Proof's Public-facing TCA: Latest Results Over One Year of Data

March 23, 2023

Abstract

Here we apply Proof's current framework for evaluating our trading performance. We have designed robustness and client privacy checks that enable us to make any stable results public, and so far we can report: a stable improvement of roughly 20% of the spread in 1 second markouts for day limit orders executed on IEX vs. other venues collectively, volume curve tracking error that is lower on average than a simulated baseline, and positive evidence that our impact model's predictions are beating a trivial baseline in the real market conditions of our trading. We also share breakdowns of our trades in terms of venues and in terms of algo components (e.g. our VWAP algo vs. our flagship Proof algo, and the liquidity seeking component of the Proof algo vs. its impact-minimizing scheduler component). We are able to report one parent order slippage stat, though only normalized by spread and rounded to a very wide range. For our Proof algo, spread-normalized slippage vs. arrival is closer to twice the spread on average than to 0 or to four times the spread.

1 Overview

Previously, we detailed our initial design of the suite of metrics that we perform to analyze our trading performance (see our previous whitepaper), and we presented initial results over six month period. Here, we apply our framework to a one year period and provide newer results.

At this point in our evolution as a company, Proof remains challenged in our analyses by our low sample sizes. We simply need more clients and more client activity before we can reach the sample sizes that appear to be needed to produce robust numbers for notoriously noisy stats like slippage vs. arrival at desired levels of precision. We have developed a metric we call "distilled impact" that removes some of the noise in arrival slippage by referencing the contemporaneously price movement in an ETF that tends to be correlated with the symbol we traded. Our distilled impact numbers do show less noise than arrival slippage, but still generally more noise than our current sample sizes can overcome. Nonetheless, our analysis framework keeps us solidly grounded through robustness checks that protect us from drawing spurious conclusions or compromising client privacy, despite our relatively small sample size.

In this report, we briefly review the design of the analyses we perform and provide the updated subset of results that currently pass our robustness checks. Our data set here will be all of Proof's trading activity in the twelve month period from February 2022 through January 2023. We chose this date range as the statistics were run during February 2023, and so January 2023 represented the most recent full month of data available, and we wanted to encompass a one year time frame.

Client Privacy There is potentially a conflict between public accountability and the privacy of our clients. Ultimately, the results of our analyses are not only a function of Proof's trading logic, but also a function of when, what, and how our clients trade. Proof has a rigorous process in place to ensure that any statistics released about our aggregate trading activity do not meaningfully compromise the privacy of our individual clients. Stats that merely describe the overall amount of trading activity at Proof, such as the number of active clients we have, the total notional value traded, our venue breakdown, etc. can simply be released. Stats that describe trading performance in some way, such as average slippage vs. a benchmark, markouts, how closely our VWAP algo follows the volume curve, etc. must be vetted before release to ensure that they are not meaningfully skewed by the trading data of any one individual client. The vetting process is:

1. Define a set of ranges for the stat. [E.g. for a stat that represents a percentage, we might specify that we will round to the nearest multiple of 10%, thereby grouping all possible answers that round to the same thing into a "range."]

2. Compute the stat on our full data set with all client data aggregated. Define the range of the result as the value that is a candidate for release.
3. For each client, remove all of that's client's trading data from the dataset. Recompute the stat and the range it belongs to. If the range does not match the value that is a candidate for release, terminate the process and do not release any value. If the range does match, restore that client's data and go on to repeat the check for the next client.
4. Once we have checked that all removals of a single client's data result in a stat that falls within the same candidate range, we may release the range.

This release process is intended to protect us from releasing statistics that depend heavily on the experience of one particular client. This protects the client's privacy, as well as protecting Proof from relying on numbers that may be misleading because they are not representative of the trading experience across clients. Clients may also opt out of being included in our aggregate analyses at all, as per our data privacy policy. [So far, no clients have taken this route.]

We perform additional robustness checks to help us gauge the level of noise in our aggregate stats. In particular, we look at how much a stat can change based on removing small segments of data. For example, we might gauge the robustness of a performance stat like slippage vs. arrival by removing a small subset of the best performing order or a small subset of the worst performing orders, and see how much the calculated value changes. If a stat exhibits a disconcertingly wide range under this check, we do not rely on that stat in our decision making, nor do we release its value externally. Due to our small trading sizes at this time, many statistics fail at least one of our privacy and robustness checks.

In the following sections, we provide: 1. a summary of the basic stats that describe Proof's overall trading activity over this time period and 2. a summary of the suite of analyses we perform to assess our activity and the subsets of results that pass our client privacy and robustness checks. For a more detailed discussion on the design of the metrics, please refer to our previous whitepaper.

