **payment received** by the trader. Because the funding rate is negative, shorts must pay longs. In this case, our **long trader receives $89.91** from a short-position holder. This incentivizes buying the perpetual, pushing P_{perp} back up towards P_{spot}.

3 THE DOUBLE-EDGED SWORD: LEVERAGE AND LIQUIDATION

To understand the power and peril of perpetuals, let's follow a single example through its entire lifecycle.

3.1 Leverage: Amplifying Your Bet

The main attraction of perpetuals is **leverage**. It allows you to control a large position with a small amount of capital, known as **margin**.

Let's imagine a trader, Alice, believes BTC is going up.

- **Alice's Capital (Margin):** $3,000
- **Leverage Chosen:** 20x
- **Resulting Position Size:** $3,000 × 20 = $60,000

When the price of BTC is $100,000, Alice's $3,000 margin now controls a position worth $60,000 (which is 0.6 BTC). This amplifies both her profits and losses, as shown in Figure 2.

- **Profit:** If BTC price rises 5% (to $105,000), her 0.6 BTC position is now worth $105,000 × 0.6 = $63,000. Her profit is $3,000. On her initial $3,000 margin, this is a **100% gain**.
- **Loss:** If BTC price drops 5% (to $95,000), her 0.6 BTC position is now worth $95,000 × 0.6 = $57,000. This is a $3,000 loss, which wipes out her entire initial margin.

3.2 Liquidation: The Automated Stop-Loss

What happens when the market moves against Alice? The exchange cannot let her losses exceed her $3,000 deposit. To protect itself, the exchange has an automated safety mechanism: **liquidation**.

As we saw, with 20x leverage, a 5% price drop creates a 100% loss of margin (since $20 \times 5\% = 100\%$).

- **Entry Price:** $100,000

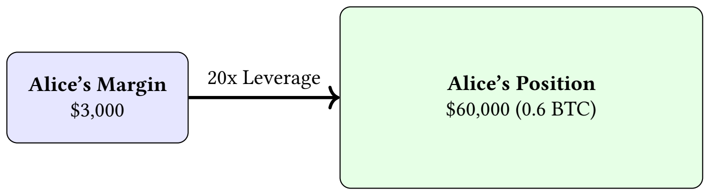

Fig. 2. Alice uses $3,000 of her own capital (margin) to control a $60,000 position.

- **Margin:** $3,000
- **Critical Loss Point:** A 5% price drop.
- **Liquidation Price:** $100,000 × (1 - 0.05) = **$95,000**

The moment the asset's price touches $95,000, the exchange's **liquidation engine** forcibly closes her position. She loses her entire $3,000 margin. This mechanism is visualized in Figure 3. (Note: In reality, the exchange also has a **maintenance margin**, meaning liquidation might happen slightly before $95,000, e.g., at $95,100, to leave a small buffer for fees. But $95,000 is the conceptual zero-point).

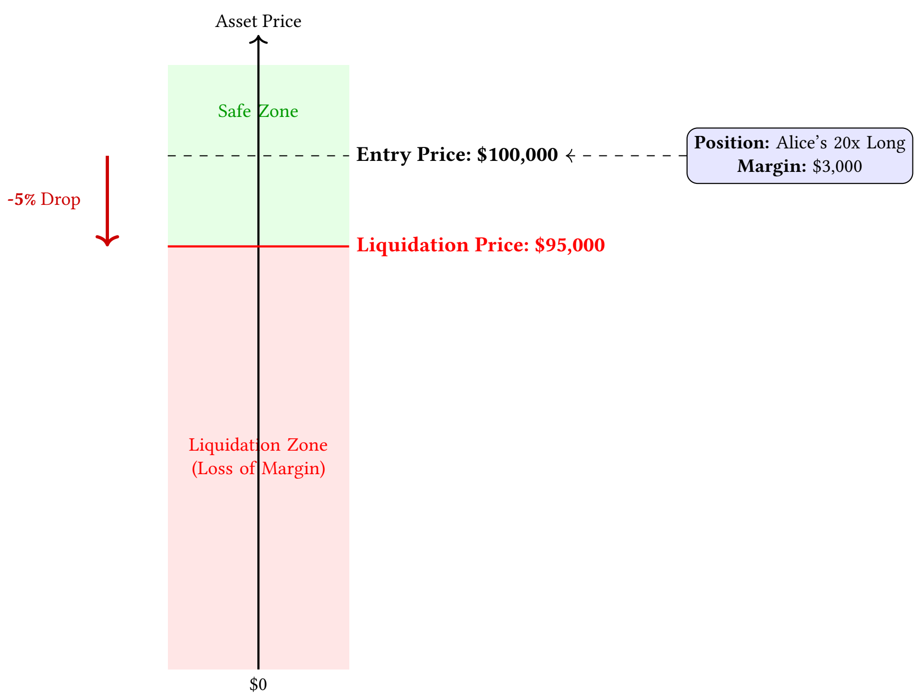

Fig. 3. For Alice's 20x long position entered at $100,000, a price drop of just 5% to $95,000 triggers liquidation, resulting in the total loss of her margin.

