

The Problem of LVR (Loss-versus-rebalancing)

First coined by a team of researchers from [Columbia University](#), LVR is a form of arbitrage that occurs whenever an AMM has an outdated (stale) price in comparison to some other trading venue.

Arbitrageurs exploit this difference by trading from the AMM to the more liquid exchange (usually a centralized exchange like Binance), correcting the arbitrage and extracting value from LPs in the process.

How LVR Works

Most AMMs (such as Uniswap) are examples of constant function automated market makers—CF-AMMs. These AMMs take in two assets and automatically re-calculate prices after each trade to ensure ample liquidity at all times. This means the AMMs experience asset price discovery as they are being traded against.

However, since crypto assets trade on various platforms including centralized exchanges, the prices on a CF-AMM may be outdated compared to the more liquid prices on a centralized exchange. This discrepancy allows arbitrageurs to trade between the liquid exchange and the outdated AMM, correcting prices and capturing arbitrage in the process.

"Loss-versus-rebalancing" is the scenario where an LP provides liquidity to a CF-AMM whose prices are stale compared to a more liquid venue like a centralized exchange. By leaving their funds in the liquidity pool, they are not making as much money as they could be if they were to constantly "rebalance" between the AMM and the exchange. Thus, they incur a "loss" compared to what they would have in a "rebalancing" strategy.

The Impact of LVR on LPs

Many liquidity providers haven't even heard of LVR, but it costs them 5–7% of their liquidity, resulting in hundreds of millions lost each year. In fact, when accounting for LVR, many of the largest liquidity pools are not profitable for LPs at all.

Due to these losses, LVR is a type of MEV (maximal extractable value) that accounts for more price exploitation than frontrunning and sandwich attacks combined. [Edit this page](#) [Previous Concepts](#) [Next How CoW AMMs Work](#)