2 Overall Activity Stats

Here is an overall summary of our trading activity for the time period from February 2022 through January 2023 (inclusive):

- Number of active clients: 9
- Total notional value traded: 1.87 billion
- Notional value traded by our VWAP algo: 1.14 billion
- Notional value traded by our Proof algo: 727 million

Our Proof algo has two components that operate in parallel: a liquidity seeker that employs high min quantities to make large trades in the dark, and an impact minimizer that trades somewhat steadily through a mix of smaller lit and dark orders. In this time period, roughly 52% of the notional value traded by the Proof algo was accomplished through liquidity seeker component, while roughly 48% was accomplished through the impact minimizing scheduler component.

2.1 Venue Breakdowns

Let's take a look at how Proof's trading distributes over venues. We will separate our liquidity seeker here from the rest of our flow, as it can be interesting to see more specifically where we are finding relatively larger blocks. Our VWAP algo and the impact minimization component of our Proof algo share a common tactical layer for low-level decision making, so we keep the flow for these together here in one category that we will refer to as the flow for our "algo schedulers." For our purposes here, we round to the nearest 1%, and so we do not list venues that represented less than half of a percent of the total. [For this reason, our displayed values can add up to slightly less than 1, and 0.00 values may be displayed when only one of the metrics for a venue rises above our threshold for inclusion.]

Here is a breakdown of what percentage of the flow traded by our algo schedulers occurred at each venue, in terms of volume and notional value:

lastMarket	nv	volume
ARCA	0.01	0.00
IEXG	0.53	0.51
SGMT	0.00	0.01
NASDAQ	0.19	0.21
NYSE	0.25	0.25

Table 1: Venue Breakdown for Algo Schedulers (units of 0.01 = 1%)

The heavy IEX component here is unsurprising, as our posting logic heavily leverages the DLIM order type at IEX. As we will detail below, we continue to find that this results in improved 1 second markouts for our passive fills.

Here is a breakdown of what percentage of flow traded by our liquidity seeker occurred at each venue, in terms of volume and notional value:

lastMarket	nv	volume
CAES	0.08	0.10
IEXG	0.50	0.48
LEVL	0.05	0.04
SGMT	0.15	0.15
UBSA	0.14	0.16
NASDAQ	0.08	0.07

Table 2: Venue Breakdown for Liquidity Seeker (units of 0.01 = 1%)

Our liquidity seeker typically rests dark orders in several venues simultaneously with high minimum quantities, so we expect these proportions to be more a reflection of where there is matching liquidity to be found, rather than a reflection of our tactics. We should remind ourselves here that the sample size so far is fairly small, so there proportions may change dramatically over time. We also expect to connect to a few more venues in the medium-term.

3 Summary of our Suite of Analyses and Results

When designing the suite of analyses we perform to evaluate our trading algos, we keep in mind the fact that an "algo" is not really a monolithic thing. There are multiple layers of decision making that translate parent orders into executions, and each layer is roughly responsible for a different time scale. These layers roughly map onto classes in the Java code that comprises our trading system. At the top layer, we have schedulers that make decisions about how much we should trade over time intervals that are typically 5-20 minutes long. For our VWAP algo, we use a scheduler that tries to predict and follow the day's volume curve. For our Proof algo, we use a scheduler that aims to minimize our expected impact. The next layer, which we call the "tactical" layer (or the "algo" class in our java code), is responsible for deciding how to trade at a time scales of minutes. It decides how much to post, and how long to wait for passive fills before shuffling some volume to a take router. Below the tactical layer is the router layer, where the detailed logic for posting and taking sits. Our post and take routers, for example, make decisions about what venues to use, and how to manage orders on the timescale of seconds. Each of our router types produces distinct types of child orders, which are then sent to venues. We rely on tools like IEX's DPEG and DLIM order types, IEX's router, Nasdaq's MELO order type and others to pursue favorable outcomes on the microsecond and millisecond time scales.

As we try to assess the performance of our algo, noise accumulates at every time scale, and is greatest at the larger time scales. This is why metrics like 1 second markouts are much less noisy than parent order slippage. Metrics like parent order slippage evaluate the joint affect of *all* of the layers of algo decision making, thus catching all sources of noise as well. Data normalization and corrections for market trends can only make so much of a dent in the overall contribution of noise, making large samples necessary for getting meaningful results here.