3.3 Cascading Liquidations: The Death Spiral

Liquidation is a risk for Alice, but it becomes a systemic risk when she is not alone. Imagine 1,000 other traders (Bob, Carol, etc.) all made the same 20x long trade as Alice, entering between $99,500 and $100,500. Their liquidation prices will all be clustered in a tight range around **$95,000**.

This creates a "liquidation wall." When an initial, normal sell-off pushes the price down to $95,000, the following feedback loop begins (visualized in Figure 4):

(1) **Trigger:** The price hits $95,000.
(2) **Mass Liquidation:** Alice's 0.6 BTC position is liquidated. Just like Alice, all 1,000 traders (600+ BTC) in the cluster are liquidated.
(3) **Forced Selling:** The exchange instantly dumps all 600+ BTC onto the market as **sell orders**.
(4) **Price Crash:** This massive, sudden supply of sell orders overwhelms the buy orders in the order book. The price instantly crashes further, say to **$90,000**.
(5) **The Cascade:** This new, lower price ($90,000) now hits the *next* cluster of liquidations. This triggers another wave of forced selling, pushing the price even lower.

This phenomenon, where one set of liquidations triggers a price drop that in turn triggers the next set, is a **cascading liquidation**. It is responsible for the sudden, violent price crashes often seen in crypto markets.

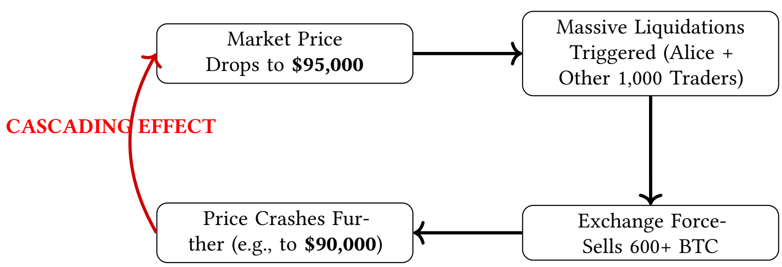

Fig. 4. The feedback loop of a cascading liquidation event, using Alice's example. One cluster of liquidations forces the price down, triggering the next cluster in a continuous, downward spiral.

4 CASE STUDY: MODELS FOR DECENTRALIZED PERPETUALS (GMX VS. HYPERLIQUID)

Decentralized perpetuals exchanges face a core design challenge: how to create a fast, liquid, and secure market? Different platforms have adopted different models. We will compare two of the most influential: GMX's "Trader vs. Pool" model and Hyperliquid's "On-Chain Order Book."

4.1 GMX: The "Trader vs. Pool" (GLP) Model

GMX, which pioneered on-chain perpetuals on L2s like Arbitrum, does not use a traditional order book. Instead, it uses a **pooled liquidity** model, centered around its **GLP token**.

- **What is GLP?** GLP is a multi-asset liquidity pool. Users (Liquidity Providers or LPs) can deposit assets like BTC, ETH, and USDC into this single "basket." In return, they receive a token (GLP) that represents their share of the entire basket.
- **How does trading work?** Traders trade directly **against the GLP pool**. The LPs, as a collective, act as the single counterparty (the "house") for every trade.
- **How is price determined?** GMX uses **oracle-based pricing**. It does not discover its own price. Instead, it relies on a high-speed external oracle (like Chainlink) that feeds it the average price from many large centralized exchanges (e.g., Binance, Coinbase) [3].

Example: GMX's "Zero Slippage". Let's see how this model enables a key feature: **zero slippage**.

- **Scenario:** A trader wants to open a 100 ETH long position.
- **Process:**

1. GMX's system pings the Chainlink oracle.
2. The oracle reports the current, volume-weighted average price of ETH is exactly $3,000.
3. GMX opens the trader's 100 ETH position at exactly $3,000.