3.1 Results for February 2022 through January 2023

Here are summary tables of our results on these metrics for the time period covering February 2022 through January 2023 (inclusive). Results that did not pass our client privacy or robustness checks are noted as "unstable" [1].

metric	rounding and units	result
slippage vs. arrival, vwap algo	nearest 10 bps	unstable
slippage vs. arrival, vwap algo	nearest 2∗spread	unstable
slippage vs. arrival, Proof algo	nearest 10 bps	unstable
slippage vs. arrival, Proof algo	nearest 2∗spread	2∗spread
slippage vs. vwap, vwap algo	nearest 1 bps	unstable
slippage vs. vwap, vwap algo	nearest 0.2∗spread	unstable
slippage vs. vwap, Proof algo	nearest 1 bps	unstable
slippage vs. vwap, Proof algo	nearest 0.2∗spread	unstable
distilled impact, vwap algo	nearest 10 bps	unstable
distilled impact, vwap algo	nearest 2∗spread	unstable
distilled impact, Proof algo	nearest 10 bps	unstable
distilled impact, Proof algo	nearest 2∗spread	2∗spread

Table 3: Parent Order Slippage Metrics

Two metrics did pass our client privacy check: spread-normalized slippage vs. arrival and spread-normalized distilled impact for the Proof algo. The range we used here is fairly wide, as we rounded to the nearest multiple of twice the spread. The positive value here indicates that we are buying higher than the benchmark and selling lower on average, by an amount that is closer to twice the spread than it is to zero or to four times the spread. It should also be noted that even this wide range was not stable under an additional robustness check, where we checked the value under removal of the best data points and then again under removal of the worst data points, removing roughly 5% of the notional value each time.

Mostly this table continues to send a clear message: we need more data before we can provide meaningful results on parent order slippage metrics.

metric	**rounding and units**	**result**
volume curve tracking error	0.02-wide ranges	0.02 - 0.04
simulated tracking error from historical curves	0.02-wide ranges	0.04 - 0.06
% liquidity seeker notional value from 10am to 3:30pm	nearest 10%	90%
1min before liquidity seeker trades	Less than 1 or greater than 1	less than 1
1min after liquidity seeker trades	less than 1 or greater than 1	less than 1
impact model features improvement over default	nearest 0.1	+0.1

Table 4: Behavioral Metrics at 1 minute plus time scales

The volume curve tracking error is a measure of the area between our own trading curve and the market's volume, and it is normalized so that 0 represents perfect tracking, and 1 is the worst possible score. This metric is primarily affected by the decisions of our scheduling layer and our tactical layer over timescales of minutes. We view our current score on this metric as acceptable, though we don't have a sense of an industry-wide standard. The dynamic volume prediction model that we have baked into our VWAP algo is intended to result in lower scores for this metric than we would expect to achieve by using a typical 20-day or 30-day average curve, and we ultimately expect this to reduce the variance of our slippage around VWAP. We have also included a "simulated tracking error" here as an attempt to see the relative contribution of our dynamic volume prediction. The simulated tracking error is computed

[1] We note that individual clients may receive detailed reports about their own trading data, and can request any raw data or analyses they want when it comes to their own data, as we view them as owners of that data. The robustness and privacy checks here apply solely to our use of aggregate data across clients (though we do insist on giving individual clients the results of robustness checks alongside the metrics we provide, so they have a proper context for interpreting their results).

by taking historical volume curves (so not applying our dynamic prediction) and simulating an algo that would follow these static curves for the same orders, trading in similar intervals/chunks. Our simulation is more rudimentary than our production algo, as it is assumes trades are scheduled into strict 10-minute time intervals, and orders are filled at the beginning of each interval. In contrast, our production algo uses interval sizes that are randomized and adjusted based on order size, and orders tend to be filled periodically throughout each interval. Nonetheless, this gives us some comfort that we are sticking closer to the curve than this default simulation on average.

We next perform some analyses to gauge behavior of our liquidity seeking component in the Proof algo. First, we look at what percentage of the liquidity it finds is traded in the middle of the day, rather than within thirty minutes of the opening or closing auction. This is roughly 90%.

The metrics here that look at price movements in the minutes before and after our liquidity seeker trades are normalized so that a score of 1 represents perfectly average movements, numbers less than 1 represent movements of less than average magnitude, and numbers greater than 1 represent movements of greater than average magnitude. **Scores greater than 1 on these metrics may indicate some level of information leakage. We are pleased here to have stable results that do not indicate this.**

The metric here that compares our impact model features to a default is intended to test that our model has predictive power for the real trading conditions we experience, rather than just in our testing on historical market data. Positive scores here indicate good evidence of predictive power, and hence our stable result of +0.1 shows we are beating our baseline in a stable way. For more detail on this metric, please see our prior whitepaper.

order type	Tif	rounding and units	result
Limit	DAY	nearest 0.1∗spread	unstable
Limit	IOC	nearest 0.1∗spread	unstable
Pegged	DAY	nearest 0.1∗spread	unstable
Pegged	IOC	nearest 0.1∗Spread	0.0

Table 5: 1 second markouts by order type and tif

To evaluate our router and venue layers, we look at 1 second markouts to mid for each fill. We have set the signs so that positive markouts represent buying lower than the later midpoint, or selling higher. Our markouts are normalized by spread, and then aggregated over fills with weighting by volume. For this data set, our 1 second markouts were largely unstable at the desired level of precision.

Exchange	rounding and units	result
IEX vs. non-IEX	nearest 0.1∗spread	0.2

Table 6: 1 second markouts for Day Limit orders, IEX vs. non-IEX

If we dig in a bit deeper into our Day Limit orders, we can see a stark and stable difference between our outcomes on IEX vs. other exchanges. We believe this difference also validates our IEX-heavy trading distribution. It is worth noting here that **the improvement we see on Day Limit markouts for IEX (roughly +0.2 of the spread) is much larger than the differences in access fees/rebates across exchanges.**

metric	rounding and units	result
percentage of intra-day shares passively filled	nearest 20%	80%
percentage of shares filled in auctions	nearest 10%	10%

Table 7: Other Metrics on Fills

Here we have a stable result that the percentage of intra day shares that are filled passively rounds to 80% as the nearest multiple of 20%. This is a wide range, but gives us some idea of our mix of passive and active fills. We can also see that our percentage of shares filled in auctions is stable at 10%, rounded to the nearest multiple of 10%.

Finally, we also perform screens to catch any unintended patterns in trades whose timing we directly control: trades where we cross the spread to take. We have added randomization at our tactical layer to

avoid, say, always taking in the first second of a minute, or always taking in minute 9 out of a 10-minute interval.

Currently, **these screens seem to validate that our current level of randomization suffices to stamp out any noticeable patterns of this form.** Running our screens on this particular data set though did reveal some interesting nuances in our design of the screens and how they interact with our algo mechanics.

More specifically, we look at individual take "events," which we define as seconds of the trading day that contain fills where we have crossed the spread. We make these seconds our unit instead of the fills themselves, as we often get several individual fills as part of the same process when we cross the spread to take. Having assembled our set of events, we look at their second timestamps (modulo 60) and their minute timestamps (modulo 10) to see if any particular value is unexpectedly prevalent.

Heavier trading into the close can skew these numbers a bit, as the steep tail of the volume curve will induce a tendency to trade closer to the end of the last 10 minute interval, as well as closer to the end of the last minute. To remove this effect, we remove trades that occurred in the last 30 minutes of the trading day from this analysis.

For the remaining data, our default expectation is that all values are equally likely to occur, due to the randomization we add to our algorithms' tactics. To decide if our data is consistent or inconsistent with this hypothesis, we run what are called "Monte Carlo" simulations. If our data includes N trade events, than each simulation will pick N random values from the default distribution where all values are equally likely. We then compare the frequency of the mode value in our real data with the modes in the simulations. We ask: what's the probability that the mode of such a simulation is at least as frequent as what we see in our real data? This probability is well estimated by running a lot of independent simulations. (In particular, we run 10,000 simulations.) If this probability is very low (say, less than 1%), this suggests that our algo tactics display a tendency towards a particular value for this feature on our flow, rather than being uniformly distributed across all possible value.

Across our full data set here, none of the modes jumped as out as being implausible to have occurred by chance. But we did happen to observe some suspicious modes when we limited to various subsets of the data (excluding certain clients, narrowing the time period of data collection, etc.) The reason for this was interesting - for relatively large orders, the algo may reasonably perform several small take waves in the course of a single minute. This happens particularly for our VWAP algo when we estimate that we are behind the volume curve, and when our posted orders are not generating trades. Overall, this behavior is reasonable and intended in these relatively rare instances. We would expect it to balance out over a large data set when it occurs in somewhat random minutes over the trading day across orders. However, especially at the small sample sizes that result when we chop our data up into smaller pieces, the relative scarcity of this phenomenon meant that the individual minutes containing many such trade events contributed a noticeable skew to the data in terms of the distribution of the minute modulo 10. If we grouped trade events by minutes instead of seconds, collapsing these bursts of activity to single events, the skew disappeared. At this point, we don't see this behavior as problematic, but we are encouraged that our timing pattern screens were sensitive enough to call our attention to the phenomenon.