The trade is executed with zero slippage. The trader gets the exact price they see, regardless of whether their trade size is 1 ETH or 1,000 ETH. When the trader wins, their profit is paid directly *out of* the GLP pool's assets. When they lose, their losses are added *to* the GLP pool.

Under this model, the LPs are betting that, in aggregate, the traders will lose more than they win.

4.2 Hyperliquid: The "On-Chain Order Book" (CLOB) Model

Hyperliquid represents a new generation that takes a different approach. Instead of a pool, it runs a **Central Limit Order Book (CLOB)** directly on its own blockchain [1].

- **What is a CLOB?** This is the traditional model used by centralized exchanges (and the stock market). It is a public ledger of all buy orders and sell orders at different prices.
- **How does trading work?** Traders trade **against each other**. When a trader places a buy order, it must be matched with another trader's sell order. Liquidity is provided by **market makers**, professional traders who place both bids and asks, hoping to profit from the price difference.
- **How is price determined?** The price is discovered directly on the platform. The market price is simply the price of the last matched trade, determined by the meeting point of the highest bid and the lowest ask.

To handle the high-speed (thousands of orders per second) requirement of a CLOB, Hyperliquid has its own **application-specific blockchain**, optimized for this single purpose.

Example. Let's run the same 100 ETH trade on Hyperliquid to see the difference.

- **Scenario:** A trader wants to open a 100 ETH long position by placing a market buy order.
- **Process:**
  1. The trader's order looks at the sell side of the order book (the list of ETH for sale), which might look like this:
     * 20 ETH for sale @ $3,000.00
     * 50 ETH for sale @ $3,000.10
     * 50 ETH for sale @ $3,000.20
  2. To fill the 100 ETH order, the system automatically buys the cheapest available ETH first. It buys all 20 ETH at $3,000.00. Then, it moves to the next price level, buying all 50 ETH at $3,000.10. It still needs 30 ETH, so it buys 30 ETH from the last level at $3,000.20.
- **Result:** The trader's 100 ETH are filled, but not at a single price. Their average price will be **$3,000.11**[1]. The difference between the best price they saw ($3,000) and the average price they actually got ($3,000.11) is called **slippage**.

4.3 The Core Trade-Off

As illustrated in Figure 5, these two models represent a fundamental difference in design:

- **GMX (Trader vs. Pool):** Prioritizes simplicity and capital efficiency. It's easy for LPs to participate: they just deposit assets and collectively act as "the house" against all traders. Traders get zero-slippage execution. Its weakness is its reliance on oracles; price discovery does not happen *on* GMX, it is imported from other exchanges.

[1] $(20 \times 3000.00 + 50 \times 3000.10 + 30 \times 3000.20)/100$

- **Hyperliquid (On-Chain CLOB):** Prioritizes performance and traditional market dynamics. It offers real-time, native price discovery, just like a centralized exchange. Its weakness is that it re-introduces slippage and requires a constant presence of professional, high-frequency market makers. Without them, the order book becomes thin (lacks buy/sell orders), and even small trades can cause massive slippage.

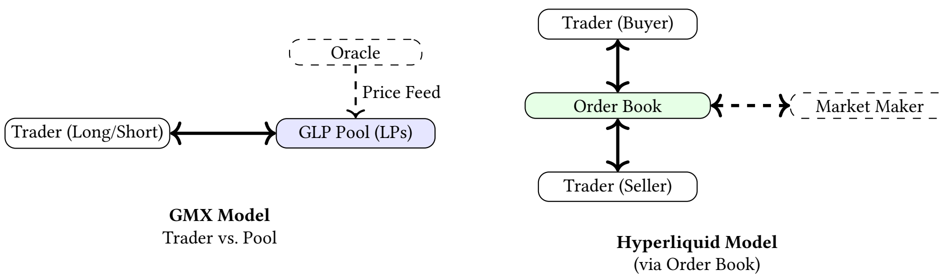

Fig. 5. A comparison of the GMX (Oracle-fed pool) vs. Hyperliquid (Order Book-centric) models.

REFERENCES

[1] 2024. Hyperliquid: Docs. https://hyperliquid.gitbook.io/hyperliquid-docs. (2024). Accessed: 2025-10-01.

[2] 2024. Perpetual Contracts Guide. https://www.bitmex.com/app/perpetualContractsGuide. (2024). Accessed: 2025-09-27.

[3] Chainlink. 2017. Chainlink: A Decentralized Oracle Network. https://chain.link/whitepaper. (2017). Accessed: 2025-09-27.